Integrated Voice/Data Network Design

Michael Robert Payette

A Major Technical Report

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

April 1987

ISBN 0-315-49063-2

ABSTRACT

Integrated Voice/Data Network Design

Michael Robert Payette

This report describes a heuristic design procedure for the design of integrated voice/data networks under fixed partitioning of bandwidth. The design heuristic is based on a channel routing heuristic developed for the design of voice networks. The heuristic is described in detail, and examples are given to illustrate its operation. The heuristic is used to assess the economic leverage of a new integrated voice/data private line service.

ACKNOWLEDGEMENTS

I wish to thank Dr. T. Radhakrishnan for his supervision of this report. His direction and encouragement through the completion of my work were invaluable. Thanks to Nathalie Rico, Larry Dunkelman, and Patrick Smith for the many valuable discussions concerning the direction of the work. Also, a special thanks to my family for their support and encouragement. I am indebted to the National Science and Engineering Council for their generous financial support.

TABLE OF CONTENTS

1.0 Introduction **1**

1.1 Private Networks 1

1.2 Integration of Voice and Data 3

1.3 Overview of the Report 4

2.0 Background **6**

2.1 Voice Network Design 6

2.1.1 Voice Networks 6

2.1.2 The Voice Network Design Problem 9

2.1.3 Voice Network Design Process 10

2.2 Data Network Design 16

2.2.1 Data Networks 16

2.2.2 The Data Network Design Problem 18

2.2.3 Data Network Design Process 19

2.3 Integrated Networks 26

2.3.1 Digital Technology 26

2.3.2 Types of Integration 30

2.3.3 Integrated Network Design 31

3.0 The Problem **37**

3.1 The Approach 37

3.2 Problem Formulation 40

3.3 The Integrated Service 40

4.0 The Solution **43**

4.1 Channel Routing 43

4.1.1 Problem Statement 43

4.1.2 Procedure 44

4.1.3 Run Time Analysis 46

4.1.4 Cost Improvement Analysis 47

4.2 Modifications 47

4.3 The Method 49

4.3.1 Starting Solution 51

4.3.2 Rerouting of Circuits 53

4.3.3 Rerouting of Links 53

4.3.4 Shortest Path 54

4.3.5 Bin Packing 55

4.4 Network Design Example 58

5.0 Analysis **63**

5.1 Network Designs 63

5.2 Run-time Analysis 74

5.2.1 Software Environment 74

5.2.2 Run-time vs Network Size 75

5.2.3 Run-time Improvements 78

5.3 Cost Savings Analysis 81

5.3.1 Cost Savings for Each Step 81

5.3.2 Savings Due to Integration 82

6.0 Future Work **86**

6.1.1 Multiplexer Placement 86

6.1.2 T1 Modularity 88

6.1.3 Iteration Over the Design 89

6.1.4 Network Reconfiguration 90

7.0 Summary **91**

8.0 References **93**

LIST OF ILLUSTRATIONS

Figure 1. Typical Route List 8

Figure 2. Tandeming 11

Figure 3. Multipoint Tree Topology 17

Figure 4. Pseudocode for Line Layout Algorithm 23

Figure 5. Steiner Point Example 24

Figure 6. Time Division Multiplexing 28

Figure 7. T1 Frame 29

Figure 8. Voice/Data Multiplexing 30

Figure 9. General Approach to IVD Network Design 39

Figure 10. Integrated Service Structure 42

Figure 11. Algorithm Steps 50

Figure 12. Algorithm Pseudo-code 52

Figure 13. Bin Packing Example 58

Figure 14. Requirements 59

Figure 15. Initial Solution for Example 60

Figure 16. Example After Rerouting of Routes 61

Figure 17. Final Solution for Example Network 62

Figure 18. Requirements for Example 1 64

Figure 19. Network Design for Example 1 65

Figure 20. Requirements for Example 2 67

Figure 21. Network Design for Example 2 68

Figure 22. Requirements for Example 3 71

Figure 23. Network Design for Example 3 72

Figure 24. CPU Tıme 76

Figure 25. Normalized CPU Time 77

Figure 26. Decomposıtion of the Network into Rate Centers 80

Figure 27. Algorithm Cost Savings 83

Figure 28. Cost Savings Due to Voice/Data Integratıon 85

Figure 29. Placement of the IVD Multiplexer 87

1.0 INTRODUCTION

1.1 Private Networks

Corporations or enterprises with many locations generally have private voice and data networks to support communication between the various locations. These networks consist of equipment at the locations, and communications links connecting the equipment. These communications links are often transmission facilities leased from the telephone company (telco).

In a private voice network, the equipment is usually a PBX (Private Branch eXchange) or Centrex. The PBX is the device that provides switched communication within a building. Centrex is a telco offered service which provides the same functionality as the PBX.

There are three basic types of services available to interconnect locations in a private voice network. Private line service provides a dedicated facility between two PBXs. WATS (Wide Area Telephone Service) provides communications between a network location and a geographic area. DDD (Direct Distance Dialing) is the use of the (pay as you use) public long distance network.

Hence, in private voice network design, one attempts to determine the correct number of transmission facilities of each type of service to connect the network, such that the grade of service (maximum percentage of calls blocked) is met while the monthly cost is minimized.

In a private data network, the equipment includes devices such as terminals, multiplexers, data switches, and computer hosts.

There are two basic service types provided for private data networks. Multipoint private lines (MPL) provide a dedicated connection between two or more devices. They are available at different transmission speeds (bits per second - bps), depending on the application. Packet switched service provides virtual connections between devices. We consider only MPL type networks in this report. There are two basic service types available for such a network. Analog private lines, through the use of modems, provide data communication at transmission speeds from 300 to 9600 bps. Digital private lines provide digital data transmission at speeds from 300 bps to 56K bps. A typical MPL data network would be composed of both analog and digital private lines, depending on the availability and cost of these services.

The objective, in MPL data network design is to determine the topology (number of circuits, number of terminals on each circuit, etc.) and transmission speeds such that the performance requirements are met (response time, throughput) for all devices, and the monthly cost is minimized.

Hence, the type of private network considered here consists of customer owned equipment to be connected by transmission facilities leased from the telco(s) on a monthly basis. We exclude transmission facilities owned by the customer. The objective of the design process is to minimize the monthly cost of leasing these facilities, while satisfying the network requirements.

1.2 Integration of Voice and Data

The characteristics of voice and data traffic, such as arrival rate, holding time, and performance criteria are quite different. Hence, the communication requirements for voice and data networks are quite different. Also, in the past, the transmission facilities underlying the telco service offerings were dedicated to voice or data. They could not transmit voice and data simultaniously, resulting in a set of services for voice and a different set of services for data. For these and other reasons, the voice and data networks were designed independently. Methods were developed for voice network design, which take into account the particular characteristics of voice traffic and voice network services. Very different methods were developed for data network design. This was a reasonable approach (separating voice and data network design) as the voice and data networks were virtually independent.

Advances in digital communications have led to transmission facilities which can carry voice and data simultaneously. This is reflected in new private network service offerings called integrated services which can be leased from the telco and used to carry both voice and data traffic simultaniously.

A great deal of effort is now going into the definition of standards for an Integrated Services Digital Network (ISDN). ISDN is defined as providing end-to-end digital connectivity supporting a wide range of voice, data, and video services, accessed through a small number of standard interfaces. The ISDN will evolve from the Integrated Digital Network (IDN). The key component then of ISDN versus IDN is services. While it is not clear at this time what these services will look like, the emergence of the IDN is resulting in new (pre-ISDN) private network services, called integrated services.

While these new services may not conform to the ISDN standards, they introduce new complexity into the private network design problem. Because these new services carry voice and data simultaneously, it is no longer valid to partition the network design process into voice network design and data network design. New techniques must be developed to design the integrated voice/data network.

This report describes an experimental procedure for the design of integrated voice/data networks with fixed partitioning of bandwidth. It is built upon an existing voice and data network design system, by introducing an additional channel routing heuristic which combines the voice and data network designs into an integrated voice/data network design. The heuristic has been implemented, and we report on the sample networks which have been designed.

1.3 Overview of the Report

This report is divided into five main chapters. Chapter 2, "Background", describes the voice and data network design problems, and voice and data network design methodologies. In addition, it describes integrated voice/data networks, the integrated network design problem and some work that has been done towards solving this problem. Chapter 3, "The Problem", describes the particular problem in integrated voice/data network design addressed in this report. The solution examined is adapted from an heuristic algorithm developed to determine near optimal channel routing in voice networks. The approach and the adaptations are described in chapter 4, "The Solution". Chapter 5, "Analysis", examines the results obtained from the heuristic. The heuristic is assessed in terms of the amount of CPU time required to arrive at the solution, and the cost savings achieved with the networks designed by the heuristic. Then, using the heuristic, we try to estimate the potential cost savings for private network customers due to the integration of voice and data onto one network.

2.0 BACKGROUND

This section describes the voice, data, and integrated voice data network design problems. The voice network and data network design methodologies developed at Bell-Northern Research are described. These are important because they are used in the integrated network design. Also, some related work in Integrated Voice Data (IVD) network design is described.

2.1 Voice Network Design

This section describes the problem addressed in voice network design, and the voice network design process, PNDS (Private Network Design System) developed at Bell-Northern Research [DUNK 86]. This process will form the first component of the integrated network design system.

2.1.1 Voice Networks

As mentioned briefly, a voice network consists of locations to be interconnected and facilities leased from the telco, interconnecting the locations.

Network locations can either be PBXs (Centrex) or NPAs (Numbering Plan Areas). The main purpose of the **PBX** is to provide the telephone switching within a building. In addition, the PBX provides access to the public network and to WATS and private line facilities.

NPAs are telephone codes corresponding to geographical areas. Thus, NPA 514 includes the southern portion of Quebec. There may be a communication requirement between an office and a geographic area, consisting of one or more NPAs.

As mentioned, there are three basic private voice network services, private line, direct distance dialing and WATS. A **private line** provides a dedicated connection between two PBXs, at a fixed monthly cost. That is, the cost is independent of how much the line is used. Therefore, the private line is a good economic choice if it will be used a lot, but a bad choice if it is under utilized.

A variation of the private line service is **Teleroute 200**. Teleroute uses special voice compression techniques to put two voice conversations on one private line. The result is a cost savings on the transmission facility (about half) at the expense of paying for the extra end equipment to do the compression.

Direct Distance Dialing (DDD) is the use of the public long distance network. Use of DDD is billed per call, according to the length (duration) of the call, and the distance between the origin and destination. It is appropriate when there is only a low volume of traffic.

Wide Area Telephone Service (WATS) provides communication between a location and a geographic area. For any given location, the country is divided into zones 1 to 6, which are like concentric circles. Each zone gives access to successively larger areas. Charging is a combination of fixed and usage sensitive. The customer pays a flat rate for the first n hours (n=5,10, or 160) and then pays an overtime rate for any usage over the base time of n hours.

An important feature of private networks which has a big impact on private network design is **Automatic Route Selection (ARS).** Modern PBXs have the ability to determine which route a call will take to get from the origin to the destination, based on the actual network conditions.

```
From:  Montreal  To:  Toronto

     1) PL -> Toronto
     2) PL -> Ottawa
     3) DDD -> Toronto
```

Figure 1. Typical Route List: This figure shows a typical route list that might be found in a Montreal PBX. A call from Montreal to Toronto first attempts to complete directly via private line (PL). If the private line is busy, the PBX will try the private line to Ottawa. The PBX in Ottawa will have a suitable route list entry to tell it how to complete the call to Toronto. As the third and last choice, long distance is used.

The PBX will have a set of route lists, one for each destination that can be reached from the PBX, and one for each origin-destination pair that might use the PBX as an intermediate point. The route list tells the PBX where to send the call, that is, to which location (final or intermediate) and using which facility (private line, WATS, or DDD). The route list will generally contain a number of possibilities (see Figure 1). The PBX will try the first, if it is not available (busy), it will try the second, and so on, until an available facility is found or the call is blocked (all entries in the route list are busy).

2.1.2 The Voice Network Design Problem

The major inputs to the design process are the locations, the traffic, and the grade of service.

The **locations** are all of the customer locations and NPAs which are either the origin or destination of traffic to be carried on the network.

The **traffic** is in the form of volume (Erlangs) [1] of traffic between all pairs of locations during the network busy hour.

The **grade of service (GOS)** is specified as the percentage of blocked calls that will be tolerated on the network.

The outputs consist primarily of the route lists and facility sizes. For each PBX, for each origin-destination pair, the **route list** describes how calls should be routed. This is basically the topology ot the network.

[1] The unit of voice traffic intensity is the Erlang. One Erlang of traffic is one hour of calling in one hour

The **facility sizes** are the number of trunks for each origin-destination pair, for each facility type.

2.1.3 Voice Network Design Process

The voice network design process consists of four major steps.

1. Traffic Matrix Generation

   The traffic is either computed from a "community of interest matrix", estimated by the customer, or determined from measurements on the existing customer network (network is being re-designed).

2. Tandem Selection

   Some PBXs are designated as tandems, such that traffic can pass through these tandems as intermediate points, on route to some other point (see Figure 2). The tandem selection process determines how many tandems the network should contain and where the tandems should be located to minimize the network cost.

   The tandems are selected from a set of tandem candidates. The network designer chooses the number of candidates, N, and the candidates are determined using the following nearest neighbor approach. Based on a distance function defined as the cost of a private line between the two locations divided by the volume of traffic between the locations, each location votes for its N nearest neighbors (N is the

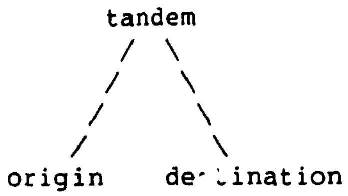

Figure 2. Tandeming: A voice call from origin to destination is routed (switched) through an intermediate point (tandem).

number of candidates to be selected). After all voting, the N locations with the largest number of votes are selected as the tandem candidates. In other words, a good tandem candidate is one which is the nearest neighbor of many other locations.

To select the tandems from the set of tandem candidates, a dynamic programming approach is adopted. First, the N possible one tandem solutions are costed and stored. Next, the algorithm considers adding each remaining tandem to each of the N one tandem networks, and saves the N cheapest two-tandem networks. From this point on, the search is not exhaustive. The algorithm tries to add each of the N tandem candidates to each of the N two-tandem solutions. So starting with candidate #1, it examines the N two-tandem solutions and tries to create three-tandem solutions. It then chooses the cheapest feasible three-tandem solution. Processing the other candidates in the same way results in N three-tandem solutions.

