(1341) (1341)

SIMULATION OF SOME DYNAMIC ROUTING ALGORITHMS

Dissertation submitted in partial fulfillment for the award of the degree of

Master of Technology

in

Computer Science

By

Filbert Minj

School of Computer & Systems Sciences

Jawaharlal Nehru University

New Delhi -110067

January 1997

Declaration

This is to certify that the dissertation entitled "*Simulation of Some Dynamic Routing Algorithms*" submitted by me to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi**, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in Computer Sciences, is a record of the original work done by me under the supervision of Dr. Sonajharia Minz, during the Monsoon Semester, 1996.

New Delhi
January 1997

Filbert Minj

Certificate

This is to certify that the dissertation entitled ***"Simulation of Some Dynamic Routing Algorithms"*** submitted by Filbert Minj to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi**, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in Computer Sciences, is a record of the original work done by him under the supervision of Dr. Sonajharia Minz, during the Monsoon Semester, 1996.

Prof. G. V. Singh
(Dean)

Dr. Sonajharia Minz
(Supervisor)

Filbert Minj
(Candidate)

Acknowledgment

I would like to express my sincere gratitude and thanks to Dr. Sonajharia Minz for her guidance, without which this dissertation would have been beyond my competence. I also wish to express my indeptness to Prof. G.V. Singh Dean, SC&SS, JNU for providing me excellent infrastructure facility which enabled the successful completion of this work.

Beside this I would like to thank Prof. P.C. Saxena and all my friends who helped me at the various stages of this work.

New Delhi,
January 1997

Filbert Minj

Abstract

Routing is an indispensable telecommunications network function that connects a call from the origin to the destination and it is the heart of any architecture, design and operation of any network. The introduction of dynamic routing into several telecommunications networks has resulted in a marked improvement in network connection availability besides simultaneously reducing the costs. This work includes a brief description of routing in general communications networks, followed by introduction to the concepts of dynamic routing in telecommunications and simulation of three dynamic routing algorithms.

The algorithms are **Dynamic Nonhierarchical Routing (DNHR), Real-Time Dynamic Traffic Routing (RTNR)** and **State and Time-Dependent Routing (STR)**. DNHR is a hybrid of *time-variable routing* and a simple implementation of adaptive routing. RTNR is real-time dynamic traffic routing strategy for *integrated class-of-service* networks. STR is *state and time-dependent* which uses distributed and centralized methods for routing.

CONTENTS

List of Figures

List of Tables

CHAPTER ONE

1. Introduction

1.1. Routing in Network

The real function of the network layer is routing the data packets from the source machine to the destination. Routing is an issue if the source and destination are not on the same network. The routing algorithm is that part of the network layer software that is responsible for deciding which output line an incoming packet should be transmitted. If the packet uses datagrams internally, this decision must be made anew for every arriving data packet. However, if the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up.

There are certain properties that are desirable in a routing algorithm:

- simplicity
- robustness
- correctness
- stability
- fairness
- optimality.

Routing in a network typically involves a rather complex collection of algorithms that work more or less independently and yet support each other by exchanging services or information [4]. The complexity is due to a number of reasons. First, routing requires coordination between all the nodes of the subnet rather than just a pair of nodes. Second the routing system must cope up with link and node failure, requiring redirection of traffic and an update its routes when some areas within the network become congested.

There are number of ways to classify routing algorithms. One way is to divide them into centralized and distributed routing algorithms [4]. In centralized algorithms, all route choices are made at a central node. The computation of routes is shared among the network nodes with information exchanged between them as necessary in distributed algorithms.

Another classification of routing algorithms relates to whether they change routes in response to the traffic input patterns [12]. These are adaptive and nonadaptive. Nonadaptive algorithms do not based their routing decision on measurements or estimates of the current traffic and topology, whereas adaptive once do. There are many algorithms in use with different levels of sophistication and efficiency. This variety is partly due to historical reasons and partly due to the diversity of needs in different networks. Another way of classification of routing algorithms is as static, quasi-static, and dynamic, according to how adaptive they are [15]. In a static routing algorithms, the choice of routes is predetermined and fixed for relatively large time periods. A dynamic routing algorithm in contrast, allow continuos changes in routing decisions to reflect the current traffic and topological changes. But such routing algorithms are often very complex and require large amount of information exchange and computation. The shortest path routing algorithm widely used today fall approximately into the quasi-static; in which the link distances remains constant for a short period of time but can be updated when significant changes occur.

1.2. Dynamic Routing

Dynamic routing describes routing strategies which are time-sensitive or possibly real-time state dependent, as opposed to hierarchical routing rules which are time fixed [1]. Dynamic routing techniques are applicable to all networks, including metropolitan networks, long distance national networks, global international networks, future broadband networks, packet switched networks, and circuit switched networks. Dynamic routing are often involved as a key to the success of internet.

1.3. Advantages of Dynamic Routing

Dynamic routing is an efficient method of traffic control in which call routing is frequently altered according to the status of the network and/or anticipated demand shifts, so that the network can respond quickly and properly to the changes in the traffic and/or facility conditions. The dynamic routing scheme increases network efficiency by routing calls away from busy areas through lightly loaded portion of the network. This additional routing intelligence maximizes utilization of all available network capacity and leads to the reduction in busy-hour blocking. Furthermore, dynamic routing enhances the robustness of networks with respect to equipment failures and unexpected traffic surges. Dynamic routing provides the opportunity for international carriers to exploit the non-coincidence of busy hours and save international network cost while maintaining a specified blocking grade of service. Alternatively, with the introduction of dynamic routing, network throughput can be significantly increased with no additional investment in network capacity.

CHAPTER TWO

2. Motivation and Background

The dynamic routing networks were first introduced by Segall and his co-workers. First implemented during the 80s, dynamic routing has made major strides in the past a few years and now deployed in four major networks AT&T USA, Stentor Canada, Bell Canada, and NTT Japan. This has provided considerable benefits in improved performance quality and reduced costs. These benefits have motivated the extension of dynamic routing to integrated networks with multiple classes-of-services and to networks with dynamically rearrangable transport capacity. Dynamic routing in telecommunication networks has now been the subject of world-wide study and interest. Service providers, equipment providers, and academic institutions throughout the world have active research programs in this area, and many networks are in planning and development stage.

The rapid changes that have occurred in telecommunication during the last decade have allowed network providers to offer their customers sophisticated service capabilities, which represent new opportunities in business as well as in social activities. Circuit-switched telecommunications have evolved from plain old telephony towards the integrated digital network, that already offer a set of new services. Consequently; this has increased customer demand, combined with higher quality of service requirements. Network management and administration, therefore represent a new challenge for network operators, that can be delt with to the technological innovations in the telecommunications industry. In the mid 70's, new routing techniques were developed to introduce flexibility in traffic control. They resulted in the conversion of the well-established hierarchical network to a nonhierarchical one. Though these techniques are traffic performance oriented, a lot of emphasis has been placed primarily on network cost savings that are obtained at the design stage. Together with decrease of transmission cost, the vulnerability of modern and the high grade of service requirements implied by the introduction of new services force us to consider again the original purpose of the new routing techniques, that is the improvement of network efficiency and robustness in most situations. Since new services induce new traffic profiles or unexpected traffic surges, other traffic controls are needed to guarantee the integrity of the network and a fair grade of service to the customer. Therefore, the integration of routing methods and traffic volume control methods allow us to consider network control in a new light.

Hierarchical networks conceptualized in the '50s with the development of common control switching systems that enabled the introduction of the alternate routing concepts based on the overflow technique [6]. As there were still technological limitations that require the guarantee of integrity between numbering, signaling, and routing networks were structured into different levels used to concentrate traffic from one region to another. To prevent a call from returning from one of the switching centers along its routing path the call looping phenomenon-overflowing and selection of alternate paths were subjected to hierarchical rules. In addition to limited measurement capabilities in the switches and lack of computing facilities, routing was determined at the design stage and remained fixed under normal conditions until the next design stage. With the advent of Stored Program Controlled switches, the emergence of common channel signaling systems and the development of data networks, new processing capabilities allowed for the evolution of traffic routing from fixed hierarchical to flexible nonhierarchical [6]. The problems that persist are identified to be:

- As traffic routing is closely related to network design in fixed hierarchical networks, it becomes clear that using a single routing pattern determined according to busy-hour, busy-season traffic measurements cannot allow the efficient accommodation of traffic in all situation.
- Network robustness is not ensured, also because of the hierarchical future of routing

In order to improve network efficiency, new routing technique referred to as dynamic routing with decreased network costs was evolved. The principle of dynamic routing is to profit from the existence of spare capacities in parts of the network while other parts are overloaded in order to maximize the use of network resources. The time evolution of traffic routing strategies is summarized in figure 2.1.

