COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS

OTTAWA - CANADA

TITLE: DISTRIBUTED DATA PROCESSING TECHNIQUES FOR VIDEOTEX INFORMATION SYSTEMS (FINAL REPORT)

AUTHORS: T.Y. Cheung (principal investigator)
L.G. Birta
J. Raymond
S.M. Wong (research assistant)

CONTRACTOR: Distributed Computing Research Group
Department of Computer Science
University of Ottawa

DEPARTMENT OF SUPPLY AND SERVICES
CONTRACT NO. : 0SU82-00119
FILE NO. : 24SU.36001-2-0894

DOC SCIENTIFIC AUTHORITY: Dr. M. Sablatash

CLASSIFICATION: unclassified

DATE: MAY 1983

INTRODUCTION AND EXECUTIVE SUMMARY

Objectives and Results

As stated in the contract, our research has two objectives :

(1) To find analytical and software engineering methods for investigating the following four distributed data processing problems which are direct applications to videotex systems: videotex file allocation, client (users and information providers) assignment, user interfaces with external databases and encryption-based security.

(2) To provide extensive bibliographies as reference resources for those distributed data processing problems which can be applied to videotex systems but whose developements are unrelated to videotex technology.

Our research began with a search over the relevant literature for analytical and software engineering approaches for solving these problems. For Objective (2), i.e., for those applications not directly relevant to videotex technologies, such as distributed database design, encryption-based network security, etc., we find that most of them have already established a good analytical foundation. Software engineering methods have also been used for the study of database query languages.

As for Objective (1), i.e., for those problems directly relevant to videotex technologies, such as functional properties of videotex networks, allocation of videotex files, assignment of users to videotex information centres, etc., we have found only a few using an analytic or software engineering approach. The relevant literature is overwhelmingly dominated by reports about marketing, social, psychological, developmental and field-trial studies of these systems. With very few exceptions, the contents of most of these reports are descriptive, failing to provide approaches for analysis and directions for analytic design. For example, a majority of the articles concerning videotex networks focus primarily on just hardware descriptions of these systems in an uncoordinated or casual manner. It is as if these networks are all different. Functional descriptions of their components are often incomplete, if not totally ignored. However, it is our belief that many design problems of videotex systems can be based on their functional or analytical specifications without explicitly referring to their hardware components.

In order to establish a foundation for our analytical study of a videotex system, we propose a network model for describing the functional relationship of its subsystems and for comparing their operational differences and similarities. The model is first applied to functionally classify many of the existing videotex systems. Several of them, though being quite different from a hardware point of view, are found to be functionally similar. Another major application of the model is that, by looking at these systems through the model, several analytical optimization methods available in the literature are found to be applicable for solving the problems of file allocation, user assignment, and information provider assignment for these systems.

Our research results are reported in two parts. The five chapters of Part I and the appendices present those results which we think are new, either in modelling or in experimentations. Part II includes five extensive bibliographies, providing a vast information resource of the well-developed analytical or software engineering techniques for distributed data processing.

PART I

Chapter I	Functional classification of videotex networks.
Chapter II	Allocation of files in videotex networks.
Chapter III	Assignment of users and information providers to the information servers.
Chapter IV	Issues of database interfaces.
Chapter V	Issues of data encryption.

PART II

Bibliography I	Distributed query processing.

Bibliography II Data, process and processor allocation.

Bibliography III Concurrency control for distributed databases.

Bibliography IV Data encryption.

Bibliography V Human factors methodology for database query languages.

Chapter I sets the foundation for our analytical investigation. Chapters II to V investigate those problems specified in the contract. A brief summary of these chapters follows.

Chapter I first presents a model for describing: (i) the functional interrelationship of the four subsystems of a videotex system, namely the subsystems of users, meta-service providers, information servers and information providers; and (ii) the functional architecture interconnecting the components of the information server subsystem. Based on this model, the major existing videotex networks are then classified into 5 types. Several of them, though being quite different from a hardware point of view, are found to be of the same type.

In an actual videotex system, information files are often duplicated at several geographical sites. For the purpose of deciding the locations of these files, Chapter II first classifies the 5 types of videotex networks mentioned in Chapter I into three groups. For Group I, allocation of the files is predetermined by the nature of the problem. Two analytical methods are presented for allocating the information files for videotex networks in the last two groups. These methods make use of the algorithms for the assignment and knapsack problems in mathematical programming. Some simulation results are included.

In a videotex system, the information retrievers (IR) provide the information retrieval services to the users and the information update servers (IUS) provide the information creation or modification services to the information providers (IP). In Chapter III, we consider the problems of how to assign the users to the IR's and the IP's to the IUS's in such a way that these services will be provided at a minimum cost. Minimization models based primarily on mathematical programming are proposed. Suggestions for simplifying the computational processes are also included.

Chapter IV studies the important problem of interface between an external database and a videotex system user. We first point out several issues of this problem which are specific to the videotex environment. One of these issues is the design of query languages utilized by users for accessing the databases. We classify query languages into 4 groups according to their complexities. Then we show how query languages may be 'executed' at a 'pure' videotex terminal by using two examples: a relational algebra of a relational database and DL/I of IBM's Information Management

System. Lastly, we briefly review some of the behavioural research works on query languages. Many references are listed in Bibliography V.

One of the methods often used in cryptanalysis is for the intruders to detect a trace of functional dependence between the plaintext and the cyphertext (for a fixed key) or between the key and the cyphertext (for a fixed plaintext). Breaking a cryptosystem may become much easier if such a dependence relationship is found. Videotex data, especially Telidon data which include publicly known opcode formats of the Picture Description Instructions, are sensitive to this kind of attacks. In Chapter V and Appendix C, we report the Chi-square test method for investigating this general problem. Experiments are run for the DES (NBS's Data Encryption Standard Algorithm).

The bibliographies of Part II are mainly for those system designers who want to pursue seriously the many distributed data processing concepts and methods available. Though we think most of the major works in these areas have been included, no claim is made of their being exhaustive.

Guideline for the Scope of Our Research

Videotex systems originated as low-cost interactive utility systems purely for the purpose of information retrieval. From a user's point of view, they are characterized by their very simple interactive page-oriented file searching techniques which are primarily based on tree-path traversal and menu-lookup. On the system side, videotex systems are frequently associated with the recent developments of videotex technologies, such as Telidon's Picture Description Instructions, Prestel's alpha-mosaic code for data representation, video terminals, videotex networks, etc.

For easy reference later, we refer to the above two characteristics as:

1. Videotex file-searching techniques (at user interface level).
2. Videotex data presentation technology.

Recently, research and field trials on videotex systems have evolved into integrated information management systems which offer, in addition to videotex information, a variety of other services, such as electronic mail, voice communications, telesoftware, etc. However, in order to claim to be a videotex system, they have to include (not exclusively) the above two characteristics as the essential factors in affecting their user interface and system design.

The above two characteristics have been used as the main guideline for determining the scope of our research. In general, we are seeking analytical or software engineering methods for solving 'pure' videotex problems, i.e., those problems directly involved with at least one of these two characteristics. Our results are collected in Part I. We do not look into, for example, a general electronic mail system. The design of such systems has very little to do with videotex technology, except perhaps at the level of user interface with a videotex system.

However, as required by the contract, we also provide in Part II of this report extensive references for some of the more general distributed data processing problems. These problems include many applications to videotex systems, though not necessarily having the above two characteristics of videotex technology.

CONTENTS

Appendix page

PART II

PART I

Chapter I

FUNCTIONAL CLASSIFICATION OF VIDEOTEX NETWORKS

1.1 INTRODUCTION

As pointed out in the Introduction and Executive Summary, there does not exist a logical foundation for analytical or functional studies of videotex systems. In this chapter, we provide such a foundation by proposing an abstract network model. This model describes the interrelationships of the different functional subsystems of a videotex network without referring to their hardware constituents. Section 1.2 describes the different subsystems of a videotex network. Section 1.3 describes the two parts of the proposed videotex network model. Part 1 is a functional architecture interconnecting the four subsystems of a videotex network. Part 2 is a functional architecture interconnecting the components of the central subsystem, namely the information server subsystem. Lastly, based on this model, the major (existing or proposed) videotex networks are classified into 5 types.

1.2 VIDEOTEX SUBSYSTEMS

Functionally, a videotex system can be considered as being composed of four subsystems, namely the subsystems of users (US), meta-service providers (MPS), information servers (ISS) and information providers (IPS). (See Figure 1.1)

A user (U), equipped with either a dumb videotex terminal or an intelligent terminal possessing videotex capabilities, requests services ranging from simple information retrieval to more sophisticated information management, such as electronic mail, voice communications, file transfer, etc.

A meta-service provider (MP) provides such services as teleconferencing, banking, etc. In general, these services are offered primarily to non-videotex users. Serving the videotex users is only their secondary role. They require special interfaces in order to communicate with the videotex users. A particular kind of meta-service provider, sometimes called third-party (TPD) or external databases in the literature, is worth special mention. These databases are organized according to a special data model (e.g., hierarchical, relational, etc.) and are usually stored in non-videotex formats. We shall return to this topic in Chapter IV.

An information server (IS) takes care of such tasks as usage accounting, statistics, and logging, information storage, retrieval and update, and meta-service switching. Depending on the implementation, an IS may perform only some of these tasks and an information server subsystem may include several types of IS´s. For example, in Britain´s Prestel System (TROU80), an update centre (UPC) performs only the information update service while an information retrieval centre (IRC) takes care of the other tasks.

An information provider (IP), as implied by its name, is responsible for creating the information. The information will be coded and stored in videotex formats in an information server for subsequent retrieval or update. Note that, in some real videotex systems, the tasks of data creation and update are often combined in a computing centre.

Note that the above scheme of classifying subsystems is functional. In actual implementation, a computer may be involved in several of these functional subsystems.

1.3 VIDEOTEX FUNCTIONAL NETWORKS

In this section, we propose a network model for describing: (1) the functional interconnection of the four subsystems mentioned in Section 1.2; and (2) the functional architecture of the information server subsystem (ISS). The model serves at least two purposes:

(i) As a basis for functional description and classification of videotex networks. In the literature, videotex networks are described uncoordinately, usually with more emphasis on their individual physical characteristics. No attempt is made to compare their functional similarities or differences or to provide a direction for their systematic investigation. We believe that many design issues can be pursued simply through their functional descriptions. Based on our model, we have been able to functionally classify the major existing or proposed videotex networks into five types. Several of these physically different systems are found to be functionally similar.

(ii) As a basis for analytical investigation of the many distributed data processing problems, such as file transfer protocol or electronic mail protocols for videotex services, etc. In particular, based on this model, this report considers the following problems for a videotex system : the allocation of files (Chapter II) and the assignment of users and information providers to information servers (Chapter III).

The model includes a description of the functional architecture of the information server subsystem. A description of the other three subsystems (US, IPS and MPS) is not included in the model because of two reasons:

(a) The model is concerned primarily with the overall functioning of a videotex system and not with the design of the individual subsystems.

(b) In a videotex network, the information server subsystem acts as the central communications controller. The other three subsystems do not communicate directly with one another. They communicate through the information server subsystem.

1.3.1 A Network Model

Our model consists of two parts. Part 1 (Figure 1.1) is a functional network architecture which describes the functional relationships among the four aforementioned subsystems, i.e., users, meta-service providers, information servers and information providers. Part 2 is a functional subnetwork architecture which describes the interconnection among the components of the information server subsystem (Section 1.3.2).

The functional network (Figure 1.1) can be realized by linking the different units of the four subsystems by means of various public or private communications networks, such as telephone networks, packet switching networks, CATV cable networks, local area networks, etc. Figure 1.2 shows an example of such a realization, in which information provider IP is connected to the information servers IS , IS and IS through an X.25 packet switching network.

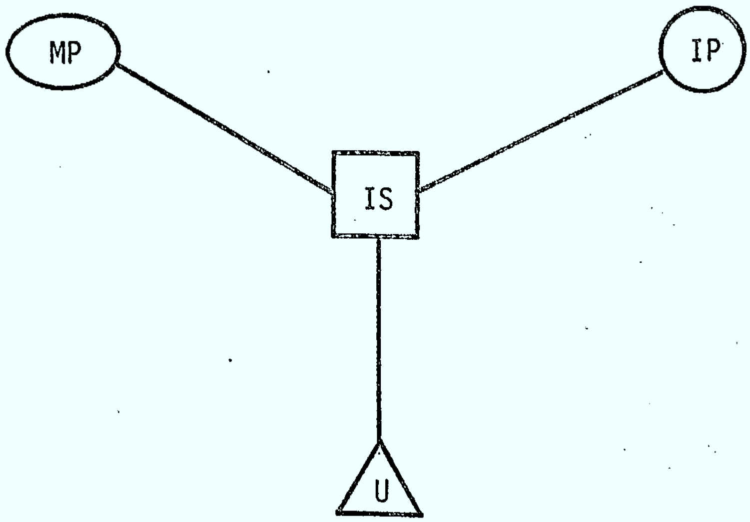


Figure 1.1 Functional network of a videotex system (IP: information provider; IS: information server; MP: meta-service provider; and U: user).

It is not within the scope of this contract to consider the design problems of these communications networks. (Readers interested in these problems can refer to (BOCH80, MANN82, WOOL79)). Instead, we are interested in those problems which can be described entirely in terms of the functional interconnection of these subsystems. Thus, we shall ignore the underlying communications networks hereafter. Figure 1.2, for instance, will then become Figure 1.3 .

Note that, as illustrated in Figure 1.3, we can assume that the units within each of the user, information provider and meta-service provider subsystems do not communicate directly among themselves or across their subsystem boundaries. This follows simply from the functional concept that communication between two such units is always through an information server.

1.3.2 Functional network architectures of the information server subsystem

In this subsection, we first describe a general functional network architecture which connects the components of the information server subsystem among themselves and with outside units. We then present several typical cases of this architecture.

General functional network architecture

In a videotex network, the information server subsystem (ISS) acts as the central communications controller. In

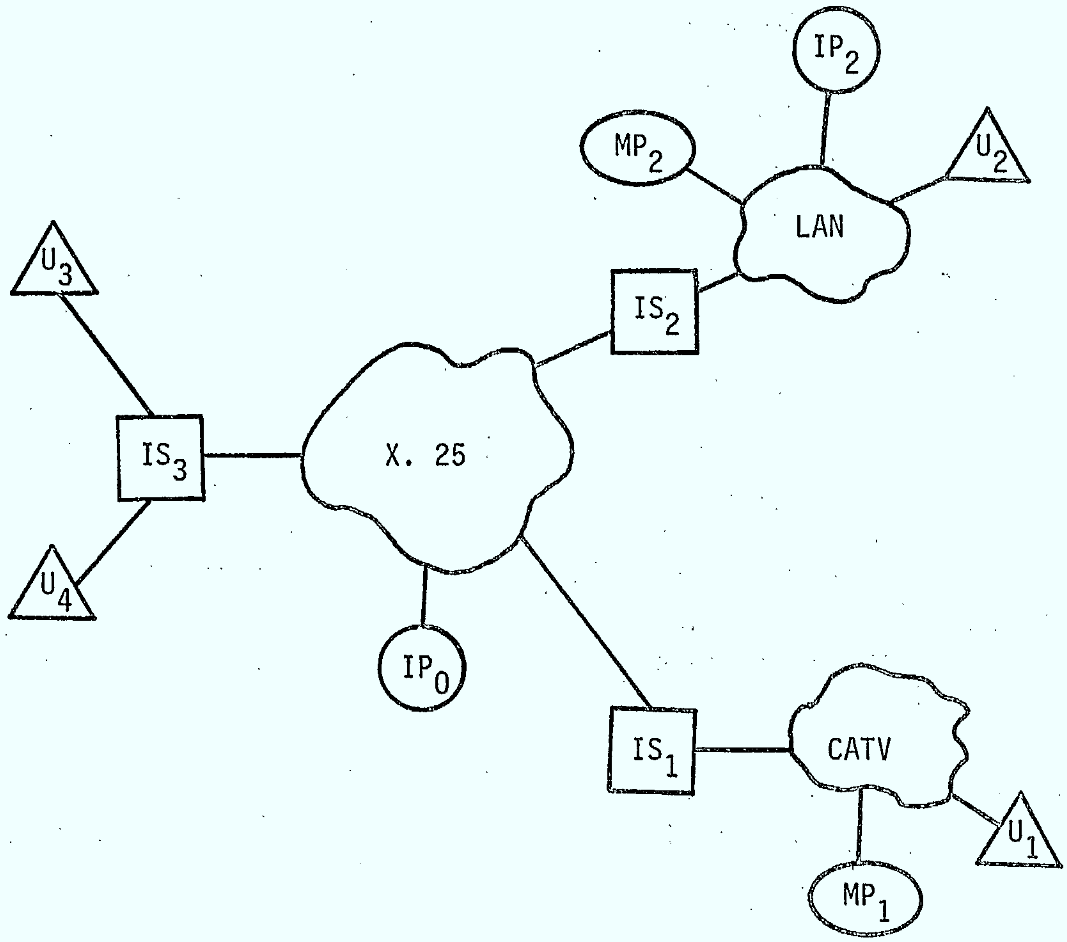


Figure 1.2 Realization of a videotex system by communications networks.

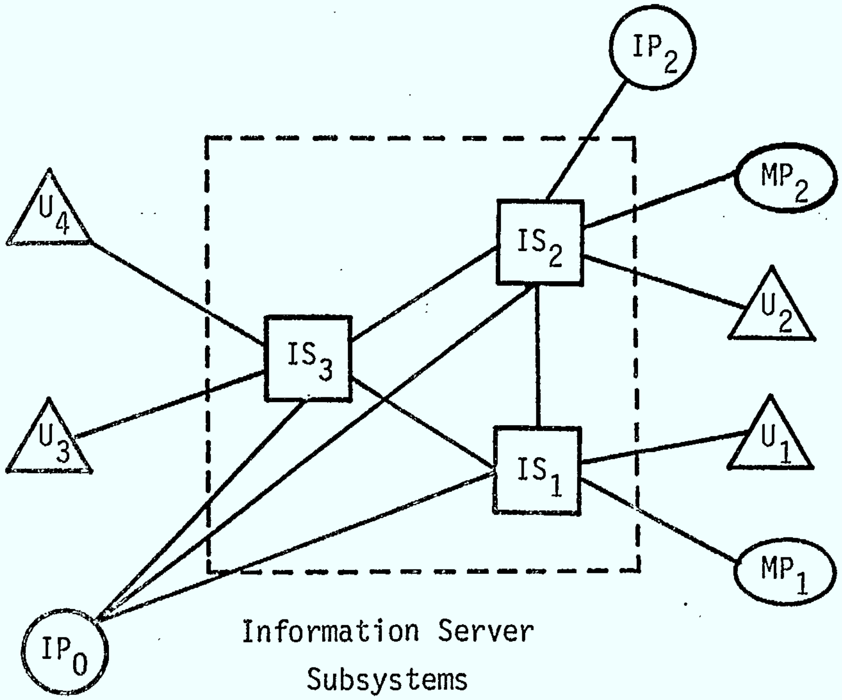


Figure 1.3 Functional network description for the interconnection of Figure 1.2.

order to interface with the three other subsystems, the components of ISS can be conceived as being clustered into three groups, namely the group of information retrievers (IR), the group of information update servers (IUS) and the group of meta-service switchers (MSS). Their functions, as shown in Figure 1.4 (a detailed version of Figure 1.1), are described below.

An information retriever (IR) interfaces with the users. It stores the videotex data and takes care of usage accounting, logging and statistics. It takes care of local data retrieval requests. It also passes control to an MSS for local or remote meta-service requests or remote data retrieval requests.

An information update server (IUS) receives data from an information provider (IP) for update. If the IUS and IP do not reside in the same computer, this can be done either off-line or on-line and either in an interactive mode (for news, stockmarket information, etc.) or in a batch mode (for more static information). If the data are already organized and coded in a videotex format, they are passed without any change onto the relevant information retrievers (IR) for storage. Otherwise, they have to be organized and coded in a suitable videotex format first.

A meta-service switcher (MSS) switches users' remote data retrieval requests or (local or remote) meta-service requests between IR's and meta-service providers (MP).

<u>Typical architectures</u>

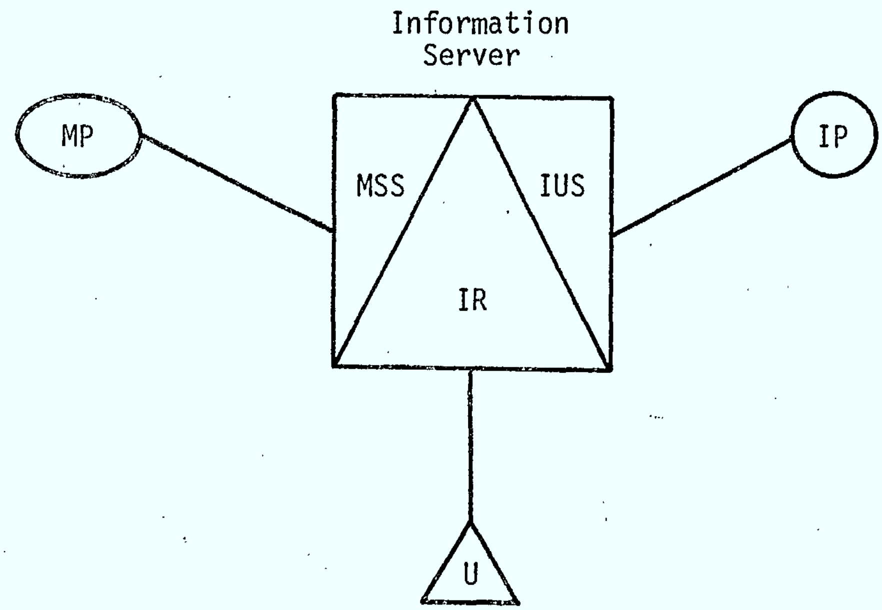


Figure 1.4 Interface components of an information server (IR: information retriever; IUS: information update server; and MSS: meta-service switcher).

The general architecture of the information server subsystem described above includes many special cases. In the following, we describe specifically a few typical ones. They represent the architectures of many well-known existing and proposed videotex networks. They also serve as the base for classifying these networks functionally. The classification scheme depends on such properties as how the components of ISS are functionally interconnected, whether all the IR's have a copy of the master database, how many IR's whose data will be updated by each IUS, etc. Solutions to two distributed data processing problems based on this classification scheme will be described in Chapters II and III, respectively.

The following notation will be used.

Notation

IP - information provider

IR - information retriever

ISS - information server subsystem

IUS - information update server

U - user

(m,n,k) - An ISS belongs to Class (m,n,k) if it has m IUS's and n IR's, and each IUS is responsible for updating the stored data of k of the IR's. If k is not known or irrelevant to the problem under consideration, it may be replaced by "?".

Type 1 For a videotex network of this type (Figure 1.5), its ISS belongs to Class (1,1,1). It has a single

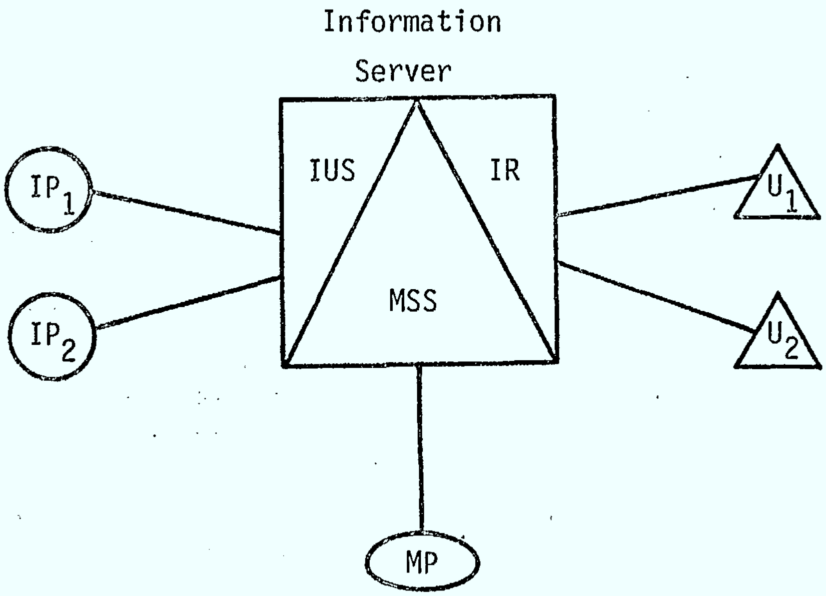


Figure 1.5 Type 1 videotex networks.

information centre which provides both information retrieval and update services to all the users and IP's.

This is the most common type of videotex networks. It includes many of the small videotex systems designed for the public or closed user groups and also many of the large-scale systems at their early stage of development.

Example 1.1 The first stage of Bell Canada's Vista System (BNR79, WILS80) consists of a single host which provides all the information providers with update service and all the users with retrieval service. (See Figure A.1 in Appendix A.)

Type 2 For a videotex network of this type (Figure 1.6), its ISS belongs to Class (m,m,1). Conceptually, it consists of m non-overlapping Type 1 subnetworks. Each subnetwork has a computing centre which functions as the information server subsystem (ISS) and has only one IR, one IUS and one MSS. Each IUS serves only the IR belonging to the same subnetwork. The MSS switches requests for data or meta-services to other subnetworks. Every file has only one copy within the entire network. According to the allocation policy of the files and users, we can

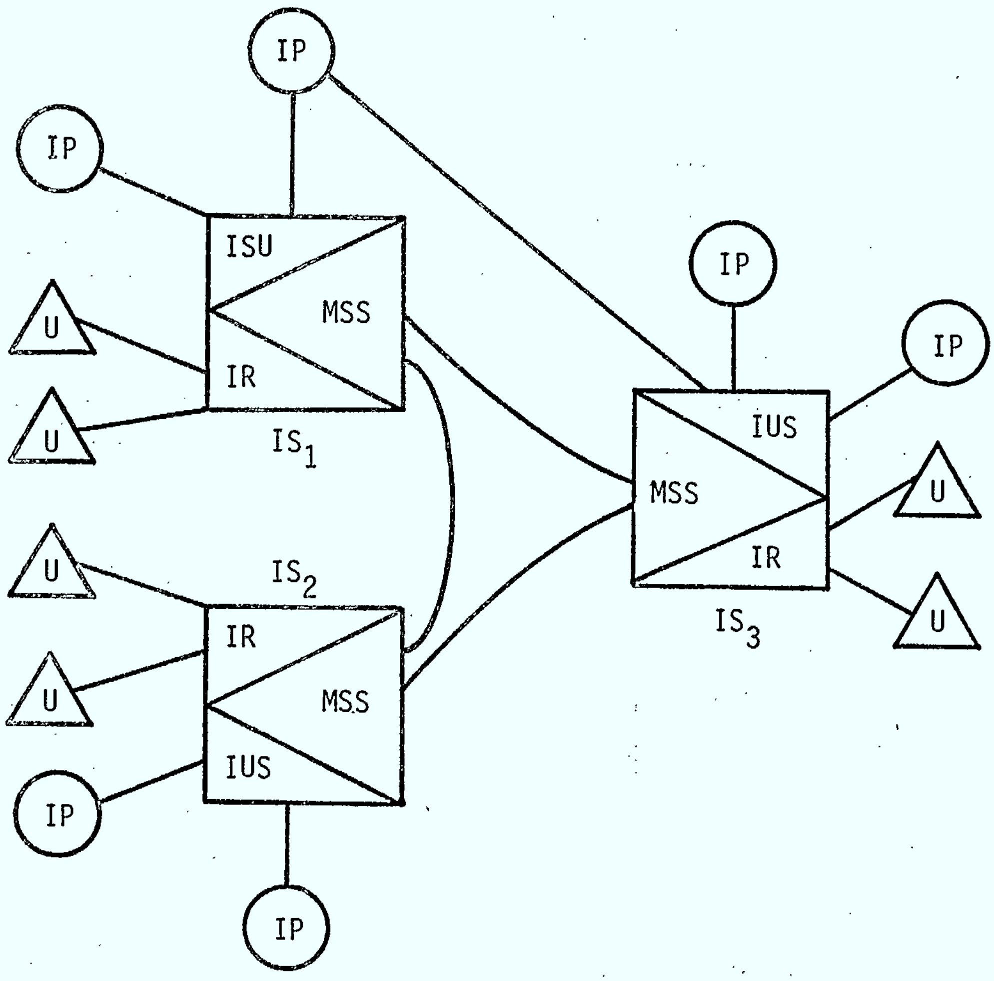


Figure 1.6 Type 2 videotex networks (compare with Figure 1.3).

distinguish between two cases of this type of networks:

Type 2.1 : A videotex network is of Type 2.1 if it belongs to Type 2 and if its files cannot be reallocated and the users cannot be reassigned from one subnetwork to another.