This process is repeated, creating 4, 5, etc. tandem solutions, until the network cost can no longer be reduced by adding a tandem. The algorithm returns the set of tanderns that resulted in the lowest cost network.

A post processing step is applied in which a pairwise exchange is made between each of the selected tandems and all other tandem candidates. An exchange is retained if it lowers the network cost, and rejected otherwise. The exchange process continues until the network cost can no longer be reduced.

3. Route List Generation

The route list generation process determines, for each origin-destination pair, what routes should be considered and in what order the routes should be tried in routing the traffic.

This is basically the network topology as it includes which facilities are to be used to connect each pair of points. It does not include the size (number of trunks) of the facilities.

Route lists are generated from a set of user defined route list rules. A particular route list rule defines the route lists for a particular class of origins to a particular class of destinations, satisfying certain relationships.

The origin and destination classes are defined by the user, through a set of attributes, including:

- Switch level (tandem or not),
- whether or not the location has ARS,
- customer location or NPA.

Relationships which can be defined between origin and destination include:

- The two locations are homed to the same switch,
- one location is an ancestor of the other,
- one location is a descendant of the other,
- number of generations apart,

and <, >, and = relations between the location class attributes.

Hence, an example of a route list rule would be: If the origin has ARS and the destination has ARS and the origin and destination are homed to the same tandem, the route list would be:

```
1) private line to the destination
2) private line to the home of the destination
3) WATS to the destination
4) DDD to the destination
```

All origin-destination pairs which satisfy these conditions will have the above route list. If the origin and destination had different home switches, the route list might be:

1) private line to the destination
2) private line to the home of the destination
3) private line to the home of the origin
4) WATS to the destination
5) DDD to the destination

Once the user has specified all of their rules, the route list generation is a simple matter of performing a table lookup, for each origin-destination pair. The correct rule is found and the route list is generated. After the route list generation process, the user is able to edit the route lists manually, to make any desired corrections.

4. Dimensioning

At this point in the process, we know the locations to be interconnected, the traffic between the locations, the routes to be considered and the transmission facilities which will be used to interconnect the network locations.

What remains to be determined is the size (number of trunks) of each facility. This is done heuristicly, based on the costs and efficiencies of alternate routes.

The heuristic is based on a generalization of the Economic CCS (ECCS) method of sizing trunk groups in hierarchical networks [TRUI 54]. Stated simply, the ECCS method balances the cost of carrying traffic on one route (called the HU - High

Usage route) against the cost of carrying it on some more expensive, but more efficient alternative route (AR route). The correct balance is achieved when

$$\frac{\text{AR cost}}{\text{HU cost}} = \frac{\text{AR efficiency}}{\text{HU efficiency}}$$

Since the costs are known, to determine the desired efficiency of the HU, we need to know the efficiency of the AR. In hierarchical networks, this is assumed to be 28 CCS per trunk. In a private network, the assumption is not necessarily valid (private networks are not necessarily hierarchical and the volume of traffic is considerably lower). This problem is approached by introducing an iteration into the process. At the Nth iteration, the efficiencies of the alternate routes are taken from the previous iteration. ECCS is applied and the trunk group is sized. Given the new sizes, the traffic flows in the network are determined and the new efficiences computed. These efficiencies are used in the next iteration. After each iteration, the network is costed. The iteration stops when the network cost can no longer be reduced, or some maximum number of iterations have been performed.

As the efficiencies are not known initially, the first few iterations assume an efficiency of 65%.

The result at the end of this step is the fully defined voice network. This includes how many of each facility will be used between each pair of locations.

The portion of the solution that is required as input to the integrated network design heuristic is the private line requirements. It is these private lines that can be integrated with data circuits on an integrated service.

2.2 Data Network Design

This section describes the problem addressed in data network design, and the data network design process, DTS (Data Tool System), developed at Bell-Northern Research [KENN 86]. This process will form the second component of the integrated network design system.

2.2.1 Data Networks

The type of network considered here is the centralized computer network. In this problem, there are a number of devices, such as terminals, controllers, and printers, to be connected to the computer host by multipoint lines. The object is to design a minimum cost network which meets the performance requirements.

The performance requirements are specified as the average delay (response time) of a message. This can be specified for each device and each type of message.

Perhaps the simplest topology is the star network, in which every device is connected directly to the host. This has the advantage of very fast response but the disadvantage of being quite expensive. More commonly, several devices are put on a single (multipoint) line, giving the network a tree structure (see Figure 3). The host or concentrator polls (requests information from) each device on the line, so the devices take turns transmitting information. This reduces network cost by allowing devices to share transmission facilities, but increases the response time as devices have to wait for their turn.

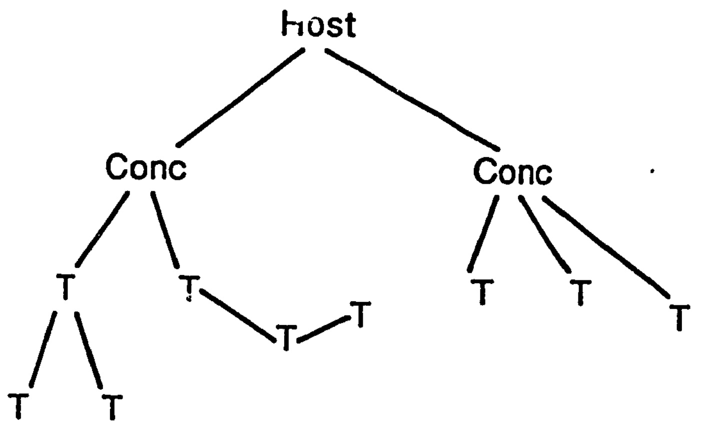

Figure 3. Multipoint Tree Topology: The use of multipoint lines give a network a tree topology.

Also, concentrators can be used to aggregate several low speed lines onto a higher speed line. This is advantageous for two reasons. First, the cost per bps is generally lower for higher speed lines. Thus, one 4800 bps line is generally cheaper than four 1200 bps iines. Second, statistical multiplexors can use the statistical nature of the traffic (the fact that not all devices need to transmit at the same time) to reduce the bandwith required. Thus, perhaps six 1200 bps lines can be multiplexed onto one 4800 bps line.

The lines themselves can be analog or digital. With the inclusion of modems and bridging equipment, analog private lines can be used for multipoint data circuits at transmission speed from 300 to 9600 bps. Digital multipoint private lines are provided under a service called Dataroute. These lines are available at different transmission speeds, 300, 1200, 2400, 4800, 9600, 19.2K, and 56K bps.

2.2.2 The Data Network Design Problem

The main inputs to the data network design process are the locations and characteristics of the devices, the traffic and the delay requirements.

The **devices** are all of the pieces of equipment to be connected on the network. This includes the host, controllers, terminals, printers and concentrators.

The **traffic** is a description of all transactions to be sent between each device and the host. Each device may send or receive several types of transactions. A single transaction can consist of several messages sent back and forth between the device and the host, and can include CPU processing time, if desired. Each message is described by a histogram, giving the different lengths (in characters) that the message can assume, and the frequency with which the message assumes each of its possible lengths.

The **delay requirement** is the maximum tolerable delay (in milliseconds) that a message should experience between each device and the host.

The main outputs from the data network design process are the concentrator locations, the network topology, the line speeds, and the expected performance of all circuits.

The number of **concentrators** required in the network and their locations are determined.

The network **topology** will consist of a number of multipoint lines, each line with one or more device on it.

The **line speed (bps)** of each line is the transmission speed of the line.

2.2.3 Data Network Design Process

The data network design process consists of four major steps.

1. Traffic Statistics

The traffic mean and variance are required for each device. These are computed from the histograms of the messages making up the transactions for the device.

2. Concentrator Location

This step determines the number and location of any concentrators to be used in the network. These concentrators are generally connected directly to the host. The algorithm used is the add algorithm of Schwartz [SCHW 77].

Concentrators are selected from a set of concentrator candidates, specified by the user. Starting from an initial configuration, where the terminals are connected directly to the host, the algorithm attempts to add concentrators to the network, one by one. Each concentrator candidate is placed temporarily into the network. For each terminal, we compute the cost trade-off between connecting the terminal directly to the host and connecting the terminal to the concentrator. The terminals for which there is the largest decrease in cost are connected to the concentrator, up to the maximum number of terminals per concentrator. Each concentrator candidate is considered in this fashion. The one that results in the lowest network cost is put into the network. This process is repeated, adding concentrators one at a time, until adding a concentrator into the network no longer reduces the network cost.

Thus, the algorithm determines the number of concentrators, the locations of the concentrators and the homing of each terminal (which concentrator or host the terminal is connected to).

3. Homing Arrangement

Given the locations of the concentrators, this step determines which (if any) concentrator each terminal should be connected to. This info was available at the end of the previous step, however, it is recomputed because the network designer may have manually selected or changed the concentrator locations.
The algorithm used here is the same algorithm used in the concentrator location step.

4. Line Layout

The line layout step determines how many lines are required, which devices should be on which lines, the speed of each line, and the way in which the devices should be interconnected.

The algorithm used is that of [ESAU 66]. The implementation, however, is that of [KERS 74]. In their article, Kershenbaum and Chou demonstrate that several minimal spanning tree design algorithms can be generalized into a unified algorithm, where each of the specific algorithms can be selected by the appropriate choice for a set of coefficients, which are input to the algorithm. We use the coefficients corresponding to the Esau-Williams algorithm.

When used to determine an unconstrained minimal spanning tree, the algorithm produces the optimal solution. However, when constraints are introduced, such as performance criteria, the algorithm is no longer optimal. Pseudo code is shown in Figure 4.

Briefly, starting with an empty tree, the algorithm adds edges to the tree, one by one, until a spanning tree has been created. The order in which the edges are considered is important. As per [KERS 74], the edges are ordered by a cost function defined as

```
            C(v1,v2) = cost(v1,v2) - W(v1)


     where  v1,v2 are vertices,
            cost(v1,v2) is the actual cost of the transmission
                        facility connecting v1 and v2,
            W(v1) is a weight.
```

Different choices for the weights result in different known minimal spanning tree algorithms. We choose

```
            W(v) = cost (v,HOST)
```

which corresponds to the Esau-Williams algorithm.

Also, edges are added to the network only if a number of constraints are satisfied. These constraints include:

```
T = empty tree
Q = queue of edges, sorted by cost
Test = Queue_top(Q)

WHILE ^ Connected(T) DO;

   IF (T U {Test} satisfies the constraints) THEN
      T = T U {Test}
   Test = Queue_top(Q)

END
```

Figure 4. Pseudocode for Line Layout Algorithm: T will contain the resulting network.

- adding the edge does not result in a loop,
- maximum number of terminals per line has not been exceeded,
- line utilization has not been exceeded,
- response time has not been exceeded.

An additional complexity in the line layout process is introduced with the concept of **Steiner points**. A Steiner point is a location on the network which does not send or receive traffic, and so, does not have to be connected on the network. There are situations where it is desireable to have such locations on the network, because they reduce the network cost. For example, in Figure 5, the cheapest way to connect the three locations, without using Steiner points, is shown in part A and has a cost of 2 units. If we are allowed a Steiner point in the middle, we can connect the locations as shown in part B, for a cost of 1.7 units. Point s does not have to be in the network, it is not the source or destination of any traffic, but including it in the network reduces the network cost. s is a Steiner point.

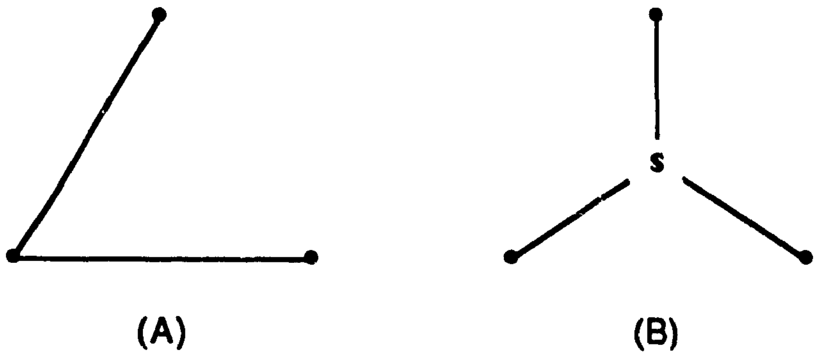

Figure 5. Steiner Point Example: The use of a Steiner point can reduce the cost of connecting network locations.

In real networks, these Steiner points are telco offices, which are not the source or destination of the customer's traffic, but whose inclusion into the network may reduce the network cost. These Steiner points are important because they are not correctly handled by the Esau-Williams algorithm. Because of the order in which the algorithm puts links into the network, the algorithm may include Steiner points which don't lower the network cost. This defficiency was recognized by Pack [PACK 75], who developed an algorithm for removing unwanted Steiner points. The Pack algorithm is run after the Esau-Williams algorithm. It works as follows.

We consider each Steiner point for removal from the network. The Steiner point is removed from the network, the network is reconnected, and the new network cost is computed. The new network cost may be higher or lower that the original. This is repeated for each Steiner point. If the removal of at least one Steiner point lowers the network cost, we remove that Steiner point which resulted in the lowest network cost. This process is repeated on the remaining Steiner points, until removing an additional Steiner point does not lower the network cost.

5. Performance Analysis

The performance analysis steps determines the expected performance of the network.

The response time computed is the delay between either the first or the last character sent and the first or the last character back. The model takes into account

- o turnaround time,
- o propagation delay,
- o CPU time to process the message,
- o transmission delay,
- o device characteristics (buffering, speed, etc.),
- o protocol characteristics (overhead, etc.)
- o and queuing delay.

The performance analysis function is used after the network has been designed, to verify that the performance requirements are being met. It is also used from within the line layout algorithm, whenever a device is added to a circuit, to ensure that performance is met.

The result at the end of this step is the number and location of all concentrators, and the topology and speed of all multipoint lines. These lines can be integrated with the voice network private lines on the integrated service.

2.3 Integrated Networks

2.3.1 Digital Technology

In the past, voice was usually transmitted as an analog signal while data might be analog or digital. A transmission facility was used for voice or data, but not both, simultaneously. Following certain advances in digital technology, digital transmission facilities can now be used to transmit both voice and data simultaneously.