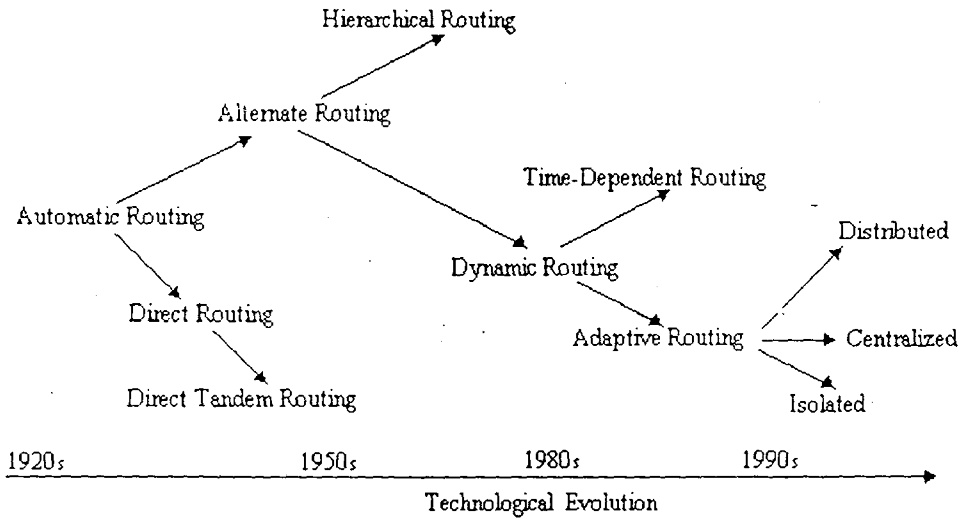

Hierarchical Routing
Alternate Routing
Time-Dependent Routing
Automatic Routing
Distributed
Dynamic Routing
Direct Routing
Adaptive Routing
Centralized
Direct Tandem Routing
Isolated
1920s
1950s
1980s
1990s
Technological Evolution

Figure 2.1

Factors driving dynamic routing evolution, include performance quality improvements, new services introduction and technological evolution [1]. Performance quality improvement is achieved when network routing increases real time adaptivity and robustness to load shifts and failure. Introduction of new services is aided when network routing extends network routing and flexible bandwidth allocation to new services within an integrated network. Technological evolution is supported when network routing capitalizes on new transmission, switching and network management technologies to achieve simpler, more automated, and more efficient networks than are now possible.

A number of dynamic routing strategies have been proposed and implemented, including dynamic nonhierarchical routing (DNHR) deployed in the AT&T network from 1984 to 1991, real-time network routing (RTNR), deployed in the AT&T network in 1991, Dynamically controlled routing (DCR), deployed in the Stentor Canada long distance network and Bell Canada Toranto metropolitan area network in 1991, dynamic alternate routing (ADR), deployed by British Telecom, state and time-dependent (STR), initially deployed by Nippon Telephone and Telegraph in NTT network in 1992, system to test adaptive routing (STAR), deployed and tested by France Telcom in the Paris network; DR-5 deployed and tested by Bellcore for application to metropolitan Local Exchange Carrier networks; and worldwide intelligent network (WIN) dynamic routing, an international dynamic routing strategy initially deployed in 1994 by conscrtium of several countries. Currently implemented dynamic routing systems use different methods: RTNR is a distributed real-time state-dependent approach, STR is a distributed learning approach, and WIN is a distributed traffic dependent approach. Ubiquitous deployment of dynamic routing networks is inhibited by the lack of internetworking of different vendor equipment to implement dynamic routing strategies. A recent ITU-TS proposal aims to standardize dynamic routing control an information exchange within four Common Channel Signaling (CCS) messages.

We elaborate on hierarchical traffic routing, dynamic traffic routing, and dynamic transport routing the three stage of routing evolution [1].

2.1. Hierarchical Traffic Routing

Network routing plans have always been closely aligned with the capabilities of technology. Such is the case with hierarchical traffic routing when in 1930 a hierarchy of switching center was established by the AT&T General Toll Switching Plane. This plan limited the number of switches in a long distance call to six in any connection including the switching centers to which the local switches connected. Routing through the network was usually handled manually by operators at specialized switchboards following rather involved operating procedures. During the late 40s, planning for nationwide customer dialing required that the 1939 network plane be revised. The new plane established ten regions and a five level hierarchy of switching systems centering on the regional switches. By 1951, this five level hierarchy of switching centers had evolved, and direct distance dialing (DDD) was introduced concurrently with automatic alternate routing [1]. The technologies which enabled these improvements include the No. 4A crossbar system, automatic message accounting, and efficient network design techniques for alternate routing networks. The planning for DDD and the five level hierarchical network was of such quality that only minor refinements were necessary until it was replaced by DHNR in the 80s.

Figure 2.2 illustrates hierarchical traffic routing which served the AT&T network into the 80s. The switching hierarchy used in the AT&T network prior to divestiture in 1984 had five levels as illustrated in figure 2.2 in which successively higher level offices concentrated traffic from increasingly larger geographical areas [1]. At the highest level of switching were ten Regional Switching Centers, or class 1 offices, shown as squares in figure 2.2. At the next level of switching were the Sectional Switching Centers, then Primary Switching Centers, then Toll Switching Centers, and at the lowest level End-Offices, or Class 5 offices, which are part of the

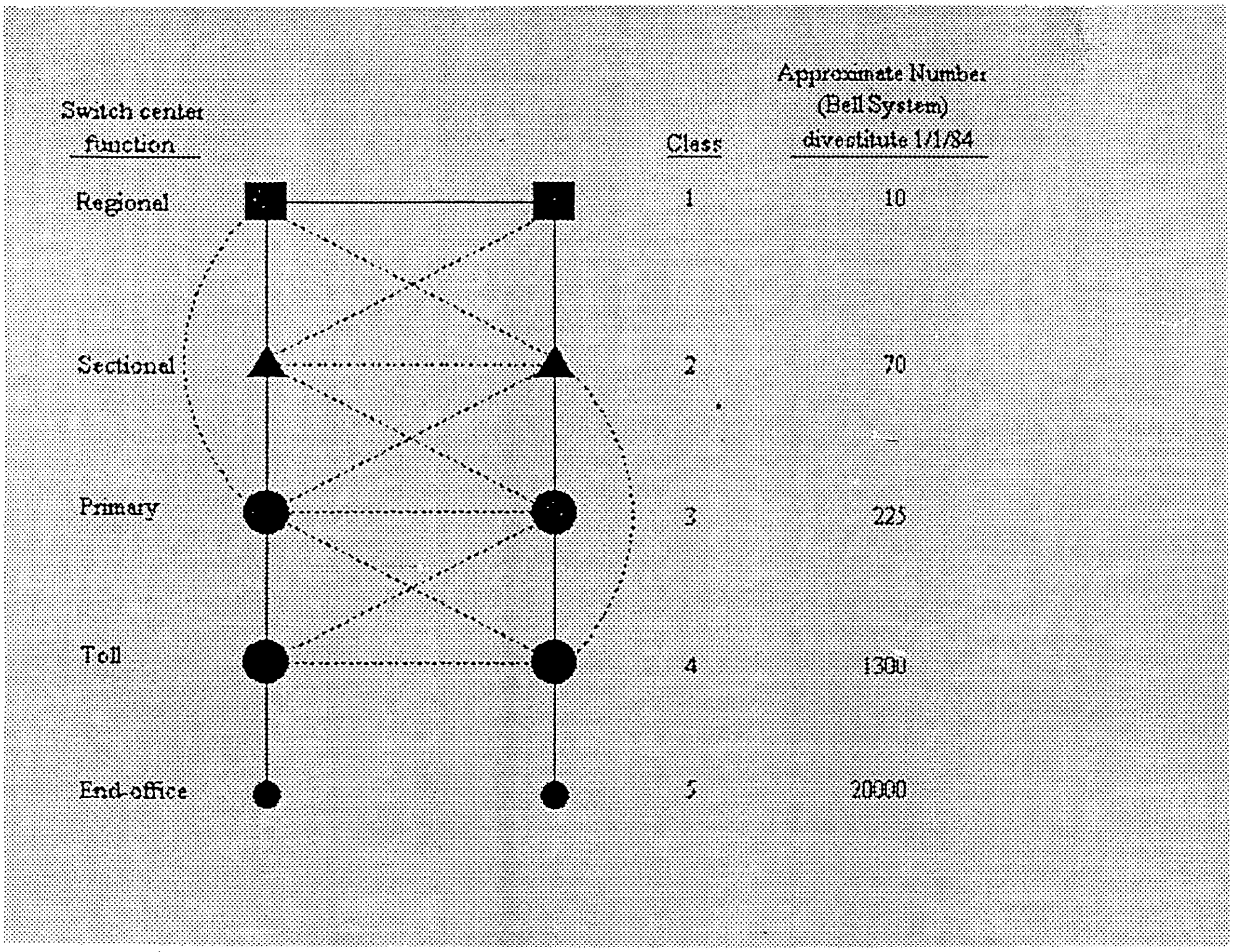

Switch center function
Class
Approximate Number (Bell System) divestitute 1/1/84
Regional
1
10
Sectional
2
70
Primary
3
225
Toll
4
1300
End-office
5
20000

Figure 2.2

local switching network. The approximate numbers of these switching centers at the time of divestiture in 1984 are given in figure 2.2.

