DESIGN AND IMPLEMENTATION OF A FAULT-TOLERANT MULTIMEDIA NETWORK AND A LOCAL MAP BASED (LMB) SELF-HEALING SCHEME FOR ARBITRARY TOPOLOGY NETWORKS

BY

ARION KO KIN WA

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF INFORMATION ENGINEERING

THE CHINESE UNIVERSITY OF HONG KONG

JANUARY 1997

Acknowledgement

In the past two years of my M.Phil. study, working till the mid-night became a daily routine for myself. During this hard period, my colleagues and friends have given me invaluable support and encouragement. Firstly, I would like to express my warmest gratitude to my advisor, Prof. Kwok-Wai Cheung, his guidance has provided me all-round training both theoretically and experimentally. During the course of study, I have learnt a lot from him about the design, analysis as well as implementation of a fault-tolerant multimedia optical network. I do believe that these precious knowledge will definitely be beneficial to my future career.

Furthermore, I would like to express my special thanks to Mr. Calvin C.K. Chan for his help, advice and valuable comments on my projects. Also, I am very grateful for all of his encouragement throughout the course.

Moreover, I would like to thank my project partner Mr. Patrick S.F. Lam for his advice and care in the past two year, Mr. Wilson Y.H. Wang for his help in consulting my simulation programs, Mr. Ringo W.K. Lam, Mr. Gordon C.T. Yeung and the rest of CUMLAUDE working group for their contributions to the CUMLAUDE NET.

Finally, I would like to express my regards to my colleagues, Mr. Alex

W.H. Siu, Miss Georgia Y.Gao, Mr. Charles C. Su, Miss Zoe C.Y. Tsang, Mr. Simon S.P. Lau, Mr. Chris L.C. Wong, Miss Samantha W.S. Chan, Mr. Ronald C.C. Hui, Mr. Simon K.Y. Hung, Mr. Freeman S.L. Hui for their helpful discussion and encouragement.

Abstract

The recent wide spread deployment of optical fiber transmission systems have aroused the concern on the survivability issues of networks. As any service disruption in these high-capacity fiber transmission systems could cause huge amount of data and revenue losses to users as well as to service providers. Such losses can be minimized by having better network survivability planning. Traditionally, this planning is divided into four phases, (1) failure prevention, (2) fault detection and alarming, (3) survivable network architecture and restoration algorithm designs, and (4) manual restoration planning. In this thesis we mainly focus on the issues of the last three phases.

We first provide the necessary background information on survivable network planning and restoration algorithm design. Then we describe our fault-tolerant multimedia network prototype - CUM LAUDE NET. Various issues concerning the network architecture, hardware design, traffic control, medium access control protocol, network fault-tolerance as well as the services supported are addressed.

Next, we propose two distributed fault-tolerant (FT) and auto-healing (AH) algorithms for dual-ring networks. These two algorithms are the generalized version of the revertive fault-tolerant scheme used in the CUM LAUDE NET. They are based on the inter-communications and hand-shaking processes among adjacent nodes to exchange network information in case of network failure. Some of their features are: (1) fast network restoration, (2) short restoration message(2-bytes) to alleviated the network loading during restoration, (3) high network availability to users during restoration and (4) hot replacement of faulty network components. The validity of the algorithms was tested on our high-speed multimedia network prototype described previously. Failure recovery time on the order of milli-second was achieved in our laboratory testing. Measures including communicative probability, survivability and average reachability are used to quantify the network reliability under the algorithms.

To facilitate real-time monitoring and control of the CUM LAUDE NET, a network management software called NETMAN is developed . It provides a centralized comprehensive network management solution for gateways and networks on a single platform with a colorful GUI (graphical user interface) based management capability. A variety of detailed views and window displays is designed for real-time monitoring of the network status.

Then we extend our network protection scope from dual-rings to arbitrary topology networks. A Local Map Based (LMB) Self-Healing Scheme, which eliminates the topological constraint imposed on the distributed fault-tolerant algorithms is proposed. It allows fast and efficient network restoration by making use of the information available on a small-scale local map. The most time-consuming alternate-path seeking process in other distributed restoration schemes are replaced by simple searching and sorting algorithm executed in the DCS (Digital Cross-connect Systems) nodes. This leads to fast network restoration, high spare resource utilization and suppressed message volume. Also, the performance of the algorithm is independent of the network size. Simulation has shown that the LMB self-healing scheme exhibits significant performance improvement over existing network restoration algorithms.

Finally, we summarize our work and give an outlook in the field of network survivability planning and possible future research directions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

The recent wide spread deployment of high-speed communication networks has significant impacts on the development of civilization, together with the use and integration of computers has brought us into the "information age" [13]. Nowadays, network communication has become a vital part of everybody's life. Service disruption is no longer tolerable by individuals, industry and business sectors due to the increased necessity of communications among bankers, purchasing managers, stock brokers, students, researchers, and so forth. In the mean time, the consequences of service disruption caused by network failures are becoming more severe. This is primarily due to the high volume of traffic being carried by the fiber systems and the arising necessity of uninterrupted communication. For example, if the network is disabled for one hour, loss of revenue more than millions of US dollars can occur in the trading and investment banking industries [41]. This still does not include the intangible losses such as legal costs, adverse customer relations, loss of competitive advantage and credibility. Therefore, rapid restoration from failures is becoming a critical challenge for network operations and management.