2.3.1.1 PCM Voice

To obtain a digital voice signal, the analog voice signal is converted to a digital signal through **Pulse Code Modulation (PCM)**. As the frequency range required for intelligible reproduction of voice is approximately 4000 Hz, the voice signal is sampled at a rate of 8000 times a second. Each sample is coded as an 8 bit sequence and the resulting bit stream is a digital representation at 64 Kbps. At the other end of the transmission facility, each 8 bit sequence is converted back to its analog equivalent, and the analog signal is recreated.

2.3.1.2 ADPCM

Because transmission facilities are expensive, much research has been conducted to try to reduce the bit rate required to transmit good quality voice. The most commonly used technique is **Adaptive Differential PCM (ADPCM)** which provides toll quality voice with 32 Kbps. This allows two voice channels on one DS-0 (64 Kbps) circuit.

Devices are also available which encode a voice signal in 24 or 16 Kbps, but these are not as common.

2.3.1.3 TDM

Time Division Multiplexing (TDM) is a method of combining two or more information streams into a single stream (see Figure 6).

Part of the history of telecommunications has been based on the need to increase the amount of information that can be carried on a transmission facility; in the past, copper wires, now and in the future, microwaves and fiber optic cables. TDM provides a means to do just that.

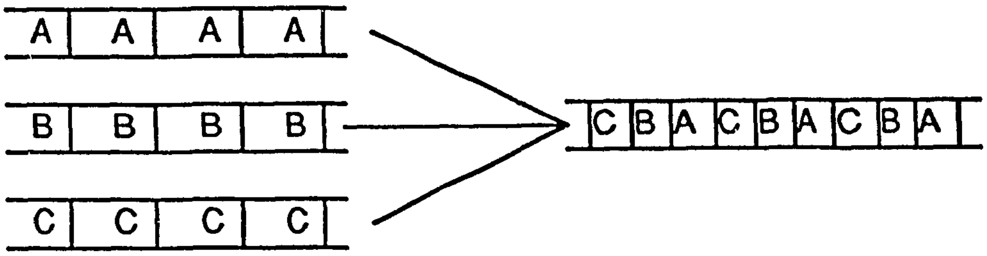

Figure 6. **Time Division Multiplexing:** Two or more information streams are combined into one.

Following a convention now referred to as T1 multiplexing, 24 digital voice channels (64 Kbps each) are multiplexed onto a single "T1" channel. The multiplexing is on a byte basis; that is, the first 8 bits are for channel 1, the next 8 bits for channel 2 and so on. For synchronization purposes, an extra bit is added after the 24th channel (see Figure 7). The resulting group of bits is called a frame, and is 193 bits long (24 x 8 + 1 = 193). The frame recurs 8000 times per second, yielding the T1 rate of 1.544 Mbps (8000 x 193 = 1544000). Each of the 24 channels gets its 8000 x 8 = 64000 bps and the additional 8000 bps is used for synchronization. The device that performs this multiplexing function is called a channel bank. The channel bank accep's 24 analog voice inputs and produces a T1 output. This T1 signal can be transmitted over two pairs of copper wire (one pair for each direction of transmission) with suitable digital repeaters (regenerators) along the way.

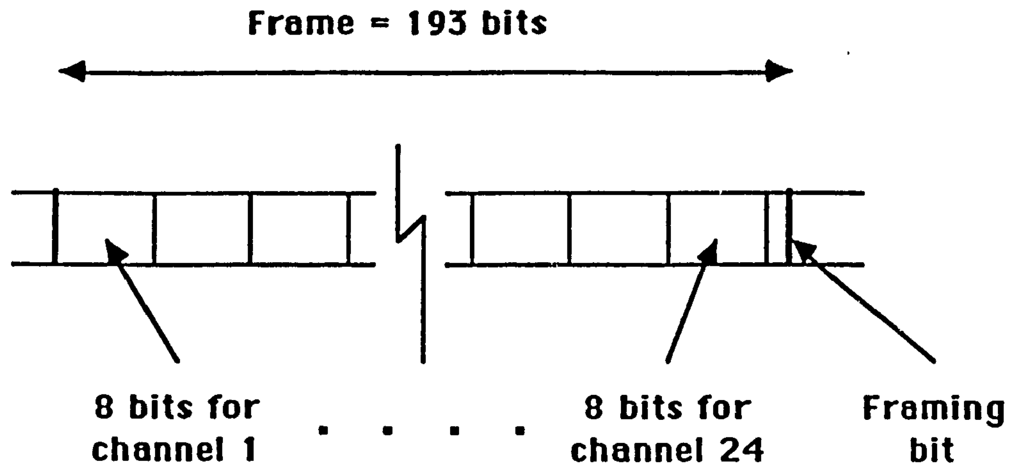

Figure 7. T1 Frame: The T1 frame includes an 8 bit "slot" for each of the 24 channels plus a synchronization bit.

2.3.1.4 IVD Multiplexers

While T1 systems were originally used for voice, it was soon observed that the bit streams carried on the T1 system could be used for voice or data transmission. Multiplexers have been designed which accept as input voice channels and different speeds of data channels, and multiplex both the voice and the data onto a T1 system.

2.3.2 Types of Integration

Figure 8 shows a method of deriving subchannels on an integrated transmission facility. The channel is segmented into frames, each frame consisting of a number of fixed duration time slots. Typically, a single slot (8 bits in T1) provides a subchannel. Wideband traffic may require multiple slots. The slots can be assigned to channels under computer control [GITM 77].

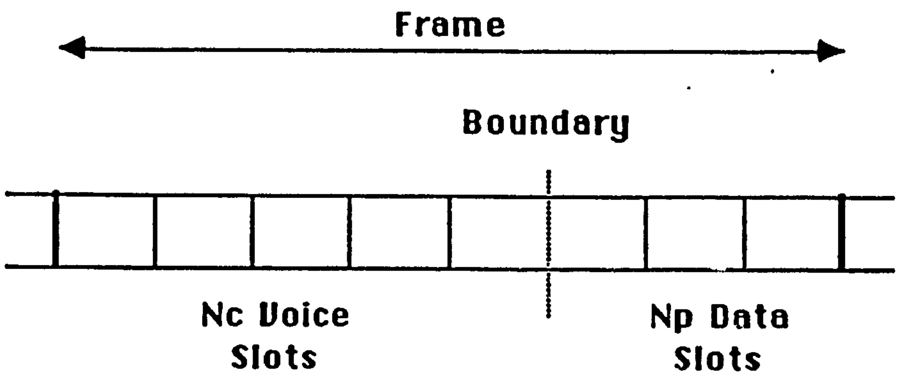

Figure 8. Voice/Data Multiplexing: The frame is divided into two groups of slots.

A boundary is maintained between slots used for voice traffic and slots used for data traffic. The boundary may or may not be fixed. If it is fixed, then the transmission capacity is divided into two groups of slots, Nc slots of the frame are reserved for voice traffic, and Np = N - Nc slots are reserved for data traffic. The multiplexing scheme is said to be of the **"fixed boundary"** type or "fixed partitioning" [GERL. 84].

To improve the utilization of transmission capacity, the boundary may be' allowed to move, depending on the instantanious traffic requirements. One group, of Np slots, is reserved for packet switched data traffic. The other group of Nc slots is available for both circuit switched voice and packet switched data traffic, with circuit switched traffic having priority. That is, packet switched traffic can use any of the circuit switched slots which are unassigned, however, a request for a circuit switched channel can preempt the packets using those circuit switched slots. This multiplexing scheme is of the "moveable boundary" type [GERL 84].

2.3.3 Integrated Network Design

The network design techniques described earlier, which are representative of techniques in general use, consider voice and data networks separately. It was possible to partition the network design problem into voice and data as the networks themselves were partitioned. The use of IVD (Integrated Voice Data) facilities invalidates to some extent the partitioning of the design problem. New design methods are required, depending on what type of networks are being integrated and what type of integration scheme is being used. Some research work has been done in this area, and results reported in the literature. We discuss three particular approaches next.

2.3.3.1 Design of Hybrid Networks

Krone [KRON 81] described an algorithm for the design of hybrid networks. Krone's hybrid network is an integrated voice·data network with circuit switched voice and packet switched data, under fixed partitioning. The input to his procedure includes

1. customer (switch) locations,
2. point to point traffic (Erlangs for voice, Kilobits for data),
3. permissible voice traffic blocking (for each point to point pair),
4. permissible packet delay (network wide average), and,
5. cost model parameters.

The procedure determines the network topology and the routing for voice and data. It is an adaptation of the "flow deviation" method used for the design of packet switched networks [GERL 73]. Starting with a trial network, the algorithm searches for a lower cost network by shifting traffic flows away from expensive routes onto lower cost routes, until no further improvement can be found. The cost of a route is the incremental cost of routing the next unit of traffic on the route, and hence, depends on the loadings on the network. These costs are re-evaluated each time the routings are changed.

The algorithm employs the concept of pseudolength. The pseudo length of a link is the incremental cost of carrying additional traffic through the link.

The algorithm can be broken down into three main steps:

1. Initial network topology

   An initial topology is generated by ordering the potential links in the network according to certain criteria, and then adding links until the desired connectivity has been achieved. The ordering is based on the traffic between the end points and the length of the link. Short links with lots of traffic are favoured. To specify the connectivity, a minimum node degree is entered by the user. Links are added to the network until all nodes have degree greater than or equal to the minimum node degree.

2. Link blocking allocation (for the voice portion)

   The allowed blocking specified is the end-to-end blocking for a point-to-point pair. As there will generally be more than one path between two points, the end-to-end blocking will be different from the link blocking. Hence a value B for link blocking is approximated which provides the required end-to-end blocking. Computation of B is determined according t the network connectivity.

3. Network improvement

   Network improvement consists of the following steps.

   a. compute initial link costs (pseudolengths),
   b. compute shortest path routings,
   c. route packet traffic along shortest path routes,
   d. route circuit traffic (traffic distribution),
   e. size each link for
      - voice, using equivalent random theory [WILK 56],

- o data, using capacity assignment [GERL 73].

f. update pseudolengths

Steps b to f are repeated until the network cost can no longer be reduced.

The process determines the bandwidth required between each pair of locations. A realization of the network must then be determined in which actual services are selected to provide the required bandwidth.

2.3.3.2 Express Pipe Networks

Pazos-Rangel [PAZO 83] investigates a number of problems in integrated network design, in particular, bandwidth allocation and routing for integrated networks (fixed boundary). Again, the integration here is between circuit switched voice and packet switched data.

The problem can be stated as: Given the voice and data traffic, the network topology, and link capacities (and hence, cost), determine the capacity assigned to circuit switched traffic, and the routing for circuit switched and packet switched traffic such that a weighted sum of the average blocking and average delay is minimized.

The problems are formulated as non-linear programming problems. Two different situations are considered:

1. For a non-linear objective function with linear constraints, the method of steepest descent is used.

2. For a non-linear objective with non-linear constraints, penalty functions are introduced, and the method of steepest descent is applied. Iteration is performed over the penalty parameter.

As the objective function is convex and the feasible set is convex, there is exactly one local minimum which is the global minimum.

2.3.3.3 Capacity Allocation

[AVEL 82] addresses the problem of capacity allocation in voice-data networks. Packet switched data and circuit switched voice are integrated under fixed partitioning. The link capacities are known, and the allocation of capacity to voice and data traffic for each link are determined such that the blocking probability for the voice traffic is minimized while the average delay experienced by data messages is constrained below a specified value.

The computation of average delay for data packets is based on the approach of Kleinrock [KLEI 76]. A correction factor is introduced to correct for the fact that only a portion of the TDM frame is available to the data packets (the rest being used for voice). The computation of blocking probability for voice is based on the Erlang B formula [MINA 74]. A key assumption made at this point is that only first offered traffic is considered.

All voice overflow traffic is carried on a separate overflow network, which is not part of the design problem. This greatly simplifies the calculation of the voice blocking probability.

The problem was formulated as a non-linear integer programming problem, and solved through the application of linear programming techniques. The capacity allocation technique was used to study several sample network problems. Capacity allocation was determined on a link basis, and compared to results obtained where the capacity allocation was constrained to be the same on all links. It was shown that capacity allocation on a link basis led to considerable improvement in network performance.

3.0 THE PROBLEM

This section describes the particular integrated voice/data network design problem addressed in this report.

3.1 The Approach

The type of integration considered here is integration of a voice network with a multipoint (centralized) data network, using the "fixed boundary" integration scheme. The approach can also be applied to the integration of a voice network with a packet switched data network, under fixed partitioning.

The assumption of fixed partitioning implies that the voice and data traffic do not interact. This means that the voice and data traffic can be analyzed separately. This is a great advantage, as the two traffic types require completely different types of considerations. Voice traffic is Poissonian with exponentially distributed holding times; calls can be blocked and the performance constraint is the percentage of calls blocked. While data traffic can be considered to be Poissonian, the message lengths are not generally exponentially distributed. Messages are never blocked (only delayed), and the performance constraint is the delay experienced by the messages.

Under fixed partitioning, voice and data are combined at the channel level. Certain bandwidth is reserved for voice channels, sufficient bandwidth to ensure the required performance. Certain bandwidth is reserved for data channels, enough to meet the delay requirements. The amount of bandwidth reserved for voice does not affect the data performance or vice versa. They will, however, both affect the total network cost and the topology of each can effect the topology of the other.

Hence, we choose to address the integration of voice and data at the channel level. We use existing design methodologies to determine the channel requirements for both the voice and data networks, and then consider these two sets of requirements together to determine the best integrated network.

The general approach is shown in Figure 9. The voice and data network design tools are used as described in the chapter "Background" on page 6. Each process results in a set of channel requirements between pairs of locations. The set of locations and channel requirements constitute the input for the channel grouping and routing step.

Channel grouping refers to the fact that voice and data channels are grouped onto integrated services, when such an action is economical. Channel routing refers to the fact that channels may have to be rerouted along different paths to achieve the appropriate channel groupings. The problem addressed in this step is defined next.