The subnetworks of this type want to exchange information and/or meta-services while maintaining their autonomy for control over their users and IP´s.

Example 1.2 A typical example of a videotex network of this type is the current iNET (FARR82). At the present time, TransCanada Telephone System´s (TCTS) Computer Communications Group (CCG) is responsible for a one-year field trial on this national network. iNET interconnects (not permanantly) the information centres of a number of Telidon-based videotex systems, such as Teleguide (by the Government of Ontario and Informart), Vista (by Bell Canada), NBTel (by New Brunswich Telephone), B.C.Tel (by British Columbia´s Telephone), etc. These videotex systems maintain their individual administrative autonomy but exchange data and services through DATAPAC or telephone networks.

Type 2.2 : A videotex network is of Type 2.2 if it is of Type 2 and if each of its files either has duplicates at all sites or has no duplication at all. The files can be reallocated and the users can be reassigned from one subnetwork to another.

The subnetworks of this type, probably all belonging to the same owner, can exchange the files they store and the users they service.

Example 1.3 During the current field trial, Germany's Bildschirmtext System (GRIE82), has two information servers (called BTX centres), one located at Berlin and another at Dusseldorf. These two BTX are independent as far as their users and information providers are concerned, but they share some of the meta-services and their files can be related from one BTX to another. (See Figure A.2 in Appendix A)

Type 3 In a videotex network of this type (Figure 1.7), its ISS belongs to Class (1,n,n). That is, the ISS has only one IUS and an arbitrary number of IR's. The number of MSS's is irrelevant. The IUS is connected to all the IR's and serves all the IP's. We distinguish between two cases of this type.

Type 3.1 A videotex network is of Type 3.1 if it is of Type 3 and each of its IR's contains a copy of the master database (i.e., data are fully duplicated at all IR's).

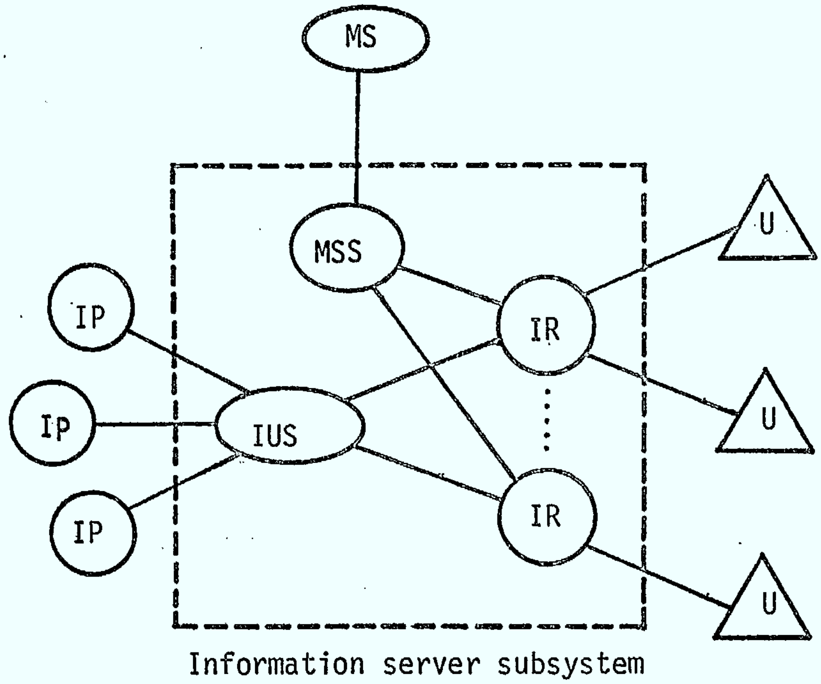


Figure 1.7 Type 3 Videotex Networks.

Example 1.4 The early stage of Britain's Prestel System (WOOL80) belongs to this type. A single update centre (UPC) was located at London. All data were updated there before being transferred to and stored at the information retrieval centres (IRC) located at London, Central Scotland, N.W. England and Birmingham. For the convenience of information management, every data frame is duplicated at all the IRC's. (See Figure A.3 in Appendix A.)

Type 3.2 A videotex network is of Type 3.2 if it is of Type 3 and only one of its IR's contains the master database, while all the other IR's contain only a portion of the whole database.

Example 1.5 At its current stage, France's Antiope System (MART79) belongs to this type. Its single information update centre (IUC) is connected to all IP's and serves all the information retrieval centre (IRC). However, unlike the Prestel System, only the IR located at the same site as the IUC contains the master database. The other IRC's contain only local information. (See Figure A.4 in Appendix A.)

Type 4 In a videotex network of this type (Figure 1.8), its ISS is of Class (m,n,n). The ISS has several IUS's, each of which is connected to a disjoint subset or

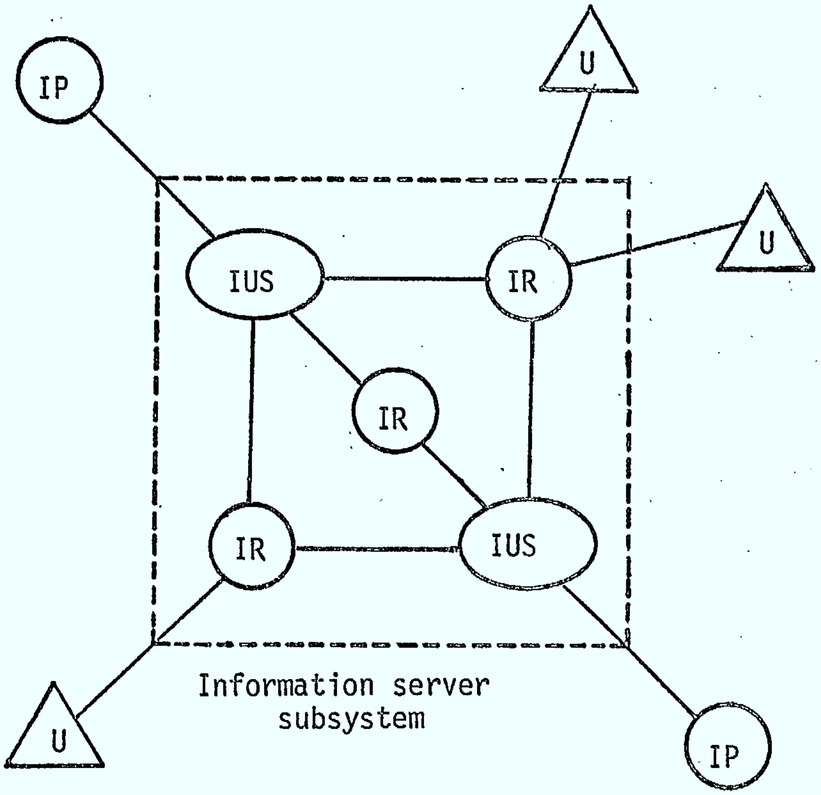


Figure 1.8 Type 4 Videotex Networks.

IP's but serves all the IR's. Each IR contains a copy of the master database.

Example 1.6 Currently, Britain's Prestel System (TROU80) is experimenting an architecture which contains up to three update centres (UPC) and many information retrieval centres (IRC). Thus, each UPC updates information for a specified group of IP's and sends a copy of the updated data to all the IRC's. Thus, each IRC contains a copy of the master database. (See Figure A.5 in Appendix A).

Type 5 In a videotex network of this type (Figure 1.9), its ISS is of Class (m,m,n). There are usually several videotex information centres. Each centre consists of an IUS and an IR (and possibly an MSS). An IUS serves not only the IR located within the same centre, but also those remote IR's where data are duplicated.

Example 1.7 Germany's new Bildschirmtext System (GRIE82), scheduled for starting services in Autumn 1983, has several videotex information centres (called BTX). Only one of these BTX's contains the master database. The other BTX's contain only local information. A data frame is updated and stored locally and a copy will be transferred to the master

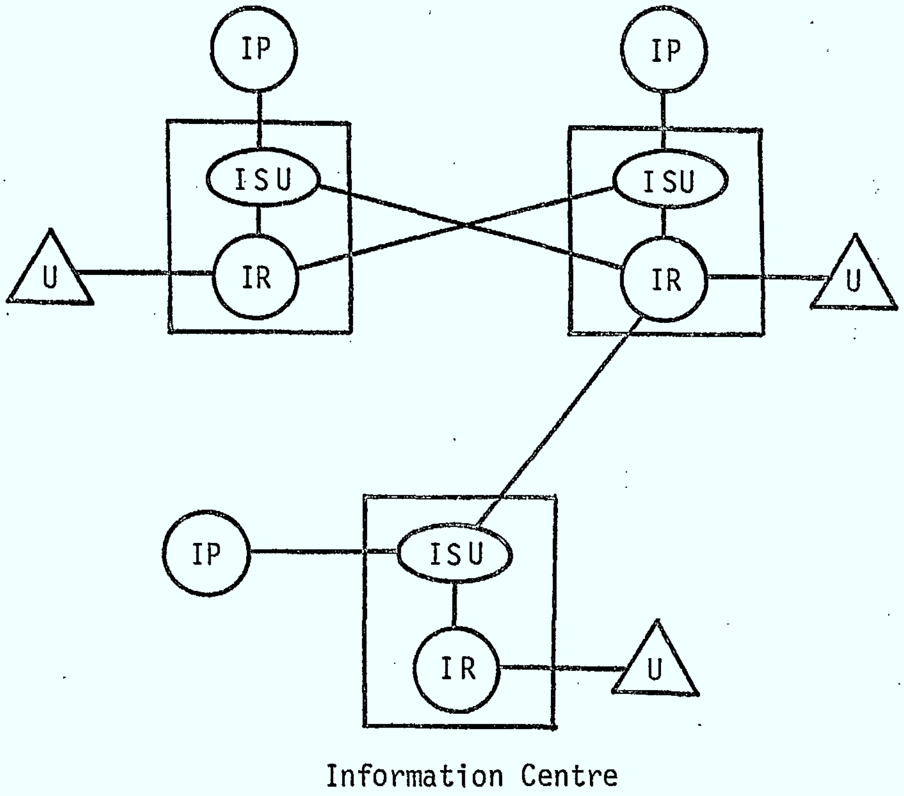


Figure 1.9 Type 5 videotex networks.

database. Other BTX´s having the same data frame will be informed and marked, but not updated immediately (for the purpose of reducing communications costs). Later, they will be physically updated from the master database when they are requested at these BTX´s. (See Figure A.6 in Appendix A.)

Chapter II

ALLOCATION OF FILES IN VIDEOTEX NETWORKS

2.1 INTRODUCTION

Normally, in a videotex system, multiple copies of each information file exist and are distributed in some ad hoc way to some (or possibly all) of the sites. In this chapter, we study the problem of how to allocate these files to the sites in a 'best' manner. Though the file allocation problem is often mentioned (BALL81, BOCH81, GECS82) as an important design issue for videotex systems, we do not know of any study of analytical methods for its solution in a videotex environment. It is not clear from the literature what criteria and methods the existing videotex systems use for allocating their files. In general, it appears to be done in a way that simply serves the convenience of administration or software design. In an analytical approach, the objective is to minimize a certain combination of communications, storage and processing costs. Though an analytical solution is usually not taken as the final solution in a design process, it often forms the initial step of an approximation procedure and provides insights into the problem which is often too complex to handle by intuition alone.

In the early stage of our search for analytical methods for these problems, one of the obstacles was the lack of a functional view of videotex systems. This led to the proposal of the network model of Chapter I. Through this model, we are now able to identify several analytical methods of solution, based mainly on mathematical programming and approximate solutions to the knapsack problem, to the file allocation problem for those types of videotex networks as classified in Chapter I. These methods will be described in the following sections.

2.2 ASSUMPTIONS AND COSTS FOR FILE ALLOCATION

In our study of the file allocation problem, we make the following general assumptions:

1. The geographical locations and network interconnection (both physical and functional) of the components of the videotex information subsystem, namely the information retrievers (IR), information update servers (IUS) and meta-service servers (MSS), are known.
2. The locations of the users are fixed and known.

3. For each file, its retrieval traffic (induced by the users) and update traffic (induced by the information providers and information update servers) can be estimated. These kinds of statistics are usually available at the information centres for billing or system analysis purposes. The communications cost for each file is therefore estimable.

4. The processing and storage capacities at each information retriever are known.

Additional assumptions may be needed for individual problems.

In order to estimate the total costs for communications, storage and processing, it is necessary to know who pays for these services. In general, this depends on the policies of the individual videotex network. The following example shows a typical case :

Example Payments by the users and information providers (IP) in the Prestel System (WOOL80) are as follows :

Users pay : i) to IRC for connection-time;

ii) to IP for frame accesses;

iii) to PO Telecommunications for telephone usage;

iv) to TV industry for adapted TV´s.

IP´s pay : i) to IRC for storage and connection-time; and

ii) to PO Telecommunications for telephone usage.

2.3 ANALYTICAL METHODS FOR FILE ALLOCATION

In general, the method for allocating files in a videotex system depends on the functional architecture of the videotex network. Since there are many such architectures (as pointed out in Chapter I), it is quite impossible to find analytical solutions for all of them. In the following, as illustrations, we shall consider three groups of videotex networks. Each group includes several types of networks as classified in Chapter I. When combined together, these groups cover most of the major existing videotex systems.

Group I This group includes networks of Types 1, 2.1, 3.1, and 4.

Examples: Bell Canada's Vista (Example 1.1), Britain's Prestel Systems (Examples 1.4 & 1.6) and Canada's iNET (Example 1.2).

Characteristics and Solutions: For videotex networks of Type 1 or Type 3.1 or Type 4, every IR contains a copy of the master database (i.e., every file is duplicated at all IR's). Thus, no decision making about file allocation is necessary.

A videotex network of Type 2.1 is an interconnection of several Type 1 subnetworks. The whole network has only one copy of every file, which can be retrieved by users both inside and outside its subnetwork, but can only be updated by the IUS´s within its subnetwork. It is not allowed to reallocate the files from one subnetwork to another. In practice, for example, the subnetworks may belong to different owners and their interconnection into a single videotex network is solely for the purpose of exchanging data and meta-services. They maintain their autonomy over their own subscribers (users and IP´s). For this kind of networks, a file is always allocated to the IR belonging to its owner´s subnetwork.

Group II This group includes networks of Type 2.2.

Example: Germany´s current Bildschirmtext System (Example 1.3).

Characteristics: In this group, the files are assumed not to be duplicated and to be relocatable from one subnetwork to another. Those files having duplicates at all sites may be excluded from consideration for our file allocation problem. In practice, for instance, the files may all belong to the same owner and their allocation to a suitable site may decrease the system´s overall expenses.

Solution: We want to allocate the different files to the sites so as to minimize the total cost for communications, storage and processing. The problem can be formulated as a generalized assignment problem as follows.

Notation	Explanation
F_i , $i=1,2,\ldots,n$	The n different files to be allocated.
s_i	The size of F_i .
S_j , $j=1,2,\ldots,m$	Sites of computers, each containing an IR and an IUS .
b_j	Storage capacity of the computer at S_j .
c_{ij}	Communications and processing cost if F_i is located at S_j , due to the retrieval traffic induced by its users and the update traffic induced by its IP .
x_{ij}	$x_{ij} = \begin{cases} 1 & \text{if } F_i \text{ is allocated at } S_j \\ 0 & \text{otherwise.} \end{cases}$

Note that the constants s_i , b_j and c_{ij} are readily available or estimable. In particular, c_{ij} are usually recorded by the system for billing or system analysis purposes. Note also that when all computers charge the same rate for processing and storage, these costs may be ignored as they do not affect the total cost no matter where the files are stored.

The optimal locations of the n files can be obtained by solving the following assignment problem :

Minimize: $\sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij}$

$$
\text{subject to} \begin{cases} \sum_{i=1}^{n} s_i x_{ij} < b_j & ,j=1,2,\dots,m \\ \sum_{j=1}^{m} x_{ij} = 1 & ,i=1,2,\dots,n \\ x_{ij} = 0 \text{ or } 1 & ,j=1,2\dots,m; \\ & j=1,2\dots,n. \end{cases} \quad (1)
$$

In fact, (1) is a special case of the generalized assignment problem where the constants s_{ij} are replaced by s_i. There exist at least two algorithms (DEMA71, SRIN73) for solving the special case (1). A branch and bound algorithm (ROSS75), based on the solution of a series of 0-1 knapsack problems for determining the bounds, has also been developed for medium-sized (up to 4000 binary variables) generalized assignment problems.

<u>GROUP III</u> This group includes videotex networks of Types 3.2 and 5.

<u>Examples:</u> France's Antiope System (Example 1.5) and Germany's proposed Bildschirmtext System (Example 1.7).

<u>Characteristics:</u> The architecture of a videotex network of this group is shown in Figure 2.1. The IR at site S_0 has a master database and the IR's at the other sites contain a portion of the database only.

For information retrieval, the files located at a local site S_j will be searched first. If the data are not found, the request will be switched to the central site S_0. Since S_0 contains the master database, no further switching is necessary (See Figure 2.1). Thus, retrieval traffic will be increased along S_0/S_j if a file is not stored at S_j.

For information update, we distinguish between two cases:

<u>Case a</u> (Figure 2.2a) This case includes networks of Type 3.2. All IP's are linked to S_0. Thus, a file can only be updated at S_0. A new copy of that file is then sent to those sites (e.g., S_1, S_3) which also store a copy of it.

<u>Case b</u> (Figure 2.2b) This case includes networks of Type 5. The IP's are attached to different sites. A file is updated at the site (e.g., S_1) associated with its IP. A new copy of that file is then transferred to S_0, from where the other sites (e.g., S_3) containing a copy of it will be updated later (e.g., when there is an actual retrieval request at S_3 for that file).

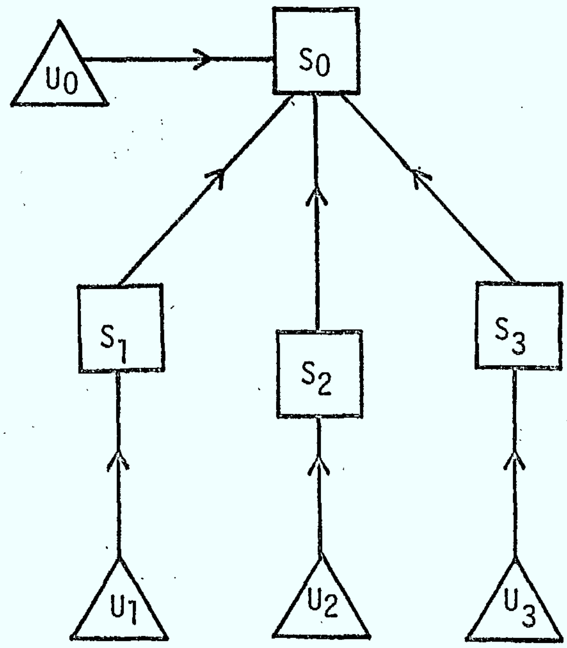


Figure 2.1 Direction of retrieval requests in Group III Networks.

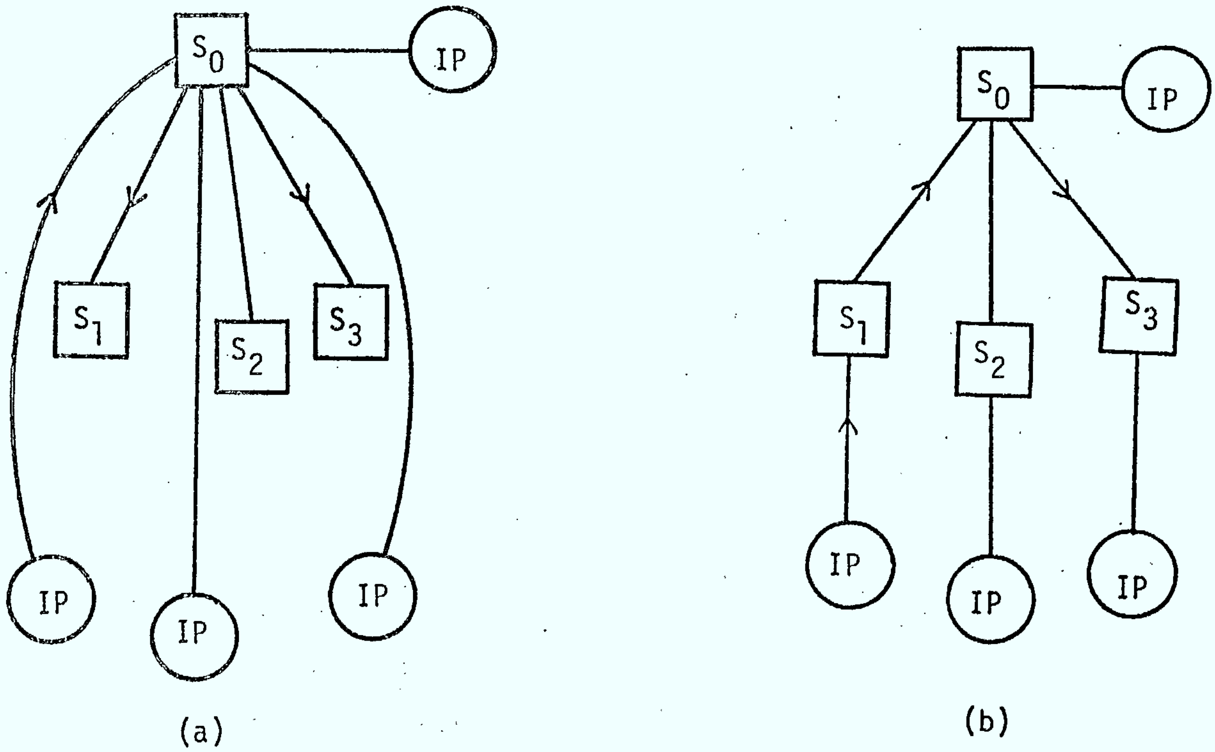


Figure 2.2 Direction of update requests in Group III Networks.

(a) All updates initiated at S_0 only.

(b) Update initiated at site S_1.

<u>Solution</u>: The file allocation problem for this group of videotex networks can be stated as follows:

"A set of videotex files F_i , i=1,2,...,n, is to be duplicated over the set of videotex information centres located at S_j , j=1,2,...,m. The main centre S_o will store all files and each regional centre S_j will store a subset of them. With the files being retrieved and updated in the way described above, we want to determine the subset of files to be stored at each of the regional centres so that the total storage and communication cost over the information server subsystem (ISS) will be minimal."

Note: For this problem, we are not concerned with the communication cost outside the ISS. For example, the cost of transmitting data from the IP's to the IUS's is not included in this formulation. It will, however, be studied in Chapter III.

This problem has been studied in a slightly more general context by Chang (CHAN75) for the case where the storage and communication costs are linear functions, and by Lam, et. al (LAMY80) for the case where these costs are step-functions. The main difference between their problem and ours lies in the file updating process rather than in the analytical method of solution. In their problem, update is usually reasonably dynamic and can be initialized by multiple users at various sites; whereas in our problem (essentially an information retrieval system), update of a file is usually less dynamic and is initialized by its IP at a specific site. As a result, statistics about update traffic can be collected more easily in our case.

In the following, as an illustration, we explain how to formulate Lam and Yu's problem in our context :

Analytic Formulation of File Allocation Problem for Group III Videotex Networks

We consider a distributed system consisting of a central computer located at site S_0 connected to a set of regional computers that are at sites $S_1, S_2, \ldots, S_m$. The overall system contains n files denoted by $F_1, F_2, \ldots, F_n$. A copy of each of these files is stored at the central computer S_0, while each regional computer S_j stores some subset of these files. The underlying problem is to determine which subset of files should be stored at each regional site. The solution is obtained in the context of minimizing storage costs and data traffic costs between S_0 and the regional sites. It is furthermore assumed that for each site S_j , the function $S=S(x)$ is known where $S(x)$ provides the cost of storing x units (bytes) of data at S_j and that the function $T=T(y)$ is known where $T(y)$ is the traffic cost of y units of data between site S_j and the central site S_0 .

We now examine the nature of the traffic between S_0 and site S_j . Two types of activity need to be separately considered.

a) Retrieval

Suppose a user connected to site S_j needs to retrieve data elements from file F_k . If this file is stored at S_j , then there is no traffic cost incurred. If, on the other hand, F_k is not stored at S_j then the request can only be serviced by accessing the copy of F_k which resides at S_0 and this results in a traffic cost on the S_0/S_j link. Clearly then, retrieval traffic costs are higher for those files not stored at S_j . (Note that there is no retrieval traffic on the S_0/S_j link due to a user at site S_i $(i \neq j)$ because a copy of all files exists at S_0 to which S_i is connected).

b) Update

Suppose a user at site S_j wishes to update data elements in file F_k . This necessarily (independent of whether or not a copy of F_k exists at S_j) results in traffic on the S_0/S_j link because S_0 contains a copy of F_k which must be updated. If a user at site S_i $(i \neq j)$ wishes to up-date F_k and a copy of F_k exists at S_j , then this also results in traffic on the S_0/S_j link. It therefore follows that update traffic is greater on the S_0/S_j link for those files that are stored at site S_j .

Thus, with respect to site S_j and file F_k , there are two traffic parameters u_k and v_k . The first characterizes the traffic on the S_0/S_j link if F_k is not stored at site S_j and the second characterizes the traffic if it is stored there. In addition, there is associated with F_k a third parameter, s_k, which is the storage space required to store F_k. This parameter is clearly not dependent on the remote site in question.

Let $d_k = 1$ if file F_k is not stored at site S_j and $d_k = 0$ if file F_k is stored at S_j, and let $D=(d_1, d_2, \ldots, d_n)$. The vector D therefore provides a characterization of which files are stored at site S_j. For any particular value of D the storage requirement at S_j is given by

$$x = x(D) = \sum_{i=1}^{n} s_i (1-d_i)$$

and the traffic on the S_0/S_j link is given by

$$y = y(D) = \sum_{i=1}^{n} \left[u_i d_i + v_i (1-d_i) \right]$$

Also

$$C = C(D) = S(x(D)) + T(y(D)) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

provides the total cost at site S_j of the file allocation characterized by D. The objective of our file allocation problem is then to choose D so as to minimize C.

A summary of the relevant notation is given below:

Notation	Explanation
S_0	Central computer site.
S_j, $j=1,2,\ldots,m$	Regional computer sites.
F_k, $k=1,2,\ldots,n$	Files to be allocated.
s_k, $k=1,2,\ldots,n$	Storage requirements (number of bytes) for file F_k.
u_k	Traffic on the S_0/S_k link if F_k is not stored at site S_j.
v_k	Traffic on the S_0/S_k link if F_k is stored at S_j.
$S(x)$	Cost of storing x units (bytes) of data at any site.
$T(y)$	Cost of y units of traffic between any two sites.
$D=(d_1,d_2,\ldots,d_n)$	Decision vector; $d_k = 1$ if file F_k is not stored at site S_j and $d_k = 0$ otherwise.
$C(D)=S(x(D))+T(y(D))$	Total cost of storing those files implicitly specified by D.

Equivalence to the Knapsack Problem

The knapsack problem in the mathematical programming literature has the following form: Let $a=(a_1,a_2,\ldots,a_n)$ and $b=(b_1,b_2,\ldots,b_n)$ be two given n-vectors of constants. Find the n-vector z which maximizes $\sum_{i=1}^{n} a_i z_i$ subject to the constraint that $\sum_{i=1}^{n} b_i z_i \leq M$ where, M is a given constant. Lam and Yu (LAMY80) have shown that, under certain assumptions about the functions S(x) and T(x) in equation (2), there is an equivalence between the file allocation problem and the knapsack problem. Specifically, if

i)

$$T(y) = \begin{cases} TM, & \text{if } y \leq \sum_{i=1}^{n} v_i + M \quad (M>0) \\ \infty & \text{otherwise} \end{cases}$$

ii) S(x) is monontonically increasing (i.e., $S(x_2) \geq S(x_1)$ for $x_2 > x_1$) and is finite for finite x, and

iii) D* is the solution to the associated file allocation problem,

then, z=D* is the solution to the knapsack problem associated with the coefficients $a_i = s_i$ and $b_i = u_i - v_i = h_i$ and vice-versa. (Note that the assumed form for T(y) implies that, if the traffic exceeds $\sum_{i=1}^{n} v_i + M$, then it cannot be handled; otherwise, the cost has a constant value of TM.)