Two types of trunk connect these switches High-usage trunk groups connect any two switches that have sufficient traffic between them to make a direct route economical. Final trunk groups are the links between each switch and its immediate superior in the hierarchy, together with the final trunk groups interconnecting all regional centers. An office connected by a final trunk group to a higher class office is said to be home on thàt office. High-usage trunk groups are sized to handle only a portion of the traffic directed to them and the switching systems redirect traffic by automatic alternate routing to a different trunk group when all circuits of a high-usage group are busy. At each stage, the alternate routing plan shifts overflow calls from the more direct route towards the final route. Final groups are designed to handle their own direct traffic plus overflow traffic from high-usage groups with low blocking, such as an average loss of one call in a hundred. The hierarchical routing pattern precludes switching or looping calls back on themselves or using an excessive number of trunks on a call.

2.2. Dynamic Traffic Routing

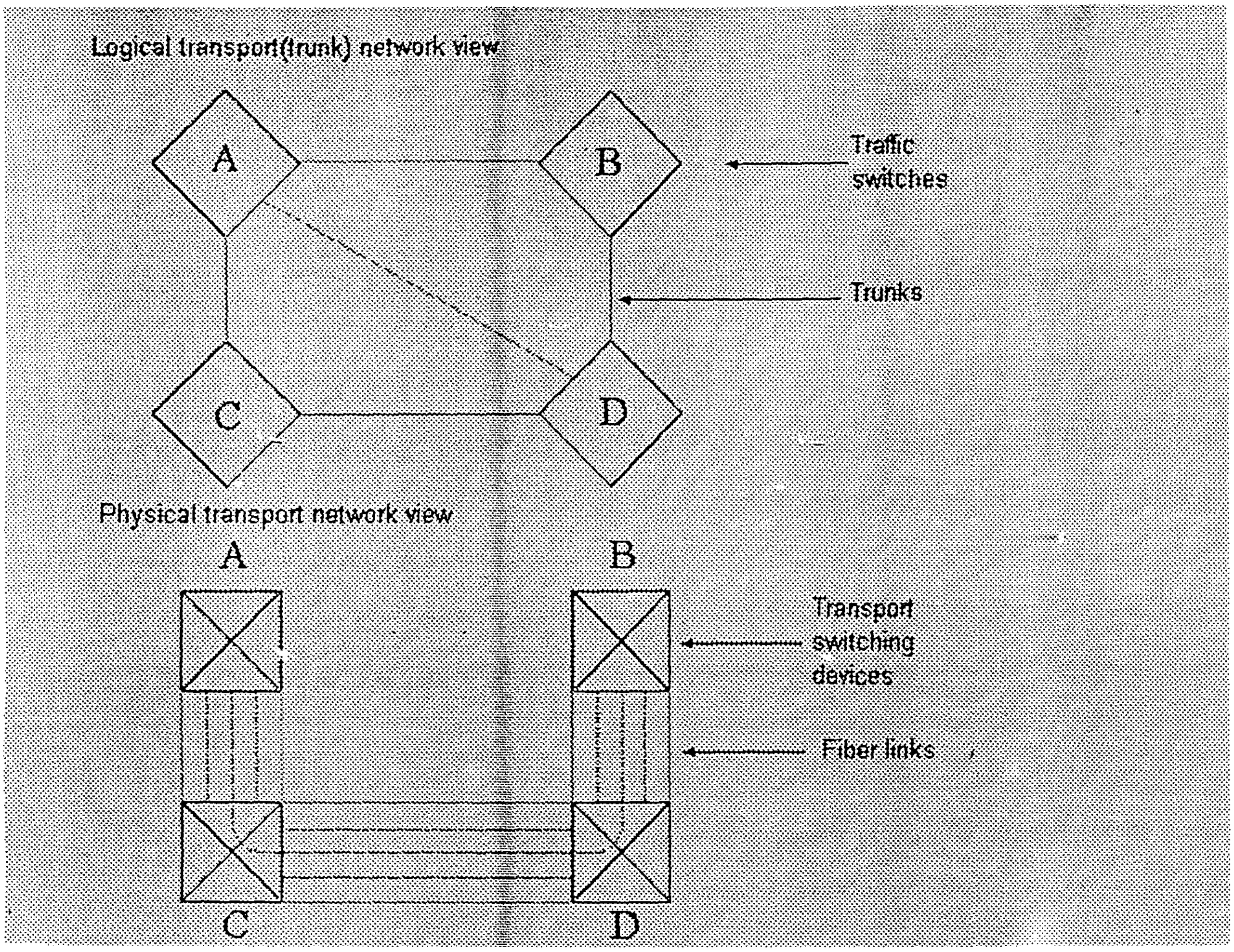
Dynamic traffic routing in telecommunications network again capitalizes technological advances. During the 70s network technology evolved rapidly toward stored program control capabilities. In the AT&T network, for example, the network evolved toward handling consideration higher traffic volumes with fewer switches because the 4ESS switch a very large switching system with more than three times the capacity of the No. 4 crossbar, became the primary switching vehicle in the AT&T network and enabled consolidation of many electromechanical switches into a single large ESS switch. ESS switches provide extensive stored program control capabilities that enable sophisticated routing instructions to be executed. In addition, the CCS network, a high-speed, high-capacity signaling system for linking ESS switches, has been deployed. CSS enables far more routing information to be sent between switches in call set up messages than was possible with in hand signaling, and allows ESS switches to make full use of their stored program control capabilities. Finally computerized network management has made major advances in automating complex network functions. These systems can process vast amount of raw data, implement very complex network design procedures, and automate the flow of routing instructions and other network management information into and out of ESS switches. Therefor the simple manual design rules for hierarchical network could be made more efficient and sophisticated with computerized implementation.

Dynamic traffic routing allows ESS routing tables to be changed dynamically, either in a preplanned time varying manner, or on-line in real time. With preplanned dynamic traffic routing (PPDTR) strategies, routing patterns contained in routing tables might change every hour or at least several times a day to respond to known shifts in traffic loads, and in general PPDTR routing patterns change with a time constant greater than a call holding time. Hence a typical PPDTR strategy may change routing patterns every hour, which is considerably longer than a typical call holding time of a few minutes. These routing patterns are preplanned, preprogrammed, and recalculated perhaps each week within the performance management routing control function.

Real-time dynamic traffic routing (RTDTR) does not depend exclusively on precalculated routing patterns. Rather, the switching system senses the immediate traffic load and if necessary searches out new paths through the network possibly, as in RTNR, on a call-by-call basis. With RTDTR strategies, routing tables change with a time constant less than a call holding time. RTNR uses real-time exchange of network status information , with CCS query and status messages, to determine an optimal route from a very large number of possible choices. With RTNR, the originating switch first tries the direct route and if it is not available, finds an optimal two-link path by querying the terminating switch through the CCS network for the busy-idle status of all trunk groups connected to the terminating switch. The originating switch compares its own trunk group busy-idle status to that received from the terminating switch, and finds the least loaded two-link path to route the call.

2.3. Dynamic Traffic and Transport Routing

As with dynamic traffic routing, the introduction of new technology opens opportunities for dynamic transport routing. Dynamic transport routing can combined with dynamic transport routing to shift transport bandwidth among nodes pairs and services, through use of flexible transport switching technology. Dynamic transport routing can provide automatic trunk provisioning, diverse trunk group routing, and rapid trunk restoration for improved transport capacity utilization and performance under stress. Figure 2.3 illustrates the difference betwwen the physical transport network and the logical transport (trunk) network. Trunks are logical connections between network switches, which are routed on the physical transport network.