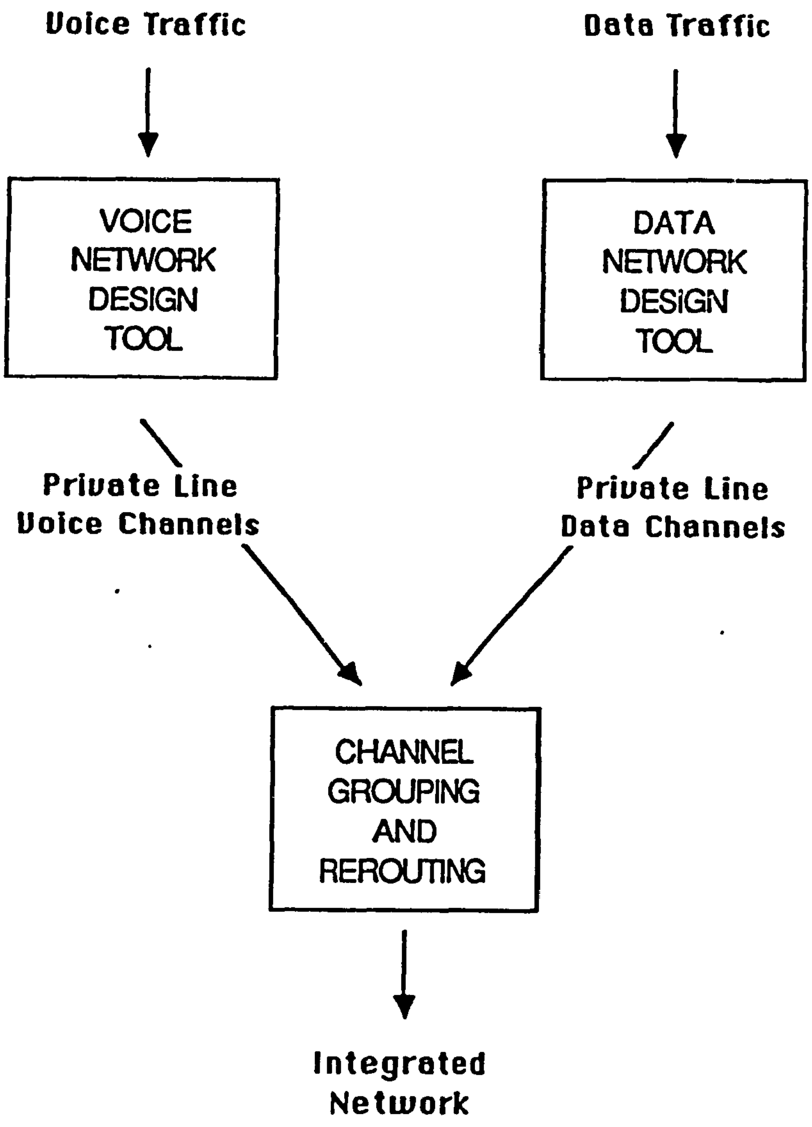
Voice Traffic
Data Traffic
VOICE NETWORK DESIGN TOOL
DATA NETWORK DESIGN TOOL
Private Line Voice Channels
Private Line Data Channels
CHANNEL GROUPING AND REROUTING
Integrated Network

Figure 9. General Approach to IVD Network Design: Voice and data networks are designed separately to give the channel requirements. These requirements are combined and the best integrated network is determined.

3.2 Problem Formulation

The problem is stated as follows:

Given:

- customer locations,
- requirements between each pair of locations for voice and all speeds of data,
- service information, including costing, rate centers, etc.,

Determine:

- a feasible network solution,
- which minimizes the total monthly cost.

A feasible network solution is one in which there is sufficient bandwidth between each pair of locations to satisfy the requirement. This implies that a routing must be found for each requirement for the total number of voice and data channels.

3.3 The Integrated Service

The advances in digital communications described earlier have allowed the Telcos to tariff new services, with new integrated voice data features for business customers.

Some of the features of one particular service, Megastream, offered by Bell Canada, are described below.

The general structure is shown in Figure 10. The service consists of an IVD multiplexer at each customer premise, an access portion to connect the multiplexer to the telco office equipment (rate center), and an interexchange portion connecting rate centers. Alternatively, the multiplexer can be placed at the rate center.

If the multiplexer is at the customer premise, the access from the customer premise to the rate center is via a DS-1 channel. If the multiplexer is at the rate center, then the customer equipment is connected to the multiplexer via regular voice and data services.

The interexchange portion connecting rate centers is based on DS-0 channels. That is, the customer leases bandwidth in multiples of 64 Kbps. There are two peculiarities which affect the design process. The first is that there is a minimum bandwidth on each cross section that the customer leases of four DS-0 channels. The second, is that the leasing of these channels is subject to a volume discount (for channels within a particular cross section). A fixed rate per DS-0 channel is charged for the first 12 channels. Channels 13 to 24 are discounted by 10%, and each channel from 25 and up is discounted 15%.

An additional feature which makes the service attractive is subrate data multiplexing. Several low speed data circuits can be multiplexed onto a single DS-0 circuit.

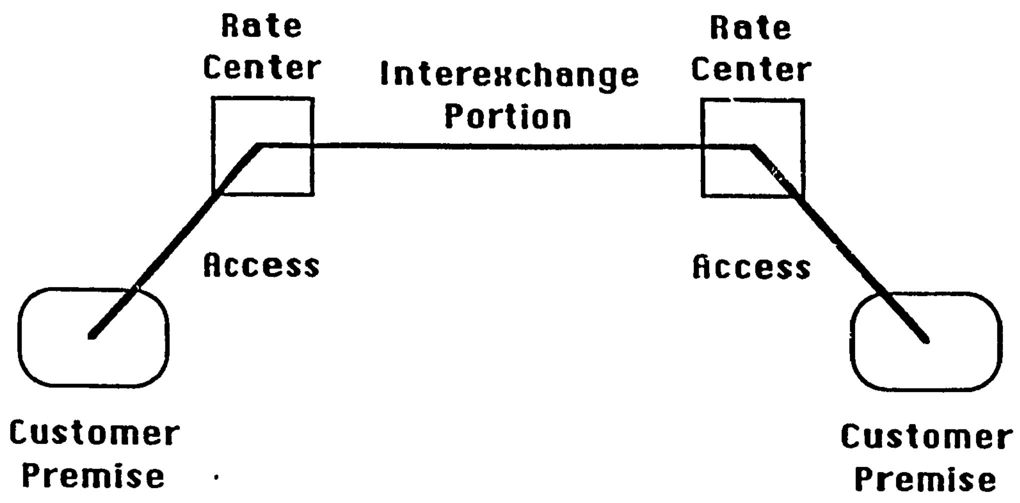

Figure 10. Integrated Service Structure: This figure shows the major components of the integrated voice/data service.

4.0 THE SOLUTION

The channel grouping and routing is based on a voice channel routing heuristic developed by Hansler.

4.1 Channel Routing

Hansler [HANS 73] addressed the problem of mapping channel requirements onto leased facilities for voice channels under the Telpak tariff[2]. He described a set of heuristic procedures which determine near minimal cost network configurations. His problem, solution and results are summarized next.

4.1.1 Problem Statement

Hansler's problem is stated as follows:

Given

- network locations,
- service requirements (number of channels) required between each pair of locations,
- and the cost function,

[2] The Telpak tariff is a private line voice service, offering volume discounts on large cross sections

find the routing which

- satisfies the requirement,
- and minimizes the monthly network cost.

Hansler made the following assumptions:

- one class of requirements (voice),
- requirements are undirected,
- only integer numbers of channels,
- no limit on link capacity,
- no steiner points,
- no connection charges,
- cost depends on location, length and capacity,
- and economy of scale (volume discounts - Telpak tariff).

4.1.2 Procedure

Hansler's procedure consisted of generating an initial solution, in which each requirement is implemented by one or more circuits in the network. A circuit is a sequence of links, and the capacity of a link is the sum of the capacities of the circuits using the link. Then, the following network improvement steps are run:

- rerouting of single circuits,
- rerouting of pairs of circuits,

- splitting of circuits,
- rerouting of links,
- optimization of triangles,
- other miscellaneous steps.

We describe only those steps which were adopted for the channel grouping and routing design procedure. Descriptions of the other steps can be found in Hansler's paper.

4.1.2.1 Starting solution

Hansler considered three different methods of generating a starting solution. The simplest method was to implement each requirement by a circuit following the direct path. This ignores any characteristics of the tariff which might allow a cost reduction. Another approach is to compute the minimal spanning tree. This generally results in a more expensive solution.

The best approach, Hansler called the method of "cheapest increment". It involves sorting the list of requirements according to the distance between the origin and destination in ascending order. Then, each requirement in turn is put onto the network on its shortest path based on the network configuration achieved so far.

4.1.2.2 Network improvement

Hansler considered a number of network improvement steps, of which we describe only two here.

Rerouting of single circuits

After a starting solution has been determined, this step considers each circuit one by one for rerouting.
A circuit is removed from the network, and the shortest path for the corresponding requirement is computed. The circuit is then put back in the network on its shortest cost path. The circuits are considered in order of increasing length.

Rerouting of links

A link can be used by several circuits, and so modifying a link requires modifying all of the circuits which use the link. To reroute a link, the link is removed from the network and the cheapest path is computed for a circuit with the capacity of this link. If a new, cheaper path is discovered, it is chosen over the direct link. Each link is considered for rerouting in this way.

4.1.3 Run Time Analysis

Hansler recorded the computing times on an IBM 370 Model 155 and observed that his run time was of order

```
r * n ** (3/2),
```

where r is the number of requirements and n is the number of network locations. As the requirements are generally of order n^2, the algorithm is approximately of order n^(7/2).

4.1.4 Cost Improvement Analysis

Hansler recorded the reduction in network cost achieved by each step of the procedure. He determined that most of the cost benefit was achieved in the starting network, using the cheapest increment approach. The other steps which resulted in significant savings were

1. rerouting of single circuits
2. rerouting of links
3. rerouting of pairs of circuits

As most of the computing time was spent in rerouting of pairs of circuits and little network cost reduction was achieved, we did not consider this step.

4.2 Modifications

We adapt Hansler's approach to solve the problem described in "The Problem" on page 37, in the following ways:

- multiple classes

  While Hansler assumed one class of requirement, voice, we must consider two classe·', voice and data.

  To handle this, whenever we determine the routing of a requirement, we consider two different cases. First, the minimum cost path is determined for the voice and data components of the requirement separately. Then, the least cost path for the two together is determined. The cheapest of the two alternatives is used.

- bin packing

  An additional problem, related to multiple classes, is that the voice requirements can be satisfied as 32 or 64 Kbps voice circuits and the data requirements can be for any of a number of different speeds (we consider 1.2, 2.4, 4.8, 9.6, 19.2, and 56K bps). When these requirements are mapped onto the integrated service, they must be packed into DS-0 channels. A bin packing algorithm is used to determine the number of DS-0 channels required to satisfy the requirement. The method is described below.

- multiple services

  Hansler considered only one service, Telpak. We consider several services, private line (Telpak), Dataroute, Teleroute 200, and Megastream. This complicates the computation of the minimum cost path as described below.

- connection charges

In computing the cost of routing a circuit from A to B, we include all connection charges. These are charges covering the connection of customer equipment to the Telco transmission facilities, and costs associated with the connecting of one type of facility to another.

The resulting method is now described, and an example given to illustrate the various steps.

4.3 The Method

The steps of Hansler's method that were chosen are (see Figure 11):

1. initial solution by the method of cheapest increment,
2. rerouting of single circuits, and,
3. rerouting of single links.

The two network improvement steps are repeated until the network cost no longer decreases.

The individual steps are now described in more detail. Pseudo-code is shown in Figure 12.

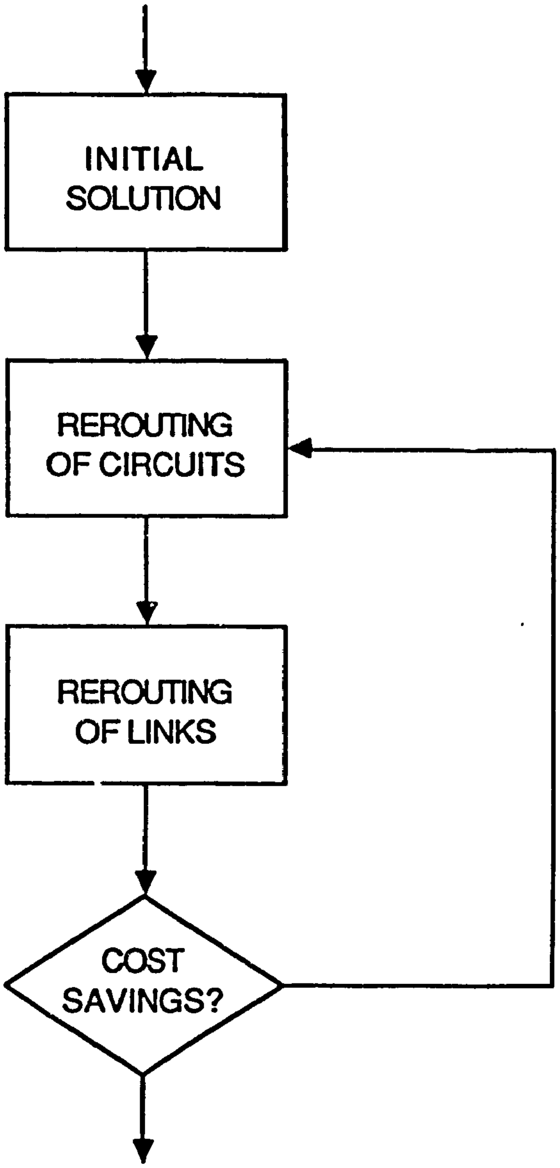

Figure 11. Algorithm Steps: The algorithm consists of three steps taken from Hansler.

4.3.1 Starting Solution

The starting solution is generated using the cheapest increment approach. Starting with an empty network, circuits are added to the network one by one. The requirements are sorted according to their ratio of length/number of channels, and are mapped onto the network in the order of increasing ratio. Each requirement is mapped onto its minimum cost path, as computed with the shortest path procedure described below. The minimum cost routings for voice and data separately, and voice and data combined are computed, and the routing that gives the cheapest cost is selected. These circuits are added to the initial network solution.