The equivalence is established in the following way. First, observe that, for any D which yields a finite C(D), it must be true that $y(D) \leq \sum_{i=1}^{n} v_i + M$. Furthermore, C(D*) is finite because $C(D_0)$ is finite (where $D_0 = (0,0,\ldots 0)$) and by definition $C(D^*) \leq C(D_0)$. In other words

$$y(D^*) = \sum_{i=1}^{n} \left[u_i d_i^* + v_i (1-d_i^*) \right] = \sum_{i=1}^{n} h_i d_i^* + \sum_{i=1}^{n} v_i \leq \sum_{i=1}^{n} v_i + M$$

which in turn means that:

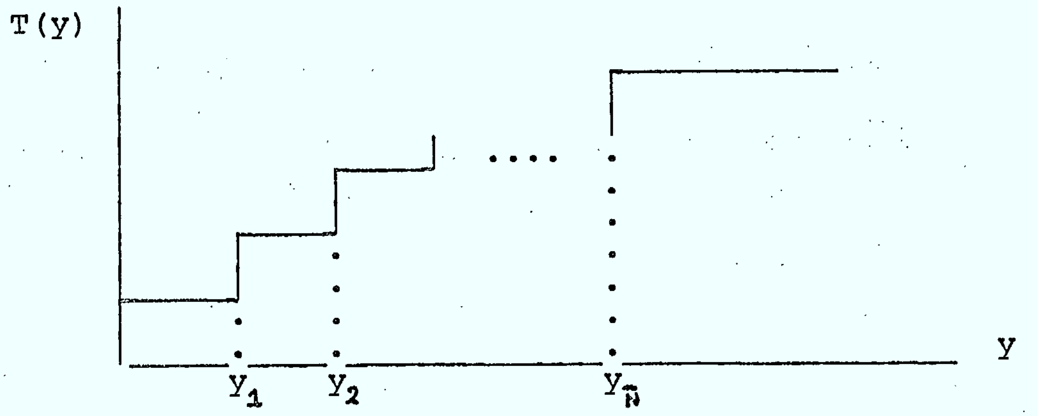
$$\sum_{i=1}^{n} h_i d_i^* \leq M$$

i.e., D* is a feasible solution to the knapsack problem in question.

Now, because T(y) is constant, the minimization of C implies the minimization of S(x(D)) in (2). But because S(x) is monotonically increasing and $x = x(D) = \sum_{i=1}^{n} s_i - \sum_{i=1}^{n} s_i \cdot d_i$ (where $\sum_{i=1}^{n} s_i$ is a constant), it follows that the minimization of S implies the maximization of $\sum_{i=1}^{n} s_i d_i$. Thus D* is the solution to the instance of the knapsack problem under consideration.

It is worth noting at this point that if, in equation (2), $h_k = u_k - v_k \leq 0$, then it means that, from a traffic point of view, there is as much (or more) cost in placing file F_k at site S_k as there is in not placing it there. Since its placement at S_k would typically introduce a storage cost as well, it follows that the best strategy is not to place F_k at S_j ; i.e., set $d_k = 1$. This observation makes it possible to reduce the dimensionality of the problem.

The above equivalence result can be utilized in the reasonably realistic context where both the storage cost function and the traffic cost function are step-functions. A typical form for T(y) is:



Here a discrete jump in traffic cost occurs when the traffic moves across each of the points in the finite set $y_1, y_2, \ldots, y_{\bar{N}}$. The solution to the file allocation problem in this framework can be obtained by solving a sequense of $\bar{N}$ knapsack problems. A description of the algorithmic procedure is given in Figure 2.3.

. For $k=1,2,...,n$, determine s_k, the storage required for file F_k.

. Let $N = \{1,2,...,n\}$.

. Repeat for $j=1,2,...,m$ (i.e., for each remote site)

. For $k=1,2,...,n$, determine values for u_k, the retrieval or update traffic on the S_0/S_j link, if file F_k is not stored at S_j.

. For $k=1,2,...,n$, determine values for v_k, the retrieval or update traffic on the S_0/S_j link, if file F_k is stored at S_j and let $z_1 = \sum_{i=1}^{n} v_i$.

. Determine the set I^* where

$$I^* = \{ i \in N: h_i = u_i - v_i \leq 0 \}$$

and let $\hat{I} = N - I^*$.

. Repeat for $\alpha = 1,2,...,\bar{N}$

. Solve the knapsack problem:

$$\max \left(\sum_{i \in \hat{I}} s_i d_i \right)$$

$$\text{subject to: } \sum_{i \in \hat{I}} h_i d_i \leq (Y_\alpha - z_1 - \sum_{i \in I^*} h_i).$$

. Denote the solution by D_α^*.

. Increment α.

. Find $\min \{ C(D_\alpha^*): \alpha = 1,2,...,\bar{N} \}$.

. Increment j

Figure 2.3 Solution Procedure for the File Allocation Problem for Group III Videotex Networks.

The procedure in Figure 2.3 is stated in terms of obtaining a true solution to each of the sequence of knapsack problems. While algorithms are available (Horowitz and Sahni), the problem is NP-hard. Approximate procedures, however, are available and can be used in a large network where computational costs may become prohibitive. Depending on the level of approximation selected such alternate procedures can have a high probability of providing a good match to the true solution.

As an illustration of the procedure in Figure 2.3, a representative numerical example has been treated.

Example We assume 6 sites (m=6) and 10 files (n=10). The storage cost function, S(z), is shown in Fig. 2.4 and the traffic cost function, T(y), is shown in Fig. 2.5. The storage requirements for each of the files are summarized in Fig. 2.6(a) and the u/v vectors for each of the sites are given in Fig. 2.6(b). The results obtained from the program based on the procedure of Fig. 2.3 are given in Fig. 2.7.

Listings of the various Fortran subprograms which were used in the computation are attached in Appendix B.

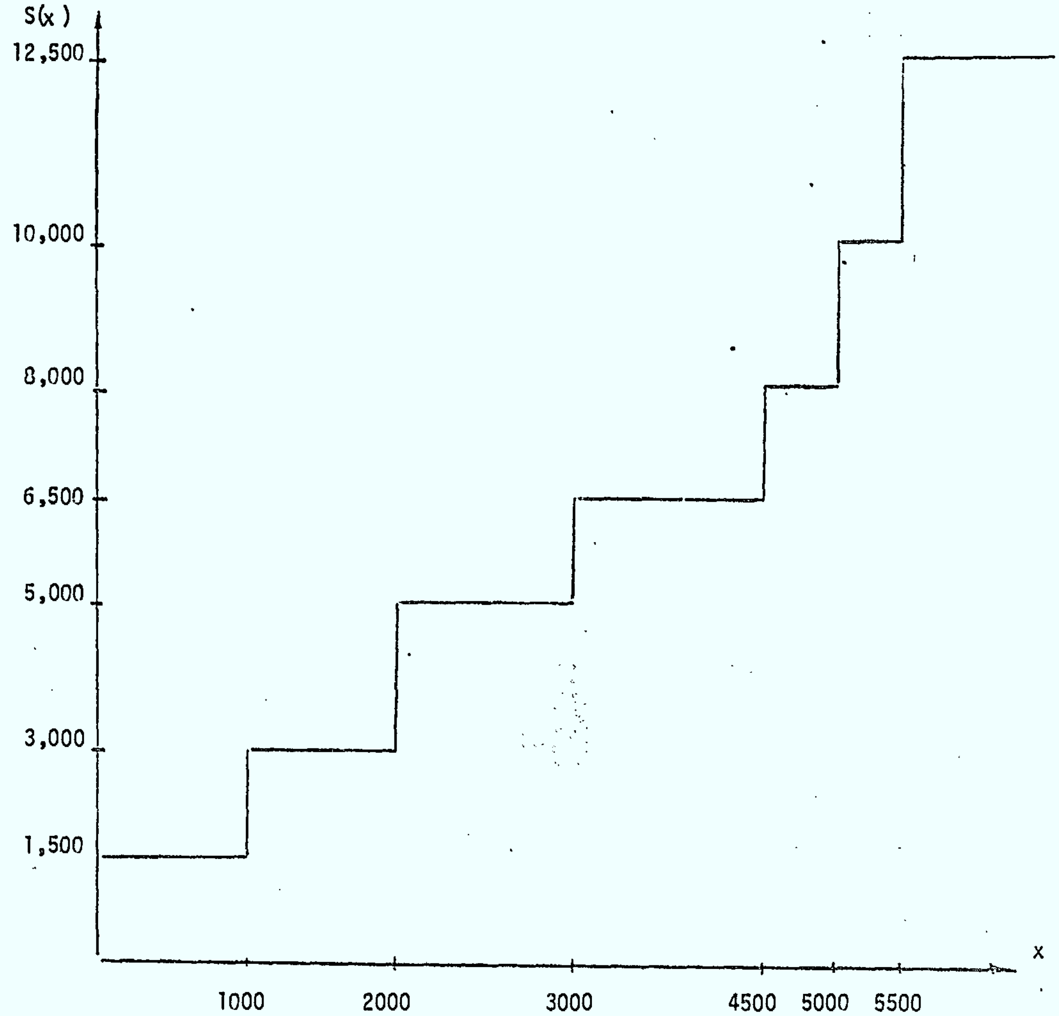


Figure 2.4

Storage Cost Function for the Example Problem

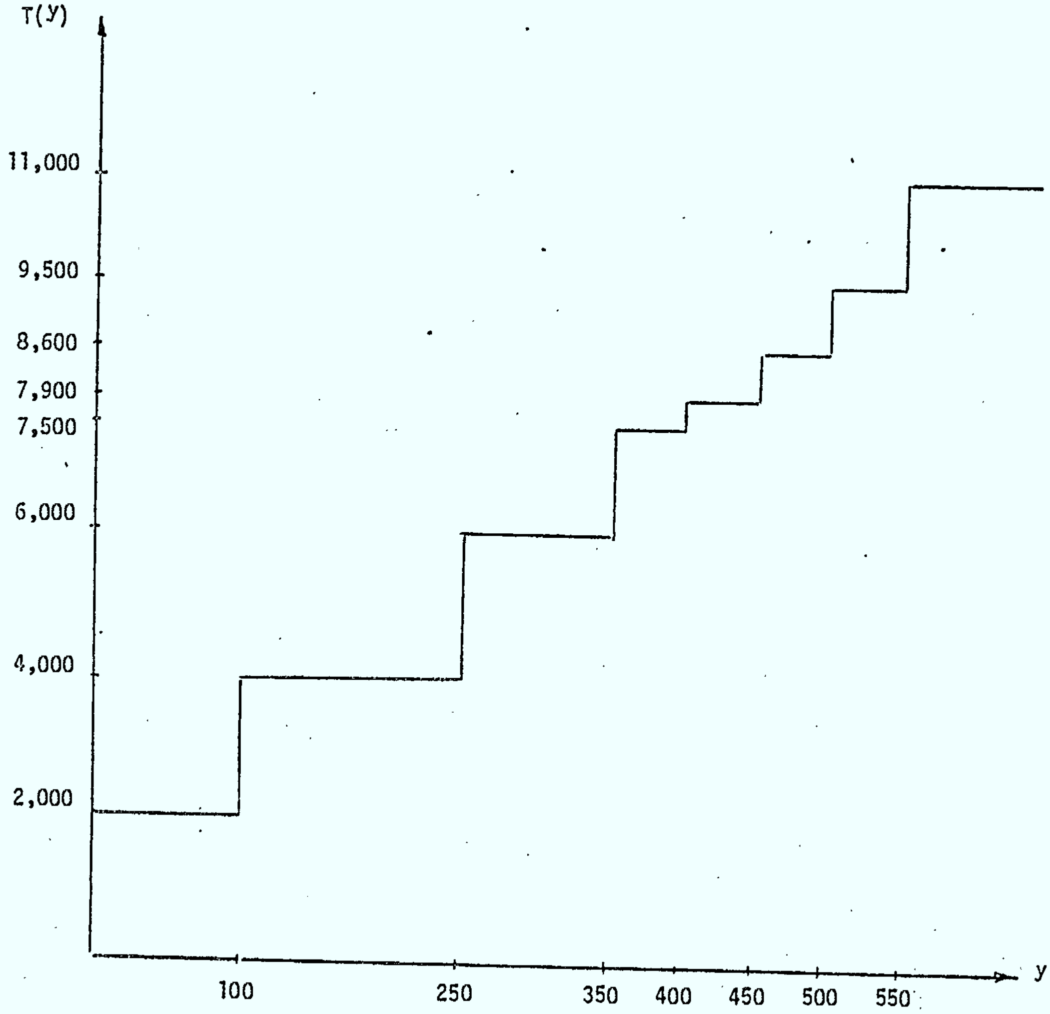


Figure 2.5

Traffic Cost Function for the Example Problem

File	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
Storage	100	200	300	400	500	600	700	800	900	1000

Figure 2.6(a)

Storage Requirements for the Files

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
S_1	70/10	60/20	50/30	40/40	30/50	20/60	10/70	100/80	90/90	10/100
S_2	10/10	100/20	90/30	80/40	70/50	60/60	50/70	40/80	30/90	10/100
S_3	50/10	40/20	30/30	20/40	10/50	100/60	90/70	80/80	30/90	10/100
S_4	30/10	20/20	10/30	100/40	90/50	80/60	70/70	60/80	30/80	10/100
S_5	60/10	50/20	40/30	30/40	20/50	10/60	100/70	50/80	30/90	10/100
S_6	40/10	30/20	20/30	10/40	100/50	90/60	80/70	50/80	30/90	10/100

Figure 2.6(b)

The u/v Vectors for the Sites

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
S_1	*	*	*					*		
S_2		*	*	*	*					
S_3	*	*				*	*			
S_4				*	*					
S_5	*									
S_6	*				*	*				

Figure 2.7

Optimal File Allocation Summary

(* indicates that the file is stored at the site)

Chapter III

ASSIGNMENT OF USERS AND INFORMATION PROVIDERS TO INFORMATION SERVERS

3.1 INTRODUCTION

Under our model of Chapter I, the Information Server Subsystem (ISS) plays the role of a communications message switcher within a videotex network. In Chapter II, the functional architecture which describes the internal relationship among the components of ISS, i.e., the information retrievers (IR), information update servers (IUS), and meta-service switchers (MSS), is applied the videotex file allocation problems. In this chapter, we turn our attention to some of the external relationships between the components of ISS and the other three subsystems (users U, information providers IP and meta-service providers MP). In fact, we investigate two problems : namely, the problem of assigning the users to the IR´s and the problem of assigning the IP´s to the IUS´s. The problem of assigning MP´s to the MSS´s is similar and thus will not be explicitly considered. In the following, we use the term ´server´ to mean either ´IR´ or ´IUS´ and the term ´ client´ to mean either ´user´ or ´IP´.

In practice, a server may service many clients located at different sites. The way in which the clients are assigned to the servers affects the total (communications, processing and/or storage) costs. Our objective is to minimize a certain combination of these costs, using analytical and approximation methods.

Notes

i) 'Assignment' here is a functional concept. Physical realization of an assignment is assumed to be available but is beyond the scope of study of this report.

ii) The minimization is from the standpoint of the entire system and is not for an individual client. A smaller total cost for the system may or may not result in lower service charges for some clients.

iii) For our study of the client assignment problems, it is assumed that the topology of the ISS and the locations of files have already been fixed. An integrated study of these problems is of extremely large scale.

In the next section, we first formulate a model for the minimization problems. We then apply this model to the different types of videotex networks as classified in Chapter I. Lastly, we consider some methods for simplifying the computations involved.

3.2 ASSIGNMENT OF USERS AND INFORMATION PROVIDERS

By the phrase ´assigning a client to a server´, we mean ´associating a client to a server for the purpose of physical connection´. Without loss of generality and just for the purpose of simplifying our problem formulation, we may assume that each client is assigned to one and only one server. More explanation follows.

In general, the data and services requested by a user may originate from several IR´s. On subscribing, a user is given the telephone numbers of one or several IR´s for connection purposes. Interactions with the other IR´s are switched through the connected IR. For our problem formulation, if a user is given the telephone numbers of more than one IR, we may either use just the primary number while ignoring the others or logically assume that there are several users.

Similarly, an IP may have to be connected to several IUS´s in order to update different files. However, we can, logically at least, partition such an IP into several IP´s so that each of them is serviced by only one IUS. Also, if there is more than one copy of the <u>same</u> file, the IP is connected to one of the IUS´s in order to directly update one of the copies. The other copies are indirectly updated through this IUS.

In the following, we first formulate the assignment problems as an analytical model and then apply the model to several types of videotex networks.

3.2.1 A model for assignment problems

We make the following assumptions for our model :

i) The functional architecture of the ISS is known. That is, we know whether each IR has a master database, whether an IUS services all or part of the IR's, etc.

ii) The locations of the files, users, IP's, MP's, IR's, IUS's and MS's are all known.

iii) Each user knows what data and meta-services (not necessarily their locations) he/she requests. Each information provider knows what data (and their locations) it provides. We shall refer to this as "client's demands".

iv) The communications and/or processing costs of servicing a client's demand by a server are known or estimable. How this is calculated depends on the problem. (See Section 3.2.2)

v) The number of clients each server can service is known. This can be estimated in terms of the hardware speeds, number of communications ports, etc., available at the server's computing facilities.

vi) Each client is assigned to one and only one server.

We shall use the following notation :

Notation	Explanation
m	Number of servers.
n	Number of clients.
c_{ij}	Total (communications and/or processing) cost of servicing the demands of the jth client by assigning it to the ith server.
b_i	Maximum number of clients the ith server can service.
x_{ij}	$x_{ij} = \begin{cases} 1 & \text{if the jth client is assigned to the ith server} \\ 0 & \text{otherwise.} \end{cases}$
$\sum_i$	scope of summation is from 1 to m.
$\sum_j$	scope of summation is from 1 to n.

Our minimization problem is then the following well known assignment problem in operations research:

$$\begin{aligned} \text{Minimize :} \quad & \sum_{i,j} c_{ij}\, x_{ij} \\ \text{subject to} \quad & \sum_i x_{ij} = 1 , && j=1,2,\ldots\ldots\ldots,n \\ & \sum_j x_{ij} \leq b_i , && i=1,2,\ldots\ldots\ldots,m \\ & x_{ij} = 0 \text{ or } 1 && i=1,2,\ldots\ldots\ldots,m; \\ & && j=1,2,\ldots\ldots\ldots,n. \end{aligned} \qquad (3)$$

There exist very efficient algorithms (DENN58) for solving (3).

3.2.2 Application to videotex networks

Based on the Minimization Model (3), we can now discuss the user and information provider assignment problems for the various types of networks as classified in Chapter I. Essentially, we explain how the cost parameters c_{ij} can be obtained in each case.

For Videotex Networks of Type 1

Since a network of this type has only one IR and one IUS, all users are assigned to the same IR and all IP's to the same IUS.

For Videotex Networks of Type 2.1

In this case, since each subnetwork has autonomy over its users and files, the users and IP's can only be assigned to the IR and IUS, respectively, of the subnetwork they subscribe to.

For Videotex Networks of Type 2.2

A network of this type has several information centres, each containing an IR and an IUS.

* A user may be assigned to any IR. If user U_j is assigned to IR_i , the parameter c_{ij} includes the communications, storage and processing costs for : (i) IR_i directly servicing U_j; and (ii) IR_i switching data and meta-services to and from other IR's.

* An IP must be assigned to the IUS whose associated IR contains those files belonging to this IP. Since each file

has only one copy in the entire network, such an assignment is unique.

For Videotex Networks of Type 3

A network of this type has one IUS and several IR's.

* A user U_j may be assigned to any IR_i. If IR_i contains a master database, the parameter c_{ij} includes only the communications and/or processing cost between IR_i and U_j. If IR_i contains only a portion of the entire database, c_{ij} should also include the cost of switching data and services between IR_i and the IR which contains the master database.

* Since there is only one IUS, all IP's should be assigned to it.

For Videotex Networks of Type 4

A network of this type has several IR's and IUS's.

* A user U_j may be assigned to any IR_i. Since all IR's contain a master database, the parameter c_{ij} includes only the communications and/or processing cost between IR_i and U_j ; i.e., no switching cost for data.

* An IP_j may be assigned to any IUS_i . The parameter c_{ij} includes the communications and/or processing costs between IUS_i and IP_j and the cost of updating (from IUS_i) the copies stored at all the IR's.

For Videotex Networks of Type 5

* For this type of networks, user assignment is the same as for Type 3.

* As for information provider assignment, it is the same as Type 4 except that a file to be updated may have copies at only some of the IR´s. (Computationally, there is no difference.)

3.2.3 Some Computational Aspects of the Model

The biggest obstacle when applying Model (3) is obviously the size of the problem, even though there exist reasonably efficient algorithms for its solution. In practice it is easier to handle the information provider assignment problem as there are usually not many IP´s. However, a videotex system, especially those open to the public, may have millions of users. Solving (3) for problems of such sizes is computationally intractible. The following observations may help reduce the sizes of the problems.

(i) Excluding some users from consideration

For some users, the selection of an IR for assigment purposes is quite obvious. For example, if a dominantly large portion of the files requested by a user U are stored at IR , U should definitely be assigned to IR . The following rough rule may be used: Let p be the cost imposed on U for retrieving data directly from IR and q be the cost for retrieving data from the other IR´s through IR . If p >> q, U should be assigned to IR and hence excluded from further consideration.

(ii) Group assignments by regional partitioning

In practice, one may assign the clients by groups instead of individually. For example, a district can be partitioned into several smaller regions according to the geographical environment, population, or management convenience of the system. All clients within a region will be assigned to the same server and thus conceptually become a single client when appling our Model. The total or average cost of servicing a region by a server can be easily estimated. There is an upper bound on the number of regions a server can service. A detailed explanation of this approximation method and its analytical foundations are given in (CHEU74).

Chapter IV

ISSUES OF DATABASE INTERFACES

4.1 INTRODUCTION

For most of the existing videotex systems, the term ´database´ means in fact a collection of conventional files whose records contain videotex information pages --hereafter we call them ´conventional files´. In this chapter, however, a database is a reservoir of data which are defined, organized and manipulated according to a data model, such as the relational model, the hierarchical model, the CODASYL model, etc.--hereafter we call them ´databases´.

The capability of a videotex system to access databases greatly enhances the system´s flexibility and sophistication. If access is limited to conventional files, the information contained in each page is pre-determined. Users have to search through numbers of pages in order to get what they want, and very often in vain. In the case of databases, users can specify the scopes and the conditions of the information they desire by means of a query language. The system then retrieves various segments of relevant information from different portions of the database and composes them dynamically into one or several pages for subsequent display at the terminals.

In a videotex environment, many issues about the design of such databases arise. Of particular concern are the following :

1. Methods for physically accessing the distributed databases.
2. Methods for allocating the distributed data and management tasks.
3. Security problems.
4. Issues of interfaces with videotex system users and information providers.

Among these 4 issues, analytical works on the first 3 are abundant. But, they are not "directly" related to videotex systems. In other words, they exist by themselves, though a videotex environment may make the problems more complicated. Thus, in accordance with our Guideline for Research Scope (see Introduction and Executive Summary), we shall simply collect their reference resources in Bibliographies I, II and III, respectively.

In this chapter, we consider the above Issue #4, namely, interfaces between the databases and the clients of a videotex system.

4.2 INTERFACES BETWEEN DATABASES AND CLIENTS

In general, a user interfaces with a database through two phases :

i) Database definition This phase includes defining the database schemas and subschemas, etc. It is known to be the most difficult job in the administration of databases and is usually handled only by experts called database adminstrators (DBA).

ii) Database manipulation This phase includes data creation, retrieval and update. Many data manipulation languages have been designed for the different well-known data models, such as relational, hierarchical, CODASYL, etc. From the users' point of view, those languages for relational databases are obviously the most powerful and easy to understand.

In many cases, a database is primarily utilized by users of a non-videotex database management system. Providing accesses to videotex system users is only one of its secondary functions. Such a database is called an 'external' or 'third-party' database by videotex system users. Thus, in designing its interface with the videotex clients (i.e., users or information providers), we should take into consideration the fact that the non-videotex users' methods for processing the databases should not be altered.

Our consideration on these interfaces is within the environment of a 'pure' videotex system, i.e., a videotex system possessing the following two characteristics which greatly influence its design :

(A) The terminals have very simple keypads and processors, small memory size, etc.

Because of these restrictions, their processing capabilities are very limited and interface procedures between clients and databases have to be very simple. We do not include integrated intelligent terminals in our consideration, because the design is then essentially for an integrated system, with the videotex technology simply fading into another format of data presentation. Thus, those techniques which require a greater processing power, such as downloading telesoftware into an intelligent terminal for handling user interfaces and imbedding a database interface in a high level language, are not within the scope of our study.

(B) The users and information providers have separate groups of operations.

The two groups of clients, the users (U) and the information providers (IP), interface with the databases by using the retrieval (output only) update (input only) operations, respectively. Such separation of operations may offset part of the disadvantages due to Property (A) mentioned above. More explanation follows.

(i) User interface

In a ´pure´ videotex system, a user is involved only with the retrieval of information, whether from conventional files or from databases. The creation and updating of data are not the users´ responsibility. In the case of databases, this fact has the following implications :

a) The users are released of the burden of defining and modifying databases. These processes are usually too complex for most of the non-technical videotex system users and should be handled by either the IP´s or the database administrators who are supposed to be more technically knowledgeable.

b) The simplicity of a videotex terminal greatly limits the data manipulation capability of a user language. However, this is compensated by a reduction in its total number of operators, because only retrieval operations are now required.

c) With slight extensions, the simple menu-lookup and indexing techniques used frequently for conventional file searching in current videotex systems can be utilized to implement quite powerful retrieval languages for databases. Examples illustrating how to do this in the cases of a relational database and a hierarchical database will be shown in the next section.

(ii) Information provider interface

If several copies of a database can be updated by different sources, one of the difficult system management problems is to maintain the consistence of these copies. In a videotex environment, however, the difficulty of maintaining data consistency may be alleviated, because, according to the current management practice of videotex systems, one and only one IP is responsible for and has sole control over the update of all copies of an information page. (Note: According to our model, an IP does not physically do the update. It sends the data to an IUS where the physical update is done. But, conceptually, the IP is reponsible for the update since it initiates the process.) More details are given in each of the following two cases where data inconsistence arises :

Case 1 Because of communications delay, two consecutive updates for the same file may reach its duplicate copies (located at different sites) in different chronological orders, resulting in permanent inconsistence of these copies. In a videotex environment, since update of all copies of a file is initiated by the same IP, the IP can easily assure their consistence by serializing the updates in the same order for each site. One way to do this is to wait until acknowledgements for one update have come back from all relevant sites before issuing the next update. (Note : This is not so simple if there is more than one source which are responsible for update.)

Case 2 Because of communications delay, an update issued from an IP may reach two different relevant sites at two different moments. Such inconsistence is temporary for the database, but may end up with the users getting two different versions of the data, even if they retrieve at the same instant of time. Such inconsistence can be avoided by using a simple locking scheme.

4.3 QUERY LANGUAGES FOR DATABASES IN A VIDEOTEX SYSTEM

A query language consists of a set of operations for information manipulation. From the users' point of view, query languages can be roughly classified into the following four groups in a decreasing degree of technical complexities:

Group 1 A language of this group is the most advanced. It includes all the facilities for defining database schemas and subschemas and for manipulating data in a highly sophisticated manner. It is usually imbedded in a high level programming language such as COBOL, FORTRAN, etc. An example is the DL/I of IBM's IMS. This group of query languages are definitely too complicated for non-techinical users. Their implementation at a 'pure' videotext terminal is either infeasible or extremely inefficient.