Logical transport(trunk) network view
A
B
C
D
Traffic switches
Trunks
Physical transport network view
A
B
C
D
Transport switching devices
Fiber links

Figure 2.3

CHAPTER THREE

3. Dynamic Routing Algorithms

3.1. Dynamic Nonhierarchical Routing (DNHR) network

For more than 50 years the hierarchical routing structure was used in AT&T. Then it converted to dynamic nonhierarchical routing (DNHR). It completed in 1987; making it the first national dynamic routing network in the world [1]. It has resulted in a marked improvement in network connection availability and robustness while simultaneously reducing network costs. The DNHR strategy is a hybrid of time-variable routing and a simple implementation of adaptive routing, combined sophisticated off-line optimization of trunking and routing pattern design [1]. The off-line systems provide centralized and highly automated network design and control, including forecasting, servicing, and network management. Embedded within the design system is the DNHR design algorithm, which determines the trunking and routing patterns for the entire DNHR network at once [2]. Real-time management of the DNHR network is controlled from a single centralized system, which uses traffic measurements gathered in each 4ESS switch and analyzes these data every five minutes to monitor and control network availability. A high degree of flexibility has been achieved in DNHR network control: routing changes to correct network service problems and maximize network completion can be made every week, if necessary. Automatic routes can be activated by the network management system every five minutes to provide automatic modification of traffic flows in the

3.1.1. Dynamic Routing Concepts in DNHR

Figure 3.1 illustrates the network structure that incorporates DNHR into the AT&T intercity network. DNHR is applicable to stored program control/common channel

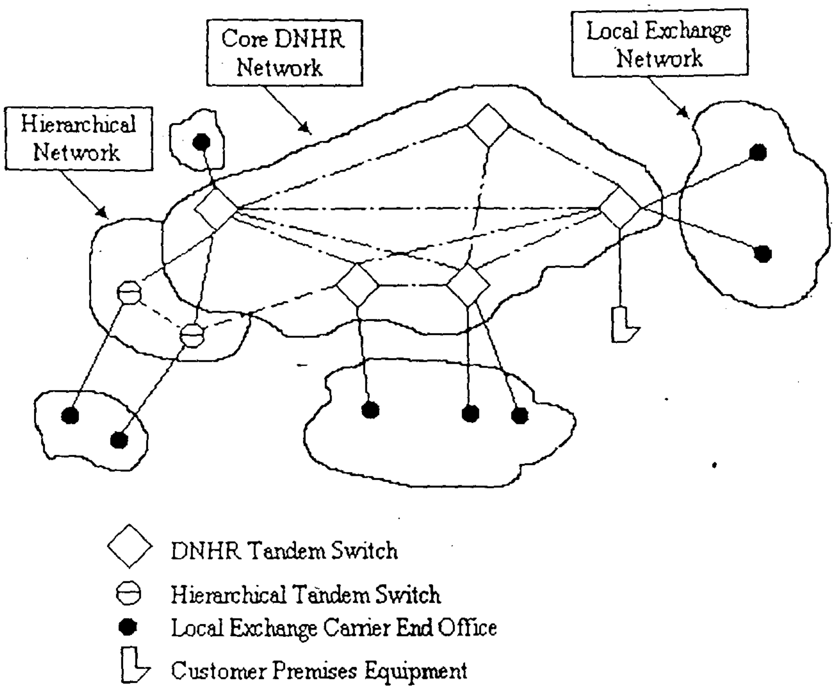

Core DNHR Network
Local Exchange Network
Hierarchical Network
DNHR Tandem Switch
Hierarchical Tandem Switch
Local Exchange Carrier End Office
Customer Premises Equipment

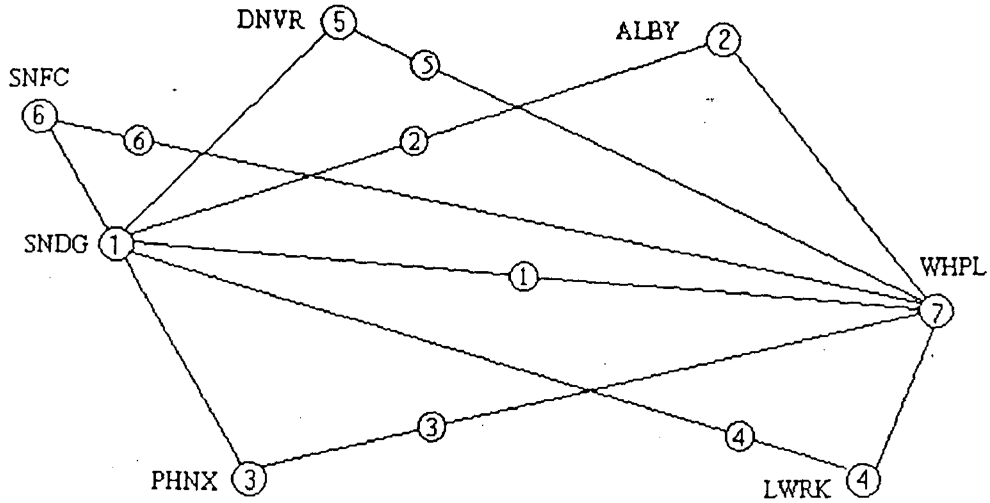
Figure 3.1

signaling network. DNHR network has only one level of tandem switching system, and local exchange carrier end offices and customer equipment are directly connected to, or home to the higher-level tandem switches in the DNHR environment, as they were in hierarchy. The local exchange carrier end offices and customer premises equipment switch customer telephone lines to each other or to the interchange carrier network. The DNHR portion of the AT&T intercity network consists of 4ESS switches interconnected by the common channel signaling network. Dynamic routing rules are used only among DNHR switches, and conventional hierarchical routing rules are used among all other pairs of switches. There are a number of small switching systems employing only the hierarchical rules. Many of these smaller systems designated at hierarchical tandem switches. From the viewpoint of the hierarchical switches, the DNHR switches appear as a large network of the highest-level hierarchical centers, and hierarchical routing patterns are therefore defined by this hybrid structure.

The DNHR network handles several categories of traffic load [2]. The first is the load that originates and terminates at local exchange carrier end offices or customer equipment homed directly on DNHR switches. The DNHR network must also route overflow and through-switched loads from the subtending hierarchical switches. A DNHR originating switch is able to identify, for a call first entering the DNHR network, the terminating switch within the DNHR network. Once a call is identified from the routing translation as being a DNHR call, dynamic routing rules are employed to complete the call on, at most, two DNHR trunk groups between the DNHR originating switch and the DNHR terminating switch.

3.1.2. Algorithm

The dynamic routing method, called two-link DNHR with crankback, is illustrated in the figure 3.2. The example is for intercity DNHR, but a similar routing technique can also be applied to metropolitan DNHR or international DNHR. The strategy capitalizes on two factors :



Traffic Allocation

SNDG WHPL Routing		Morning	Afternoon	Evening	Weekend
Routing Sequence # 1	①→②→③→④→❺→❻	✓			
Routing Sequence # 2	①→③→❷→❹→❺→❻		✓		
Routing Sequence # 3	①→④→③→❷→❺→❻			✓	
Routing Sequence # 4	③→④→❶→❷→❺→❻				✓

○ Engineered Path

● Real-Time Path

Figure 3.2

- Selection of minimum-cost paths between the originating and terminating switches
- Design of optimal, time-varying routing patterns to achieve minimum-cost trunking by capitalizing on noncoincident network busy periods.

Keeping above point in mind we write the algorithms as follows:

```
Done ← FALSE;
Initialize;
While (not Done)
Begin
        Select candidate paths;
        Optimize path flows;
         Define sequential routes;
          Dimension network;
          If (call not blocked) then
           Begin
                  Call is set up;
                   Done ← TRUE;
           End
          Else
                Update optimal link blockings;
End
```

The dynamic, or time-varying, nature of the routing scheme is achieved by introducing several route choices [2]. The DNHR routes consist of different sequences of paths, and each path has one or, at most, two links or trunk groups in tandem. In figure 3.2 the originating switch at SNDG retains control over a dynamically routed call until it is either completed to its destination at WHPL or blocked. A call overflowing at any link of the two link connection is returned to the originating switch, for possible alternating routing. Control is returned by sending messages to the originating switch by common channel crankback signal.

Each of the four routing sequences illustrated in the figure 3.2 uses a subset of the six paths in a different order. Each routing sequence results in a different allocation of link flows, for example the first choice paths normally carry the maximum flow of switch-to-switch load, but all satisfy a switch-to-switch blocking requirements. The paths used in various time periods need not be the same. Since many intercity traffic demands change with time in a reasonably predictable manner, the routing also changes with time to achieve maximum trunk utilization and minimum network cost.