As each subsequent requirement is considered for addition into the network, those circuits already in the network can affect the shortest path route of the new requirement being considered. The routings may be influenced by spare capacity or volume discounts on some of the previously placed links. Hence, the order in which the requirements are considered is important. Requirements which are short, or of large number of channels are routed first. This is because channels which are short are least likely to be able to benefit from the rerouting available later on, and so, are installed in the network first. The large requirements are the ones which will provide the volume discounts which other requirements will be able to capitalize on, and hence, are put into the network as early as possible.

```
Sort requirements by order of
   increasing ratio of length/# of channels

FOR each requirement DO
   Determine the minimum cost circuit
   Add the circuit to the initial solution
END

DO UNTIL no more cost savings
   Sort circuits by decreasing number of channels
   FOR each circuit DO
      Remove the circuit from the network
      Determine the shortest path circuit based
         on the remaining network
      IF a new cheaper circuit is found
        Add it to the network
      ELSE
        Put the old circuit back into the network
   END

   FOR each link DO
      Compute shortest path for link
      Cost savings = cost of link - cost of shortest path
   END


   Create list of links sorted by decreasing cost saving

   DO UNTIL cost savings of first link in list <= 0
      Select first link in list
      Remove from list
      Recompute cost savings for the link removed
      IF New cost savings >= cost savings of next link
         Reroute link
      ELSE
         Insert into list at proper place
   END

END
```

Figure 12. Algorithm Pseudo-code: Pseudo-code is given for the main steps of the algorithm. See the text for a more detailed explanation.

4.3.2 Rerouting of Circuits

In this step, circuits are considered one by one for rerouting along cheaper paths. The circuits in the current solution are sorted by decreasing number of channels. Then each circuit in turn is considered for rerouting. The circuit is removed from the network. Based on the remaining network, the shortest path is computed for this circuit. Again, the shortest path is computed for voice and data separately, and voice and data together. If this results in a new, cheaper path, the new circuit is added to the network solution, otherwise, the old circuit is retained.

4.3.3 Rerouting of Links

In this step, each link is considered for rerouting. As a link may be used by several circuits, rerouting a link means replacing the link by some other path, in each of the circuits that use the link. For each link, we compute the cost saving due to rerouting the circuits on the link. This is computed by first, removing the link from the network, and then, determining the shortest path routing for the link. The cost saving is the cost of the link minus the cost of the shortest path for the link. Note that this cost saving is often zero, as the least cost path is often the direct link itself. This cost saving is computed for all links (order is not important). We then select the link which resulted in the largest cost saving (if any), and reroute the circuits on this link. If the rerouting results in any loops, these must be removed. This is repeated until no more cost savings can be achieved.

After each link is rerouted, the cost saving should be recomputed for each link, as the change in the network may affect the cost saving. However, because of the large amount of computation involved, a heuristic approach is adopted. After the cost saving has been computed for each link, the links are put into a list, sorted in order of decreasing cost saving.

In the general step, the link at the head of the list (greatest cost saving) is considered for rerouting. The link is removed from the list and its cost saving is recomputed. If this link's cost saving is still the greatest, the link is rerouted. Otherwise, it is inserted into the list at the correct place according to its newly computed cost saving. This is repeated until no more links show cost savings.

4.3.4 Shortest Path

The fundamental operation of the algorithm is the computation of shortest path. In each of the algorithm steps, the minimum cost (shortest path) for each requirement, circuit, or link is determined. This is the shortest path for an incremental circuit on the network. In other words, the coster which generates the costs used in the shortest path returns the cost of adding the circuit onto the existing network design. This takes into account any spare capacity already installed in the network.

The shortest path procedure is based on Dijkstra's shortest path algorithm [DIJK 59]. Some modification of the procedure was required as we determine shortest path on a multi-graph and take into account interface and termination costs.

Because there are up to NbServe (number of services) arcs between each pair of nodes, we create NbServe copies of each node. The original nodes are called base nodes, and associated with each base node are NbServe copies, one for each service. An array, COST(i) is used to store the cost of reaching node i from the origin. The COST(i) are initialized to infinity, except for COST(Origin) which is assigned zero. The shortest path will be stored in the arrays PREV_NODE(i) and PREV_EDGE_TYPE(i). If node i is on the shortest path, then PREV_NODE(i) gives the node preceding i in the shortest path (PREV_NODE(Origin) = 0) and PREV_EDGE_TYPE(i) gives the service that was used to reach node i on the shortest path.

We visit nodes in an order determined by the COST(i). We visit that node which has the lowest COST of all nodes not yet visited. In visiting a node, called the current node, we compute the cost of reaching every other node (called adjacent nodes) from the current node. This value is called NEW_COST(j). NEW_COST(j) includes the connection charge incurred at the current node to get to the adjacent node. If NEW_COST(j) is less than COST(j), then we have found a cheaper way to get from the origin to node j. We replace the value in COST(j) with that in NEW_COST(j) and update PREV_NODE and PREV_EDGE_TYPE.

We now select the next current node (lowest COST(i)) and visit it. This continues until we reach the destination node. The shortest path can then be reconstructed from PREV_NODE and PREV_EDGE_TYPE.

4.3.5 Bin Packing

As mentioned above, the integrated service is available in multiples of DS-0 (64 Kbps) bandwidth. If the requirement is for voice, the situation is simple. Either 1 or 2 voice requirements can be put on each DS-0 (64 or 32 Kbps encoding). Because of the wider range of data speeds available, the situation for data requirements or mixed voice and data requirements is slightly more complex.

This problem is an example of a bin packing problem. While methods providing optimal solutions to the bin packing problem exist (dynamic programming, integer programming), they are too slow for our purposes here. Instead, we choose a heuristic approach. Based on the work of [GOLD 76], our choice is first fit, considering the requirements in decreasing order of required bandwidth. We procede by packing requirements onto DS-0 channels, one by one, starting with the highest speed requirement. The DS-0 channels are numbered 1 to N, where N is to be determined. The requirements are put on DS-0 #1 until it is full, then DS-0 #2 and so on, until the requirement for this speed has been satisfied. Then, the next lower speed is considered in the same way, starting again with DS-0 #1. The process is repeated until all speeds have been processed.

For example, consider the following requirements to be packed onto DS-0 channels:

Speed	Number of channels
56 Kbps data	1
32 Kbps voice	3
19.2 Kbps data	2
9.6 Kbps data	4

The order in which the requirements are listed is the order that they would be considered for bin packing. They are mapped onto DS-0 channels as follows, with the final result shown in Figure 13. The 56 Kbps data requirement would occupy all of the first DS-0. The three voice channels are considered next. Two can fit on DS-0 number 2, the third will go on DS-0 number 3. The first 19.2 Kbps requirement will fit on DS-0 number 3, but the second must go on DS-0 number 4. Last, the 9.6 Kbps requirements are considered. There is room for the first on DS-0 number 3. The other 3 will fit on DS-0 number 4. Hence, 4 DS-0 channels are required. In computing the allocation of channels to DS-0s, any required overhead bandwidth is for the channel is also included.

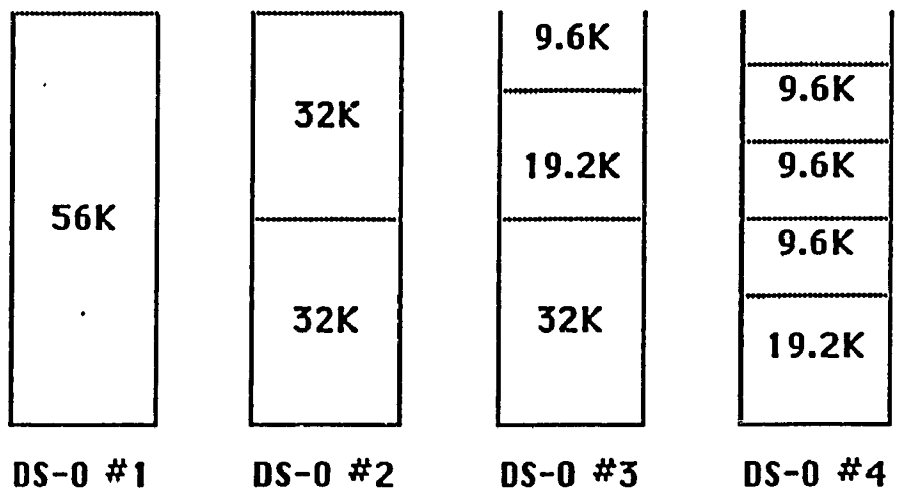

Figure 13. Bin Packing Example: This figure shows how the requirements would be mapped onto DS-0 channels.

4.4 Network Design Example

The following network design example illustrates the steps of the algorithm. The network consists of four locations, A, B, C, and D. The requirements are shown in Figure 14.

Identifier		Distance	Requirement	dist/# chan	# chan
R1	A-C	150	20 voice	7.5	20
R2	B-C	120	10 voice 4 9.6K data	8.57	14
R3	B-D	130	16 voice	8.125	16

Figure 14. Requirements: The network requirements along with the data required for sorting the requirements.

For the starting solution, the requirements are ordered by increasing ratio of length/number of channels. Hence, they are considered in the order R1, R3, R2. The shortest path determined for R1 is 20 channels on Teleroute, for R3, 16 channels on Teleroute, and for R2, the 10 voice channels on Teleroute and the 4 data channels on analog private line (PL). The resulting initial solution is shown in Figure 15.

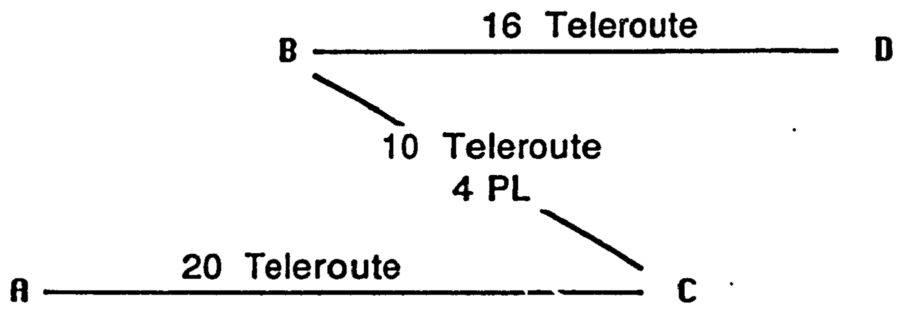

Figure 15. **Initial Solution for Example:** Network determined by the first step of the algorithm.

For the second step, rerouting of circuits, the circuits are ordered by decreasing number of channels and so considered in the order R1, R3, R2. R1 is removed from the network and the shortest path for the requirement is determined. Because of the B-C circuit, the shortest path is A-B-C using Telerout.ȝ and the volume discount available on link B-C. Similarly, R3 is routed B-C-D. The shortest path for R2 is still B-C. The new network is shown in Figure 16.

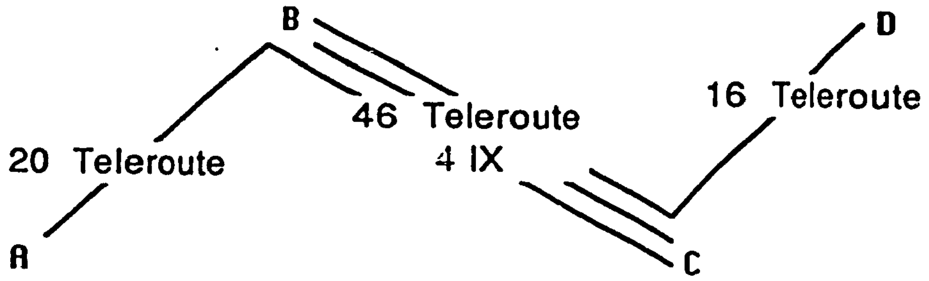

Figure 16. **Example After Rerouting of Routes:** This figure shows the example network after the second step of the algorithm. Both A-C and B-D have been rerouted through link B-C.

For the third step, rerouting of links, the possible cost saving is computed for each link. Note that the only links with non-zero capacity arɔ A-B, B-C, and C-D. For A-B and C-D, no cost savings are possible. For link B-C, the cost can be reduced by using the integrated service. The 46 voice channels on TR and the 4 data channels on private line are replaced by 24 DS-0 channels on Megastream. 23 of the DS-0s carry the 46 voice channels and the 24th carries the 4 9.6 Kbps data channels. The resulting network is shown in Figure 17.

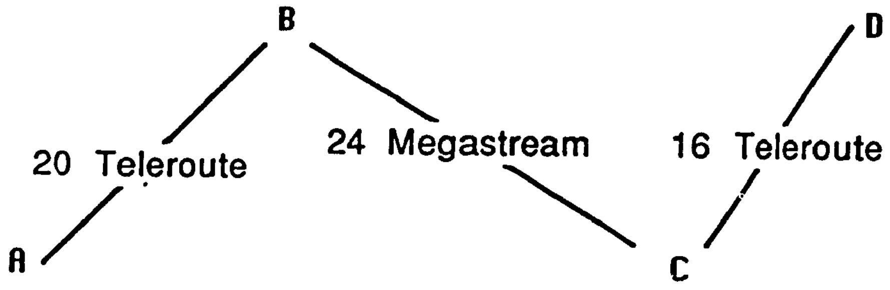

Figure 17. Final Solution for Example Network: Final solution, after rerouting of link B-C.

5.0 ANALYSIS

This sectic ` analyzes the method in terms of the network designs produced and the efficiency of the design procedure.

5.1 Network Designs

The network design heuristic was used to design several sample networks. Some of the smaller networks are shown, to give a flavour of the types of designs produced. The larger networks were too difficult to display in this fashion. The results from these network designs are summarized.

5.1.1.1 Example 1

Example 1 was a very small network, with four locations in Toronto, Lachine, Montreal and Sherbrooke. The requirements are shown in Figure 18.