Group 2 A language of this group is usually a user-friendly data manipulation language of a certain data model (mostly relational). They can be easily learned by most users. Included in this group are SQL and SQUARE (REIS75, REIS81 in Bibliography V), QBE (REIS81, THOM75, ZLOO75 in Bibliography V), etc. However, in their existing forms, these languages cannot be implemented ´directly´ for ´pure´ videotex terminals, because these terminals lack the capability of inputting alphabetical character strings or special functions required in these languages.

Group 3 This group includes those languages of either Group 1 or Group 2, but implemented in an environment where only simple menu-lookup types of searching techniques and numeric key inputs are used. Though at least two languages of this sort (DBM82, RMS83 in Bibliography V) exist for the APPLE microcomputers, we do not know of any of them designed specially for videotex systems.

Group 4 This group includes the menu-lookup searching techniques for conventional files, as used in most of the existing videotex systems.

Since we limit our research to ´pure´ videotex terminals (as mentioned in the Introduction and Executive Summary), languages of Group 1 and 2 are not within our scope of study. Group 4 does not concern databases. In the following, we shall first illustrate by two examples how to execute a language of Group 3 in a videotex environment (Section 4.3.1) and then mention some activites of behavioural research on these languages (Section 4.3.2).

4.3.1 Execution of database manipulation languages by menu-lookup techniques

Even with limited capabilities, 'pure' videotex terminals can still implement many of the well known database query languages in a manner familiar to the videotex system users, i.e., by using the techniques of menu-lookups, indexing, etc. The following two examples illustrate the concepts of this approach without giving the details.

Execution of a relational algebra

A relational database consists of a set of table-like relations. A relational algebra is a set of operations, such as projections, restrictions, join, etc., on the relations. In a videotex environment, these operations can be executed as follows :

All operations begin with the display of the following page of information :

Operators :

1. Difference ; 2. Intersection ;
3. Join ; 4. Projection ;
5. Restriction ; 6. Union ;

......

Relations :

1. R1(A1, A2,.....);

2. R2(B1, B2,.....);

......

One way to project relation R1 onto attributes A2 and A4, for example, is to enter the sequence (4..1.2.4) through the keypad. (See the Note below.) In this sequence, the digit (here 4) in front of the two dots '..' refers to the operator (here Projection) in the displayed list. Its operand is indicated by the subsequence (1.2.4), in which the first digit indicates the relation (here R1) in the dispayed list of relations, and the remaining digits indicate the positions of the attributes (here the 2nd and the 4th) within that relation.

Similarly, a join operation of relations R1 and R2 over their attributes A1 and B2 can be executed by entering the sequence (3..1.1..2.2) on the keypad.

Note : Other symbols such as a comma ',', if available in the keypad, may be used to replace the double dots '..'. Some keypads (e.g., Vista's keypad), however, do not have the keys for these special symbols.

Execution of IBM's DL/I

DL/I is the data definition and manipulation language of IBM's IMS (Information Management System) system for implementing (essentially) hierarchical databases. Briefly, each database within IMS is a hierarchy of segments (i.e., records). DL/I consists of a set of operations such as GU (get unique), GN (get next), GNP (get next within parent), etc.

Execution of any operation begins with the display on the terminal the following page of information :

<u>Operators</u> :

1. Get Unique; 2. Get Next;

3. Get Next Within Parent;

<u>Databases</u> :

0. SO(A1, A2,.....)

1. S1(B1, B2,.....)

1.1 S1.1(C1, C2,.....)

2. S2(D1, D2,.....)

2.1 S2.1(E1, E2,.....)

2.2 S2.2(F1, F2,.....)

To execute the operation GN S1.1 (get the next segment occurrence of segment type S1.1), for example, a user enters the sequence 2..1.1 through the keypad.

4.4 BEHAVIOURAL RESEARCH ON QUERY LANGUAGES

Users' behaviour in response to executing a query language is one of the important criteria for evaluating how successful the language is. Behavioural researchers have recently done quite a lot of studies on these languages. Language characteristics such as ease-of-use, procedurality, data-retrieval-power, error-sensitivity, etc., are investigated using human factors methodology. Since the methodology used is essentially not analytical and thus is not within our scope of research, we simply mention in the following some of the main results without giving the details. A list of references are given in Bibliography V.

For languages of Group 1, Lochowsky (LOCH77,78 in Bibliography V) evaluated the user performance for several query languages of the relational, hierarchical and CODASYL data models. These languages are implemented in APL (EDBS75). For Group 2, QBE was evaluated by Thomas (THOM75 in Bibliography V) and SQL (known earlier as SEQUEL) by Reisner (REIS75 in Bibliography V). An excellent study on evaluating the performances of several query languages of Group 2 was made by Reisner (REIS81 in Bibliography V). The study reviews the methods for evaluating several query languages (e.g., QBE, SQL, SQUARE, TABLET, etc.), compares their results, and proposes models for evaluation. As for languages of Group 4, user behaviour of all existing videotex systems is measured during their field trials. In particular, Lee, et. al. (LATR81, LEE80, WHAL80 in Bibliography V), Phillips (PHIL81 in Bibliography V), and Mills (MILL81 in Bibliography V) have done behavioural research on videotex tree indices. Unfortunately, only some fragmentary statistical results of these evaluations are reported in the publicly available literature.

Chapter V

ISSUES OF DATA ENCRYPTION

5.1 INTRODUCTION

The security of a modern cryptosystem depends on several factors: the strength of its methods for key distribution and message authentication and the robustness of its encryption/decryption algorithms.

There exist many encryption-based techniques for the distribution of conventional or public keys and for the authentication of the participants' identities in a general communication network environment. Bibliography IV contains a long list of articles about these techniques. Since a videotex network is just a particular type of communication network, these techniques should, theoretically at least, also be applicable. However, in practice, the following four characteristics of a videotex system may affect the design and/or implementation of these algorithms:

i) A 'pure' videotex terminal is very simple

A videotex terminal has very limited storage space and processing power. Its keypad has only a few numeric and functional keys.

ii) Most of the users are not technical people

They get used only to very simple interactive processing techniques, such as menu-lookup types of searching, etc.

iii) A 'pure' videotex system is essentially an information retrieval system

Most of the data move in one direction, from the information centre to the user. There are no interactions among the users.

iv) Videotex data usually have special patterns

For example, Telidon data are coded in special formats of the Picture Description Instructions. These patterns are usually readily avaibable.

Because of the first three features mentioned above, implemention of encryption-based techniques in a videotex network is possible only if there are both hardware and software adjustments in the system. On the hardware side, some recent developments are moving in this direction. Two LSI chips, one for the DES (NBS's Data Encryption Standard) Algorithm (DATA82) and another for the MIT Algorithm (RIVE78) are being designed. Their speeds, however, may still be too slow for application in a real-time environment such as videotex networks. We do not know of any hardware developments for the purpose of key distribution or message authentication. As for the software aspects, adjustments will be similar to the interface techniques for database query languages as described in Chapter IV.

5.2 ROBUSTNESS OF ENCRYPTION DECRYPTION ALGORITHMS

The robustness of an encryption/decryption algorithm is related to its protective capability against cryptanalytic attacks.

In classical cryptography, the aim of a cryptanalyst is to discover the mathematical transformations for encryption. For modern cryptosystems, since the mathematical transformations are usually publicly known, the aim is either to discover the keys or to recover the plaintexts directly. Three approaches are commonly used for cryptanalytic attacks: ciphertext only attacks, known plaintext attacks and chosen plaintext attacks.

Surprisingly, though modern cryptography provides an elegant approach for building cryptosystems, there exist only a few 'strong' encryption algorithms, e.g., the DES Algorithm and, MIT Algorithm. On the other hand, there does not seem to be many analytical methods for 'proving' the strength of these algorithms. In the following, we shall consider such an analytic approach.

What we are interested in concerns the functional dependence between the plaintext and ciphertext (for a fixed key) and between the key and the ciphertext (for a fixed plaintext). If the attacker finds any such functional dependences, it will be much easier to break the system. This is specially true when the attacker knows that the plaintext is composed of data structured in special patterns, as in the case of most videotex data.

In order to test the independence of the data mentioned above, S.M. Wong has used the Chi-square test method and has done some computational experiments based on the DES Algorithm. (The details are given in Appendix C.) Though her method and results are not specifically for videotex data (partly because of inavaibability of the videotex hardware and data), we believe that her approach is a useful analytical method for testing the robustness of an encryption algorithm.

Appendix A

CONFIGURATIONS OF MAJOR VIDEOTEX SYSTEMS

This appendix consists of the configurations of several major videotex systems. They illustrate the different types of videotex networks under the classification scheme based on the model proposed in Chapter I.

Figure A.1 The first stage of Bell Canada's Vista System.

Figure A.2 Germany's current Bildschirmtext System.

Figure A.3 The early stage of Britain's Prestel System.

Figure A.4 The current stage of France's Antiope System.

Figure A.5 The current stage of Britain's Prestel System.

Figure A.6 Germany's new Bildschirmtext System.

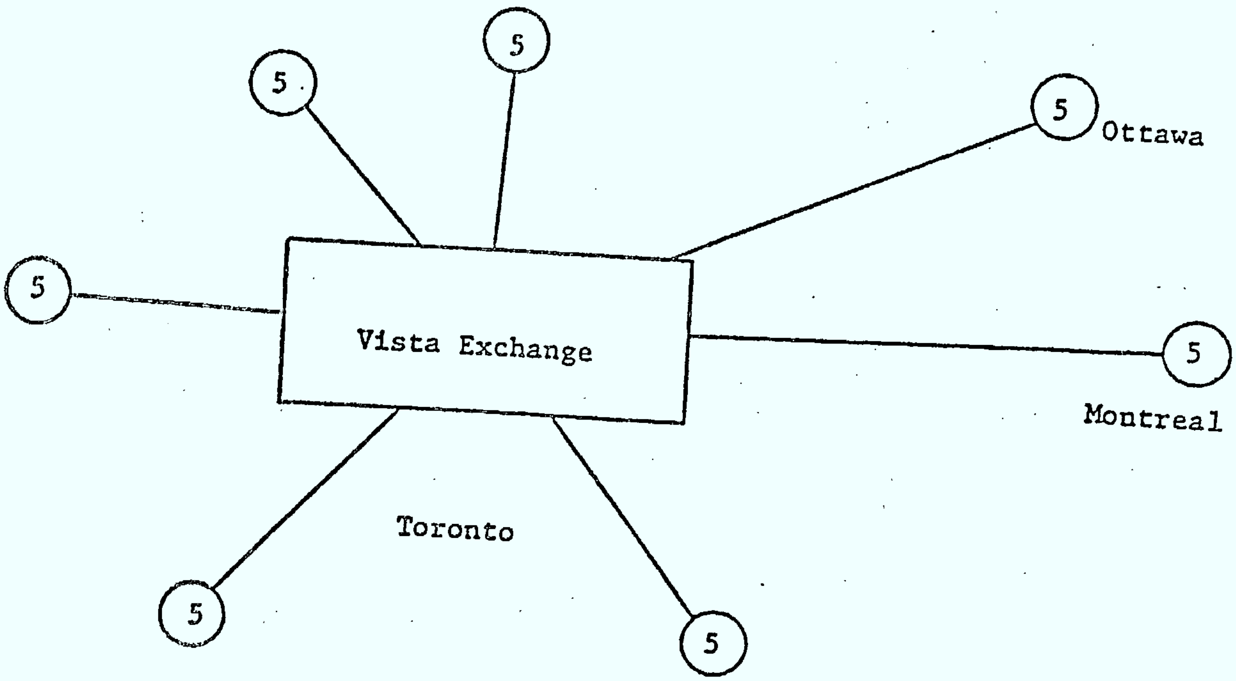
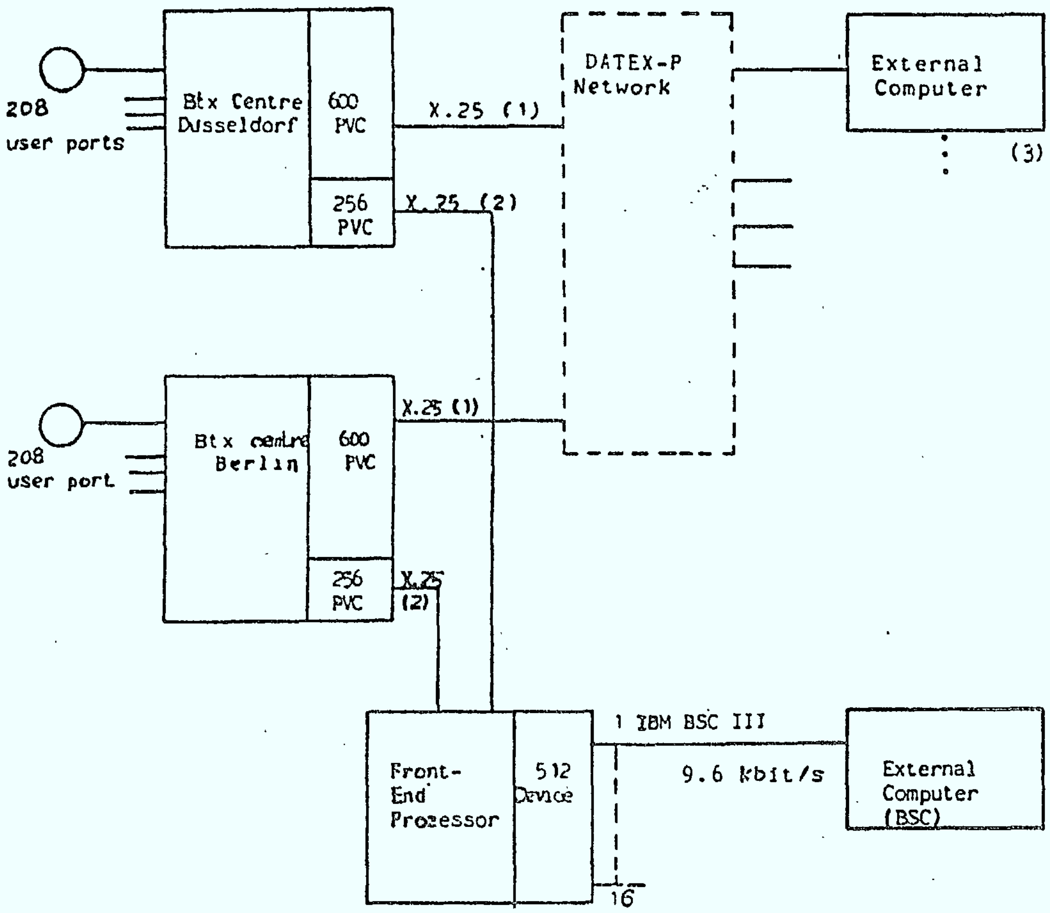


Figure A.1 The first stage of Bell Canada's Vista System.
(extracted from [MANN82])



(1) 5 links each 9.6 kbit/s
(2) 3 links each 9.6 kbit/s
(3) number of ext. computers only restricted
by total number of PVC:

Figure A.2 Germany's current Bildschirmtext System.
(extracted from [GRIE82])

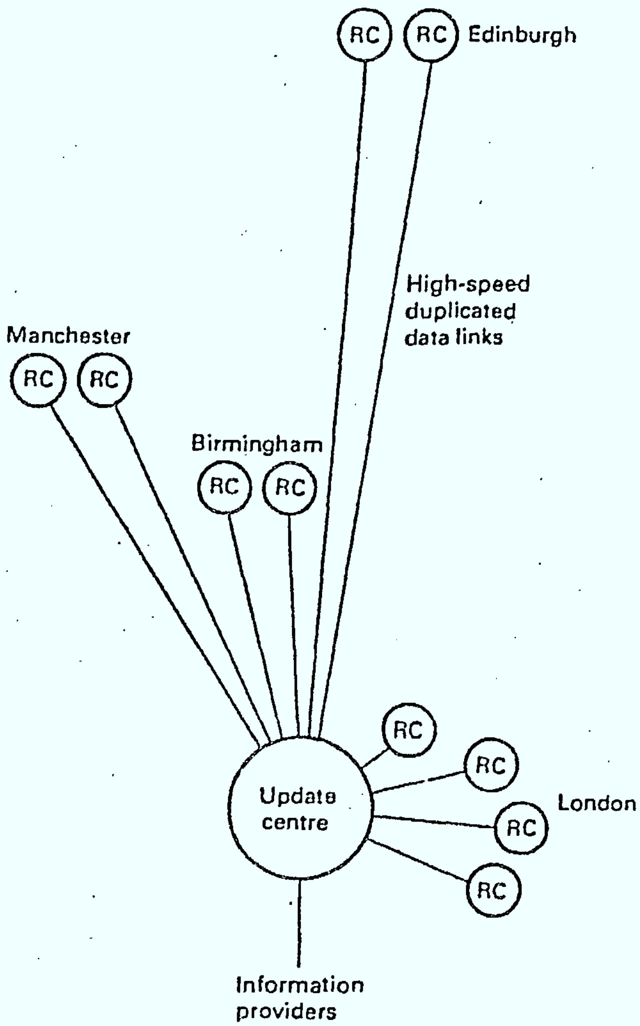


Figure A.3 The early stage of Britain's Prestel System.
(extracted from [CLAR81])

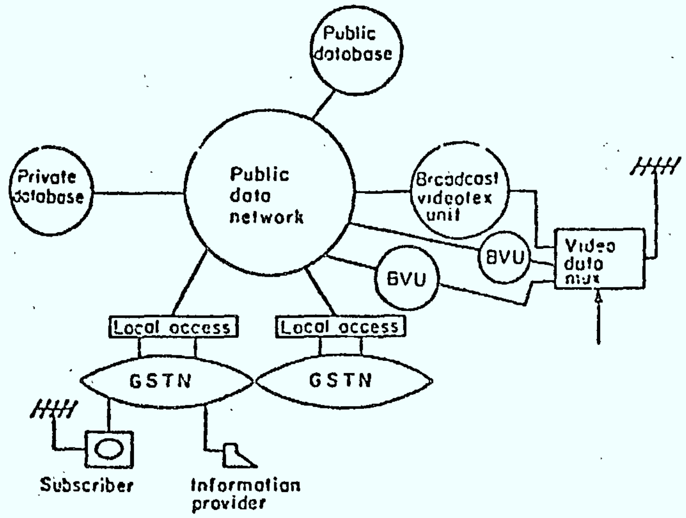


Figure A.4 The current stage of France's Antiope System.
(extracted from [MART79])

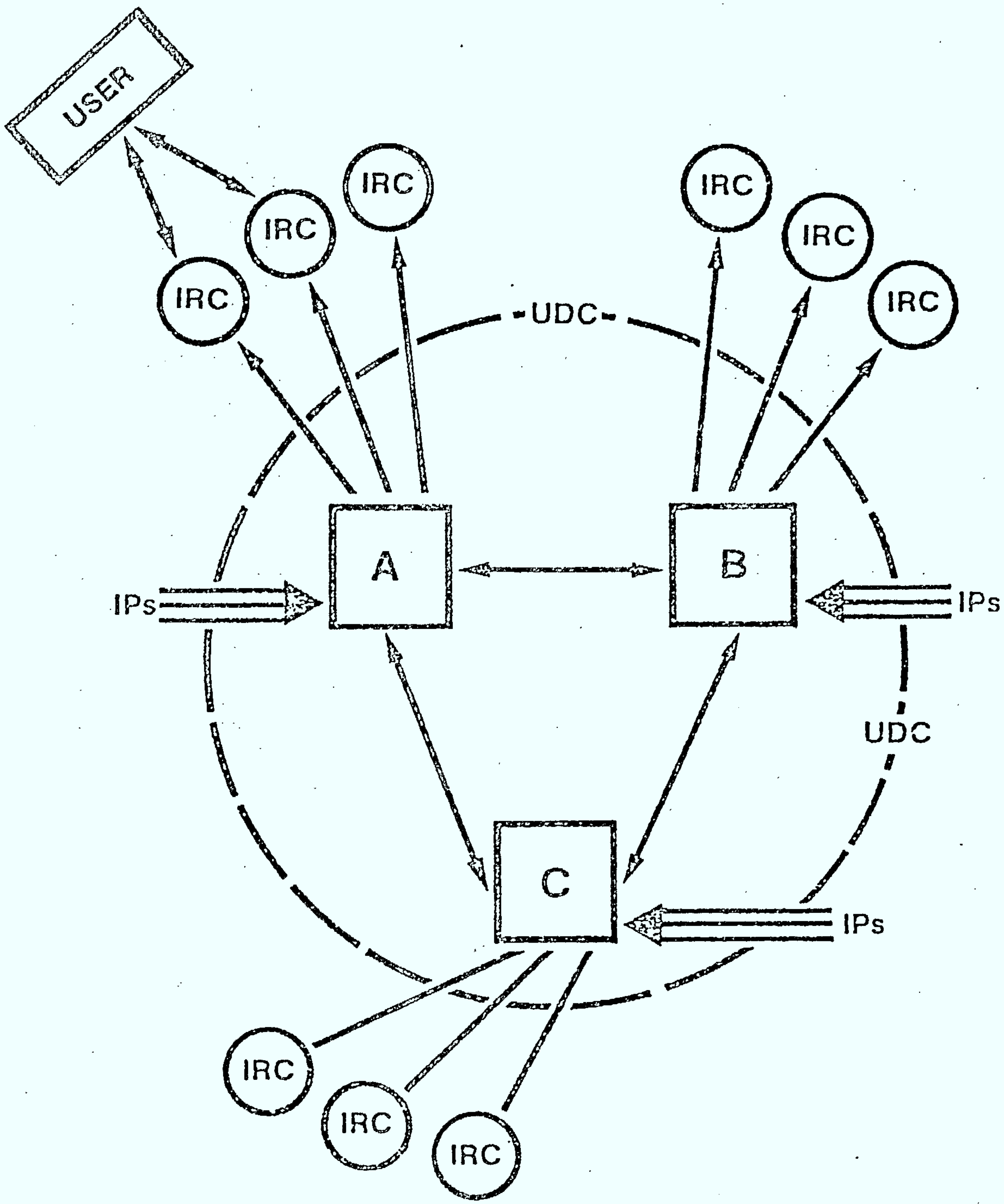


Figure A.5 The current stage of Britain's Prestel System.
(extracted from [TROU80])

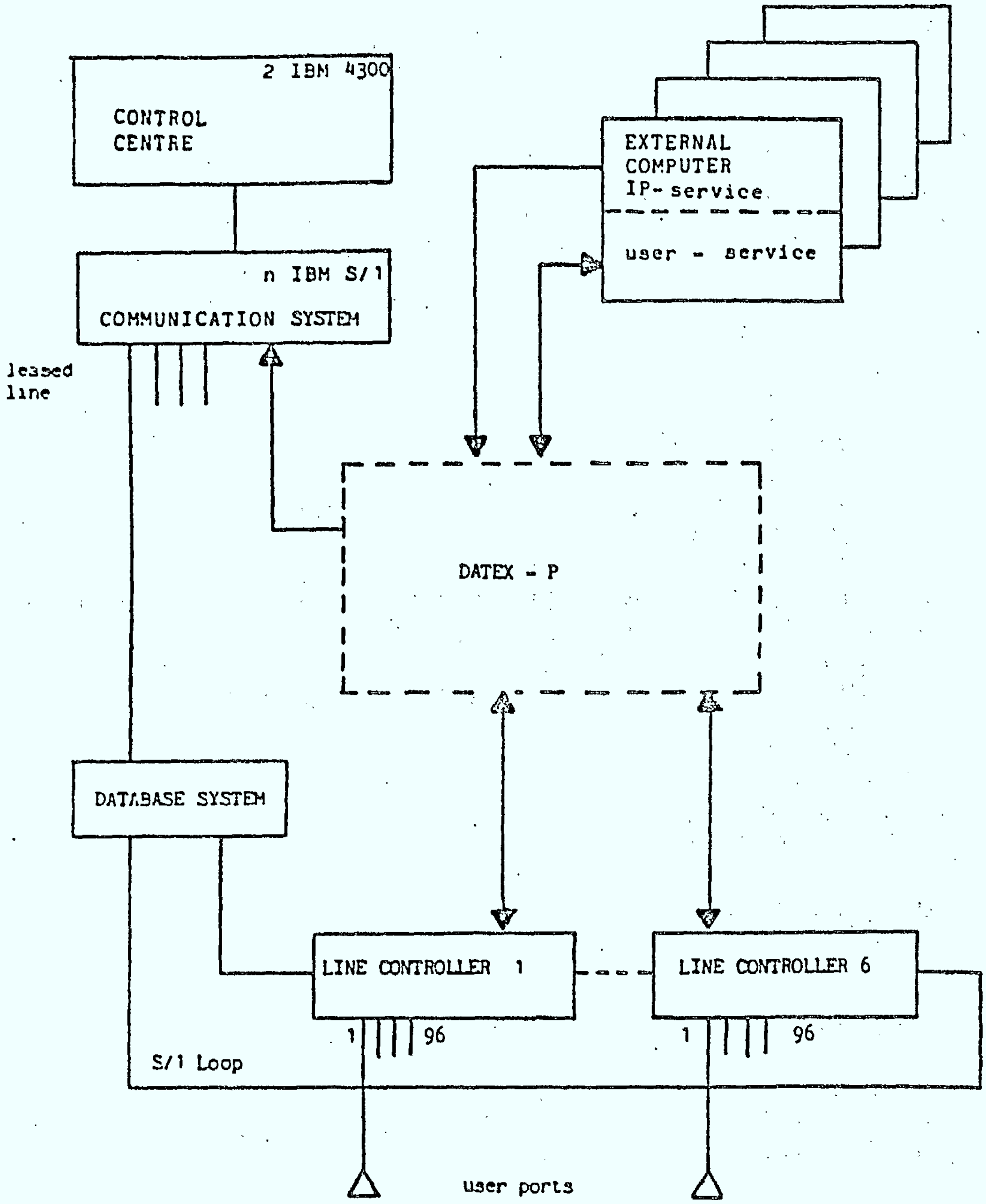


Figure A.6 Germany's new Bildschirmtext System.
(extracted from [GRIE82])