The only decisions necessary in real time involve network conditions that become known in real time, such as actual loads, failure, and overloads. The real-time, traffic-sensitive(or adaptive) aspect of the routing strategy, involves the use of additional real time paths for possible completion of calls that overflow the preplanned routing sequences. The preplanned, or engineer, paths are designed to provide the objective blocking grade of service. The candidate real-time paths, which are determined by the central forecasting servicing system, provide additional possible completion of calls that would otherwise be blocked, subject to trunk reservation restriction. Trunk reservation requires that one more than a specified number of trunks the reservation level be free on each trunk before a real-time connection is allowed. This prevents calls that normally use a trunk group from being swamped by real-time-routed calls. Heavy overloads and certain failure may result in traffic overflowing the real-time paths. When this occurs, additional paths may be implemented automatically or manually as the result of network management.

The design of network routing patterns depend primarily on performance off-line, centralized computations that determine the optimal routing patterns from a very large number of possible alternatives in order to maximize network throughput as well as minimize network costs. Rather than search for optimal routing patterns in real-time, we perform this search off line using a centralized forecasting system and a centralized servicing system, which employ the DNHR design algorithms. The algorithm is given in pseudo-code. DNHR has led to fundamental changes in network operations, since DNHR requires the centralized co-ordination and increased automation of forecasting, servicing and network management.

3.1.3. Design and Control of Dynamic Routing Networks

The entire DNHR network is forested in the network forecasting system and serviced in the serving system. Within this support system the DNHR design algorithm is embedded, which determines the trunking and routing for the entire DNHR network for the network servicing design, the algorithm determines the routing that makes best use of the available network capacity in order to solve blocking problems. If routing changes alone cannot solve all blocking problems, the DNHR design algorithm determines cost effective larger number of trunk needed to correct the blocking problems. In order to achieve network design savings the following several techniques are used:

- Use time-varying routing so that trunks are shared to the greatest extent possible with the variation of loads in the network.
- Equalizes the loads on links throughout the busy periods on the network.
- Routes traffic on least costly paths.
- Select an optimum number of trunks on each link to optimally divide the traffic between the direct link and the alternate paths used by the traffic.

The step of the design algorithm consists of :

- Select a routing module.
- A dimension network module and an update optimal link blocking module.

Input parameters are:

- Link-cost
- Switch-to-switch offered loads
- Maximum switch-to-switch blocking level

The select routing module determines the optimal DNHR routes for design hour by executing the three steps:

- Select candidate paths
- Optimize path flows
- Define sequential routes

The select candidate paths steps finds the least-cost paths between switches in the network. This step considers paths. The optimize path flows step assigns flow (carried traffic) to the candidate paths to minimize network cost, and the third step of the select routing block finds the path sequence that best match the optimal path flows. The optimal routing is then provided to the dimension network module, which sizes each link in its peak hour and calculates blocking estimates in side hours. Network dimensioning is accomplished through use of a two-parameter traffic model. After completion of the dimension network step, the cost of the resulting network calculated. If the network cost is still decreasing, the update optimal link blocking module computes a new estimates of the optimal link blocking using the optimization method of Truitt [14].

3.2. Real-Time Dynamic Traffic Routing

Since DNHR implementation provided significant benefits for voice services, there was strong motivation to extend dynamic routing to all class of voice and data services on an integrated transport network. Motivators for this evolution were as follows:

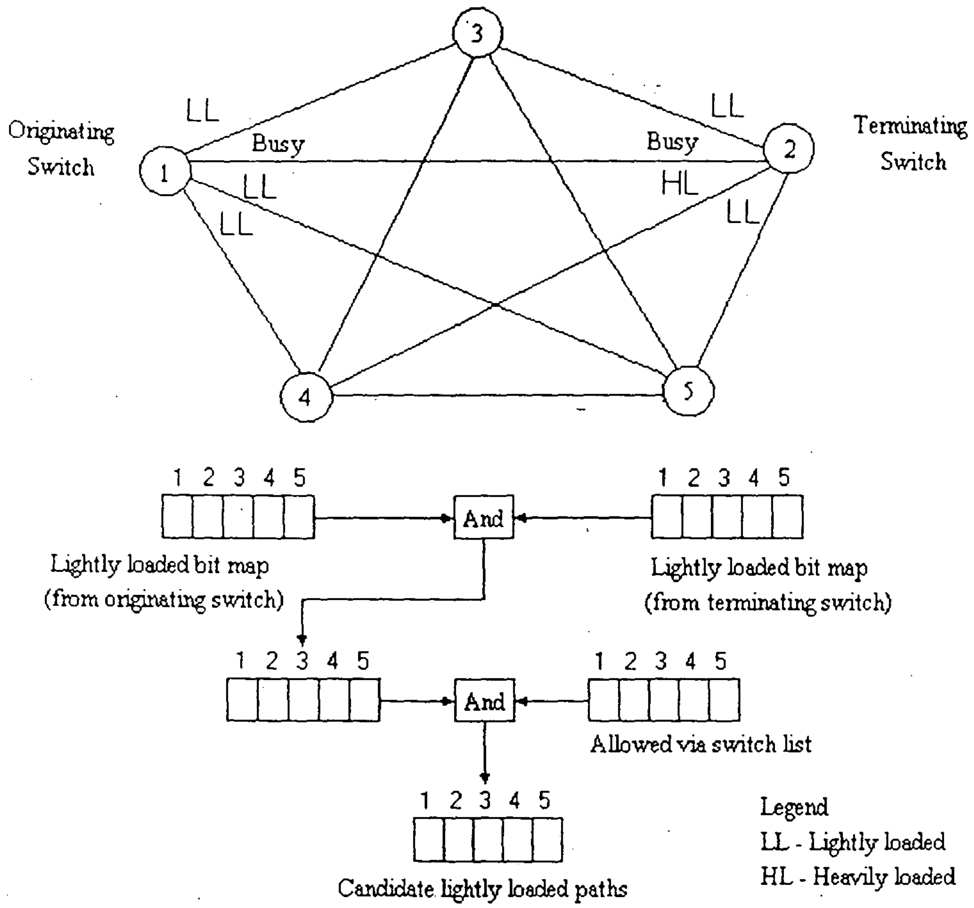
- Provide a standardized class-of-service routing structure for all current and future voice and data services
- Introduce new services using dynamic traffic routing as the routing strategy of choice.
- Provide new routing capabilities, such as priority (key service) routing for new enhanced services.
- Allow common network management system to support all classes-of-services.
- Allow an integrated shared transport network across all classes-of-services.

Example of voice and data services supported by RTNR include 64kb/s, 384kb/s, and 1536kb/s switched digital services, international long distance services, priority routing services, 800 services, virtual private services, and many others. RTNR class-of-service routing allows these individual voice and data services to utilize dynamic routing and share network bandwidth across class-of-services. This network provides connection for voice, data, and wideband services on a shared transport network. These connections are distinguished by resource requirements, traffic characteristics, and design performance objectives. Class-of-service routing allows independent control, by class-of-service, service-specific performance objectives, routing rules/constraints, and traffic data collection. Class-of-service routing allows the definition of virtual networks by class-of-service. Each virtual network is allotted a predetermined amount of total bandwidth. This bandwidth can be shared with other class-of-service, is protected under conditions of network overload or stress. RTNR provides the platform of the class of services real-time dynamic traffic routing strategy and also provides a multiple access/egress routing arrangement to ensure reliability and flexibility for international and access/egress networks.

3.2.1. Algorithm

```
n ← Number of nodes;
For each call
  If ((direct path exists) and (not busy)) Then
        path ← direct path
  Else
      For i ← 1 to n
      Begin
            If (path exists between the origin(o)
                         and destination(d)) Then
            If (o,i) ∈ LL(o) and (i,d) ∈ LL(d) Then
                (* Weight < Threshold *)
                Store the path
            Else          Path does not exists
      End;
   If (more than one path exists) Then
            path ← least loaded path;
           (* Least weight of the path *)
   Else
         If (only one path exists) Then
             path ← Stored path
         Else
              No suitable path exists;
            (* Call is blocked in that case*)
```

As illustrated in the figure 3.3 an available two-link path from originating switch OS_j goes through a via switch to which both links and have idle trunks, that is neither trunk is busy. An available two-link path is considered to be lightly loaded if the number of idle trunks on both links exceeds a threshold level. In order to determine all of the switches in the network which satisfy this criterion, the originating switch sends a message to TS_k over the CCS network, requesting TS_k to send the list of switches to which it has lightly loaded trunk groups. Upon receiving this list of switches from Ts_k, originating switch OS_j compares this list with its own



3
LL
Originating
Switch
1
Busy
LL
LL
Busy
LL
Terminating
Switch
2
HL
LL
4
5
1 2 3 4 5
And
1 2 3 4 5
Lightly loaded bit map
(from originating switch)
Lightly loaded bit map
(from terminating switch)
1 2 3 4 5
And
1 2 3 4 5
Allowed via switch list
1 2 3 4 5
Candidate lightly loaded paths
Legend
LL - Lightly loaded
HL - Heavily loaded

Figure 3.3

list of switches to which it has lightly loaded trunk groups. Any switch that appears in both list currently has lightly loaded trunk groups to both OS_j and TS_k and therefore can be used as the via switch for a two-link connection for this call.