The resulting network design is shown in Figure 19. The following costs and CPU times were recorded for the steps of the heuristic:

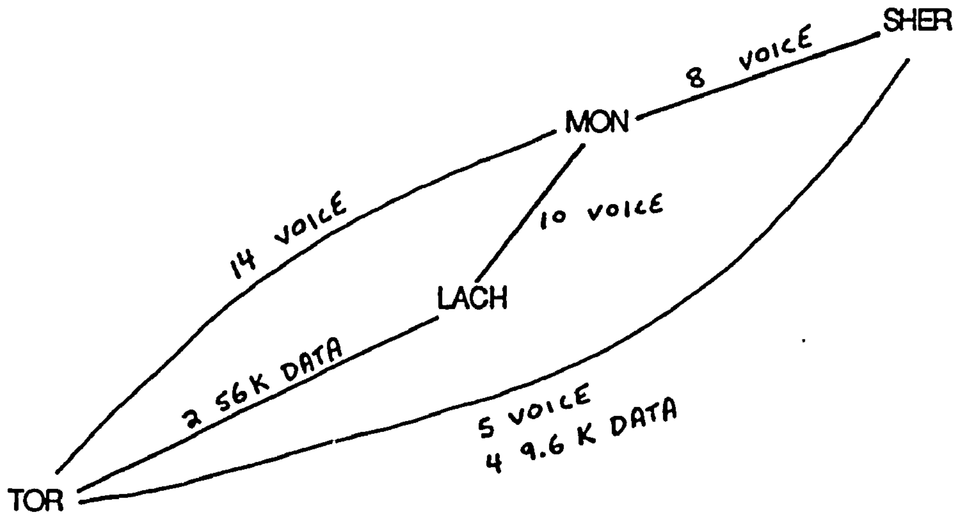

Figure 18. Requirements for Example 1

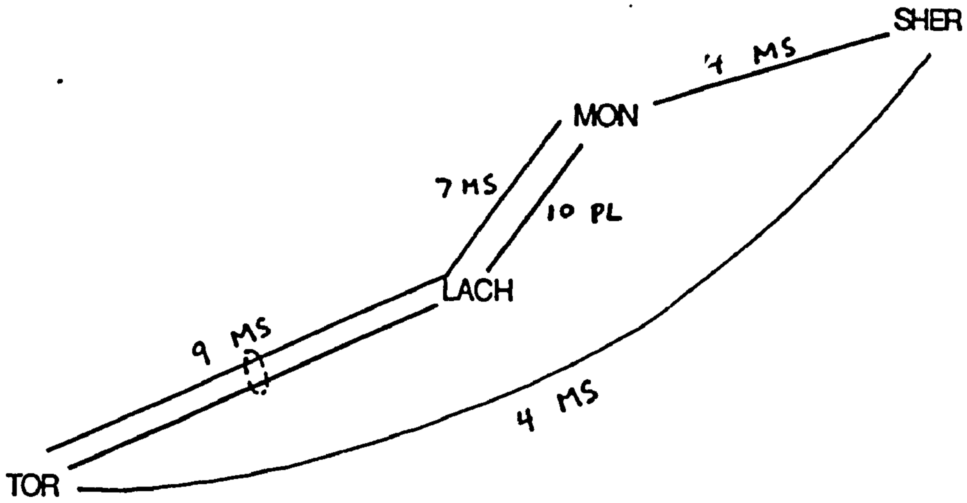

Figure 19. Network Design for Example 1

Step	Network Cost ($)	CPU Time (sec)
Reference Cost	45679.30	
Initial Solution	43020.30	0.05
Rerouting of routes	38510.35	0.29
Rerouting of links	38510.35	0.57

		0.91

Observations:

While no cost improvement was found by the rerouting of links step, the starting solution and rerouting of circuits step produced the following groupings and reroutings.

- The Montreal-Toronto channels were rerouted through Lachine because of spare capacity on the Lachine-Toronto link.
- The 5 voice and 4 data channels between Toronto and Sherbrooke were grouped, as 4 DS-0s on Megastream are cheaper than 5 on Teleroute and 4 on private line.
- It is interesting to note that the Lachine - Montreal channels remained on IX, even though the Toronto - Lachine - Montreal channels are on Megastream. This is because the distance from Lachine to Montreal is very short, and so, the cost of the end equipment for either Megastream or Teleroute is not justified by the savings in transmission facilities.

5.1.1.2 Example 2

The network for example 2 had seven locations. The requirements are shown in Figure 20 and the resulting network design in Figure 21.

The following costs and CPU times were observed:

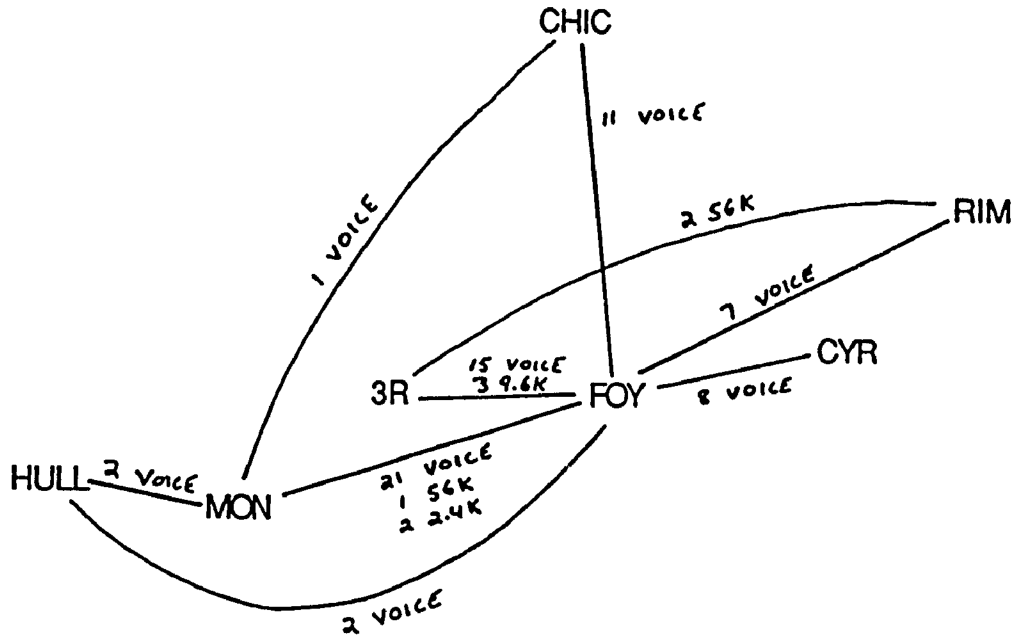

Figure 20. Requirements for Example 2

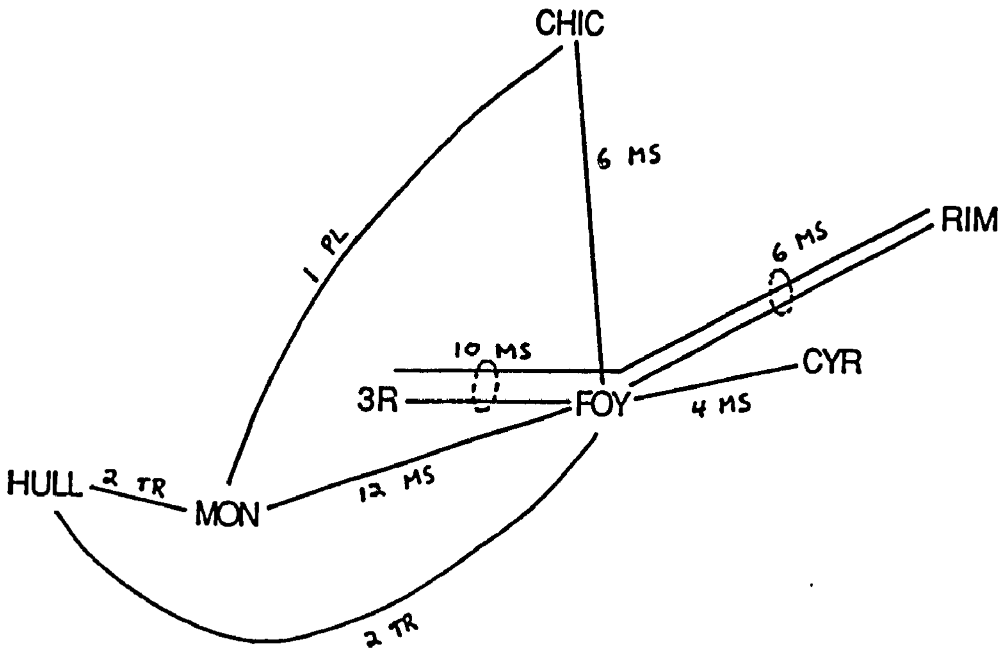

Figure 21. Network Design for Example 2

Step	Network Cost ($)	CPU Time (sec)
Reference Cost	75642.79	
Initial Solution	67165.80	0.16
Rerouting of routes	65329.80	1.48
Rerouting of links	65329.80	2.81

		4.45

Observations:

Again, the rerouting of links step failed to find any network improvement. The first two steps found the following groupings and reroutings:

- The Montreal - Ste Foy channels were grouped on Megastream.
- The Three Rivers - Rimouski channels were rerouted through Ste Foy.
- The Three Rivers - Ste Foy channels were grouped with the Three Rivers - Rimouski channels.
- The Ste Foy - Rimouski channels were grouped with the Ste Foy - Rimouski channels.
- The Ste Foy - Ste Cyr channels were put on Megastream. For this distance, 4 DS-0s on Megastream were cheaper than 8 voice channels on private line or Teleroute.
- The 11 voice channels from Chicoutimi to Ste Foy were put on 6 DS-0s on Megastream. This leaves 32 Kbps of spare capacity.

- o It was expected that the Hull - Ste Foy channels would be grouped with the Montreal - Ste Foy channels. This was not the case, as for this distance, Teleroute was cheaper.

5.1 1.3 Example 3

Example 3 had ten locations. The requirements are shown in Figure 22 and the resulting network design is shown in Figure 23.

The following costs and CPU times were recorded for the steps of the heuristic:

Step	Network Cost ($)	CPU Time (sec)
Reference Cost	110809.46	
Initial Solution	108508.96	0.20
Rerouting of routes	107406.86	4.08
Rerouting of links	107406.86	6.47

		10.75

Observations:

Again, rerouting of links failed to find any network improvement, but the initial solution and rerouting of circuits found many economic groupings and one rerouting, as shown in the network design.

Although we do not try to enumerate them, one interesting example is the circuit from Rimouski to Montreal. The circuit is routed through Quebec. From Rimouski to Quebec,