Appendix B

PROGRAM FOR FILE ALLOCATION PROBLEMS

```
      REAL M,FW,FP,FS,FT
      REAL PFTIN(100),WGTIN(100),OCOST,OS,OT,OTCOST,OSCOST,COST
      REAL PROFIT(100),WEIGHT(100),PFTWGT(100)
      REAL S(100),U(100),V(100),H(100),TX(10),TY(10),SX(10),SY(10)
      REAL TEMP,Z1,Z2,Z3,Z4
      INTEGER N,I,J,MAXT,MAXS,NBAR,SITE
      INTEGER*2 D(100),Y(100),X(100),DPTR(100),PTR(100),OX(100),Z(100)
C     M, MAXIMUM WEIGHT/TRAFFIC FOR KNAPSACK
C     FW, FINAL WEIGHT/TRAFFIC FROM KNAPSACK - NOT USED
C     FP, FINAL PROFIT/STORAGE FROM KNAPSACK - NOT USED
C     FS, FINAL STORAGE VALUE
C     FT, FINAL TRAFFIC VALUE
C     PFTIN, SUBSET OF TRAFFIC H VECTOR; VALUES GT 0
C     WGTIN, CORRESPONDING STORAGE VALUES
C     OCOST, OPTIMAL TOTAL COST OF A SITE
C     OS, OPTIMAL STORAGE VALUE OF A SITE
C     OT, OPTIMAL TRAFFIC VALUE OF A SITE
C     OTCOST, OPTIMAL TRAFFIC COST OF A SITE
C     OSCOST, OPTIMAL STORAGE COST OF A SITE
C
      DATA PROFIT/100*0/,WEIGHT/100*0/,PFTWGT/100*0/
C
C     PROFIT, SORTED PFTIN ;  SORT IS SUCH THAT P/W RATIO
C     WEIGHT, SORTED WGTIN ;  IS ASCENDING;NECESSARY FOR KNAPSACK
C     PFTWGT, PROFIT/WEIGHT RATIO ARRAY
C
      DATA V/10,20,30,40,50,60,70,80,90,100,90*0/,Z1/550/,DPTR/100*0/
      DATA S/100,200,300,400,500,600,700,800,900,1000,90*0/,Z2/5500/
      DATA D/100*0/,X/100*0/,Y/100*0/,Z/100*0/
C     V, TRAFFIC COST FOR STORING A FILE
C     S, STORAGE COST FOR STORING A FILE
C     Z1, TOTAL OF V VECTOR
C     Z2, TOTAL OF S VECTOR
C     DPTR, POINTER FOR H VECTOR TO MAP INTO ORIGINAL U,V VECTORS
C     D,X,Y,Z - BINARY VECTORS TO DENOTE SELECTION OF A FILE
C
      DATA TX/100,250,350,400,450,500,550,3*0/,MAXT/7/
      DATA TY/2000,4000,6000,7500,7900,8600,9500,3*0/
      DATA SX/1000,2000,3200,4500,5000,5500,4*0/,MAXS/6/
      DATA SY/1500,3000,5000,6500,8000,10000,4*0/
```

```
C
C      TX, TRAFFIC VALUE/NODE ALONG X-AXIS
C      TY, TRAFFIC COST  ALONG Y-AXIS
C      SX, STORAGE VALUE/NODE ALONG X-AXIS
C      SY, STORAGE COST ALONG Y-AXIS
C      MAXT,MAXS  MAXIMUM NUMBER OF STEPS IN  TX AND SX
C               THE ABOVE GRAPH IS A STEP FUNCTION
C
C      OUTPUT THE TRAFFIC, STORAGE COST FUNCTION
C
       WRITE(6,67)  (TX(I),I=1,MAXT)
       WRITE(6,68)  (TY(I),I=1,MAXT)
       WRITE(6,49)  (SX(I),I=1,MAXS)
       WRITE(6,48)  (SY(I),I=1,MAXS)
 67    FORMAT('1TX = ',10(1X,F6.0))
 68    FORMAT(' TY = ',10(1X,F6.0))
 49    FORMAT(' SX = ',10(1X,F6.0))
 48    FORMAT(' SY = ',10(1X,F6.0))
C
C      CREATE DIFFERENT U VECTORS BY ROTATING V VECTOR
C      CREATE H VECTOR CONTAING U-V .G.T 0
C      ALSO DETERMINE CORRESPONDING TRAFFIC ,STORAGE VALUES AND COSTS
C
       DO 65 SITE=1,30
       N=10
       CALL UVECTR(U,V,N)
       CALL HVECTR(U,V,H,S,N,NBAS,DPTR,Z,PFTIN,WGTIN,Z1,Z2,Z3,Z4)
C
C      OUTPUT SITE,U,V,S,H VECTORS AND THE TRAFFIC,STORAGE VALUES/COSTS
C
       WRITE(6,85)  SITE
       WRITE(6,91)  (S(I),I=1,N)
       WRITE(6,92)  (V(I),I=1,N)
       WRITE(6,93)  (U(I),I=1,N)
       WRITE(6,73)  (H(I),I=1,N)
       WRITE(6,74)  Z1,Z2,Z3,Z4
 85    FORMAT('0',' ******   EXAMINE SITE ',I5)
 91    FORMAT(' ','S = ' ,10(1X,F5.0))
 92    FORMAT(' ','V = ' ,10(1X,F5.0))
 93    FORMAT(' ','U = ' ,10(1X,F5.0))
 73    FORMAT(' ','H = ' ,10(1X,F5.0))
 74    FORMAT(' ','Z1 = ',F5.0,3X,'Z2 = ',F5.0,3X,'Z3 = ',F5.0,
      13X,'Z4 = ',F5.0)
```

```
C
C     SET INITIAL VALUE AS OPTIMUM VALUE
C
      OT=Z3
      OS=Z4
      CALL FNDCST(TX,TY,MAXT,OT,OTCOST,SX,SY,MAXS,OS,OSCOST,OCOST)
      DO 27 II=1,N
        OX(II)=Z(II)
 27   CONTINUE
C
C     ALL ITEMS SELECTED SINCE H IS NEGATIVE; THIS IS OPTIMUM VALUE
C
      IF (NBAR.EQ.0) GO TO 28
C
C     SORT PROFIT(STORAGE)/WEIGHT(TRAFFIC) RATIO ; NECESSARY FOR KNAPSAC
C
      CALL SETPW(PFTIN,WGTIN,PROFIT,WEIGHT,PFTWGT,PTR,NBAR)
C
C     FIND THE OPTIMUM KNAPSACK SOLUTION USING BOUND/BACKTRACKING
C     PROCEDURES DEFINED BY HOROWITZ & SAHNI IN
C     "FUNDAMENTALS OF COMPUTER ALGORITHMS"
C
C     TRY EACH STEP ON THE X-AXIS OF TRAFFIC FUNCTION
C     IF M.LT.O, KNAPSACK CANNOT GIVE A SOLUTION;SO TRY NEXT STEP
C
      DO 10 J=1,MAXT
      M=TX(J)-Z3
      IF (M.LE.0) GO TO 10
      DO 60 II=1,N
      D(II)=Z(II)
 60   CONTINUE
      CALL KNAPSK(M,NBAR,PROFIT,WEIGHT,FW,FP,X,Y)
C
C     USING X, CONTAINING THE ITEMS CHOSEN FOR KNAPSACK
C     AND PFTWGT CONTAING INDEX FOR ORIGINAL INPUTS
C     PRINT OUT THE CORRESPONDING PROFITS & WEIGHTS
C
      FT=Z3
      FS=Z4
      WRITE(6,53) J,TX(J)
      CALL FNDCS(X,PFTIN,WGTIN,DPTR,PTR,D,FT,FS,NBAR)
 53   FORMAT('OKNAPSACK SOLUTION FOR ALPHA = ',I3,3X,'(',F8.0,')')
```

```
C
C     FIND THE COST OF THE TRAFFIC AND STORAGE AND OUTPUT  THEM
C
      CALL FNDCST(TX,TY,MAXT,FT,TCOST,SX,SY,MAXS,FS,SCOST,COST)
      CALL OUTCST(D,FS,SCOST,FT,TCOST,COST,N)
C
C     DETERMINE IF OPTIMUM;SAVE IF SO
C
      IF(COST.GE.OCOST) GO TO 11
      OCOST=COST
      OTCOST=TCOST
      OSCOST=SCOST
      OS=FS
      OT=FT
      DO 64 II=1,N
        OX(II)=D(II)
 64   CONTINUE
      GO TO 10
 11   CONTINUE
C
C     TERMINATE IF ALL ITEMS SELECTED AND NOT LAST NODE
C
      IF (FS.EQ.0.AND.J.NE.MAXT) GO TO 12
 10   CONTINUE
      GO TO 28
 12   WRITE(6,66)
 66   FORMAT(' <<< TERMINATED BECAUSE ALL ITEMS SELECTED')
C
C     OUTPUT OPTIMUM COST FOR THE SITE;INCLUDE D-VECTOR AND THE
C     TRAFFIC/STORAGE VALUES AND COSTS
C
 28   CONTINUE
      WRITE(6,62) SITE
      IF (Z1.EQ.Z3) GO TO 42
 62   FORMAT('0 ******   OPTIMUM ALLOCATION FOR SITE = ',I5,30X,'****')
      CALL OUTCST(OX,OS,OSCOST,OT,OTCOST,OCOST,N)
      GO TO 65
C
C     TERMINATE SINCE THE MAX. VALUE OF TRAFFIC IS NOT SUFFICIENT
C
 42   WRITE(6,63)
 63   FORMAT(' <<< COULD NOT PROCEED BECAUSE OF INSUFFICIENT ',
     1       'NUMBER OF TX STEPS >>>')
 65   CONTINUE
      STOP
      END
```

```
      SUBROUTINE KNAPSK(M,N,PROFIT,WEIGHT,FW,FP,X,Y)
C
C     M, THE SIZE OF THE KNAPSACK
C     N, THE NUMBER OF WEIGHTS AND PROFITS
C     PROFIT(1:N), THE CORRESPONDING PROFITS
C     WEIGHT(1:N), THE WEIGHTS
C     ****  NOTE THAT P(I)/W(I) >= P(I+1)/W(I+1); PRESORTED ****
C     FW, THE FINAL WEIGHT OF THE KNAPSACK
C     FP, THE FINAL MAXIMUM PROFIT
C     X(1:N) EITHER 0 OR 1. X(K)=0 IF W(K) IS NOT IN KNAPSACK,
C     ELSE X(K)=1
C     Y(1:N) TEMPORARY WORKING ARRAY FOR X
C
      INTEGER N,K,I
      REAL FW,FP,CW,CP,M,WEIGHT(100),PROFIT(100)
      INTEGER*2 Y(100),X(100)
C
C     INITIALIZE
C     CW, THE CURRENT WEIGHT
C     CP, THE CURRENT PROFIT
C     K,  THE CURRENT NODE
C     FP, MAXIMUM PROFIT (SET NEGATIVE)
C     N1, ADD 1 TO THE NUMBER OF ITEMS SO THAT LAST ITEM IS TESTED
C
      DO 12 I=1,N
         X(I)=0
         Y(I)=0
 12   CONTINUE
      CW=0
      CP=0
      K=0
      FP=-1
      FW=0
      N1=N+1
      PP=0
      WW=0
      IBND=0
      J=0
C
C     FIND A BOUND FOR THE CURRENT PATH
C     TERMINATE PRESENT PATH IF BOUND<=FP
C     SINCE THE PATH CANNOT LEAD TO A BETTER SOLUTION
C
 10   CONTINUE
      BND=BOUND(CP,CW,K,PP,WW,J,M,N1,PROFIT,WEIGHT,Y)
      IBND=INT(BND)
      IF (IBND.GT.FP) GO TO 50
 20   CONTINUE
```

```
C
C     FIND THE LAST WEIGHT INCLUDED IN THE KNAPSACK SO AS TO
C     TRACE BACK ALONG THE RECENT PATH TOMOST RECENT
C     NODE FROM WHICH AN UNTRIED MOVE CAN BE MADE
C
      IF (K.EQ.0) RETURN
      IF (Y(K).EQ.0) GO TO 30
      GO TO 40
 30   CONTINUE
      K=K-1
      GO TO 20
C
C     ALL NODES TRIED SINCE WE ARE BACK TO THE FIRST NODE;
C     OPTIMUM SOLUTION IS FP
C
 40   IF (K.EQ.0) GO TO 70
C
C     REMOVE ITEM K FROM THE KNAPSACK
C
      Y(K)=0
      CW=CW-WEIGHT(K)
      CP=CP-PROFIT(K)
      GO TO 10
C
C     PLACE ITEM K INTO KNAPSACK
C
 50   CONTINUE
      CP=PP
      CW=WW
      K=J
C
C     IF K<=N THEN ITEM K DOES NOT FIT; MAKE A RIGHT CHILD MOVE
C
      IF (K.GT.N) GO TO 60
      Y(K)=0
      GO TO 10
C
C     UPDATE FOR CURRENT OPTIMUM SOLUTION
C
 60   CONTINUE
      FP=CP
      FW=CW
      K=N
C
C     TRANSFER FROM TEMP TO PERM TO INDICATE THE ITEM CHOSEN
C
      DO 80 I=1,N
         X(I)=Y(I)
 80   CONTINUE
      GO TO 10
 70   CONTINUE
      RETURN
      END
```

```
      FUNCTION BOUND(P,W,K,PP,WW,I,M,N,PROFIT,WEIGHT,Y)
C
C     TREE ORGANIZATION IS USED TO FIND THE BOUND BY BACKTRACKING
C     AND KILLING AND EXPANDING NODES OF THE TREE.
C
C     P, THE CURRENT PROFIT
C     W, THE CURRENT WEIGHT
C     K, THE CURRENT NODE
C     PP, THE NEW PROFIT CORRESPONDING TO THE LAST LEFT CHILD MOVE
C     WW, THE NEW WEIGHT CORRESPONDING TO THE LAST LEFT CHILD MOVE
C     I, INDEX OF THE FIRST INDEX OBJECT THAT DOES NOT FIT; N+1 OF NONE
C     M, THE SIZE OF THE KNAPSACK
C     N, THE NUMBER OF WEIGHTS AND PROFITS IN THE PROBLEM + 1
C     **** PLUS ONE SO THAT LAST NODE IS NOT IGNORED ****
C     PROFIT(1:N), THE CORRESPONDING PROFITS IN THE PROBLEM
C     WEIGHT(1:N), THE CORRESPONDING WEIGHTS IN THE PROBLEM
C     Y(1:N) EITHER 0 OR 1. Y(K)=0 IF W(K) IS NOT IN KNAPSACK,
C     ELSE Y(K)=1
C
C
      INTEGER K,I,J,N
      REAL P,W,PP,WW,M,PROFIT(100),WEIGHT(100)
      INTEGER*2 Y(100)
C
C     SET THE CURRENT PROFIT & WEIGHT IN THE KNAPSACK
C
      PP=P
      WW=W
C
C     LOCATE THE NEXT NOTE AS A CANDIDATE FOR THE KNAPSACK
C
      J=K+1
C
C     FIT AS MANY ITEMS IN THE KNAPSACK UNTIL FULL
C
      DO 10 I=J,N
        IF(WW+WEIGHT(I).GT.M) GO TO 20
        WW=WW+WEIGHT(I)
        PP=PP+PROFIT(I)
        Y(I)=1
 999    GO TO 10
 20     CONTINUE
        BOUND=PP+(M-WW)*PROFIT(I)/WEIGHT(I)
        RETURN
 10   CONTINUE
      BOUND=PP
 60   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PART(A,M,P)
C
C     THE ELEMENTS IN THE ARRAY A ARE REARRANGED IN SUCH A WAY
C     THAT IF INITIALLY T=A(M)
C     THEN AFTER COMPLETION A(Q)=T FOR M<=Q<=P-1,
C     A(K)<=T FOR M<=K<=Q AND A(K)>=T FOR Q<K<P
C     THE FINAL VALUE OF P IS Q
C
C     A, CONTAINS ELEMENTS TO BE PARTITIONED
C     **** NOTE THAT N+1 ELEMENT MUST BE ZERO ****
C     M, NUMBER OF ELEMENTS TO BE PARTITIONED
C     P, POSITION IN A WHERE PARTITION OCCURS
C     V,I,TEMP WORKING VARIABLES
C
      INTEGER P,I
      REAL A(100),V,TEMP
C
C     INITIALIZE TO DEFINE THE FIRST & LAST
C     ELEMENTS FOR PARTITION
C
      V=A(M)
      I=M
C
 10   CONTINUE
C
C     I MOVES LEFT TO RIGHT
C
      I=I+1
      IF (A(I).LE.V) GO TO 20
      GO TO 10
 20   CONTINUE
C
C     P MOVES RIGHT TO LEFT
C
      P=P-1
      IF (A(P).GE.V) GO TO 30
      GO TO 20
 30   CONTINUE
      IF (I.GE.P) GO TO 40
C
C     INTERCHANGE A(I) & A(P)
C
      TEMP=A(I)
      A(I)=A(P)
      A(P)=TEMP
      GO TO 10
C
C     PARTITION IS AT POSITION P
C
 40   CONTINUE
      A(M)=A(P)
      A(P)=V
      RETURN
      END
```

```
      SUBROUTINE SORT(A,N)
C
C     THE ELEMENTS IN THE ARRAY A ARE REARRANGED IN SUCH A WAY
C     THAT IF INITIALLY T=A(M)
C     THEN AFTER COMPLETION A(Q)=T FOR M<=Q<=P-1,
C     A(K)<=T FOR M<=K<=Q AND A(K)>=T FOR Q<K<P
C     THE FINAL VALUE OF P IS Q
C
C     A, CONTAINS ELEMENTS TO BE SORTED
C     **** NOTE THAT N+1 ELEMENT MUST BE ZERO ****
C     N, NUMBER OF ELEMENTS IN THE ARRAY
C
C     STACK, CONTAINS A PAIRED SET DEFINING THE FIRST & THE LAST
C        OF A FOR PARTITIONING
C     TOP, THE CURRENT INDEX FOR STACK
C     J,P,Q, WORKING VARIABLES
C
      INTEGER STACK(100)/100*0/,TOP
      INTEGER N,J,P,Q
      REAL A(100)
C
C     INITIALIZE
C
      TOP=0
      P=1
      Q=N
C
 10   CONTINUE
      IF (P.GE.Q) GO TO 40
C
C     PARTITION THE ARRAY BETWEEN A & P
C     J RETURNS THE INDEX WHERE PARTITION TAKES PLACE
C
      J=Q+1
      CALL PART(A,P,J)
C
C     DEPENDING ON THE INDEX J, SAVE IN THE STACK THE FIRST & LAST
C     ELEMENTS OF THE PARTITION
C
      IF ((J-P).GE.(Q-J)) GO TO 20
      STACK(TOP+1)=J+1
      STACK(TOP+2)=Q
      Q=J-1
      GO TO 30
 20   CONTINUE
      STACK(TOP+1)=P
      STACK(TOP+2)=J-1
      P=J+1
 30   CONTINUE
```

```
C
C     UPDATE LOCATION IN STACK
C
      TOP=TOP+2
C
C     NOW SORT THE SMALLER SUBFILE
C
      GO TO 10
 40   CONTINUE
C
C     NOTHING IN STACK; ARRAY SORTED
C
      IF (TOP.EQ.0) GO TO 60
C
C     NOW SORT THE ELEMENTS DEFINED BE STACK;THOSE IN ARRAY
C     ALREADY SORTED
C
      Q=STACK(TOP)
      P=STACK(TOP-1)
      TOP=TOP-2
      GO TO 10
C
 60   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FNDCST(TX,TY,MAXT,T,TCOST,SX,SY,MAXS,S,SCOST,COST)
C
C     THE TRAFFIC COST, STORAGE COST AND THE TOTAL COST ARE COMPUTED
C     FOR THE FILES INCURRING A TRFFIC OF T AND STORAGE OF S
C
C     TX, TRAFFIC VALUES ALONG THE X-AXIS
C     TY, TRAFFIC COST ALONG THE Y-AXIS
C     MAXT, MAXIMUM TRAFFIC VALUE
C     T,  CURRENT TRAFFIC
C     TCOST,  TRAFFIC COST CORRESPONDING TO THE CURRENT TRAFFIC
C     SX, STORAGE VALUES ALONG THE X-AXIS
C     SY, STORAGE COST ALONG THE Y-AXIS
C     MAXS, MAXIMUM STORAGE VALUE
C     S,  CURRENT STORAGE
C     SCOST,  STORAGE COST CORRESPONDING TO THE CURRENT STORAGE
C     COST,  TOTAL COST = TCOST+SCOST
C
      INTEGER I,MAXT,MAXS
      REAL TX(100),TY(100),SX(100),SY(100),T,S,TCOST,SCOST,COST
C
C     DETERMINE TRAFFIC COST
C     IF NOT WITHIN THE RANGE SPECIFIED, THEN MAXIMUM COST
C
      DO 10 I=1,MAXT
         IF (TX(I).GE.T) GO TO 20
 10   CONTINUE
      TCOST = TY(MAXT)
      GO TO 30
 20   TCOST = TY(I)
C
C     DETERMINE STORAGE COST
C     IF NOT WITHIN THE RANGE SPECIFIED, THEN MAXIMUM COST
C
 30   CONTINUE
      DO 40 I=1,MAXS
         IF (SX(I).GE.S) GO TO 50
 40   CONTINUE
      SCOST = SY(MAXS)
      GO TO 60
 50   SCOST = SY(I)
C
C     SET TOTAL COST
C
 60   COST=TCOST+SCOST
      RETURN
      END
```

```
      SUBROUTINE OUTCST(D,S,SCOST,T,TCOST,COST,N)
C
C     THE D-VECTOR, STORAGE AND TRAFFIC AND THE CORRESPONDING COSTS
C     ARE OUTPUTTTED
C
C     D,  BINARY VECTOR DEFINING WHETHER A FILE IS STORED OR NOT
C     S,  STORAGE VALUE
C     SCOST,  STORAGE COST CORRESPONDING FOR S
C     T,  TRAFFIC VALUE
C     TCOST,  TRAFFIC COST CORRESPONDING FOR T
C     COST, TOTAL COST
C     N, NUMBER OF ELEMENTS IN D
C
      INTEGER N,I
      INTEGER*2 D(100)
      REAL S,SCOST,T,TCOST,COST
      WRITE(6,10)  (D(I),I=1,N)
      WRITE(6,20)  S
      WRITE(6,30)  SCOST
      WRITE(6,40)  T
      WRITE(6,50)  TCOST
      WRITE(6,60)  COST
 10   FORMAT('0D = ',10(1X,I5))
 20   FORMAT(' X(D)  = ',F10.0)
 30   FORMAT(' S(X)  = ',F10.0)
 40   FORMAT(' Y(D)  = ',F10.0)
 50   FORMAT(' T(Y)  = ',F10.0)
 60   FORMAT(' C     = ',F10.0)
      RETURN
      END
```

```
      SUBROUTINE HVECTR(U,V,H,S,N,NBAR,DPTR,Z,PFTIN,WGTIN,Z1,Z2,Z3,Z4)
C
C     THE H-VECTOR IS COMPUTED,AS U-V
C     ARE OUTPUTTTED
C
C     U,  VECTOR CONTAINING TRAFFIC COST FOR FILES NOT STORED
C     V,  VECTOR CONTAINING TRAFFIC COST FOR FILES STORED
C     H,  (U-V) .GT. 0
C     S,  VECTOR CONTAINING STORAGE COST FOR FILES STORED
C     N,  NUMBER OF FILES UNDER CONSIDERATION; NUMBER OF ITEMS
C         IN THE U,V,S VECTORS
C     NBAR, NUMBER OF ITEMS
C     DPTR, POINTERS FOR H, TO THE ORIGINAL LOCATION IN U,V
C     Z, BINARY VECTOR DEFINING WHICH FILES ARE TO BE STORED/NOT STORED
         1 - IF NOT STORED;ELSE 0
C     PFTIN, PROFIT FOR SUSEQUENT KNAPSACK PROBLEM; STORAGE COST
C     WGTIN, WEIGHT FOR SUSEQUENT KNAPSACK PROBLEM; TRAFFIC COST
C     Z1, TOTAL TRAFFIC COST FOR STORING FILES
C     Z2, TOTAL STORAGE COST
C     Z3, TRAFFIC COST FOR NOT STORING ALL FILES
C     Z4, STORAGE COST FOR NOT STORING ALL FILES
C
      INTEGER NBAR,N
      REAL U(100),V(100),H(100),S(100),PFTIN(100),WGTIN(100)
      REAL DIFF,Z1,Z2,Z3,Z4
      INTEGER*2 DPTR(100),Z(100)
C
C     INITIALIZE
C
      SUMD=0
      Z3=0
      Z4=0
      NBAR=0
C
C     FOR H, CHOOSE ALL FILES WHERE U.GT.V
C     ALSO KEEP RUNNING TAB ON THE STORAGE AND TRAFFIC COSTS
C
      DO 10 I=1,N
         DIFF=U(I)-V(I)
         H(I)=DIFF
         IF(DIFF.GT.0)GO TO 50
         Z(I)=1
         Z3=Z3+DIFF
         Z4=Z4+S(I)
         GO TO 10
 50      CONTINUE
         Z(I)=0
         NBAR=NBAR+1
         DPTR(NBAR)=I
         WGTIN(NBAR)=DIFF
         PFTIN(NBAR)=S(I)
 10   CONTINUE
      Z3= Z1+Z3
      Z4= Z2-Z4
      RETURN
      END
```

```
      SUBROUTINE UVECTR(U,V,N)
C
C     THE U-VECTOR IS DERIVED BY ROTATING THE V-VECTOR
C
C     U,  VECTOR CONTAINING TRAFFIC COST FOR FILES NOT STORED
C     V,  VECTOR CONTAINING TRAFFIC COST FOR FILES STORED
C     N,  NUMBER OF FILES UNDER CONSIDERATION; NUMBER OF ITEMS
C         IN THE U,V VECTORS
C
      INTEGER N,ISTP/100/,N1/0/,N2,INXT,IBEG
      REAL U(100),V(100)
C
C     INITIALIZE
C
      IF(ISTP.LE.N) GO TO 30
      N2=N-N1
      N1=N1+1
      ISTP=N1+1
   30 CONTINUE
      INXT=N+N1-ISTP
      DO 10 I=1,N2
         IBEG=INXT-I
         IF(IBEG.LE.0) IBEG=IBEG+N
         U(I)=V(IBEG)
   10 CONTINUE
      ISTP=ISTP+1
      RETURN
      END
```

```
      SUBROUTINE SETPW(PFTIN,WGTIN,PROFIT,WEIGHT,PFTWGT,PTR,N)
C
C     PFTIN, THE STORAGE COST
C     WGTIN, THE TRAFFIC COST
C     PROFIT, THE SORTED STORAGE COST
C     WEIGHT, THE SORTED TRAFFIC COST
C     PFTWGT, PROFIT/WEIGHT RATIO FOR SORTING
C     PTR,    POINTERS TO THE PRESORTED ARRAYS
C     N, NUMBER OF ITEMS IN THE ARRAYS
C
      INTEGER N,I,J
      REAL PFTIN(100),WGTIN(100)
      REAL PROFIT(100),WEIGHT(100),PFTWGT(100)
      REAL TEMP
      INTEGER*2 PTR(100)
C
C     CREATE PROFIT/WEIGHT RATIO ARRAY(1:N) FOR SORTING
C     SUCH THAT PROFIT(I)/WEIGHT(I) >= PROFIT(I+1)/WEIGHT(I+1)
C
      DO 10 I=1,N
        PFTWGT(I)=PFTIN(I)/WGTIN(I)
 10   CONTINUE
      PFTWGT(N+1)=0
C
C     SORT THE PROFIT/WEIGHT RATIO IN DESCENDING ORDER
C
C
C     SORT THE PROFIT/WEIGHT RATIO IN DESCENDING ORDER
C     USING QUICKSORT/PARTITION
C     PROCEDURES DEFINED BY HOROWITZ & SAHNI IN
C     "FUNDAMENTALS OF COMPUTER ALGORITHMS"
C     *** NOTE THAT N+1 ELEMENT IN PFTWGT MUST BE 0
C
      CALL SORT(PFTWGT,N)
C
C     CREATE PROFIT & WEIGHT ARRAYS AS DEFINED BY THE SORTED
C     PROFIT/WEIGHT RATIO
C     PFTIN POINTERS IN PTR ARRAY TO KEEP TRACK OF ORGINAL ORDER
C     IN PROFIT & WEIGHT
C
      DO 30 I=1,N
        TEMP=PFTIN(I)/WGTIN(I)
        DO 20 J=1,N
          IF (TEMP.EQ.PFTWGT(J)) GO TO 40
 20     CONTINUE
 40     PTR(J) = I
        PFTWGT(J) = -I
        PROFIT(J) = PFTIN(I)
        WEIGHT(J) = WGTIN(I)
 30   CONTINUE
C
C     LAST ITEMS MUST BE 0 FOR PROPER SORTING
C
      PROFIT(N+1)= 0
      WEIGHT(N+1)= 0
      RETURN
      END
```

```
      SUBROUTINE FNDCS(X,PFTIN,WGTIN,DPTR,PTR,D,FT,FS,N)
C
C     THE FINAL TRAFFIC VALUE AND THE STORAGE VALUES ARE DETERMINED
C     AS DEFINED BY THE BINARY VECTOR X FROM PFTIN & WGTIN VECTORS
C
C     X, BINARY VECTOR FROM THE KNAPSACK DEFINING FILES TO BE SELECTED
C     PFTIN, STORAGE VALUES
C     MAXT, MAXIMUM TRAFFIC VALUE
C     WGTIN, TRAFFIC VALUES
C     DPTR, POINTER TO THE ORIGINAL U,V VECTORS
C     PTR, POINTER TO THE PRESORTED H VECTOR
C     D, OPTIMUM BINARY D-VECTOR
C     FT, THE FINAL TRAFFIC COST
C     FS, THE FINAL STORAGE COST
C     N, NUMBER OF ITEMS IN X
C
      INTEGER N
      REAL PFTIN(100),WGTIN(100),FS,FT
      INTEGER*2 IPTR,I,J,X(100),DPTR(100),PTR(100),D(100)
C
C     CHOOSE ONLY IF X=1; FINF THE INDEX FROM PTR;THEN FROM DPTR
C     KEEP RUNNING TOTAL OFTRAFFIC AND STORAGE
C
      DO 10 I=1,N
        IF (X(I).EQ.0) GO TO 10
        J=PTR(I)
        IPTR=DPTR(J)
        D(IPTR)=1
        FT=FT+WGTIN(J)
        FS=FS-PFTIN(J)
   10 CONTINUE
      RETURN
      END
```

Appendix C

CHI-SQUARE TEST FOR INDEPENDENCE OF ENCRYPTED DATA

(S.M. Wong)

C.1 INTRODUCTION

The following two problems are relevant to those schemes cryptanalysts often use for breaking a cryptosystem which uses a publicly available encryption algorithm:

<u>Problem #1</u> For a fixed but arbitrary key, is there any relationship between a certain portion of the plaintext and a certain portion of the ciphertext?