Each switch in the RTNR network is assigned a unique network switch number (NSN); these NSNs are used as switch identifiers. In the example depicted in the figure 3.3, there are five switches in the network which are assigned NSNs, and the list of switches can therefore be represented by a bit map made in the bit map for each NSN having a lightly loaded trunk group to TS_k and OS_j, also maintains its own bit map listing each NSN having a lightly loaded trunk group to OS_j. Using bit maps makes it very easy and efficient for the originating switch to find all lightly loaded two-link paths. The originating switch simply takes AND operation with the bit map it receives from TS_k, which lists all of the lightly loaded trunk groups out of TS_k, with its own bit map to produce a new bit map which identifies all the via switches with lightly loaded trunk groups to both OS_j and TS_k. Bit maps also are a very compact way to store a list of switches. This is an important consideration since the list is sent in a CCS message. In the simplest case, only 16 bytes of data are needed for a network with 128 switches.

Since some of the available two-link paths may not provide good voice transmission quality, network managers can restrict path selection to the two-link paths which provides good transmission quality through use of another bit map, the allowed via switch list, which specifies the accepted via switches from OS_j to TS_k. Performing AND operation with this bit map with the bit map containing the via switches of all of the available two-link paths removes the via switches of paths with unacceptable transmission quality. When two or more available paths have the same load status, the originating switch randomly picks one of these paths to use for the call. To pick a path for this particular call to TS_k, the originating switch starts a circular search through the bit map list of paths with the same load status, beginning with the entry immediately following the via switch it last used for a call to TS_k.

3.3. State and Time-Dependent Routing

DNHR if effective for its time-dependent routing capability, where pre-specified routing patterns are changed as frequently as every hour to respond to forecasted traffic conditions. But now the telecommunications network is in traditional period, with a wider range of telecommunication services and use pattern as well as new common carriers, which result greater and more complex variations. So more powerful real-time traffic control capability such as state-dependent routing are needed [8]. The purpose of STR is to achieve high performance despite unprintable real-time traffic fluctuations, with minimum changes to existing switching software and operations systems. STR uses an appropriate combination of the two kinds of control, centralized and distributed [8]. The centralized methods use a routing calculation center that collects network-status information from switches and returns routing recommendations to the switches in a pre-determined update-cycle. Here a very short cycle, on the order of five to ten seconds is required for updating routing recommendations to achieve high performance. To obtain optimal routing pattern according to real-time conditions in a large-scale network, the centralized methods required very large computing power and very high data transfer speed.

In a distributed methods, routing patterns are determined autonomously in each switch based on the network status, which can be obtained by the switch itself in the call connection procedure. The distributed methods also carry out the decision on the basis of network status information distributed from other switches. These methods can be easily implemented on the existing switching software with some changes and additions. Generally more network status information may be available for centralized methods than for distributed methods. In the STR network, a set of alternate routes between each origin and destination node pair is determined on a time dependent basis by a control center using a centralized control method. For call level routing, state-dependent routing is achieved by using a distributed control method to determine an alternate route for each overflow call.

3.3.1. Overview of STR

The network is assumed to have a mesh topology in which each origin destination node pair is connected by a direct route. For each originating call, the first choice is the direct route. If the direct route is busy then the second choice is one of the alternate routes between the origin and destination node pair. If the alternate route is also busy then the call is blocked.

STR is the combination of two dynamic process [8]. The first is the routing pattern generation, in which an ordered set of possible alternate routes is determined for each origin destination pair. The routing pattern is determined according to the network capacity and traffic level for specific time periods, such as day or night. Alternate routes between an origin and destination node pair are restricted to link routes. The second control process is call-level routing, where an alternate route is determined for each call overflowing from the direct route.

3.3.2. Routing Pattern Generation

In the routing pattern generation process, the flow between each origin-destination node pair is assigned to two-link alternate routes in a way that maximizes the total amount of flow carried under the given constraints of link capacity and the number of alternate routes per O-D node pair. In general, the amount of CPU time and memory required to find a solution increases significantly as the number of network nodes increases. In this process of STR a heuristic algorithm employing an iterative procedure is used in order to rapidly find the near-optimal flow assignment. In addition, a simplified network model is used to minimize the complexity of the flow assignment problem in a realistic large-scale network. This process is effective in not only simplifying call level routing but also in improving performance.

3.3.3. Call-Level Routing Scheme

The requirement for implementation are to minimize additional processing power requirements, and to minimize changes in existing switching software and operations systems. To ensure the stability of network performance, a trunk reservation method gives priority to directly routed call connections.

3.3.4. Algorithm

The basic algorithm is as follows:

```
Done ← FALSE;
Find routing pattern;
Set one of alternate routes as currently assigned alternate route;
For each originating call
   Select direct route ;
    If  direct route is busy Then
        While ((not Done) and No more input))
         Begin
                Select currently assigned route;
                 If overflowing call is completed Then
                  Keep this route as the currently assigned route;
                  If any of the two - link is busy Then
                          The call is blocked;
                      Choose the next alternate route in the set of alternate
                       routes as currently assigned alternate route;
                   If path is found Then
                          Done ← TRUE;
         End
```

The only information obtained through the call-connection process in switches is used to select a near optimal alternate route from possible alternate routes. A set of possible alternate routes is set for each O-D node pair, and one of the alternate routes is specified as the currently assigned alternate route. A call-connection process is carried out according to the routing sequence, for example, the marked routing sequence b in (1) of Fig 3.4. The desirable alternate route set, that is, the routing pattern {4,5,6}, is set in the originating switch. For each originating call, the first choice is the direct route {1}. If this direct route is busy , the second choice is the currently assigned alternate route, for example, route {5}. If the overflowing call is completed, this alternate route {5} is kept as currently assigned alternate route and is used for the next overflowing call. If the first or second link on the currently assigned alternate route {5} is busy, the call is blocked and the next alternate route {6} in the set of possible alternate route becomes the currently assigned alternate route. Thus the routing sequence c in (1) of Fig. 3.4 is used for the next call. The originating switch is informed of blocking in the second link on the currently assigned alternate route by using the trunk-release signal.

Two advanced schemes for decreasing the incidence of call blocking while needing only minimal changes in exchange software and signaling systems are described below.

3.3.5. Single-Overflow STR Using Congestion Threshold (STR-S)

In the basic scheme, a new alternate route for each O-D node pair is assigned only after a call is blocked in the first or second link of the currently assigned alternate route. In the single-overflow advanced scheme, a new alternate route as assigned when the number of idle trunks in the first or second link of the currently assigned route is less than a pre-determined value (congestion threshold). This is done to prevent the next call from being lost. The conditions for changing the currently assigned alternate route are Thus

$$x_{i1} < K + T$$

or

$$x_{i2} < K + T$$

where x_{i1} is the number of idle trunks in the first link of the currently assigned alternate route i, x_{i2} is the number of idle trunks in the second link of the currently