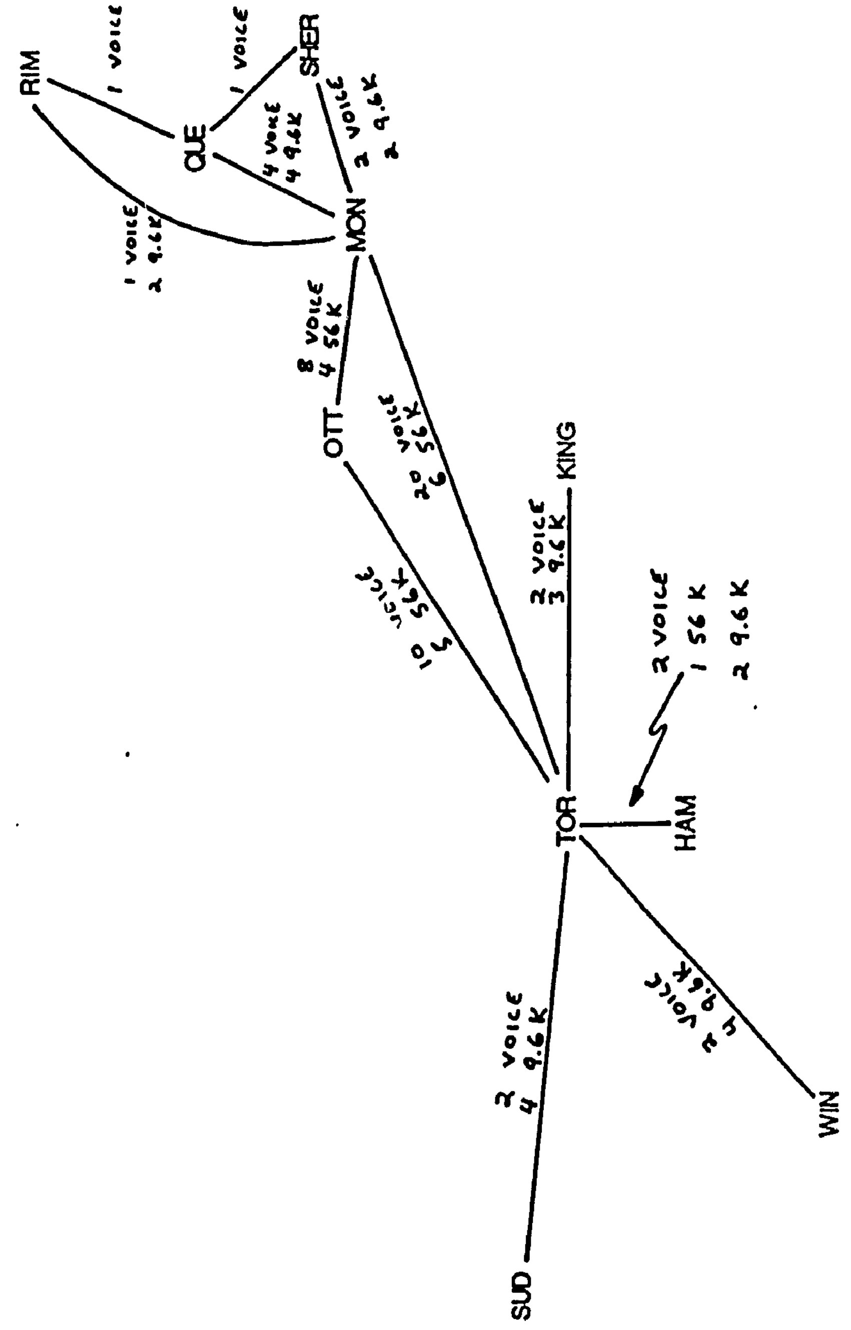

Figure 22. Requirements for Example 3

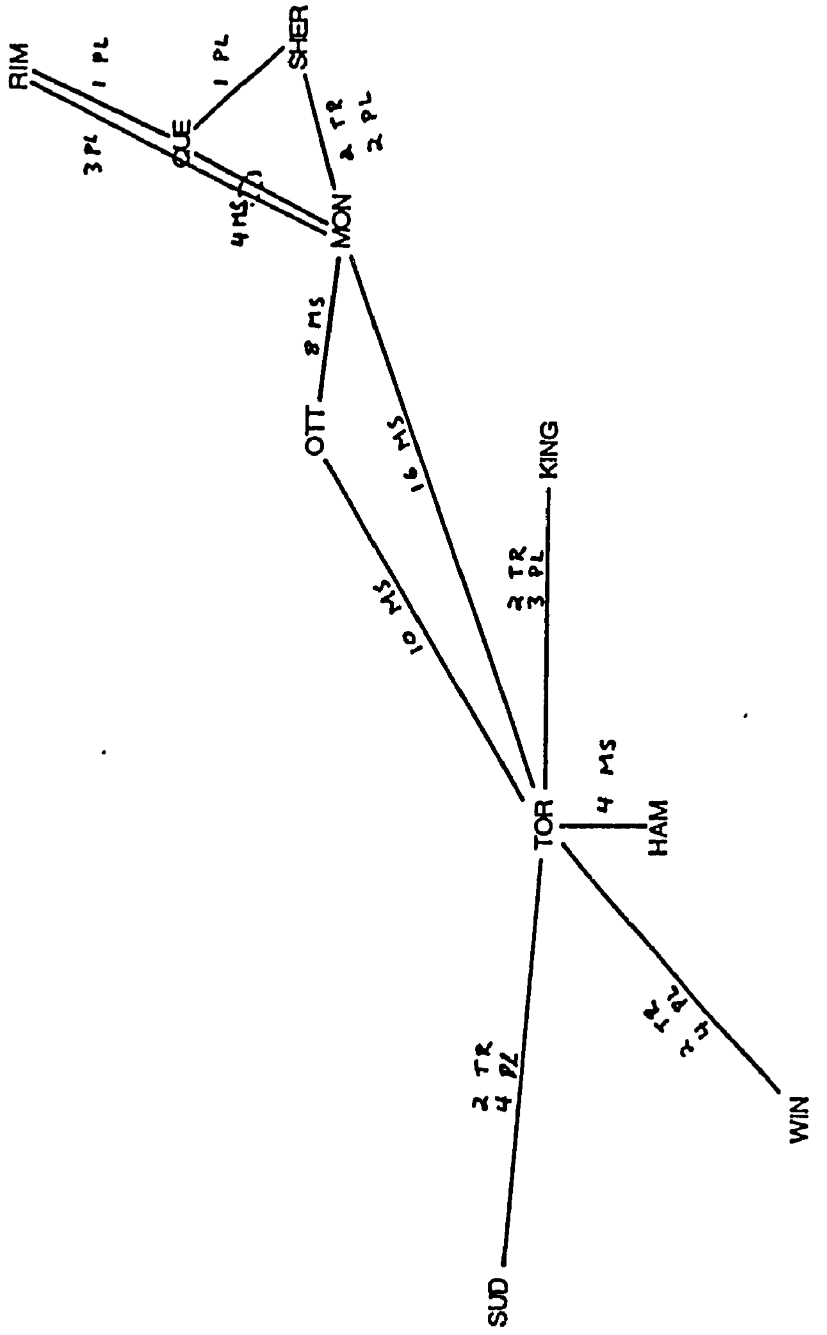

Figure 23. Network Design for Example 3

the circuit consists of 3 private lines (one for voice, two for data). From Quebec to Montreal, the Rimouski - Montreal channels are grouped with the Quebec - Montreal channels on Megastream.

5.1.1.4 Example 4

Example 4 had 15 locations. The following costs and CPU times were recorded for the steps of the heuristic.

Step	Network Cost ($)	CPU Time (sec)
Reference Cost	9936245.99	
Initial Solution	9527451.51	44
Rerouting of routes	9526927.41	47
Rerouting of links	9526927.41	62

		153

Again, rerouting of links failed to find any cost savings. Most of the cost savings was found in the initial solution.

5.1.1.5 Example 5

Example 5 consisted of 24 locations. The following costs and CPU times were recorded for the steps of the heuristic.

Step	Network Cost ($)	CPU Time (sec)
Reference Cost	20850943.09	
Initial Solution	20671153.17	263
Rerouting of routes	20670629.07	392
Rerouting of links	20670629.07	314

		969

Rerouting of links found no network improvement, and as above, most improvement was found by the initial solution.

5.2 Run-time Analysis

5.2.1 Software Environment

The described heuristic was implemented for the purposes of studying the heuristic procedure and the network designs produced. The implementation was built into an existing network design environment, which currently houses the voice and data network design tools. The environment runs on an IBM-type mainframe, under CMS.

The environment consists of a set of relational databases, a file management system which keeps track of the users various studies, and the design programs themselves. The databases are implemented under a BNR developed database management system called GERM (Generalized Entity-Relationship Model). The file management system is written in REXX (IBM System Product Interpreter) and the design programs are written in

PL/I. The channel grouping and routing program consists of about 15000 lines of PL/I code.

5.2.2 Run-time vs Network Size

The CPU time, on an Amdahl V7 using 8 megabytes of virtual storage, for each step was recorded for several different network sizes, and the results are shown in Figure 24. Applying linear regression shows that the slope of the line is 3.7, and therefore the algorithm is of order n**3.7 in complexity.

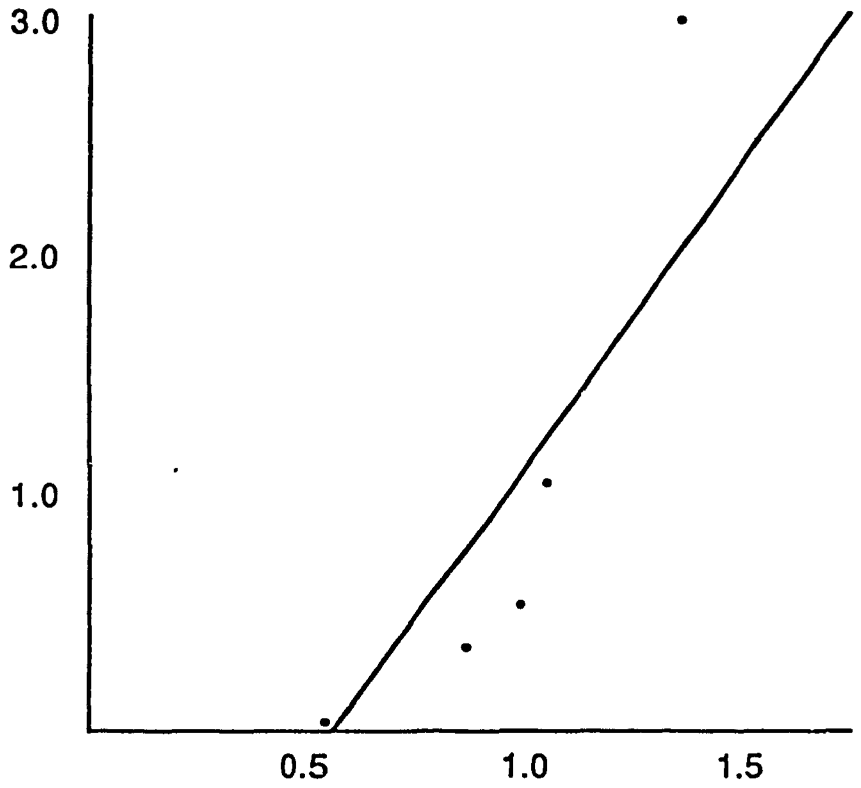

Figure 24. CPU Time: The logarithm of CPU time plotted against the logarithm of network size for the five examples.

The points do not fit a straight line very well. This is because two of the networks, examples 4 and 5, are core networks from much larger networks (700 - 1200 nodes). These networks have a much larger number of point-to-point requirements. To account for this, we divided the run time by the number of requirements for each example, and plotted the results in Figure 25 on page 77.

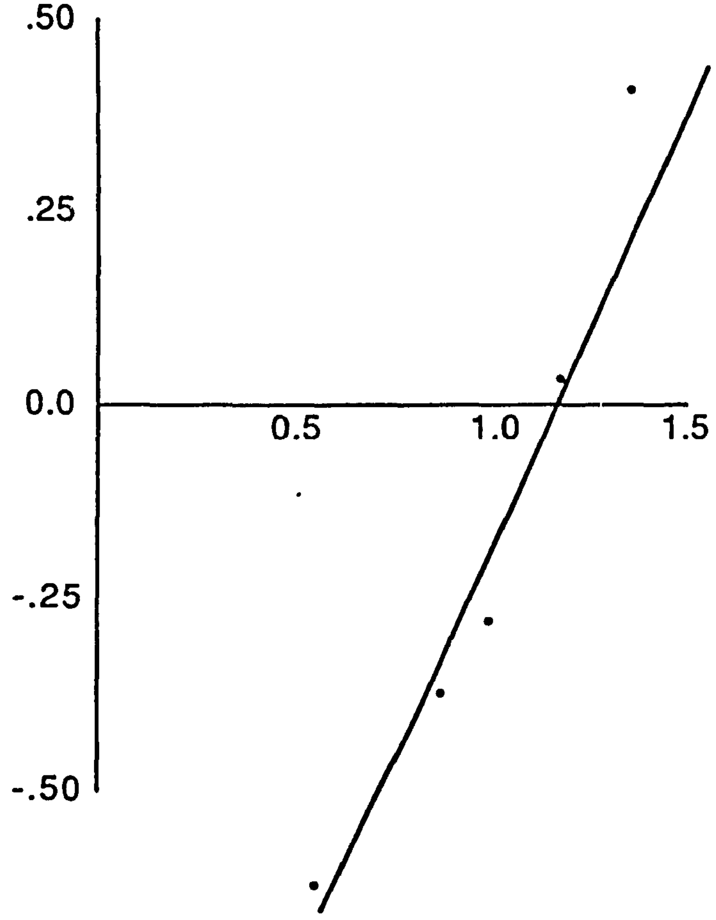

Figure 25. **Normalized CPU Time:** The logarithm of CPU time divided by the number of requirements is plotted against the logarithm of network size.

The slope of the new line is 1.4. If we assume that the requirements are of order n**2, then the algorithm is of order n**3.4. This is consistent with Hansler's results.

5.2.3 Run-time Improvements

As the CPU time was discovered to be quite high, even for relatively small networks (24 nodes), we have given some thought to how the run time might be reduced, to allow the design of larger networks.

A run time profile of the software was obtained, to try to identify procedures which were using excessive amounts of CPU time. It was discovered that the bin packing routine was using approximately 25% and the costing routines about 15% of the total CPU time. In retrospect, this is not surprizing. Each time a requirement is to be added onto the network, the procedure must compute the cost of routing the requirement along each link of the network (in order to apply the shortest path algorithm). To cost the link, for the integrated service, the number of DS-0 channels required must be computed through bin packing.

While the costing and bin packing procedures are fairly well optimized, it may be possible to reduce the number of times these procedures are being called. At present, when determining the shortest path between A and B, all links in the network are considered. This means that the costing and bin packing must be applied to all network links. It should be possible, for a given origin-destination pair, to eliminate some portion of the links that are not "likely" to be on the shortest path, particularly for origin-destination pairs that are close together.

For example, if we are computing the shortest path from Montreal to Ottawa, a link from Calgary to Vancouver is not likely to be on the shortest path. Eliminating these links from the shortest path determination for this origin-destination pair would reduce the number of calls to the costing and bin packing routines, and would speed up the shortest path process itself.

A more sirnificant (though more complex) run-time improvement could be achieved through decomposition of the network into smaller pieces. At present, the network locations consist of the customer locations, and the telco rate centers where the customer locations get access to the services.

The network could be decomposed according to these rate centers (see Figure 26). Each rate center covers a certain geographic area (for instance, the island of Montreal) for which it provides a set of services. There may be many customer locations in a given rate center. The network optimization would be broken down into two phases.

In the first phase, the point-to-point requirements would be aggregated to produce a set of rate center to rate center requirements. This network of rate centers would then be designed according to the procedures previously discussed. In phase 2, the portions of the network within each rate center would be designed. Thus, a large network design is broken down into several small network designs. In the example given, the 15 location problem is broken down into the phase 1 problem, with 3 locations (rate centers) and three phase 2 problems, each with 5 locations.

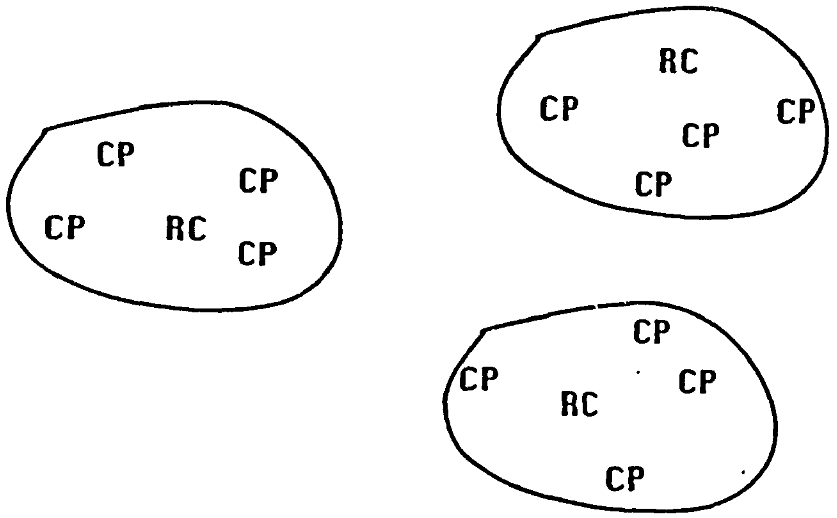

Figure 26. Decomposition of the Network into Rate Centers: The network is decomposed into disjoint sets of locations in the same rate centers.

While some loss of optimallity would be expected, such a technique would clearly be required for large networks (> 100 nodes).

5.3 Cost Savings Analysis

5.3.1 Cost Savings for Each Step

The objective of this section is to measure the cost savings achieved by each step in the algorithm.

The reference cost is determined by satisfying each requirement on its cheapest direct path. Multiple requirements between the same pair of nodes are each routed independently on their least cost direct path. For instance, if between A and B, we require ten voice channels and two 56Kbps data channels, we compute the best routing for the 10 voice channels and then the best routing form the 2 56K data channels separately We do not look for the best service for the two together. We leave that for the algorithm to find.

Each step of the algorithm then produces a new network configuration of lower or equal cost.

We consider three different cases, when all services are available, when only the regular services (not the integrated service) are available, and when only the integrated service is available. The results are shown in Figure 27. For each network, for each of the three cases, we show the percentage cost reduction achieved by each step of the algorithm compared to the reference cost. The average across the sample networks is shown at the bottom of the figure. The results suggest that when all services are available, steps one and two produce significant cost savings, but step three (rerouting of links) could probably be omitted. When designing a network with only the regular services, no savings are obtained from the procedure. When designing networks where only the integrated service is available, only the first step (initial solution) is required.

5.3.2 Savings Due to Integration

The objective of this section is to assess the advantage (cost savings) to the customer due to integrating voice and data onto one network.

This can easily be done by comparing the cost of the voice network using regular voice services (private line, TR) plus the cost of the data network using regular data services (private line, DR) to the cost of the integrated network solution, where all services are available (private line, TR, DR, MS).

This would show the advantage to the customer of using the integrated service, but would not measure the value of integration. Because all of the services are costed differently, the cost differences above will reflect differences due to costing as well as the effect of integration. As the integrated service is significantly cheaper than the regular services, the integrated networks will have a lower cost. This tells us that it is good for the customer to use the integrated service, but not how much of the cost savings was due to integration by the customer.