To ensure service continuity, network service providers have paid more efforts to alleviate the adverse effects caused by network failures. However, providing protection against fiber network failures could be very expensive due to the high costs associated with fiber transmission equipment. Therefore it is highly desirable to have a cost-effective network management strategy offering an acceptable level of survivability. Different restoration technologies [40] [43], strategies [35] and survivable architectures [22] [38] have been developed to overcome this challenge. The issues described above are often referred as network or service survivability planning.

1.2 Service Survivability Planning

In service survivability planning, both technological considerations and regulatory realities have to be taken into account. There are four common phases in survivability planning to ensure service continuity and minimize the level of impact caused by service disruption. As shown in Figure 1.1, these phases are (1) prevention, (2) failure detection and alarming, (3) network self-healing through robust network design, and (4) manual restoration. The first phase focuses on the prevention of network failures caused by people and the environment (e.g., fire and earthquake). The second phase focuses on the techniques used in network failure detection and alarming. It is often considered together with network administration and management. The third phase concerns about the problem

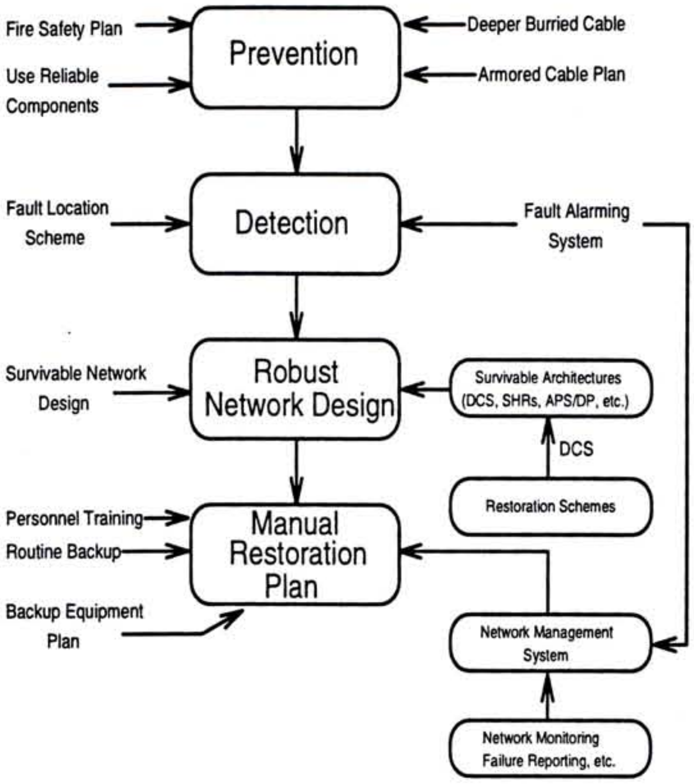


Figure 1.1: Network Service Survivability Planning.

of maximizing the self-healing capability of networks during failures. The last phase emphasizes the efficient utilization of available work forces, facilities and equipment to restore the network when the failures cannot be fixed by the automatic restoration plan. In this thesis, we will focus mainly on the issues related to the last three phases.

1.3 Categories of Outages

Communication networks are vulnerable to many threats: natural disasters [12], intentional sabotages, hardware or software failures. Depending on the failure scale, duration, the priority weighting of lost traffic and the level of restoration, the following categories can be defined for both natural and human caused failures.

- **Catastrophic**: This type of failure affects a large number of network users with a relative long duration(days to weeks). Usually, only a small part of service can be restored automatically and most of the service need to be restored manually. Some typical examples are multi-CO (Central Office) failure caused by earthquake, flood, hurricane, tornado, global hardware/software defect, act of war, etc.

- Major: The failure scale and duration of a major network failure are less than a catastrophic network failure. A large percentage of service may be restored automatically while the rest would be restored manually. Usually, it takes less than a couple of days for complete restoration of failures belong to this type. Some obvious examples are single-CO failures, software defects, fiber cuts, etc.

- Minor: Service disruption caused by equipment or component failures, which affects a relative small population of the network users are included in this category. It may be caused by a single fiber cut or a line equipment failure. In this type of failures, service restoration can be fully automatic and virtually transparent to the users.

1.4 Goals of Restoration

The ultimate network reliability goal is to make all failures imperceptible to users. An interim goal is to reduce the impact of a failure so that calls will not be dropped (in Telecom networks) and data sessions are not prematurely terminated (in computer networks). Data transmission protocols can maintain sessions for seconds after an interruption in transmission. Declarations of the carrier group alarm that two seconds after a failure can cause a voice call to be disconnected by network switches [4]. Also, literatures [6] [20] [41] have pointed out that most existing services will not be adversely affected when the service outage lasts for less than 2 seconds. Accordingly it may be necessary to restore the circuits within the 2-second objective to achieve failure transparency to the customers.