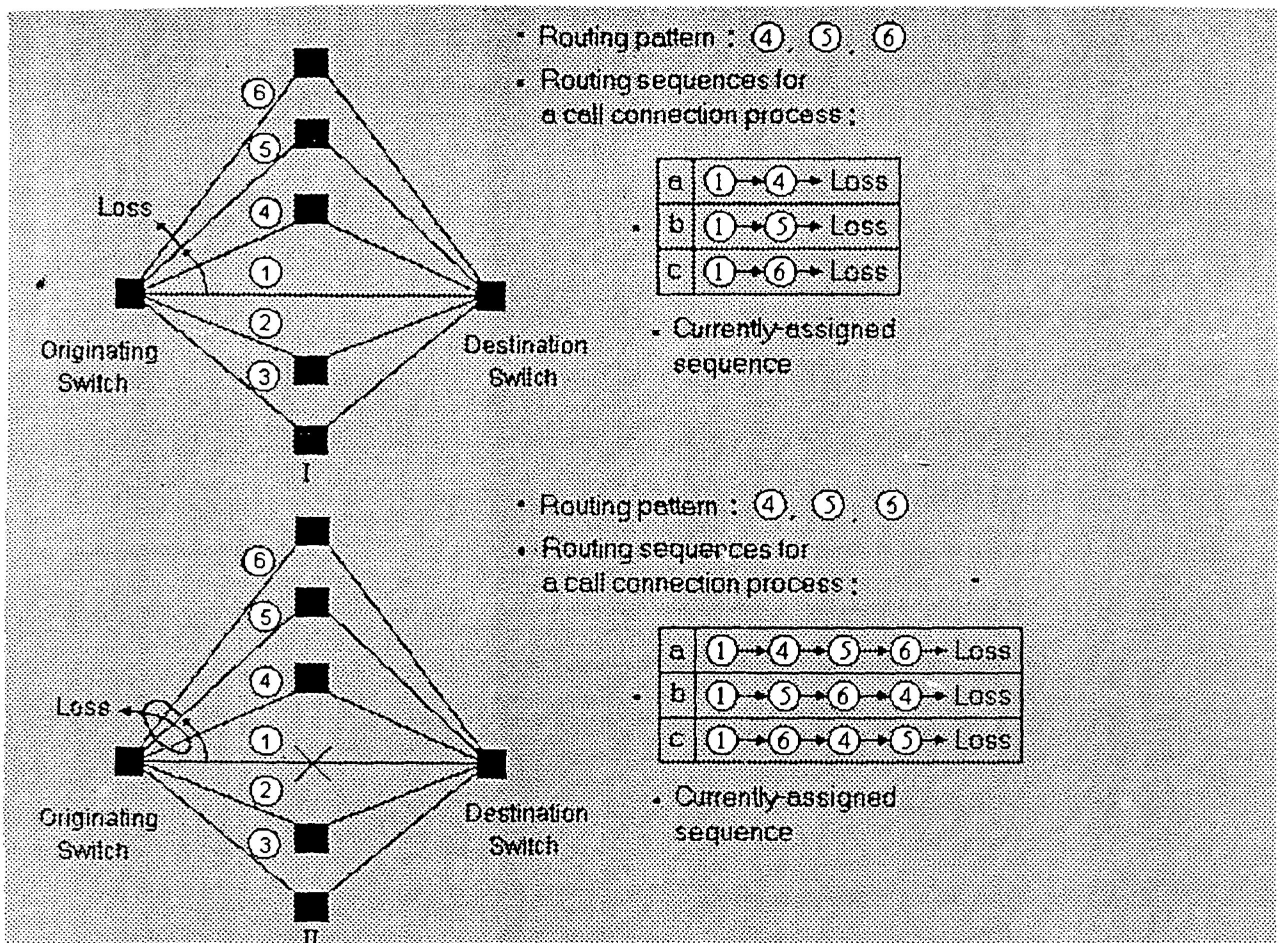

Routing pattern : ④, ⑤, ⑥
Routing sequences for a call connection process :
a ①→④→ Loss
b ①→⑤→ Loss
c ①→⑥→ Loss
Currently-assigned sequence
Loss
Originating Switch
Destination Switch
I
Routing pattern : ④, ⑤, ⑥
Routing sequences for a call connection process :
a ①→④→⑤→⑥→ Loss
b ①→⑤→⑥→④→ Loss
c ①→⑥→④→⑤→ Loss
Currently-assigned sequence
Loss
Originating Switch
Destination Switch
II

Figure 3.4

assigned alternate route i. K is the number of reserved trunks for directly routed calls, and T is the congestion threshold for changing the alternate route assignment.

This single-overflow types STR using a congestion threshold is expressed as STR-S(N,T), where N is the number of possible alternate routes and T is the congestion threshold. Except for the condition of changing the currently assigned alternate route, STR-S(N,T) is the same as the basic scheme. In other words, with the congestion threshold zero, that is STR-S(N,0), it is the same as the basic scheme.

3.3.6. Multiple-Overflow STR(STR-M)

The second advanced scheme, STR-M(N,T), is the same as STR-S(N,T), except that it allows sequential multiple overflow. In STR-S(N,T), a call overflowing from the direct route is blocked if the first link of the currently assigned alternate route is busy, where busy means that the number of idle trunks reserved for directly routed calls. In STR-M(N,T), on the other hand a different alternate route whose first link is not busy, is sought according to the currently assigned routing sequence.

CHAPTER FOUR

4. Simulation and Performance Evaluation

The program for simulation is written in C++ using DEC-Alpha platform running OSF/1 operating system.

4.1. Dynamic Nonhierarchical Routing (DNHR)

In dynamic Nonhierarchical network, each node represents the switch through which a call is completed to its destination. The design algorithm of DNHR presented in section 2.1.2 for determining routing sequences for different load conditions at different time periods is simulated in this section. Since the calls at a switch vary time-to-time, so routing sequences also vary. The paths used for call completion at various time periods need not be the same.

The network structure used is shown in the figure 3.2. Here each link represents trunk group and the nodes represent the switches in the network, assuming, each link has a predetermined capacity. The program uses the class defined as below:

```
class DNHR {
        private:
                int capacity[MAXNODE][MAXNODE];
                // MAXNODE is the maximum nodes in the network
                int flow[MAXNODE][MAXNODE];
                int *tflow;
                int s,t;
```

```
public:
        void const_graph();
        void read_capacity();
        void genrate_routing_sequence();
        int any(int endpath[]);
        void find_maxflow(int cap[][],int s,int t,
        int flow[][],int *ptotflow);
};
```

The input parameters are:

- capacity of all the links in the network
- s and t

Capacity represents a capacity function defined on a weighted graph, s and t represent the origin and destination node. Output parameters are:

- flow
- tflow

Flow represents the maximum the flow function and tflow is the amount of flow from origin to destination under the flow function flow.

Member function const_graph() initializes the graph. Read_capacity() function reads the capacities of the various links of the network and stores them in the matrix capacity[MAXNODE][MAXNODE]. Generate_routing_sequence() calls all the member functions to perform the required task, and generates the routing sequences. The find_maxflow() is the main function, which finds the flows assignments of the links and total flow of the network. Find_maxflow() function uses Ford_fulkerson algorithm [7]. After getting the flow assignment, and maximum flow of the network, it is very easy to generate the routing sequences. Routing sequences results in a different allocation of link flows. The first choice paths carry the maximum flow of a switch-to-switch load.

4.1.1. Input/Output

Let the ordered pair (a,b) denote the link from node a to node b. We assign capacities to all the links in the network in figure 3.2. All the links in the network are numbered for the ease of computation in the following manner:

1 : (1,2)

2 : (1,3)

3 : (1,4)

4 : (1,5)

5 : (1,6)

6 : (1,7)

7 : (2,7)

8 : (3,7)

9 : (4,7)

10: (5,7)

11: (6,7)

Table 4.1: DNHR-runs

Link	Capacities Assigned	s,d	Total Flow	Path Sequences
(1,2,3,4,5,6,7,8,9, 10,11)	(3,4,3,5,2,5,8,2,4,7,2)	1,7	20	(,1,5,2,4,3,6)
(1,2,3,4,5,6,7,8,9, 10,11)	(8,5,6,3,6,7,5,2,6,110,4,)	1,7	27	(1,4,26,5,3)
(1,2,3,4,5,6,7,8,9, 10,11)	(5,8,4,0,2,3,5,9,4,3,7)	1,7	20	(2,4,1,3,6,5)

The set of inputs for various runs of the simulation program of DNHR are as in the table 4.1. The corresponding output path sequences are also indicated in the table.

4.2. *Real-Time Dynamic Traffic Routing (RTNR)*

The algorithm for simulation of RTNR is given in section 3.2.1. It uses a complete graph, so when we look it into graph theory, the problem becomes simply the path finding problem between the origin and destination nodes. Here each node represents a switch in real-time network, and link represents trunk group.

The program written for simulation uses the algorithm given in section 3.2.1. Each link of the network is leveled with weights, which represent traffic loads in that link. If the load exceeds some preassigned value then it is assumed to be busy. Similarly a threshold number is assigned, for each link, when the load is less than that threshold number then it is assumed to be lightly loaded, otherwise it is heavily loaded. A path is said to be found between the origin and destination, if both link of any path is lightly loaded. In other words a call can be set up through that path.