For example, if a pure voice network is designed using regular services giving design D1, and with the integrated service, giving design D2, the cost of D2 will be less than the

Number of nodes	Step	All Services	Regular Services	Integrated Services
4	Step 1	5.8	0.0	7.1
	Step 2	9.9	0.0	0.0
	Step 3	0.0	0.0	0.0
7	Step 1	11.2	0.3	13.0
	Step 2	2.4	0.0	0.0
	Step 3	0.0	0.0	0.0
10	Step 1	2.1	0.0	7.8
	Step 2	1.0	0.0	0.0
	Step 3	0.0	0.0	0.0
15	Step 1	4.1	0.0	0.2
	Step 2	0.0	0.0	0.0
	Step 3	0.0	0.0	0.0
24	Step 1	0.9	0.6	0.9
	Step 2	0.0	0.0	0.0
	Step 3	0.0	0.0	0.0
Average	Step 1	4.8	0.2	5.8
	Step 2	2.7	0.0	0.0
	Step 3	0.0	0.0	0.0

Figure 27. Algorithm Cost Savings: The%cost savings achieved by each step of the algorithm are shown.

cost of D1, because the service is cheaper. As no voice data integration is involved, none of the cost reduction can be attributed to integration.

What is required is a way to factor out the advantage of the integrated service due to its lower cost, and see if any cost saving remains to be attributed to voice data integration. The comparison we choose to make is between the sum of the cost of the voice network designed with all services plus the cost of the data network designed with all services and the cost of the integrated network designed with all services available. The results are shown in Figure 28.

On the average, network cost was reduced by a total of 27.1%. Of this, 21.5% could be attributed to the lower cost of the integrated service, and 5.6% to the benefit of integrating voice and data.

Nodes	Regular Cost	Best Separate Cost		Integrated Cost	
4	53,788.35	44,171.30	17.9	38,510.30	28.4
7	81,865.16	75,642.49	7.6	65,329.80	20.2
10	195,850.52	113,946.50	41.82	107,406.86	45.16
15	11,971,421.73	9,618,697.67	19.65	9,526,927.41	20.42
24	26,292,644.42	20,886,744.81	20.56	20,670,629.07	21.38
			21.50		27.11

Figure 28. Cost Savings Due to Voice/Data Integration: This figure shows the percentage cost savings due to voice/data integration.

6.0 FUTURE WORK

While this method provides adequate network designs, there are features which could be added to solve the problem more fully. Three of these features are discussed below. Also, as network technology advances, new capabilities can be expected to appear in the networks. In particular, the next major advance will come with the use of digital cross-connect devices which will allow customers to reconfigure their network on demand, or on a scheduled time of day basis.

6.1.1 Multiplexer Placement

Under the existing tariff, the IVD multiplexer can be placed at the customer premise or at the rate center. The IVD multiplexer would be placed at the customer premise if there were a sufficient number of requirements to justify the high bandwidth DS-1 connection between the customer premise and the rate center. If there is insufficient bandwidth required to justify it, the IVD mux is placed at the rate center, and lower bandwidth services are used to connect the customer premise to the rate center.

An advantage of placing the IVD mux at the rate center is that it can be shared with other locations or even other customers, and therefore, the customer is tariffed for only that portion which he is actually using (see Figure 29).

In our algorithm, we have assumed that the IVD mux is always at the rate center. The rationale for this is as follows: At any intermediate step in the algorithm, it is difficult to

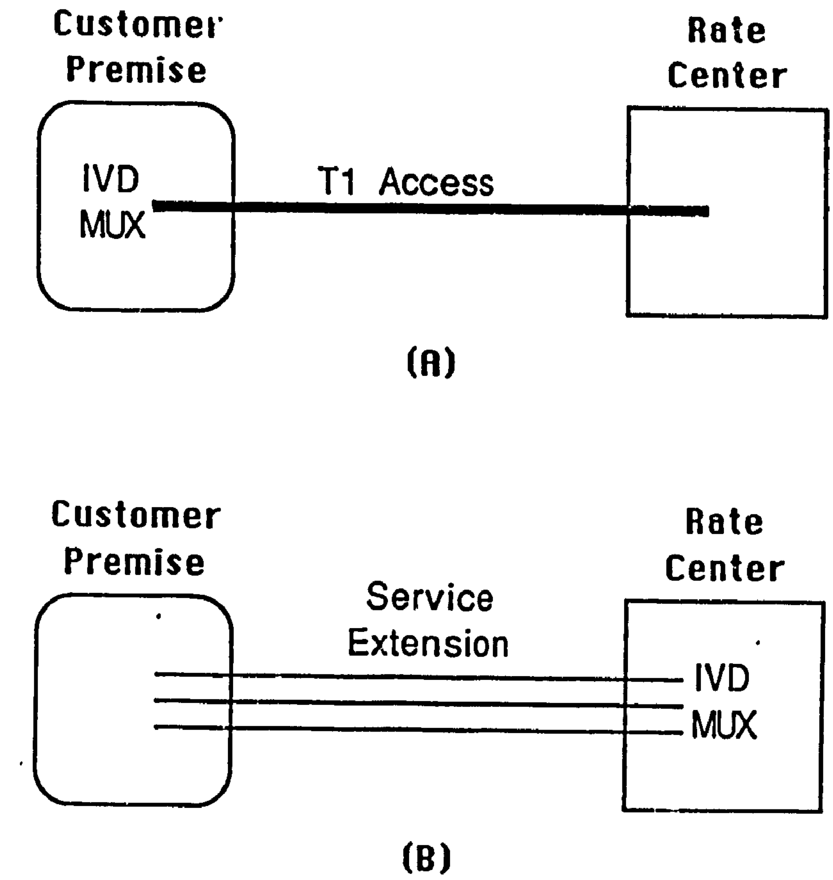

Figure 29. **Placement of the IVD Multiplexer:** The IVD multiplexer can be placed at the customer premise or at the rate center.

know where the IVD mux should be placed, as not all requirements have been processed. If the IVD mux is assumed to be at the customer premise, the integrated service may not prove in because the DS-1 access is not cost justified. By assuming that the IVD mux is at the rate center, we allow locations which can't justify the DS-1 access, to use the service. To identify those locations which can justify the DS-1 access, we suggest the following post process.

After the network has been designed, examine each location which uses the integrated service. For each such location, compute the access cost with service extension (IVD mux at the rate center) and standard access (DS-1). If the DS-1 access is cheaper, place an IVD mux at the location and adjust the network cost. Note that changing the access from service extension to standard does not affect the cost of other locations that may have been using service extension at that rate center.

6.1.2 T1 Modularity

The integrated service, Megastream, which is available in Canada, is available in multiples of DS-0 bandwidth. This has the effect of giving a reasonably "smooth" cost function. Currently, in the United States, integrated services are available only in DS-1 units, that is, T1 channels. This introduces large steps into the cost function

Our heuristic is sensitive to the shape of the cost function. Because circuits are added incrementally to the network, the heuristic performs well under a smooth cost function. When applied under the step cost function (with large steps), the heuristic fails to find economic solutions. In cases where two or more requirements combined could cost justify a T1 (but neither on their own can justify it), the heuristic does not find this solution because it looks at the requirements one by one.

One approach might be to smooth out the T1 cost function, by using a cost per bit sec of bandwidth, and allow the heuristic to install "partial" T1 circuits. A post process would then be applied to determine, for each partial T1, whether a T1 does in fact prove in. If not, then alternative services would be used in place of the partial T1. More investigation is required into this idea.

6.1.3 Iteration Over the Design

Reviewing the design process shown in Figure 9, we see that the process consists of running independent voice and data network designs, and combining those results in a channel grouping and rerouting step.

The private line requirements which form the input to the channel grouping and rerouting step, are determined in the voice and data network design steps, based on the economic trade off between private line services and alternatives. The channel grouping and rerouting step then finds cheaper ways of providing the private line services, primarily by combining voice and data on integrated services. This reduction in the cost of providing the private line facilities can affect the economic trade-offs between the private line services and the non-private line services that were determined in the voice and data design steps.

Hence, we suspect that the voice and data network designs should be repeated, based on the expected costs of providing the private line services as determined in the channel grouping and rerouting step. If the economic trade-offs change, the private line requirements coming out of the voice and data design steps will change. This implies that the channel grouping and rerouting step should also be repeated

Hence, we can introduce an iteration into the process. Only one example of this type was attempted, and a cost reduction of about 1.5% was achieved. More analysis should be performed to determine the value of this iteration. If it is of value, then the controls on the iteration process and the conditions of convergence would have to be explored.

6.1.4 Network Reconfiguration

The network designs considered here, are static networks. That is, the design is performed for one set of traffic input, and the resulting network design does not change over time. In real networks, traffic patterns change over time. With the introduction of digital cross connect devices, the network topology and capacities can change over time also, to best meet the changing traffic demands.

This completely changes the nature of the design problem. Traffic for several time periods of the day are required. Assuming that multi-hour voice and data dimensioners were available, determination of the amount of bandwidth required on a link would have to take into account channel requirements which were changing on an hourly basis. It is not clear that this heuristic can be adapted to handle this type of network design.

7.0 SUMMARY

In this report, we have described a problem in integrated voice and data private network design. Circuit switched voice networks are integrated with multipoint data networks, under fixed partitioning of bandwidth. This integration is possible due to new private integrated network services offered by the telephone companies.

A design methodology has been proposed which is based on existing voice and data network design capabilities. Voice and data network designs are determined independently, resulting in separate voice and data networks. These networks are then combined into an integrated network, using a channel grouping and rerouting heuristic.

The channel grouping and rerouting heuristic is composed of three steps. In the first step, initial solution, circuits are added incrementally to the network (starting with an empty network), along their shortest path routings. In the rerouting of circuits, circuits are examined one by one, removed from the network, and then put back into the network along their shortest path routing (which may be different from the original routing). In rerouting of links, each link is considered for rerouting, the link is removed from the network, and then put pack into the network along its shortest path route.

A prototype of the algorithm has been implemented and the prototype has been used to design sample networks to assess the design heuristic and the network cost reductions due to the integration of voice and data on private leased integrated services. Five sample networks were designed and the heuristic was demonstrated to design cost effective integrated networks.

The steps of the heuristic were analyzed in terms of the cost savings achieved and the CPU time required to produce network designs. It was determined that the first (initial solution) and second (rerouting of circuits) steps produced significant cost savings. The third step (rerouting of links) did not produce significant cost savings, and could be removed from the heuristic. The average cost saving observed in the sample networks was 27.1% of the monthly lease cost. The CPU time required to run the algorithm is approximately of order n**3.4, where n is the number of nodes in the network.

We also partioned the cost saving into a portion due to the integration of voice and data, and a portion due to the inexpensive nature of the integrated service. We presume that the integrated service is less expense than regular services due to the telephone company's ability to reduce operating costs by integrating voice and data on the physical facilities underlying the private network leased service offerings. We observed that of the 27.1% total cost savings, 21.5% was due to the low cost of the integrated service, and 5.6% was due to the integration of voice and data bye the customer.

8.0 REFERENCES

[AVEL 82] O.A. Avellaneda, J. Hayes, M.M. Nassehi. **A Capacity Allocation Problem in Voice-Data Networks.** IEEE Transactions on Communications, Vol. 30, No. 7, July 1982, 1767-1772.

[DIJK 59] E.W. Dijkstra. **A Note On Two Problems in Connection With Graphs.** Numerische Mathematik 1, 1959, 269-271.

[DUNK 86] L. Dunkelman. **PND User's Guide and Reference Manual.** Internal Report, Bell-Northern Research, 1986.

[ESAU 66] L.R. Esau, K.C. Williams. **On Teleprocessing System Design Part II A Method for Approximating the Optimal Network.** IBM Systems Journal, Vol.5, No. 3, 1966, 142-147.

[GERL 73] M. Gerla. **The Design of Store-and-Forward (S/F) Networks For Computer Communications.** Ph.D. Thesis, Computer Science Department, School of Engineering and Applied Science, University of California, Los Angeles, 1973.

[GERL 84] M. Gerla, R.A. Pazos-Rangel. **Bandwidth Allocation and Routing in ISDN'S.** IEEE Communications Magazine, Vol. 22, No. 2, February 1984, 16-26.

[GITM 77] I. Gitman, H. Frank, B. Occiogrosso, W. Hsieh. **Issues in Integrated Network Design.** Proceedings of the International Conference of Communications, June 1977, 38.1, 36-43.

[GOLD 76] B.L. Golden. **Approaches to the Stock Cutting Problem.** AIIE Transactions, Volume 8, No. 2, June 1976, 265-274.

[HANS 73] E. Hansler. **A Heuristic Approach to the Multi-terminal Telpak Problem.** International Symposium on Information Theory, 1973, C2 2.

[KENN 86] T. Kennard, A. Madras, J. Hickin, M. Rivet. **DTS Functional Specification.** Technical Report TL86-0059, Bell-Northern Research, 1986.

[KERS 74] A. Kershenbaum, W. Chou. **A Unified Algorithm for Designing Multidrop Teleprocessing Networks.** IEEE Transactions on Communications, Vol. COM-22, No. 11, November 1974, 1762-1771.

[KLEI 76] L. Kleinrock. **Queuing Systems, Volume 2, Computer Application.** New York: Wiley, 1976.

[KRON 81] M. Krone. **Hybrid Circuit-Packet Network Design Algorithm.** BNR Technical Memorandum TM 34039, September 1981.

[MINA 74] R. Mina. **Introduction to Teletraffic Engineering.** Telephony Publishing Corporation, Chicago, Illinois, 1974.

[PACK 75] C. Pack. **Configuring a Private Line Circuit According to the Hi-Lo Tariff - a Minimal Steiner Tree Problem.** Proceedings of the 4th Data Communications Symposium, ACM. Quebec, Canada, 1975, 6-12 - 6-18.

[PAZO 83] R. Pazos-Rangel. **Evaluation and Design of Integrated Packet Switching and Circuit Switching Computer Networks.** Dissertation, UCLA, 1983.

[SCHW 77] M. Schwartz. **Computer-Communication Network Design and Analysis.** Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

[TRUI 54] C.J. Truitt. **Traffic Engineering Techniques for Determining Trunk Requirements in Alternate Routing Trunk Networks.** The Bell System Technical Journal, March 1954, 277-302.

[WILK 56] R.I. Wilkinson. **Theories for Toll Traffic Engineering in the U.S.A.** The Bell System Technical Journal, March 1956, 421-514.