<u>Problem #2</u> For a fixed but arbitrary plaintext, is there any relationship between a certain portion of the key and a certain portion of the ciphertext?

If the answer to Problem #1 is 'yes', a cryptanalyst will be able to derive that portion of the plaintext by analyzing the corresponding portion of the ciphertext for a cryptosystem which is using a fixed key (though unknown to the attack). Although that portion may be just a small part of the entire plaintext, an experienced cryptanalyst may find it very helpful for conjecturing or deducing the whole plaintext. Similarly, if the answer to Problem #2 is 'yes', a cryptanalyst will be able to derive the key by applying the algorithm on a known plaintext. In short, if some portions of the key, the plaintext and the ciphertext are found to be dependent, the encryption algorithm will become unsecure or even totally useless.

Thus, these two problems should be an important concern for encryption algorithm designers and users. However, as far as we know, very little research has been done with the objective of showing, either mathematically or experimentally, that the existing encryption algorithms are insensitive to this kind of troubles. Because of the complexity of these algorithms, rigorous mathematical proofs seem to be out of reach. In the following, we investigate these two problems by means of a statistical technique, namely the Chi-square test for data independence. The technique is applicable to a general encryption algorithm. However, our tests are done on the DES (Data Encryption Standard) algorithm, because it is well known.

In Section C.2, we review briefly the method of Chi-square test of independence. Section C.3 describes our test process of applying this technique on the DES Algorithm. Section C.4 presents the test results and discussions.

C.2 CHI-SQUARE TEST OF INDEPENDENCE

For completeness, we briefly review the Chi-square test here.

Suppose a sample of N outcomes have been collected from a population. These outcomes are grouped according to their values of two nominal variables describing the population. For example, a population of students may be described by their examination marks and their heights. We want to find out whether these two variables are independent or not by applying the Chi-square test on the observed frequencies of these groups.

Procedure of the Chi-square test

In a Chi-square test, we begin with the null hypothesis that the two nominal variables are independent. The test is then used to determine whether the hypothesis should be accepted or rejected. The process has the following steps:

i) Form a contingency table of cells using the two nominal variables as its row and column indices. The pair of values of the two variables for an outcome specify the cell it belongs to. After N outcomes are measured, the observed frequency O_{ij}, i.e., the number of outcomes belonging to cell (i,j), is recorded.

ii) From the observed values of this contingency table, calculate the expected frequencies E_{ij} as follows:

$$E_{ij} = \frac{R_i C_j}{N} , \qquad i = 1,2,\ldots\ldots r. \quad j = 1,2,\ldots\ldots c.$$

where E_{ij} = expected frequency for cell (i,j),

R_i = sum of the frequencies in row i,

C_j = sum of the frequencies in column j,

N = sum of the frequencies in all cells,

r = number of rows of the contingency table,

c = number of columns of the contingenct table,

iii) Compute the Chi-square value, x^2, by the following formula

$$x^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} = observed frequency in cell (i,j) of the contingency table.

iv) Compare the computed Chi-square value x^2 with a critical value obtained from a Chi-square table (also called rejection regions) at a specified significance level α and degree of freedom, df. Only a one-tailed test is appropriate.

v) If the computed Chi-square value equals or exceeds the Chi-square table value, the null hypothesis of independence is rejected. Otherwise, it is accepted.

C.3 APPLICATION OF CHI-SQUARE TEST ON THE DES ALGORITHM

We have conducted two sets of tests, one for Problem #1 and another for Problem #2. The test processes are explained below while the results will be described in Section C.4.

i) Experimental process for Problem #1 (fixed key)

For Problem #1, with the key unchanged, the set of all possible (plaintext, ciphertext) pairs form the population. The two norminal variables describing the population are a specified portion of the plaintext and a specified portion of the ciphertext.

Each experiment consists of N cycles. Each cycle applies the DES algorithm once to get a (plaintext, ciphertext) pair as an outcome. The first cycle chooses an arbitary key and plaintext, both 64 bits long. In the other cycles, the same key is used but the ciphertext of the previous cycle is used as the plaintext. (It is reported (GAIT77) that the DES algorithm can be used as a random bit generator. Thus, a plaintext generated in this way can be considered as random.)

In each cycle, the value of a specified portion of the plaintext and the value of a specified portion of the ciphertext are used as the row and column numbers, respectively, for determining the cell location in the contingency table. The frequency of that cell is then increased by one.

After N cycles, the Chi-square value is computed and the hypothesis is tested according to the method described in Section C.2.

ii) Experimental process for Problem #2 (fixed plaintext)

The same concept and process as for Problem #1 are adopted for Problem #2, except that the plaintext and the key exchange their roles. The plaintext is now fixed in all cycles while, in each cycle, the ciphertext of the previous cycle is used as the new key.

C.4 TESTS AND DISCUSSIONS

Tests: (see also Section C.3)

The data and results of our test are explained in the following:

i) Since software for the DES encryption and decryption algorithms is not available in the literature, a lot of efforts have been spent in their coding. Table C.1 shows part of our tests on the validity of our coding. The 64-bit key and plaintext are chosen arbitrarily. The encryption algorithm is then applied to generate the 64-bit ciphertext. Then, this ciphertext is used as the plaintext for the decryption algorithm. The same key as for the encryption test is used. The output of the decryption test shows the complete recovery of the plaintext in the encryption test by the decryption algorithm.

ii) Twenty-one tests have been performed for each of Problem #1 and Problem #2. Results are summarised in Tables C.2 and C.3, respectively. In these tables, each row shows the data and the results of one test. The total frequency (i.e., sample size) for each test has been chosen between 500 to 3000, depending on the degree of freedom used. The bit positions of the plaintext and the ciphertext are chosen arbitrarily. Rejection regions are obtained from the Table of Chi-square distribution (BEYE) with 1% significance level. The Chi-square values are calculated by the method described in Section C.2.

For a statistical test, increase in the sample size will decrease the two types of errors mentioned in Section C.2. Also, according to Roscoe (ROSC69), a sample size between 30 to 500 is usually adequate for this type of behavioural research. The sample size for our Chi-square data independence tests are between 500 to 3000. These are much greater than this requirement and should therefore give us a high level of confidence about our test results.

As shown in Tables C.2 and C.3, the computed Chi-square values are all less than the rejection regions at a significance level of 1%. Therefore, the hypothesis of independence between the specified portion of the plaintext and the specified portion of the ciphertext is accepted. The hypothesis of independence between the specified portion of the key and the specified portion of the ciphertext is also accepted in all of our tests.

iii) As illustrations, two typical examples of our Chi-square tests for Problem #1 and #2 are given and shown in Figures C.1 and C.2. These figures are self-explanatory. The inputs and outputs for both examples are explained below and Figure C.3 shows the actual input data of Example 2.

<u>The following data are input</u> (see Figure C.3):

* First line: Total frequency, Nnmber of bits to be tested in the plaintext or key, and number of bits to be tested in the ciphertext.
* Secone line: Positions of the bits in the plaintext/key.
* 3rd line: Positions of the bits in the ciphertext.
* 4th line: Fixed key? Print table? ('1' for 'yes', '0' for 'no').

* 5th line: significance level and rejection region.
* Lines 6 to 86:Data required for the DES algorithm (NBS 77).
* Last 4 lines: 64-bit long plaintext and key.

<u>The following data are output</u> (see Figures C.1 and C.2):

* Statement of hypothesis.
* The 64-bit fixed key (for Problem #1) or plaintext (for Problem #2) used.
* Contingency table (output is optional).
* Sum of the frequencies in all cells.
* Positions ofthe specidied but fixed portion of the plaintext/key.
* Positions of the specified but fixed portion of the ciphertext.
* Degrees of freedom.
* Significance level.
* Rejection regions.
* Computed Chi-square value.
* Conclusion about the hypothesis.
* Total CPU time for the run.

The results recorded in Figure C.1 show that the hypothesis of independence between the specified portion (bits 1 and 2) of the plaintext and the specified portion (bits 3, 4 and 5) of the ciphertext is accepted. Also, the results recorded in Figure C.2 show that the hypothesis of independence between the specified portion of the ciphertext (bits 1, 2 and 3) and the specified portion (bits 1, 2 and 3) of the key is accepted.

Table C.1 Data of the DES Encryption and Decryption Algorithms.

Data for the DES encryption algorithm:

Key	10101110	00100110	10110000	00111100
	01001111	11110010	10011010	00001111
Input (Plaintext)	11001000	00111010	10111010	01111100
	10100101	10000001	11101010	01100011
Output (Ciphertext)	11100010	11000111	01101100	01101011
	00101100	10000101	01001101	00100011

Data of the DES decryption algorithm:

Key	10101110	00100110	10110000	00111100
	01001111	11110010	10011010	00001111
Input (Ciphertext)	11100010	11000111	01101100	01101011
	00101100	10000101	01001101	00100011
Output (Plaintext)	11001000	00111010	10111010	01111100
	10100101	10000001	11101010	01100011

Table C.2 For a fixed key, Chi-square Independence Test between any portion of the plaintext and any portion of the ciphertext of the Data Encryption Standard Algorithm (DES).

Test No.	Total Frequency	Bit Positions of Plaintext	Bit Positions of Ciphertext	Degree of Freedom	Rejection Regions	Chi-square	Significance Level	Hypothesis
1	500	1,2	1,2	9	21.67	6.53	0.01	accepted
2	500	7,8	7,8	9	21.67	13.94	0.01	accepted
3	500	6,7	6,7	9	21.67	3.67	0.01	accepted
4	500	1,2	3,4	9	21.67	10.93	0.01	accepted
5	500	1,2	3,4,5	21	38.93	22.02	0.01	accepted
6	500	3,6	1,2,5,8	45	69.96	36.90	0.01	accepted
7	500	5,7	1,2,3,4,5,6	189	238.28	192.73	0.01	accepted
8	1000	4,5	1,2,3,4,5,6,7	381	468.72	427.60	0.01	accepted
9	2000	1,7	1,2,3,4,5,6,7,8	765	843.00	783.60	0.01	accepted
10	500	1,2,3	1,2,3	49	74.92	45.90	0.01	accepted
11	500	3,4,5	3,4,5	49	74.92	47.08	0.01	accepted
12	500	1,3,5	2,4,6,8	105	141.62	100.34	0.01	accepted
13	500	3,4,5	1,2,3,4,5	217	271.12	222.34	0.01	accepted
14	1000	1,2,3	1,2,3,4,5,6	441	522.72	473.80	0.01	accepted
15	1500	1,2,3	1,2,3,4,5,6,7	889	1001.61	921.77	0.01	accepted
16	2000	1,3,5	1,2,3,4,5,6,7,8	1785	2150.00	1730.59	0.01	accepted
17	500	2,5,7,8	2,5,7,8	225	277.29	191.85	0.01	accepted
18	500	1,2,3,4	5,6,7,8	225	277.29	220.30	0.01	accepted
19	2000	1,3,4,5	1,2,3,4,5,6,7,8	3825	4210.80	3774.35	0.01	accepted
20	1000	1,2,3,4,5	1,2,3,4,5	961	1054.31	985.53	0.01	accepted
21	3000	1,2,3,4,5,6,7	1,2,3,4,5,6,7	16129	22700	16000	0.01	accepted

Table C.3 For a fixed plaintext, Chi-square Independence Test between any portion of the key and any portion of the ciphertext of the Data Encryption Standard Algorithm (DES).

Test No.	Total Frequency	Bit Positions of Key	Bit Positions of Ciphertext	Degree of Freedom	Rejection Fegions	Chi-square	Significance Level	Hypothesis
1	500	1,2	1,2	9	21.67	10.28	0.01	accepted
2	500	7,8	7,8	9	21.67	8.67	0.01	accepted
3	500	6,7	6,7	9	21.67	5.58	0.01	accepted
4	500	1,2	3,4	9	21.67	8.64	0.01	accepted
5	500	1,2	3,4,5	21	38.93	18.42	0.01	accepted
6	500	3,6	1,2,5,8	45	69.96	43.20	0.01	accepted
7	500	5,7	1,2,3,4,5,6	189	238.28	204.33	0.01	accepted
8	1000	4,5	1,2,3,4,5,6,7	381	468.72	385.88	0.01	accepted
9	2000	1,7	1,2,3,4,5,6,7,8	765	843.00	806.55	0.01	accepted
10	500	1,2,3	1,2,3	49	74.92	46.71	0.01	accepted
11	500	3,4,5	3,4,5	49	74.92	40.30	0.01	accepted
12	500	1,3,5	2,4,6,8	105	141.62	89.23	0.01	accepted
13	500	3,4,5	1,2,3,4,5	217	271.12	237.79	0.01	accepted
14	1000	1,2,3	1,2,3,4,5,6	441	522.72	454.61	0.01	accepted
15	1500	1,2,3	1,2,3,4,5,6,7	889	1001.61	812.89	0.01	accepted
16	2000	1,3,5	1,2,3,4,5,6,7,8	1785	2150.00	1852.18	0.01	accepted
17	500	2,5,7,8	2,5,7,8	225	277.29	258.97	0.01	accepted
18	500	1,2,3,4	5,6,7,8	225	277.29	204.16	0.01	accepted
19	2000	1,3,4,5	1,2,3,4,5,6,7,8	3825	4210.80	3812.26	0.01	accepted
20	1000	1,2,3,4,5	1,2,3,4,5	961	1054.31	927.31	0.01	accepted
21	3000	1,2,3,4,5,6,7	1,2,3,4,5,6,7	16129	22700	16200	0.01	accepted

Example 1

HYPOTHESIS : FOR A FIXED KEY, A CHOSEN BUT FIXED PORTION OF CIPHERTEXT IS INDEPENDENT TO A CHOSEN BUT FIXED PORTION OF THE PLAINTEXT.

FIXED KEY : 10101110 00100110 10110000 00111100 01001111 11110010 10011010 00001111

CONTINGENCY TABLE :

	0	1	2	3	4	5	6	7	TOTAL
0	13	17	15	27	17	9	10	14	122
1	9	16	17	16	22	14	14	14	122
2	20	15	22	16	19	14	15	13	134
3	19	18	18	14	11	19	14	9	122
TOTAL	61	66	72	73	69	56	53	50	500

SUM OF THE FREQUENCIES FOR ALL CELLS : 500

POSITIONS OF THE CHOSEN BUT FIXED PORTION OF THE PLAINTEXT: 1 2

POSITIONS OF THE CHOSEN BUT FIXED PORTION OF THE CIPHERTEXT: 3 4 5

DEGREE OF FREEDOM : 21

SIGNIFICANCE LEVEL: 0.0100

REJECTION REGIONS : 38.9300

CHI SQUARE : 22.0154

CONCLUSION FROM THE TEST ABOUT THE HYPOTHESIS: THE HYPOTHESIS OF INDEPENDENCE IS ACCEPTED.

TOTAL CPU TIME USED : 21.2200 SECONDS.

Figure C.1 Computer output of the Chi-square independence test between the plaintext and the ciphertext (for a fixed key) of the DES Algorithm.

Example 2

HYPOTHESIS : FOR A FIXED PLAINTEXT, A CHOSEN BUT FIXED PORTION OF THE CIPHERTEXT
IS INDEPENDENT TO A CHOSEN BUT FIXED PORTION OF THE KEY.

FIXED PLAINTEXT : 11001000 00111010 10111010 01111100 10100101 10000001 11101010 01100011

CONTINGENCY TABLE :

	0	1	2	3	4	5	6	7	TOTAL
0	10	9	11	15	6	7	5	6	69
1	6	6	12	5	5	11	7	9	61
2	10	7	8	9	6	3	8	6	57
3	12	7	10	7	4	13	12	7	72
4	8	8	4	5	7	10	7	4	53
5	6	8	5	9	6	10	11	10	65
6	10	7	3	14	10	5	9	9	67
7	7	10	4	8	9	5	8	5	56
TOTAL	69	62	57	72	53	64	67	56	500

SUM OF THE FREQUENCIES FOR ALL CELLS : 500

POSITIONS OF THE CHOSEN BUT FIXED PORTION OF THE KEY: 1 2 3

POSITIONS OF THE CHOSEN BUT FIXED PORTION OF THE CIPHERTEXT: 1 2 3

DEGREE OF FREEDOM : 49

SIGNIFICANCE LEVEL: 0.0100

REJECTION REGIONS : 74.9200

CHI SQUARE : 46.7069

CONCLUSION FROM THE TEST ABOUT THE HYPOTHESIS: THE HYPOTHESIS OF INDEPENDENCE IS ACCEPTED.

TOTAL CPU TIME USED : 21.2580 SECONDS.

Figure C.2 Computer output of the Chi-square independence test between the key and the ciphertext (for a fixed

```
300 5 5
1 2 3
1 2 3
0 1
0.01 74.92
38 50 42 54 26 18 10 2
60 32 44 56 28 20 12 4
62 34 46 58 50 22 14 6
64 36 48 40 52 24 16 8
37 49 41 55 25 17  9 1
39 51 45 55 27 19 11 3
61 35 45 57 29 21 13 5
63 55 47 59 31 23 15 7
40  8 48 16 36 24 64 52
39  7 47 15 55 25 63 31
38  6 46 14 54 22 62 50
37  5 45 15 53 21 61 29
36  4 44 12 52 20 60 28
35  5 43 11 51 19 59 27
34  2 42 10 50 18 58 26
35  1 41  9 49 17 57 25
32  1  2  5  4  5
 4  3  6  7  8  9
 8  9 10 11 12 13
12 13 14 15 16 17
16 17 18 19 20 21
20 21 22 23 24 25
24 23 26 27 28 29
28 29 30 51 32  1
14  4 13  1  2 15 11  8  5 10  6 12  5  9  0  7
 0 13  7  4 14  2 15  1 10  6 12 11  9  5  5  8
 4  1 14  8 13  6  2 11 15 12  9  7  5 10  5  0
13 12  8  2  4  9  1  7  5 11  5 14 10  0  6 15
13  1  8 14  6 11  5  4  9  7  2 15 12  0  5 10
 3 15  4  7 15  2  8 14 12  0  1 10  6  9 11  5
 0 14  7 11 10  4 13  1  5  8 12  6  9  5  2 15
13  8 10  1  5 15  4  2 11  6  7 12  0  5 14  9
10  0  9 14  6  3 15  5  1 15 12  7 11  4  2  8
13  7  0  9  5  4  6 10  2  8  5 14 12 11 15  1
13  6  4  9  8 15  5  0 11  1  2 12  5 10 14  7
 1 10 13  0  6  9  8  7  4 15 14  5 11  5  2 12
 7 13 14  5  0  6  9 10  1  2  8  5 11 12  4 15
13  8 11  5  6 15  0  5  4  7  2 12  1 10 14  9
10  6  9  0 12 11  7 13 15  1  5 14  5  2  8  4
 3 15  0  6 10  1 15  8  9  4  5 11 12  7  2 14
 2 12  4  1  7 10 11  6  8  3  5 15 15  0 14  9
14 11  2 12  4  7 13  1  5  0 15 10  5  9  8  6
 4  2  1 11 10 13  7  8 15  9 12  5  6  5  0 14
11  8 12  7  1 14  2 13  6 15  0  9 10  4  5  5
12  1 10 13  9  2  6  8  0 15  5  4 14  7  5 11
10 13  4  2  7 12  9  5  6  1 15 14  0 11  5  8
 9 14 13  5  2  8 12  5  7  0  4 10  1 15 11  6
 4  3  2 12  9  5 15 10 11 14  1  7  6  0  8 15
 4 11  2 14 13  0  8 15  5 12  9  7  5 10  6  1
```

```
13  0 11  7  4  9  1 10 14  5  5 12  2 15  8  6
 1  4 11 13 12  5  7 14 10 15  6  8  0  5  9  2
 6 11 13  8  1  4 10  7  9  5  0 15 14  2  5 12
13  2  8  4  6 15 11  1 10  9  5 14  5  0 12  7
 1 13 15  8 10  5  7  4 12  5  6 11  0 14  9  2
 7 11  4  1  9 12 14  2  0  6 10 13 15  5  5  8
 2  1 14  7  4 10  8 13 15 12  9  0  5  5  6 11
16  7 20 21
29 12 28 17
 1 13 25 26
 3 18 51 10
 2  8 24 14
32 27  5  9
19 13 50  6
22 11  4 23
37 49 41 55 25 17  9
 1 38 50 42 54 26 18
10  2 39 51 45 55 27
19 11  3 60 52 44 56
63 55 47 59 31 23 15
 7 62 34 46 58 50 22
14  6 61 35 45 57 29
21 13  5 28 20 12  4
14 17 11 24  1  3
 3 28 15  6 21 10
23 19 12  4 26  8
16  7 27 20 13  2
41 32 51 57 47 55
30 40 51 45 53 48
44 49 39 56 54 53
46 42 30 56 29 52
1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1
1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 1 0 1
0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 1 1
1 0 1 0 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0
0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 1 1
```

Figure C.3 Example of input data.

Discussions:

From the results of our tests, we can make the following remarks:

i) The DES algorithm is insensitive to cryptanalytic attacks

The results as recorded in Table C.2 show that, at a 1% significance level, there is no close relationship between a portion of the plaintext and a portion of the ciphertext. Thus, a cryptananalyst will not be able to derive the plaintext from the ciphertext without knowing the encryption key. This implies that a DES cryptosystem can be ruled out from ciphertext-only attacks.

Similarly, the results as recorded in Table C.5 show that, for a fixed plaintext, a portion of the key is independent of a portion of the ciphertext. Thus, it is impossible for a cryptanalyst to derive the key by applying the DES algorithm on a known plaintext. In other words, the known plaintext attack and chosen plaintext attack can also be ruled out for the DES algorithm.

ii) Applicability of our method to Telidon instructions

A Telidon system is more susceptible to ciphertext-only attacks because Telidon data streams have special structures.

A Telidon PDI has only 32 different opcode formats. They are also publicly known. If the opcode portion of a PDI is found to be related with a certain portion of the ciphertext, a cryptanalyst will be able to deduce the most important part, i.e., the opcode, of the PDI without having to know the key being used. The probability of success in this kind of attacks is high since the number of possible PDI variations is small and they usually appear in certain patterns.

Our results imply that DES is a secure cryptosystem for data composed of Telidon instructions.

REFERENCES

BALL80 Ball, A.J.S. and Gecsei, J., "Videotex networks", IEEE Trans. on Consumer Electronics, Vol.CE-25, No.3, (1980), 8-14.

BALL81 Ball, A.J.S., Bochmann, G.V. and Gecsei, J., "Videotex Networks", IEEE Computer, Vol.C-13, No.12, (1980), 8-14.

BEYE Beyer, W.H., Handbook of Tables for Probability and Statistics, Second edition, The Chemical Rubber Co.

BNR79 "Vista gives television a new face", BNR Telesis, Vol.6, No.1, (Feb. 1979), 29-31.

BOCH81 Bochmann, G., "Overview of protocols in distributed videotex systems", Technical report, Department of Communications, Federal Government of Canada, (June 1981).

CHAN75 Chang, S.K., "Database decomposition in a hierarchical computer system", Proc. ACM SIGMOD Inter. Conf. on Management of Data, (May 1975), 48-52.

CHEU74 Cheung, T., "An interactive graphic display for regional partitioning by linear programming", COMM. of ACM, Vol.17, No.9 (1974), 513-516.

CLAR81 Clarke, K., "The design of a viewdata system", Viewdata in Action : A comparative study of Prestel (1981), ed. R. Winsbury, 33-59.

DATA82 Data Star, "Unit encrypts eight bytes in 25 microseconds", IEEE Computer, (June 1982), 103.

DEMA75 De Maio, A. and Roveda, G., "An all zero-one algorithm for a certain class of transportation problems", Operations Research, Vol.19, (1971), 1406-1418.

DENN58 Dennis, J.B., "A high-speed computer technique for the transportation problem", J. ACM. Vol.5, No.1 (Jan. 1958), 132-153.

EDBS75 Educational Data Base System, by Computer Systems Research Group, Univ. of Toronto, Toronto, Ontario, CANADA, (1975).

FARR82 Farrell J., "INET", Videotex Canada, Vol.1, No.1, (May 1982), 34-35.

GAIT77 Gait, J., "A new nonlinear pseudorandom number generator", IEEE Trans. on Software Engineering Vol.SE-3, No.5, (Sept. 1977), 359-363.

GECS82 Gecsei, J., The Architecture of Videotex Systems, Prentice-Hall, Inc. (1982).

GEOF69 Geoffrion, A.M., "An improved implicit enumeration approach for integer programming", Operations Research, Vol.17, (1969), 437-454.

GRIE82 Grieble, L., "Computer networking for Bildschirmtext", Videotex ´82, (1982), 441-451.

HOFF77 Hoffman, L.J., Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., (1977).

HORO78 Horowitz, E. and Sahni, S., Fundamentals of Computer Algorithms, Computer Science Press, (1978).

LAMY80 Lam, K. and Yu, C.T., "An approximation algorithm for a file-allocation problem in a hierarchical distributed system", Proc. ACM SIGMOD Inter. Conf. on Management of Data (1980), 125-132.

MANN82 Manning, E., Tompa, F.W., Gonnet, G.H., Williams, R.R., and DiCiccio, V.F., "The establishment of a Telidon technical centre", Waterloo Research Institute, Office of Research Administration, Univ. of Waterloo, Project #908-01, (1982).

MART79 Marti, B., Poignet, A., Schwartz, C., and Michon, V., "The Antiope videotex system", IEEE Trans. on Consumer Electronics, Vol.CE-25, No.3, (1979), 326-332.

NBS 77 National Bureau of Standards, U.S. Department of Commerce, USA, "Encryption algorithm for computer data protection", FIPS 46. Also in Tutorial on Computer Security and Integrity, IEEE Computer Society, VII, (1977), 14-28.

RIVE78 Rivest, R.L., Shamir, A., and Adleman, L., "A method for obtaining digital signatures and public-key cryptosystems", Comm. ACM, Vol.21, No.4, (Feb. 1978), 120-126.

ROSC69 Roscoe, J.T., Fundamental Research Statistics for the Behavioural Sciences. Holt, Rinehart and Winston, Inc. (1969).

ROSS75 Ross, G.T., and Soland, R.M., "A branch and bound algorithm for the generalized assignment problem", Mathematical Programming Vol.8, (1975), 91-103.

SRIN73 Srinivasan, V. and Thompson, G., "An algorithm for assigning uses to sources in a special class of transportation problems", Operations Research Vol.21, No.1 (1973), 284-295.

TANE77 Tanenbaum, A.S., Computer Network, Pentice-Hall, Inc. (1981), 387-436.

TROU80 Troughton, P., "Prestel operational strategy", Videotex, Viewdata, Teletext (1980-1981), 51-62.

WILS80 Wilson, L., "Bell Canada's VISTA project", Proc. of Inside Videotex, Informart, (March 1980).

WOOL80 Woolfe, R., Videotex the new Television/Telephone Information Services, Heyden & Son Ltd, (1980).

PART II

BIBLIOGRAPHY I

DISTRIBUTED QUERY PROCESSING

APER82 Apers, P.M.G., " Query Processing and Data Allocation in Distributed Database Systems", Ph.D thesis, Mathematisch Centrum, Amsterdam, (Sept. 1982).

APER83 Apers, P.M.G., Hevner, A.R. and Yao, S.B., "Optimization Algorithms for Distributed Queries", IEEE Trans. on Software Engineering Vol.SE-9, No.1, (Jan. 1983), 57-83.