The program uses the class defined as:

```
class RTNR {
        private:
                int node[MAXNODE][MAXNODE];
                // MAXNODE is the maximum number of nodes in the graph
                int Wt[MAXNODE][MAXNODE];
        public:
                void const_graph();
                void assign_load();
                check_2link_path(int,int);
                void find_via_switches(int via[],int,int);
                void find_path();
                void find_allowd_via(int via[]);
                void find_o_LLlist(int o_LLlist[],int);
```

```
        void find_t_LLlist(int t_LLlist[],int);
        void fin_AND(int o_LLlist[],int t_LLlist[]);
        void find_least_loaded_path(int path[],int,int);
        void print_path(int paths[],int,int);
};
```

The graph is represented by adjacency matrix, node[MAXNODE][MAXNODE]. Wt[MAXNODE][MAXNODE] is used for storing the link load that is weights. Member function const_graph() initializes the graph, check_2link_path() function checks whether two-link path between the origin and destination exits, if exist returns true otherwise false. Find_via_switches() function finds all the via nodes which are lightly loaded, and stores in a array. Find_path() function finds all the possible paths between the origin and destination node. Find_allowed_via() function uses the array of via nodes, and computes the possible via nodes. Find_o_LLlist() computes the bitmap of nodes connected to origin node, which are lightly loaded from the origin node. Similarly, find_t_LLlist() function computes the bit map of nodes connected to destination node, which are lightly loaded from the destination node. Find_least_loaded_path() function computes only one path which is least loaded, and finally print_path() function simply prints the computed path.

The overall idea of the program is that it takes a undirected graph representing a network of any number of nodes and stores the existing links in a matrix. If the link exists between any pair then the value of that link in the matrix is set to be 1 or 0 otherwise. The program prompts for weights for those existing links and stores it in another matrix. The weights for each existing link can be read from a file also.

After completion of the above operations the task to find the path between the origin and destination is performed. For various runs the origin and destination node could be varied and be specified in the input. In order to find the path the program carries out the following sequences of tests and operations. Initially a direct path is searched, if it exists then the busyness of the direct link is checked by comparing the weight assigned to that link and the predetermined value of busyness. If the path is not busy then it is marked selected and program terminates by printing the direct link that is the actual path for call completion.

If direct path does not exist then it checks all the existing two-link paths between the originating and terminating nodes. Then computes the lightly loaded node list. The procedure for finding the lightly loaded node list is as follows:

Each node is assigned some unique number i.e. Network Switch Number (NSN). Switch list is represented by bit map, where each bit position represents the node. Number of bits in the bitmap depends upon the number of nodes in the network. Entry "1" is made in the bit map for each node having a lightly loaded link to the terminating node. Originating node also maintains its own bit map listing each NSA having a lightly loaded link to originating node. Weight assigned to each link is compared to the threshold number and thus lightly loaded node list from originating to terminating node is computed. After having done this it performs the AND operation between the bit maps from the originating and terminating nodes. The gives separate bit pattern. The bit map from the allowed via node is also computed. After this again AND operation is performed with the bit map generated from the originating node and terminating node and the allowed via node list. This gives the final bit map. 1s in this bit map represents the via nodes (corresponding to the 1's bit position) through which path to the terminating node is computed.

It may possible that more than one via node exist through which path goes. Program then finds the least loaded path. That is the path having least weight. That is the final path through which a call to the terminating switch can be completed. If all the bits in the candidate via node are 0 then no lightly loaded path exists.

4.2.1. Input/output

Let the ordered pair (a,b) denote the link from node a to node b. We assign weights to all the links in the network in figure 3.3. All the links in the network are numbered for the ease of computation in the following manner:

1 : (1,2)
2 : (1,3)
3 : (1,4)
4 : (1,5)
5 : (1,6)
6 : (1,7)
7 : (2,7)
8 : (3,7)
9 : (4,7)
10: (5,7)
11: (6,7)
.
.
.

BUSY = 7

THESHOLD = 5

Table 4.2: RTNR-runs

Links	Weights Assigned	Origin and Destination Nodes	Paths
(1,2,3,4,5,6,7,8,9,10)	(7,4,2,4,3,8,4,5,4,6)	(1,2)	(1,3,2)
(1,2,3,4,5,6,7,8,9,10)	(7,4,3,4,5,8,3,6,6,5)	(1,2)	(1,4,2)
(1,2,3,4,5,6,7,8,9,10,11, 12,13,14,15,16,17,18)	(7,4,5,4,8,3,6,8,2,5,7,7,2,6,2,4 ,6,3)	(1,3)	No idle path exists

The set of inputs for various runs of the simulation program of RTNR are as in the table 4.2. The corresponding output path are also indicated in the table.

4.3. State and Time-Dependent Routing

The basic concept of routing pattern generation is same as the routing pattern generation of Dynamic Nonhierarchical Routing. After generating the routing pattern, call-level routing scheme is applied. The simulation of this algorithm in figure 3.4 is under development stage.

CHAPTER FIVE

5. Conclusion and Future Work

Dynamic traffic routing implements an integrated class of service routing features for extending dynamic routing to emerging services and provides a self healing network capacity to ensure a network-wide path selection and immediate adaptation to failure. Dynamic traffic routing brings benefits to customers in terms of new service flexibility and improved service quality and reliability, at reduced cost. The various dynamic routing strategies have lead to increased efficiency of the networks in spite of increase in traffic load. A detail survey of the Dynamic Nonhierarchical Routing (DNHR) Real-Time Dynamic Traffic Routing (RTNR) and State and Time-Dependent Routing (STR) and their corresponding simulation highlight their respective features. RTNR is distributed and real-time state-dependent. This feature has been successfully simulated as for given input the program computes showing how it can handle the traffic at different situations of load conditions. The time-varying nature of DNHR is exhibited from the computation of the paths output for various capacities of the links depicting the traffic load at various time.

Problems in the simulation occur if the number of nodes in the network is large and connectivity of the network changes. Modifications for better performance of the algorithms with change in nodes and change in connectivity can be taken as future work.

References

[1]. Ash Gerald R., "Dynamic Network Evolution, With Examples from AT&T Evolving Dynamic Network", *IEEE Communication Magazine* Vol. 33, No. 7, July 1995.

[2]. Ash Gerald R. "Design Control of Networks with Dynamic Nonhierarchical Routing," *IEEE Communication Magazine*, Vol. 28, No. 10, October 1990.

[3]. Bahk Saewoog, Zarki El Magda, "Dynamic Multipath Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide-Area Networks", *ACM Computer Communication Review SIGCOMM*, 1992.

[4]. Bertrekas Dimitri, Gallager Robert, **Data Networks**, Englewood Cliffs, NJ: Prentice-Hall, 1987.

[5]. Caron Alain, Choquette Jean, Bedard Frncois, Regnier Jean, "Dynamically Controlled Routing in Networks with Non-DCR-Compliant Switches", *IEEE Communication Magazine*, Vol. 33, No. 7 July 1995.

[6]. Chemouil Prosper, Filipiak Janusz, Gauthier Paul, "Performance Issues in the Design of Dynamically Controlled Circuit-Switched Networks", *IEEE Communication Magazine*, Vol. 28, No. 10, October 1990.

[7]. Ford L.R, Fulkerson D.R, **Flows in Networks**, Princeton University, Princeton, N.J 1972.

[8]. Kawashima Konosuke, Inoue Akiya, "State-and-Time-Dependent Routing in the NTT Network", *IEEE Communication Magazine*, Vol. 33, No. 7, July 1995.

[9]. Key Peter B., Cope Graham A., "Distributed Dynamic Routing Scheme", *IEEE Communication Magazine*, Vol, 28,No. 10, October 1990.

[10]. Krishnan K. R., "Markov Decision Algorithms for Dynamic Routing", *IEEE Communication Magazine*, Vol. 28, No. 10, October 1990.

[11]. Regnier Jean, Cameron W. Hugh, "State-Dependent Dynamic Traffic Management for Telephone Networks", *IEEE Communication Magazine*, Vol. 28, No. 10, October 1990.

[12]. Tanenbaum A. S., **Computer Networks**, Englewood Cliffs, NJ: Prentice-Hall, 1985.

[13] Tropper Carl, Glazer W. David, "A new Metric for Dynamic Routing Algorithms", *IEEE Transaction on Communication*, Vol. 38, No. 3, March 1990.

[14]. Truitt C. J. "Traffic Engineering Techniques for Determing Trunk-Requirements in Alternate Routed Networks", *BSTJ*, Vol. 31, No. 2, March 1954.

[15]. Wang Zheng, Crowcroft Jon, "Analysis of Shortest Path-Routing Algorithms in a Dynamic Networks Environment", *Computer Communications Review, SIGCOMM*, Vol. 22, No. 2, April 1992.

[16]. Watanable Yu, Toshikane Oda, "Dynamic Routing Schemes for International Networks", *IEEE Communication Magazine*, Vol. 28, No. 10, October 1990.

[17]. Watanable Yu, Kashper N. Arik, "Dynamic Routing in the Multiple Carrier International Network", *IEEE Communication Magazine*, Vol. 33, No. 7, July 1995.