APER83 Apers, P.M.G., Hevner, A.R. and Yao, S.B., "Optimization Algorithms for Distributed Queries", IEEE Trans. on Software Engineering, Vol.SE-9, No.1, (Jan. 1983), 57-68.

BABB79 Babb, E., "Implementing a Relational Database by Means of Specialized Hardware", ACM Trans. on Database Systems, Vol.4, No.1, (March 1979), 1-29.

BALD79 Baldissera, C., Bracchi, G. and Ceri, S., "A Query Processing Strategy for Distributed Database", Proc. EURP-IFIP, (1979), 667-678.

BERN79a Bernstein, P.A. and Goodman, N., " The Theory of Semi-joins", Technical Report CCA-79-27, (Nov. 1979), Computer Corporation of America.

BERN79b Bernstein, P.A. and Goodman, N., "Full Reducers for Relational Queries Using Multi-Attribute Semi-Joins ", Proc. NBS Symposium on Computer Networks, (Dec. 1979), 206-215.

BERN81a Bernstein, P.A. and Chiu, D.M.W., " Using Semi-Join to Solve Relational Queries", J. ACM, Vol.28 (Jan. 1981), 25-40.

BERN81b Bernstein, P.A., Goodman, N., Wong, E., Reive, C.L. and othnie, J., "Query Processing in a System for Distributed Databases (SDD-1)", ACM Trans. on Database Systems, Vol.6, No.4, (Dec. 1981).

BLAC82 Black, P.A. and Luk, W.S., "A New Heuristic for Generating Semi-join Programs for Distributed Query Processing", Research Report TR82-4, Dept. of Computer Science, Simon Fraser Univ., Burnaby, British Columbia, Canada, (1982).

BLAS76 Blasgen, M.W. and Eswaran, K.P., "On the Evaluation of Queries in a Relational Database System", IBM Research Report RJ1745, (April 1976).

CERI79 Ceri, S., "A Query Processing Strategy for Distributed Databases", Proc. IFIP 79 European Conf. on Applied Technology, London, (1979).

CERI80 Ceri, S., "Query Processing Strategies for Distributed Databases Systems", Distributed Databases, ed. Draffan and Poole, (1980).

CHEU80 Cheung, T.Y., "Two Statistical Models for Estimating the Number of Records in a Relational Database", Technical Report 80-11, Dept. of Computer Science, Univ. of Ottawa, Ottawa, Ontario, Canada, (1980).

CHEU82 Cheung, T.Y., " A Method for Equi-join Queries in Distributed Relational Databases", IEEE Trans. on Computers, Vol.C-31, No.8, (Aug. 1982), 746-751.

CHIU80 Chiu, D.M.W. and Ho, Y.C., "A Methodology for Interpreting Tree Queries into Optimal Semi-join Expressions", Proc. ACM SIGMOD Inter. Conf. on Management of Data, (1980), 169-178.

CHU79 Chu, W.W. and Hurley, P., "A Model for Optimal Processing for Distributed Databases", Proc. 18th IEEE COMPCON, (1979), 116-122.

CHUW82 Chu, W.W. and Hurley, P., "Optimal Query Processing for Distributed Database Systems", Proc. IEEE Trans. on Computers, Vol.C-31, No.9, (Sept. 1982), 835-850.

DANI83a Daniels, D., "Query Compilation in a distributed Database System", IBM Research Report #RJ3423, (1983).

DANI83b Daniels, D., Selinger P., Haas Laura, Lindsay B., Mohan C., Walker A., and Wilms P., "An Introduction to Distributed Query Compilation in R*", IBM Research Report #RJ3497, (1983).

EPST77 Epstein, R., Stonebraker, M. and Wong, E., "Distributed Query Processing in a Relational

Database System", Proc. ACM SIGMOD Inter. Conf. on Management of Data, (1977), 169-180.

EPST78 Epstein, R. and Stonebraker, M., "Analysis of Distributed Database Processing Strategies", Proc. Inter. Conf. on Management of Data, Austin, Texas, (June 1978).

HEVN79a Hevner, A. and Yao, S.B., "Query Processing in Distributed Database Systems", Berkeley Workshop on Distributed Data Management and Computer Networks, Berkeley, (1978).

HEVN79b Hevner, A. and Yao, S.B., "Query Processing in Distributed Database Systems", IEEE Trans. on Software Engineering, Vol.SE-5, No.3, (May 1979), 177-187.

HEVN80 Hevner, A., " The Optimization of Query Processing on Distributed Database Systems", Ph.D thesis, Dept. of Computer Science, Purdue Univ., Lafayette, Indiana, USA, (1980).

HEVN83 Hevner, A.R., "Data Allocation and Retrieval in Distributed Database Systems", Advances in Database Management, Vol. II, Heydon and Sons, (1983).

HSIA79 Hsiao, D. and Menon, J., "Post Processing Functions of a Database Machine", Technical Report, Ohio State Univ., Columbus, Ohio, USA, (July 1979).

KAMB82 Kambayashi, Y., Yoshikawa, M. and Yajima, S., "Query Processing for Distributed Databases", Proc. ACM-SIGMOD, (June 1982), 2-4.

KAMB83 Kambayashi, Y. and Yoshikawa, M., "Query Processing Utilizing Dependencies and Horizontal Decomposition", Proc. ACM-SIGMOD Inter. Conf. on Management of Data, (May 1983), 24-26.

KERS79 Kershberg, L., Ting, P., Yao, S.B., "Query Optimization in a Star Network", Bell Lab. Tech. Report, (1979).

KIM81a Kim, W., " On Optimizing an SQL-like Nested Query", IBM Research Report #RJ3063, (1981).

KIM81b Kim, W., "Query Optimization for Relational Database Systems", IBM Research Report #RJ3081, (1981).

KLUG82a Klug, A., "Access Paths in the "ABE" Statistical Query Facility", Technical Report #474, Computer

Science Dept., Univ. of Wisconsin, Madison, Wis. USA, (May 1982).

LIU82 Liu, A.C. and Chang, S.K., "Site Selection in Distributed Query Processing", Proc. Inter. Conf. on Distributed Computing Systems, IEEE computer Society, (Oct. 1982), 7-12.

KLUG82b Klug, A., "On Conjunctive Queries Containing Inequalities", Technical Report #477, Computer Science Dept., Univ. of Wisconsin, Madison, Wis. USA, (May 1982).

LUK81 Luk, W.S. and Black, P.A., "On Cost Estimation in Processing a Query in a Distributed Database System", Research Report TR81-7, Dept. of Computer Science, Simon Fraser Univ., Burnaby, British Columbia, Canada, (Dec. 1982).

NG82 Ng, P., "Distributed Compilation and Recompilation of Database Queries", IBM Research Report #RJ3375, (1982).

OZKA77 Ozkarahan, E.A., Schuster, S.A. and Sevcik, K.C., "Performance Evaluation of a Relational Associative Processor", ACM Trans. on Database Systems, Vol.2, No.2, (June 1977), 175-195.

OZSO80 Ozsoyoglu, M., "Query Processing in Distributed Databases", Ph.D thesis, Dept. of Computing Science, Univ. of Alberta, Edmonton, Alberta, Canada, 1980.

PAPA82 Papakonstantinou, G., "Optimal Evaluation of Queries", Computer Journal, Vol.25, No.2, (1982), 239-241.

PELA78 Pelagatti, G. and Schreiber, F.A., "Comparison of Different Access Strategies in a Distributed Database", Proc. Inter. Conf. on Databases : Improving Usability and Responsiveness, Haifa, Isreal, (Aug. 1978), 399-410.

PELA79 Pelagatti, G. and Schreiber, F.A., "Evaluation of Transmission Requirements in Distrbuted Database Access", Proc. ACM SIGMOD Conf. on Management of Data, (1979), 102-108.

PERR82 Perrizo, W. K., "A Method for Processing Distributed Database Queries", NDSU preprint, Fargo, ND, (1982).

PERR83 Perrizo, W., "A distributed database query processing algorithm for guaranteed response time", submitted for publication on ACM Trans. on Database Systems.

PLAT82 Plateau, D., "Compiling Relational Queries for a Finite State Automaton Hardware Filter", IRIA, Technical Report No.171 (Nov. 1982).

ROSE80 Rosenkrantz, D.J. and Hunt, H.B. III, "Processing Conjunctive Predicates and Queries", Proc. Sixth Inter. Conf. on Very Large Data Bases, Montreal, Quebec, Canada, (Oct. 1980), 64-72.

ROTH77 Rothnie, J.B. and Goodman, N., "A Survey of Research and Development in Distributed Database Management", Proc. Third Very Large Data Bases, Tokyo, Japan, (Oct. 1977), 48-62.

SACC82 Sacco, G.M. and Yao, S.B., "Query Optimization in Distributed Database Systems", Advances in Computers, Vol.21, New York, (1982).

SELI79 Selinger, P.G., Astrahan, M.M., Chamberlain, D.D., Lorie, R.A. and Price, T.G., "Access Path Selection in a Relational Database Management System", Proc. ACM SIGMOD Conf. on Management of Data, (1979), 23-34.

SPAC82 Spaccapietra, S., "A Review of Distributed Query Processing", Distributed Databases, Online Publications Ltd., U.K., 1982.

STON79 Stonebraker, M.R., Wong, E., Kreps, P. and Held, G.D., "Design and Implementation of INGRES", ACM Trans. Database Systems, Vol.1, No.3, (Sept. 1976).

TAKI80 Takizawa, M., "Operational Query Decomposition Algorithm in Distributed Databases", Japan Inform. Processing Develop. Centre, Technical Report TR80/2, (1980).

TOAN80 Toan, N.G., "Decentralized Dynamic Query Decomposition for Distributed Database Systems", Proc. ACM Pacific Conf., San Fracisco, Calif., USA (1980).

WONG76 Wong, E. and Yousefi, K., "Decomposition - A Strategy for Query Processing", ACM Trans. on Database Systems, Vol.1, No.3, (Sept. 1976).

WONG77 Wong, E., "Retrieving Dispersed Data from SDD-1: A System for Distributed Databases", Second Berkeley

Workshop on Distributed Data Management and Computer Networks, Berkeley, (1977), 217-235.

YAO81 Yao, A., " An Optimal Data-structure for One-Dimensional Range Queries", IBM Research Report #RJ3205, (1981).

YU79a Yu, C.T. and Ozsoyoglu, M., "An Algorithm for Tree-Query Membership of a Distributed Query", Proc. IEEE Third Inter. Conf. on Computer Software and Applications, (Nov. 1979), 306-312.

YU79b Yu, C.T. and Ozsoyoglu, M., " On Determining Tree-Query Membership of a Distributed Query", Technical Report, Dept. of Information Engineering, Univ. of Illinois at Chicago Circle, Chicago, Illinois, (Nov. 1979).

YU80 Yu, C.T., Lam K. and Ozsoyoglu, M.Z., "Distributed Query Optimization for Tree Queries", Technical Report, Dept. of Information Engineering, Univ. of Illinois at Chicago Circle, Chicago, Illinois, (July 1980).

YU82a Yu, C.T., Lam, K., Chang, C.C. and Chang, S.K., "Promising Approach to Distributed Query Processing", Proc. Sixth Berkeley Workshop on Distributed Data Management and Computer Networks, (Feb. 1982), 363-390. YU82b Yu, C.T. and Lin, Y.C., "Some Estimation Problems in Distributed Query Processing", Proc. Inter. Conf. on Distributed Computing Systems, IEEE Computer Society, (Oct. 1982), 13-19.

BIBLIOGRAPHY II

DATA, PROCESS AND PROCESSOR ALLOCATION PROBLEMS

ANDR82 Andrews, G.R. and Dobkin, D.P., and Downey, P.J., "Distributed Allocation with Pools of Servers", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Canada, (Aug. 1982), 73-83.

BAND72 Bandy, H., " An Approach to Resource Allocation in a Computer-Communication Network", Ph.D thesis, Case Western Reserve Univ., (1972).

BELC72 Belckrinitskaya et. al., "Distribution of Functions between Central Processor and Peripheral Computer", Computing Techniques in Automatical Control, UDC 681.325, 142-151, translated from "Automatika I Telemekhamika 1", (Jan. 1972), 161-170.

BELF76 Belford, G.G., " Optimization Problems in Distributed Data Management", Proc. Third Inter. Conf. on Computer Communication, Toronto, (1976), 297-302.

BUCC77 Bucci, G. and Golinelli, S., "A Distributed Strategy for Resource Allocation in Information Networks", Proc. Inter. Computing Symposium, North-Holland, Amsterdam, (1977), 345-356.

CASE72 Casey, R,G., "Allocation of Copies of a File in a Information Network", AFIPS SJCC 40, (1972), 617-625.

CASE73 Casey, R.G., "Design of Tree Networks for Distributed Data", AFIPS NCC 42, (1973), 341-348.

CERI79 Ceri, S. and Pelagatti, G., "Optimal File Allocation for a Distributed Database on a Network of Minicomputers", Proc. of First Inter. Conf. on Data Bases, (1979).

CERI81 Ceri, S. and Pelagatti, G., "Allocation of Operations in Distributed Database Access", IEEE Trans. on Computers, Vol.C-31, No.2, (1981), 119-128.

CERI82a Ceri, S. and Pelagatti, G., "A Solution Method for the Non-additive Resource Allocation Problem in Distributed System Design", Information Processing Letters, Vol.15, No.4, (Oct. 1982), 174-178.

CERI82b Ceri, S. and Pelagatti, G., "Optimal File Allocation in a Computer Network", Computer Networks, Vol.6, No.5, (1982), 345-357.

CERI83 Ceri, S. and Pelagatti, G., "Distribution Design of Logical Database Schemas", IEEE Trans. on Software Engineering, Vol.SE-9, No.2, (1983).

CHAN72 Chang, J.H. and Gorenstein, S., "A Disk File System Shared by Several Computers in a Teleprocessing Environment", Proc. Symposium on Computer-communications Networks & Teletraffic, New York 1972, ed. Jerome Fox, Polytechnic press.

CHAN75 Chang, S.K. and Tang, D.T., "Processor Allocation in a Distributed Computer System", Proc. Symposium on Computer Networks, Trends and Applications, Gaithersburg, Maryland (1975), 47-54.

CHAN76 Chang, S.K., "A Model for Distributed Computer System Design", IEEE Trans. on Systems, Man and Cybernetics, Vol.SMC-5, No.6, (May 1976), 344-359.

CHAN77 Chandler, J.S., and Delutis, T.G., "A Methodology for Multicriteria Information System Design", AFIPS, NCC, (1977), 895-905.

CHEN76 Cheng, T., "Design Considerations for Distributed Databases in Computer Networks", Ph.D thesis, Ohio State Univ., Ohio, USA, (1976)

CHEU79a Cheung, T.Y., "Physical Database Design by Mathematical Programming", Technical Report TR79-03, Dept. of Computer Science, Univ. of Ottawa, Ottawa, Ontario, Canada.

CHEU79b Cheung, T.Y., "Data Process and Processor Allocation in a Distributed Computer System -- A Review", Proc. Session 80, Conf. of Canadian Information Processing Society, (May 1980), 23-31.

CHOU78 Chou, W., Ferrante, F., Balagangadhar, M. and Gerke, L., "An Integrated Approach to Optimally Locating Network Access Facilities", Proc. Fourth Inter. Conf. on Computer Communication, (1978), 335-334.

CHOW81 Chow, W.M.,"Allocation of Computational Elements in Multiprocessor Systems", IBM Research Report #RC9152, (Nov. 1981).

CHU69 Chu, W.W., "Optimal File Allocation in a Multiple Computer System", IEEE Trans. on Computers, Vol. C-18, No.10, (1969), 885-889.

CHU73 Chu, W.W., "Optimal File Allocation in a Computer Network", Computer-Communications Networks, ed. N. Abramson and F.F. Kuo, Prentice-Hall, (1973), 82-94.

CLAR82 Clarke, E.M. and C.N. Nikolaou, "Distributed Reconfiguration Strategies for Fault-Tolerant Multiprocessor Systems", IEEE Trans. on Computers, Vol.C-31, No.8, (Aug. 1982), 771-784.

COFF81 Coffman., E.G. ,E. Gelenbe, and B. Plateau, "Optimization of the Number of Copies in a Distributed Data Base", IEEE Trans. on Software Engineering, Vol.SE-7, No.1, (Jan. 1981), 78-84.

DOWD82 Dowdy, L.W. and Foster, D.V., "Comparative Models of the File Assignment Problem", ACM Computing Survey, 14, 2(June 1982), 287-313.

EDWA77 Edwards, B.J., "Choice of Block Sizes for Magnetic Tape files", Computer Journal, 20, 1(Feb. 1977), 10-14.

EFE82 Efe, K., "Heuristic Models of Task Assignment Scheduling in Distributed Systems", IEEE Computer, Vol.15, No.6, (June 1982), 50-56.

EFRO66 Efroymson, M.A. and Ray, T.L., "A Branch-and-bound Algorithm for Plant Location", Operations Research, 14, 3 (1966), 361-368.

ESAU66 Esau, L.R. and Williams, K.C., "On Teleprocessing System Design: Part 2, a Method for Approximating the Optimal Network", IBM System Journal, 5, 3(1966), 142-147.

ESWA74 Eswaran, K.P., "Placement of Records in a File and File Allocation in a Computer Network", Proc. IFIP Second Inter. Conf. on Computer Communications, Stockholm, Sweden (1974), 304-307.

FISC80 Fischer, M.J., Griffeth, N.D., Guibas, L.J. and Lynch, N., "Optimal Placement of Identical Resourses in a Distributed Network", Techincal Report GIT-ICS-80/13, School of Information and

Computer Science, Georgia Institute of Technology, (1980).

FOLE74 Foley, J.D. and Brownlee, E.H., "A Model of Distributed Processing in Computer Networks, with Application to Satellite Graphics", Proc. Second Inter. Conf. on Computer Communications, Stockholm, (1974), 1-5.

FOST81 Foster, Dowdy, L.W. and Ames, J.E. IV, "File Assignment in a Computer Network", Computer Networks 5, (1981), 341-349.

FRAN79 Franca, P.M.B., " Processor Allocation in a Computer Network with Distributed Ownership", Ph.D thesis, Univ. of California, Berkeley, Calif., USA, (1979).

GEIS82 Geist, R.M. and Trivedi, K.S., "Optiomal Design of Multilevel Storage Hierarchies", IEEE Trans. on Computers, Vol.C-31, No.3, (March 1982), 249-260.

HOFR78 Hofri, M. and Jerry, C.J., " On the Allocation of Processes in Distributed Computing Systems", IBM Research report #RZ905, (1978).

JENN76 Jenny, C.J. and Kuemmerle, K., "Distributed Processing Within an Integrated Circuit/Packet Switching Node", IEEE Trans. on Communications, Vol.COMM-24, No.10, (Oct. 1976), 1089-1100.

JENN77 Jenny, C.J., "Process Partitioning in Distributed Systems", Proc. Nat. Telecom. Conf., Los Angeles, (1977). Also available as IBM Research Report #RZ873, (1977).

JENN79 Jenny, C.J., "Partitioning and Allocating Computational Objects in Distributed Computing Systems", IBM Research Report #RZ984, (1979).

JENN82 Jenny, C.J., "On the Allocation of Computational Objects in Distributed Systems", IBM Research Report #RZ1123, (1982).

JENN83a Jenny, C., "Methodologies for Placing Files and Processes in Systems with Decentralized Intelligence", IBM Research Report #RZ1139, (1983).

JENN83b Jenny, G.J., "Placing Files and Processes in Distributed Systems: A General Method Considering Resources with Limited Capacity", IBM Research Report #RZ1157, (1983).

KIMB75 Kimbleton, S.R., "Network Performance User Satisfaction, and Data Base Access", Research Report, USC/Information Science Institute, Univ. of Southern Calif., Calif, USA, (Aug. 1975).

KIMB77 Kimbleton, S.R., "A Fast Approach to Network Data Assignment", Proc. Second Berkeley Workshop on Distributed Data Management and Computer Networks, Lawrence Berkeley Lab., Univ. of Calif., (1977), 245-256.

KOLL81 Kollias, J.G. and Hatzopoulos, M., "Criteria to Aid in Solving the Problem of Allocating Copies of a File in a Computer Network", The Computer Journal, Vol.24, No.1, (Feb. 1981), 29-30.

LAM80 Lam, K. and Yu, C.T., "An Approximation Algorithm for a File Allocation Problem in a Hierarchical Distributed System", Proc. ACM-SIGMOD Inter. Conf. on Management of Data, (1980), 125-132.

LAM81 Lam, C.M. and Fung, K.T., "A Quadratic Programming Model for Optimal Data Distribution", BIT 21, (1981), 294-304.

LANN78 Lann, G.L., "Pseudo-Dynamic Resource Allocation in Distributed Databases", Proc. Fourth Inter. Conf. on Computer Communications, (1978), 245-252.

LEVI74 Levin, K.D.,"Organizing Distributed Data Bases in Computer Network", Ph.D thesis, Univ. of Pennsylvania, (1974).

LEVI75a Levin, K.D., and Morgan, H.L., "Optimizing Distributed Data Bases - A Framework for Research", AFIPS, NCC, (1975), 473-478.

LEVI75b Levin, K.D., "Two Algorithms for Optimal File Assignment in Heterogeneous Computer Networks", Technical Report 75-08-02, Wharton School, Univ. of Pennsylvania, (1975).

LEVI78 Levin, K.D., "A Dynamic Optimization Model for Distributed Databases", Operations Research, 26, 5(1978), 824-835.

LOOM76 Loomis, M.E., "Data Base Design: Object Distributed and Resource-Constrained Task Scheduling", Ph.D thesis, Univ. of California, Los Angeles, Calif., USA, (1976).

MA82 Ma, P.Y., Lee, E.Y.S., and Tsuchiya, M., "A Task Allocation Model for Distributed Computing

Systems", IEEE Trans. on Computers, Vol.C-31, No.1, (Jan. 1982), 41-47.

MAHM75 Mahmoud, S.A., " Resource Allocation and File Access Control in Distributed Information Networks", Ph.D thesis, Carleton Univ., Ottawa, Ontario, Canada (1975).

MAHM76 Mahmoud, S.A. and Riordon, J.S., "Optimal Allocation of Resources in Distributed Networks", ACM Trans. on Database Systems, Vol.1, No.1(1976), 66-78.

MATS82 Matsushita, Y., Yoshida, M., Wakino, A. and Beng, L.T., "Allocation Schemes of Multiple Copys of Data in Distributed Database Systems", Proc. Inter. Conf. on Distributed Computing Systems, IEEE Computer Society, (Oct. 1982), 250-256.

MORG77 Morgan, H.L. and Levin, K.D., " Optimal Program and Data Locations in Computer Networks", COMM. of ACM 20, 5(1977), 315-322.

REIF82 Reif, J. and Spirakis, P., "Real Time Resource Allocation in Distributed Systems", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Canada, (Aug. 1982), 84-94.

ROTH77 Rothnie J.B. and Goodman, N.,"A Survey of Research and Development in Distributed Database Management", Proc. Third Inter. Conf. on Very Large Databases, Tokyo, Japan, (1977), 48-62.

TAKI78 Takizawa, M., Hamanaka, E. and Ito, T., "Resource Integration and Data Sharing on Heterogeneous Resource Sharing System", Proc. Fourth Inter. Conf. on Computer Communication", (1978), 253-258.

URAN74 Urano, Y., Ono, K., and Inoue, S., " Optimal Design of Distributed Networks", Proc. Second Inter. Conf. on Computer Communications, Stockholm, (1974), 413-420.

WHIT70 Whitney, V.K.M., "A Study of Optimal File Site Assignment and Communication Network Configuration in Remote-Access Computer Message Processing and Communication Systems", Ph.D thesis, Univ. of Michigan, Michigan, USA, (1970).

WILL75 Willcox, D., " Optimal Query Strategies", Internal Technical Memo, Center for Advanced Computation, Univ. of Illinois at Urbana-Champaign, Illinois, USA, (1975).

BIBLIOGRAPHY III

CONCURRENCY CONTROL FOR DISTRIBUTED DATABASES

AGRA82 Agrawal, R. and DeWitt, D.J., " Further Optimism in Optimistic Methods of Concurrency Control ", Technical Report #470, Computer Sciences Dept., Univ. of Wisconsin, Madison, Wis., USA, (March 1982).

BALT82 Balter, R., Berard, P. and Decitre, P., " Why Control of the Concurrency Level in Distributed Systems is more Fundamental than Deadlock Management", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Ontario, Canada, (Aug. 1982), 183-193.

BARU82 Barua, G., " Mutual Consistency of Copies of Files Based on Request Characteristics ", Proc. Sixth Berkeley Workshop on Distributed Data Management and Computer Networks, (Feb. 1982), 95-115.

BERN82 Bernstein, P.A. and Goodman, N., " Concurrency Control Algorithms for Multiversion Database Systems ", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Ontario, Canada, (Aug. 1982), 209-215.

BREI82 Breitwieser, H. and Leszak, M., " A Distributed Transaction Processing Protocol Based on Majority Consensus ", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Ontario, Canada, (Aug. 1982), 224-237.

CARE83a Carey, M., " Granularity Hierachies in Concurrency Control ", Project INGRES, Electronics Research Lab, Univ. of California, California, USA, (Jan. 1983).

CARE83b Carey, M., " An Abstract Model of Database Concurrency Control Algorithms ", Project INGRES, Electronics Research Lab, Univ. of California, California, USA, (Jan. 1983).

CERI82 Ceri, S. and Owicki, S., " On the Use of Optimistic Methods for Concurrency Control in Distributed

Databases", Proc. Sixth Berkeley Workshop on Distributed Data Management and Computer Networks, (Feb. 1982), 117-130.

CHEN80 Cheng, W.K. and Belford, G.G., "Analysis of Update Synchronization Schemes in Distributed Database", Proc. COMPCON 80, (Fall 1980), 450-455.

CHEN82 Cheng, W.K, and Belford, G.G, "The Resiliency of Fully Replicated Distributed Databases ", Proc. Sixth Berkeley Workshop of Distributed Data Management and Computer Networks, (Feb. 1982), 23-44.

DEWI80 DeWitt, D.J. and Wilkinson, W.K., " Database Concurrency Control in Local Broadcast Networks ", Technical Report #396, Computer Sciences Dept., Univ. of Wisconsin-Madison, Wis., USA, (Aug. 1980).

EAGER81 Eager, D.L., " Robust Concurrency Control in Distributed Databases ", Technical Report CSRG#135, Computer Systems Research Group, Univ. of Toronto, Toronto, Ontario, Canada, (Oct. 1981).

FLE82 Fle, M.P. and Roucairol, G., " On Serializability of Iterated Transactions ", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Combuting, Ottawa, Ontario, Canada, (Aug. 1982), 194-200.

GALL82 Galler, B.I., " Concurrency Control Performance Issues ", CSRG Technical Report #147, Computer Systems Research Group, Univ. of Toronto, Toronto, Ontario, Canada, (1982).

GELE78 Gelenbe, E. and Sevcik, K., " Analysis of Update Synchronization for Multiple Copy Data-Bases ", Research Report #322, Laboratoire de Recherche en Informatique, Univ. of Paris, Sud, 91405 Orsay, France, (Sept. 1978).

HAML82 Hamlin, G. and George, J.E., "Experiences with Distributing Graphic Software between Processors", Proc. Inter. Conf. on Distributed Computing Systems, IEEE Computer Society, (Oct. 1982), 486-492.

ISLO79 Isloor, S.S., " Consistency Aspects of Distributed Databases ", Technical Report TR79-4, Ph.D. thesis, Univ. of Alberta, Edmonton, Alberta, Canada, (Aug. 1979).

KAMO82 Kamoun, F., Djerad, M.B. and Lann, G.L., "Queueing Analysis of the Ordering Issue in a Distributed

Database Concurrency Control Mechanism: A General Case", Proc. Inter. Conf. on Distributed Computing Systems, IEEE Computer Society, (Oct. 1982), 447-452.

KLUG80 Klug, A., " Locking Expressions for Increased Database Concurrency ", Technical Report #400, Computer Sciences Dept., Univ. of Wisconsin, Madison, Wis., USA, (Oct. 1980).

KULI81 Kulikowski, J.L., "Semantical and logical consistency requirements for distributed data bases", Proc. Network from the User's Point of View, edited by Csaba, L., Szentivanyi, T. and Tamay, K., IFIP, (1981).

LANN81 Le Lann, G., " Consistency Issues in Distributed Databases ", Distributed Database, Online Publications Ltd, U.K., (1981).

LELA81 LeLann, G., "Synchronization", Lecture Notes in Computer Science, Vol.105, (1981), 266-282

HOLL81 Holler, E., "Multiple Copy Update", Lecture Notes in Computer Science, Vol.105, (1981), 284-303.

MILN82 Milner, R., " Four Combinators for Concurrency ", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Ontario, Canada, (Aug. 1982), 104-110. OZSU82 Ozsu, M.T. and Weide, B.W., "Modeling of Distributed

PETE79 Peterson, G.L., " Concurrency and Complexity ", Ph.D thesis, Technical Report #59, Computer Science Dept., Univ. of Rochester, New York, USA, (Aug. 1979).

RAMA80 Ramamoorthy, C.V. and Ho, G.S., " Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets ", IEEE Trans. on Software Engineering, Vol.SE-6, No. 5, (Sept. 1980), 440-449. SCHL82 Schluter, R.G., Shih, J.C. and Machleit, T.L., "Effect of Resource Allocation on Distributed System Response- A Case Study", Proc. Inter. Conf. on Distributed Computing Systems, IEEE Computer Society, (Oct. 1982), 892-898.

SCHN80 Schneider, F.B., " Ensuring Consistency in a Distributed Database System by Use of Distributed Semaphores ", Proc. of Inter. Symposium on Distributed Databases, Versaille, France, (March 1980).

SILB82 Silberschatz, A., " A Multi-Version Concurrency Control Scheme with No Rollbacks ", Proc. ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Ontario, Canada, (Aug. 1982), 216-223.

SINH83 Sinha, M., " Constraints: Consistency and Integrity ", ACM SIGMOD RECORD, Vol.13, No.2, (Jan. 1983), 60-63.

TAYL80 Taylor, R.N. and Osteweil, L.J., " Anomaly Detection in Concurrent Software by Static Data Flow Analysis ", IEEE Trans. on Software Engineering, Vol.SE-6, No.3, (May 1980), 265-277.

URAL82b Ural, H. and Santoro, N., " Oligrachy: A General Concurrency Control for Replicated Databases ", Technical Report #TR-82-04, Computer Science Dept., Univ. of Ottawa, Ottawa, Ontario, Canada, (March 1982).

WILK81 Wilkinson, W.K., " Database Concurrency Control and Recovery in Local Broadcast Networks ", Technical Report #448, Computer Sciences Dept., Univ. of Wisconsin, Madison, Wis., USA, (Sept. 1981).

BIBLIOGRAPHY IV

DATA ENCRYPTION

ADLE79 Adleman, L., "A subexponential algorithm for the discrete logarithm problem with applications to cryptography", Proc. Twentieth IEEE Symposium on Foundations of Computer Science, (Oct. 1979), 55-60. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 172-177.

AKL 81a Akl, S.G., "On digital signatures, blindfolded arbitrators and hybrid crytosystems", Technical Report No.81-130, Dept. of Computing & Information Science, Queen's Univ., Ont., Canada, (Nov. 1981), 1-22.

AKL 82b Akl, S.G. and Taylor, P.D., " Cryptographic solution to a multilevel security problem ", Technical Report No. 82-142, Dept. of Computing and Information Science, Queen's Univ., Ont., Canada, (July 1982).

AMES77 Ames, Jr. S.R., "User interface multilevel security issues in a transaction-oriented data base management system", Proc. Trends & Applications: Computer Security and Integrity, IEEE Computer Society, (1977), 120-124. (Carleton Univ. Lib. QA 76.9 A25 T74 1977).

AMES81 Proc. 1981 Symposium on Security and Privacy, IEEE Computer Society, (1981), Chairman: S.R. Ames, Jr.

BABA80 Babad, Y.M. and Hoffer, J.A., "Data element security and its file seqmentation", IEEE Trans. on Software Engineering, Vol.SE-6, No.5, (Sept. 1980), 402-410.

BAYE76 Bayer, R. and Metzger, J.K., " On the encipherment of search trees and random access file ", ACM Trans. on Database Systems, Vol.1, No.1, (March 1976), 37-52.

BEKE83 Beker, H. and Piper, F. Cypher System, the Protection of Communications, John Wiley & Son, Inc., (Jan. 1983).

BERR81 Berry, D.M. and Monica, S., "The application of the formal development methodology to date base design and integrity verification", UCLA Computer Science Dept. Quanterly, Vol.9, No.4, (Fall 1981), 63-96.

BOEB77 Boebert, W.E., Bonneau, C.H. and Carnall, J.J., "Secure computing", Proc. Trends & Applications: Computer Security and Integrity, IEEE Computer Society, (1977), 49-63. (Carleton Univ. Lib. QA 76.9 A25 T74 1977).

BOOT81 Booth, K.S., "Authentication of signatures using public key encryption", COMM. of ACM, Vol.24, No.11, (Nov. 1981), 772-774.

BRAN75a Branstad, D.K., "Data protection through cryptography", Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VII25-28. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977)

BRAN75b Branstad, D.K., "Encryption protection in computer data communications", Proc. Fourth Data Communications Symposium, (Oct. 1975), 8-1--8-7. Also in Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VII29-35. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977)

CARR70 Carroll, J.M. and McLelland, P.M., "Fast infinite-key private transformation for resource-sharing systems ", Proc. Fall Joint Computer Conf., (1970) 223-230.

CHAM78 Chamberlin, D., Gray, J.N., Griffiths, P.P., Mresse, M. Traiger, I.L. and Wade, B.W., "Data base system authorization", Foundation Of Secure Computation, ed. Demillo, R.A., Dobkin, D.P., Jones, A.K., and Lipton, R.J., Academic Press, (1978), 39-55. (Carleton Univ. Lib. QA 76.9 A25 F66 1978)

CHEH81 Cheheyl, M.H., Gasser, M., Huff, G.A. and Millen, J.K., "Verifing security", ACM Computing Surveys, Vol.13, No.3, (Sept. 1981), 279-340. (This article includes a long list of references)

COST81 Costas, J.P., "The hand-held calculator as a cryptographic machine", Cryptologia, Vol.5, No.2, (April 1981), 94-117.

COTT75 Cotton, I.W. and Meissner, P., "Approaches to controlling personal access to computer terminals", Computer Networks Symposium, National Bureau of Standards and IEEE Computer Society, (June 1975), 32-39. Also in Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VI 42-49. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977).

DATA82 Data Star, "Unit encrypts eight bytes in 25 microseconds", IEEE Computer, (June 1982), 103.

DAVI79 Davies, D.W. and Price, W.L., "A protocol for secure communication", NPL Report NACS 21/79, (Nov. 1979).

DAVI80a Davies, D.W., Price, W.L. and Parkin, G.I., "An evaluation of public key cryptosystems", NPL Report CTU 1, (April 1980), 1-31.

DAVI80b Davida, G.I., Demillo, R.A. and Lipton, R.J., "Protecting shared cryptographic keys", IEEE Symposium, Data Security and Privacy, IEEE Computer Society, (1980), 100-102. (Carleton Univ. Lib. QA 76.9 A25 S89 1980).

DAVI80c Davies, D.W. and Price, W.L., "The application of digital signatures based on public key cryptosystems", Proc. Fifth ICCC, North Holland Publishing Co., (Oct. 1980), 525-530. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 204-209.

DAVI80d Davies, D.W. and Price, W.L., "Selected papers in cryptography and data security", NPL Report DNACS 38/80, (Nov. 1980), 1-22. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 217-238.

DAVI81a Davida, G.I., Wells, D.L. and Kam, J.B., "A database encryption system with subkeys", ACM Trans. on Database Systems, Vol.6, No.2, (Jun. 1981), 313-328.

DAVI81b Davies, D.W., "Enhancement of teletex procedures to incorporate encipherment and signatures", NPL Report DNACS 42/81, (April 1981).

DAVI81c Davies., D.W., Tutorial: The security of data in Networks , IEEE Computer Society, (1981).

DEMI78 DeMillo, R., Lipton R. and McNeil, L., "Proprietary software protection", Foundations of Secure Computation, ed. Demillo, R.A., Dobkin, D.P., Jones, A.K. and Lipton, R.J., Academic Press, (1978), 115-131.

DENN78 Denning, D.E., "A review of research on statistical data base security", Foundations of Secure Computation, ed. Demillo,R.A., Dobkin, D.P., Jones A.K. and Lipton, R.J., Academic Press, (1978), 15-25. (Carleton Uni. Lib. QA 76.9 A25 F66 1978).

DENN79 Denning, D.E. and Denning, P.J., "Data security", ACM Computing Surveys, Vol.11, No.3, (Sept. 1979), 224-249.

DENN81 Denning, D.E. and Sacco, G.M., "Timestamps in key distribution protocols", COMM. of ACM, Vol.24, No.8, (Aug. 1981), 523-536.

DIFF76 Diffie, W. and Hellman, M.E., "New directions in cryptography", IEEE Trans. on Information Theory, Vol.IT-22, No.6, (Nov. 1976), 644-654. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 135-145.

DIFF77 Diffie, W. and Hellman, M.E., "Exhaustive cryptanalysis of the NBS encryption standard", IEEE Computer, Vol.10, No.6, (June 1977), 74-84. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 75-85.

DIFF79 Diffie, W. and Hellman, M.E., "Privacy and authentication: an introduction to cryptography", Proc. of the IEEE, Vol.67, No.3, (March 1979), 397-427. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 18-48.

DOLE82a Dolev, D. and Strong, H.R., "Research report: authenticated algorithms for Byzantine agreement", IBM Research Report #RJ3416, (1982).

DOLE82b Dolev, D., "An Efficient Byzantine Agreement without Authentication", IBM Research Report #RJ3428, (1982).

EHRS78 Ehrsam, W.F., Matyas, S.M., Meyer C.H. and Tuchman, W.L., "A cryptographic key management scheme for implementing the data encryption standard", IBM Systems Journal, (1978), 106-125. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 94-114.

EVER78 Everton, J.K., "A hierarchical basis for encryption key management in a computer communication network", Proc. Trends and Appliation: Distributed Processing, (1978), 25-32. (Ottawa Univ. Vanier Lib. QA 76.9 D5 T73 1978)

FEIS75 Feistel, H., Notz, W.A. and Smith, J.L., "Some cryptographic techniques for machine-to-machine data communications", Proc. of the IEEE, Vol.63, No.11, (Nov. 1975), 1545-1554. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 49-58.

FERN81 Fernandez, E.B., Summers R.C. and Wood, C. Database Security and Integrity, Addison-Wesley, (1981).

FRAN77 Franking, N.A., Nellen, E.P., Inselberg A.D. and Olson, A.K., "Providing data integrity and security through software interfaces", Proc. Trends and Applications: Computer Security and Integrity, IEEE Computer Society, (1977), 102-105. (Carleton Univ. Lib. QA 76.9 A25 T74 1977).

GAIT82 Gait, J., "Universal test sets for the standard encryption algorithm", IEEE Trans. on Reliability, Vol.R-31, No.1, (April 1982), 5-8.

GIFF82 Gifford, D.K., "Crytographic sealing for information secrecy and authentication", COMM. of ACM 25, 4 (April 1982), 274-286.

GLIG79 Gligor, V.D. and Lindsay, B.G., "Object migration and authentication", IEEE Trans. on Software Engineering, Vol.SE-5, No.6, (Nov. 1979), 607-611.

GOLD81 Goldsmith, L.H., "Dynamic protection of objects in a computer utitity", Technical Report CSRG130, Computer Science Research Group, Univ. of Toronto, (1981), Ont., Canada.

GUDE76 Gudes, E., Koch, H.S. and Stahl, F.A., "The application of cryptography for database security", Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VII3-12. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977)

GUDE80 Gudes, E., "The design of a cryptography based secure file systems", IEEE Trans. on Software Engineering, Vol.SE-6, No.5, (Sept. 1980), 411-419.

HERL78 Herlsetam, T., "Critical remarks on some public-key cryptosystems", BIT, Vol.18, (1978), 493-496. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society (1981), 198-201.

HOFF77 Hoffman, L.J., Modern Methods for Computer Security and Privacy, Prentice Hall, (1977). (Carleton Univ. Lib. QA 76.9 A25 H63 1977).

HSIA79a Hsiao, D.K., Kerr D.S. and Madnick, S.E., "Cryptographic transformations",, Computer Security, ACM Monograph Series, Academic Press, (1979), 135-164. (Ottawa Univ. Vanier Lib. QA 76.9 A25 H74 1979).

HSIA79b Hsiao, D.K., Kerr, D.S. and Madnick, S.E., "Database security", Computer Security, ACM Monograph Series, Academic Press, (1979), 221-275. (Ottawa Univ. Vanier Lib. QA 76.9 A25 H74 1979).

KAM 78 Kam, J.B. and Davida, G.I., "A structured design of substitution - permutation encryption network", Foundations of Secure Computation, ed. Demillo, R.A., Dobkin, D.P., Jones, A.K. and Lipton, R.J., Academic Press, (1978), 95-113. (Carleton Univ. Lib. QA 76.9 A25 F66 1978).

KENT77 Kent, S.T., "Encryption-based protection for interactive user/computer communication", Proc. Fifth Data Communications Symposium, (Sept. 1977), 5-7--5-13. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 87-93.

KENT81 Kent, S.T., "Security requirements and protocols for a broadcast scenario", IEEE Trans. on Computers, Vol.C-29, No.6, (June 1981), 778-786.

KOUH81 Kouheim, A., Cryptology: A Primer, John-Wiley and Sons, (1981).

LAND81 Landwehr, C.E., "Formal models for computer security", ACM Computing Surveys, Vol.13, No.3, (Sept 1981), 247-278. (This article includes a long list of references)

LEHM82 Lehmann, R.L., "Tracking potential security violations", ACM SIG Security Audit & Control Review, Vol.1, No.1, (Winter 1981), 26-39.

LEMP79 Lempel, A., "Cryptology in transition", ACM Computing Surveys, Vol.11, No.4, (Dec. 1979), 285-303.

LENN81 Lennon, R.E., Matyas, S.M. and Meyer, C.H., "Cryptographic authentication of time-invariant quantities", IEEE Trans. on Computers, Vol.C-29, No.6, (June 1981), 773-777.

LIEN74 Lientz, B.P. and Weiss, I.R., "On the evaluation of reliability and security measures in a computer network", Office of Naval Research, Arlington, Va., (Dec. 1974), 28. Also in Tutorial on Computer

Security and Integrity, IEEE Computer Society, (1977), VI17-41. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977).

LIPN75 Lipner, S.B., "Secure computer systems for network applications", Fourth Data Communications Symposium, (Oct. 1975), 8-8--8-12. Also in Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VI12-16.

MACE81 MacEwen, G.H., "The design for a secure system based on program analysis", Technical Report No.81-118, Dept. of Computing & Information Science, Queen's Univ., Ontario, Canada, (July 1981), 1-28.

MART73 Martin, J., Security, Accuracy and Privacy in Computer Science , Prentice-Hall (1973).

MEIJ81 Meijer, H. and Akl, S., "Digital signature schemes for computer communication networks", Proc. Seventh Data Communications Symposium, (1981), 37-43.

MELL73 Mellen, G.E., " Cryptology, computers and common sense ", Proc. FIPS, Vol.42, (1973), 569-579.

MERK78a Merkle, R.C., "Secure communications over insecure channels", COMM. of ACM, Vol.21, No.4, (April 1979), 294-299.

MERK78b Merkle, R.C. and Hellman, M.E., "On the security of multiple encryption", COMM. of ACM, Vol.24, No.7, (July 1981), 465-467.

MERK78c Merkle, R.C. and Hellman, M.E., "Hiding information and signatures in trapdoor knapsacks", IEEE Trans. on Information Theory, Vol.IT-24, No.5, (Sept. 1978), 525-530. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 152-157.

MERK80 Merkle, R.C., "Protocols for public key cryptosystems", IEEE Symposium, Data Security and Privacy, IEEE Computer Society, (1980), 122-136. (Carleton Univ. Lib. QA 76.9 A25 S89 1980).

MEYE81 Meyer, C.H. and Matyas, S.M., "Some cryptographic principles of authentication in eletronic fund transfer systems", Proc. of Seventh Data Communications Symposium, (1981), 73-88.

MEYE82 Meyer, C. and Matyas, J.M., Cryptology: A New Dimension in Computer Data Security , John-Wiley and Sons, (1982).

MILL77 Mill, J.K., "Formal specifications for security", Proc. Trends & Applications: Computer Security and Integrity, IEEE Computer Society, (1977), 115-119. (Carleton Univ. Lib. QA 76.9 A25 T74 1977).

MIRA77 Miranda, S.M., "Aspects of data security in general purpose data bases", Proc. of Data Security and Privacy, IEEE Computer Society, (1980), 46-58. (Carleton Uni. QA 76.9 S89 1980).

NBOS77 U.S. Department of Commerce, National Bureau Of Standards, "Encryption algorithm for computer data protection", FIPS 46. Also in Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VII14-24. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977).

NEED78 Needham, R.M. and Schroeder, M.D., "Using encryption for authentication in large networks of computers", COMM. of ACM, Vol.21, No.12, (Dec. 1978), 993-999. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 210-216.

NELS77 Nelson, R. and Jarzembowski, J., "Multi-level security: an overview and new directions", Proc. Trends & Applications: Computer Security and Integrity, IEEE Computer Society, (1977), 41-48. (Carleton Univ. Lib. QA 76.9 A25 T74 1977).

PIRO77 Pirola, G.C. and Sanguienetti, J.W., "The protetion of information in a general purpose time-sharing environment", Proc. Trends & Appications: Computer Security and Integrity, IEEE Computer Society, (1977), 106-114. (Carleton Univ. Lib. QA 76.9 A25 T74 1977).

POHL78 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, No.1, (Jan. 1978), 106-110. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 167-171.

POPE78 Popek, G.J. and Kline, C.S., "Encryption protocols, public key algorithms, and digital signatures in computer networks", Demillo, R.A., Dobkin, D.P., Jones, A.K. and Lipton,R.J, Foundations of Secure Computation, Academic Press, (1978), 133-154. (Carleton Univ. Lib. QA 76.9 A25 F66 1978).

POPE79 Popek, G.J. and Kline, C.S., "Encryption protocols, public key networks", ACM Computing Surveys, Vol.11, No.4, (Dec. 1979), 331-356.

PRIC81a Price, W.L. and Davies, D.W., "Issues in the design of a key distribution centre", NPL Report DNACS 43/81, (April 1981), 1-14.

PRIC81b Price, W.L., "Data security in distributed computing systems", Proc. NATO Conf., New Advances in Distributed Computer Systems, ed. Beauchamp, K.G., (1981), 279-292.

PRIC81c Price, W.L., "The data encryption standard and its modes of use", Proc. NATO Conf., New Advances in Distributed Computer Systems, ed. Beauchamp, K.G., (1981), 293-310.

PRIC81d Price, W.L., "Standardization and implementation of data encryption", Proc. NATO Conf., New Advances in Distributed Computer Systems, ed. K.G. Beauchamp, (1981), 311-326.

PRIC81e Price, W.L., "Public key cryptosystems, authentication and signatures", Proc. NATO Conf., New Advances in Distributed Computer Systems, ed. Beauchamp, K.G., (1981), 327-340.

RABI78 Rabin, M.O., "Digitalized signatures", Foundations of Secure Computation, ed. Demillo, R.A., Dobkin, D.P., Jones, A.K. and Lipton, R.J., Academic Press, (1978), 155-168. (Carleton Univ. Lib. QA 76.9 A25 F66 1978).

RIVE78a Rivest, R.L., Shamir, A. and Adleman, L., "A method for obtaining digital signatures and public key cryptosystems", COMM. of ACM, Vol.21, No.2, (Feb. 1978), 120-126.

RIVE78b Rivest, R.L., Adleman, L. and Dertouzos, M.L., "On data banks and privacy homomorphisms", Foundations of Secure Computation, ed. R.A. Demillo, D.P. Dobkin, A.K. Jones and R.J. Lipton, Academic Press, (1978), 169-179. (Carleton Univ. Lib. QA 76.9 A25 F66 1978).

RIVE78c Rivest, R.L., "Remarks on a proposed cryptanalytic attack on the M.I.T. public-key cryptosystem", Cryptologia, Vol.2, No.1, (Jan. 1978), 62-65. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 194-197.

RIVE79 Rivest, R.L., "Critical remarks on "critical remarks on some public-key cryptosystems" by T. Herlestam", BIT, Vol.19, (1979), 274-275. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 202-203.

SHAM79 Shamir, A., "How to share a secret", COMM. of ACM, Vol.22, , No.11, (Nov. 1979), 612-613.

SHAM80 Shamir, A. and Zippel, R.E., "On the security of the Merkle-Hellman cryptographic scheme", IEEE Trans. on Information Theory, Vol.IT-26, No.3 , (May 1980), 339-340. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 165-166.

SIMM77 Simmons, G.J. and Norris, M.J., "Preliminary comments on the M.I.T. public-key cryptosystem", Cryptologia, Vol.1, No.4, (Oct. 1977), 406-414. Also in Tutorial: The Security of Data in Networks, IEEE Computer Society, (1981), 185-193.

SIMM79 Simmons, G.J., "Symmetric and asymmetric encryption", ACM Computing Surveys, Vol.11, No.4, (Dec. 1979), 305-330.

SMID81 Smid, M.E., "Integrating the data encryption standard into computer networks", IEEE Trans. on Computers, Vol.C-29, No.6, (June 1981), 762-772.

SMIT72 Smith, J.L., Notz, W.A. and Osseck, P.R., "An experimental application of cryptography to a remotely accessed data system", Proc. ACM 27th National Conf., (1972), 282-297.

STAH77 Stahl, F., Gudes, E. and Koch, H., "The coordination of cryptographic and traditional access control techniques for protection in computer systems", Proc. Trends and Applications: Computer Security and Integrity, IEEE Computer Society, (1977), 86-91. (Carleton Uni. Lib. QA 76.9 A25 T74 1977).

TURN73 Turn, R., "Privacy transformations for databank systems", Proc. AFIPS, Vol.42 , (1973), 589-601.

WILM81 Wilms, P.F. and Lindsay, B.G., "A database authorization mechanism supporting individual and group authorization", IBM Research Report #RJ3137, (May 1981).

WINK74 Winkler, S. and Lee, D., "Data security in the computer communication environment", Computer, (Feb. 1974), 23-31. Also in Tutorial on Computer Security and Integrity, IEEE Computer Society, (1977), VI3-11. (Ottawa Univ. Vanier Lib. HF 5548.2 T88 1977).

BIBLIOGRAPHY V

HUMAN FACTORS METHODOLOGY FOR DATABASE QUERY LANGUAGES

ATWO79 Atwood, M.E., Ramsey, H.R., Hooper, J.N. and Kullas, D.A., "Annotated bibliography on human factors in software development", ARI Technical Report 79-1, U.S., Army Research Institute, Alexandria, Va., USA, (June 1979).

BROS78 Brosey, M. and Shneiderman, B., "Two experimental comparisons of relational and hierarchical database models", Int. J. Man-Machine Studies, Vol.10, (1978), 625-637.

CARD80 Card, S.K., Moran, T.P. and Newell, A., "The keystroke-level model for user performance time with interactive systems", COMM. of ACM, Vol.23, (July 1980), 396-410.

CHAP72 Chapanis, A. and Van Cott, H.P., "Human engineering tests and evaluations", Human Engineering Guide to Equipment Design, American Institutes for Research, Wash. D.C., (1972).

DEGR70 De Greene, K.B., Systems Psychology, McGraw-Hill, New York, (1970).

DURD77 Durding, B.M., Becker, C.A. and Gould, J.D., "Data organization", Human Factors, Vol.19, (1977), 1-14.

GOUL75 Gould, J.D. and Ascher, R.N., "Use of an IQF-like query language by non-programmers", IBM Research Report #RC5279, (Feb. 1975).

GREE78 Greenblatt, D. and Waxman, J., "A study of three database query languages", Databases: Improving Usability and Responsiveness, Academic Press, N.Y., (1978).

KLUG82 Klug, A., "A query language for constructing aggregates-by-example", Technical Report #475, Computer Science Dept., Univ. of Wisconsin-Madison, Wis., USA, (May 1982).

LATR81 Latremouille, S. and Lee, E., "The design of Telidon tree indexes: Improving tree indexes by testing naive users", Technical Memorandum, Dept. of Communications, Ottawa, Canada, (1981).

LEE80 Lee, E. and Latremouille, S., "Evaluation of tree structured organization of information on Telidon", Technical Memorandum No.BRG79-12, Dept. of Communications, Ottawa, Canada, (Dec. 1979), 231-242.

LOCH77 Lochovsky, F.H. and Tsichritzis, D.C., "User performance considerations in DBMS selection", Proc. ACM SIGMOD Inter. Conf. on Management of Data, (1977), 128-134.

LOCH78 Lochovsky, F.H., Data Base Management System User Performance, Ph.D thesis, University of Toronto, Toronto, Ont., Canada, (1978).

MAYE79 Mayer, R.E., "A psychology of learning BASIC", COMM. of AMC, Vol.22, No.11, (Nov. 1979), 589-593.

MAYE81 Mayer, R.E., "The psychology of learning computer programming by novices", ACM Computing Surveys, Vol.13, No.1, (Mar. 1981).

MCCO70 McCormick, E.J., Human Factors Engineering, McGraw-Hill, N.Y., (1970).

MCEW81 McEwen, S.A., "An investigating of user search performance on the Telidon information retrieval system", Technical Memorandum, Dept. of Communication, Ottawa, (1981).

MILL81 Mills, M.I., "A study of the human response to pictorial representations on Telidon", Telidon Behavioural Research 3, Dept. of Communications, Ottawa, Canada, (1981).

PHIL81 Phillips, D., "The design of videotex tree indexes", Telidon Behavioural Research 2, Dept. of Communications, Ottawa, Canada, (May 1981).

RAMS78 Ramsey, H.R., Atwood, M.E. and Kirshbaum, P.J., A Critically Annotated Bibliography of the Literature of Human Factors of Computer Systems, Science Applications Inc., Englewood, Colorado, USA, (May 1978).

REIS75 Reisner, P., Boyce, R.F. and Chamberlin, D.D., "Human factors evaluation of two data base query languages - Square and Sequel", Proc. of NCC, Montvale, AFIPS Press, (1975), 447-452.

REIS77 Reisner, P., "Use of psychological experimentation as an aid to development of a query language", IEEE Trans. on Software Eng., Vol.SE-3, (May 1977), 218-299.

REIS81 Reisner, P., "Human factors studies of database query languages: A survey and assessment", IBM Research Report #RJ3070, (March 1981).

RMS83 Relational Manual System, Hello Software, 8380 Roanne Drive, Orlando, Florida, USA, (1983).

SHNE78 Shneiderman, B., "Improving the human factors aspect of data base interactions". ACM Trans. on Database Systems, Vol.3, (Dec. 1978), 417-439.

THOM75 Thomas, J.C. and Gould, J.D., "A psychological study of query by example", Proc. of NCC, Montvale, AFIPS Press, (1975), 439-445.

THOM77 Thomas, J.C., "Psychological issues in data base management", Proc. Third Inter. Conf. on Very Large Data Bases, Tokyo, (Oct. 1977), 169-185.

THOM80 Thomas, J.C., "Psychological issues in the design of data-base query languages", Communication with Computers, ed. M. Sime and M. Fitter, Academic Press, London, (1980).

WELT79 Welty, C., A Comparison of a Procedural and a Nonprocedural Query Language: Syntactic Metrics and Human Factors, Ph.D thesis, University of Massachusetts, Amherst, (May 1979).

WELT81 Welty, C. and Stemple, D.W., "Human factors comparison of a procedural and a non-procedural query language ", ACM Trans. on Database Systems.

WHAL80 Whalen, T. and Latremouille, S., "The effectiveness of a tree-structured index when the existence of information is uncertain", Technical Memorandum No.BRIC80-3, Dept. of Communications, Ottawa, Canada, (Sept. 1980).

ZLOO75 Zloof, M.M., "Query by example", Proc. of NCC, Montvale, AFIPS Press, (May 1975), 431-437.