1.5 Technology Impacts on Network Survivability

Technological advances play a crucial role in implementing cost-effective survivable fiber networks. Among these technologies, Digital Cross-connect System(DCS) and active/passive optical technology have been shown to reduce the costs of survivable fiber networks. The former allows rapid reconfiguration of networks to adapt failures and efficient use of spare capacities. Many DCS-based restoration algorithms have been proposed [5] [7] [8] [16] [21] [42]. They all lead to drastic improvement in both restoration time and spare facility utilization compared to the conventional protection technique based on dual-homing and diverse routing. The latter includes optical switches, power splitters, wavelength division multiplexers, and optical amplifiers. Optical switching and optical amplifier technologies have been used to implement a cost-effective point-to-point system with 1:1 diverse protection against potential fiber cable cuts and a cost-effective SONET self-healing ring architecture [43]. Power splitters have been suggested for use in a cost-effective, optical, dual-homing protection architecture that may prevent major office failures [37]. Wavelength division multiplexers and optical amplifiers have been used to implement optical add-drop multiplexers for a self-healing ring application to accommodate network growth demands in metropolitan intraLATA networks [39].

1.6 Performance Models and Measures in Quantifying Network Survivability

There are two basic approaches to perform survivability analysis. The first approach employs a probabilistic view to define network survivability. It uses probability of network failures and, possibly, rates of repair/restoration to calculate various network survivability measures such as availability or unservability (e.g., the expected amount of time during which a network is unavailable). The second one is a conditional or combinatorial approach. By studying the possible combinations of different failure scenarios, network measures are defined and analyzed by some deterministic methods assuming that the failure is occurred. Both approaches can be used to evaluate different restoration, repair or preventive methods depending on which types of comparison characteristics are the most critical.

1.7 Organization of Thesis

The thesis is divided into five chapters, Chapter 2 covers various design and implementation issues addressed in building a survivable dual-ring multimedia network prototype called CUM LAUDE NET [27]. They include the network architecture, the protocol layering, the network restoration scheme, etc. . Also included in this chapter is the performance evaluation of the restoration scheme and some prelimenary measurements on the prototype.

Chapter 3 describes a follow-up work based on the achievements presented in chapter 2. A network management software called NETMAN is designed to demonstrate the fault-tolerant and auto-healing ability of the survivable network prototype described previously. Besides, it is also capable of performing a variety of network management tasks.

In Chapter 4, we propose a local map based (LMB) self-healing scheme for DCS-based fiber networks with arbitrary topologies. The scheme is devised based on the distributed algorithm design experience achieved in chapter 2. It aims to provide fast and efficient restoration from different types of failures. Simulation based on different sets of parameters are performed to test the validity of the scheme. Finally, the thesis is concluded in chapter 5.

Chapter 2

Design and Implementation of A Survivable High-Speed Multimedia Network

2.1 An Overview of CUM LAUDE NET

The CUM LAUDE NET is a large group effort at the Chinese University of Hong Kong, supervised by Prof. K.W.Cheung, to prototype a multi-gigabit/sec multimedia integrated network. It is designed to support high-speed, real-time multimedia services with maximum compatibility to IP-based networks. The motivation is that Internet is a worldwide network service, has a very broad user base, and yet, Internet does not support real-time multimedia service. Thus, the design could provide an easy upgrade for IP-based networks to the future multimedia networks. The whole project consists of two phases. The phase I objective is to demonstrate a practical, low-cost and fault-tolerant integrated network that can provide real-time video and voice conferencing. Phase II objective is to construct a Gb/s fault-tolerant dual-ring backbone for the provisioning of real-time multimedia services for a metropolitan area. Phase I construction of CUM LAUDE NET is completed in 1996. Supporting software included a fully operational video and voice conferencing utility, a gateway to the public switched telephone network, voice mail services as well as other Internet services. Also, a network management software called NETMAN is designed for CUM LAUDE NET to facilitate real-time network monitoring. The next phase construction has started and is scheduled to be completed soon.

In the CUM LAUDE NET project, I had played the roles of a network developer as well as a system integrator. As a developer, I designed, implemented and tested the Fault-tolerant and Auto-healing algorithms, as well as the algorithms for buffer transmission priority assignment, congestion control and packet buffering. Besides, I had written the control software for the router-nodes and built the network manager - NETMAN for the CUM LAUDE NET. As a system integrator, I integrated all the efforts contributed from former colleaques, and prototyped a fully operational 100 Mb/s survivable dual-ring multimedia network testbed.

2.2 The Network Architecture

2.2.1 Architectural Overview

The prototype that is being constructed consists of two hierarchies as shown in Figure 2.1. The level-2 hierarchy is a Gb/s backbone connected in a fault-tolerant dual-ring. Each router-node on the backbone will be able to route one

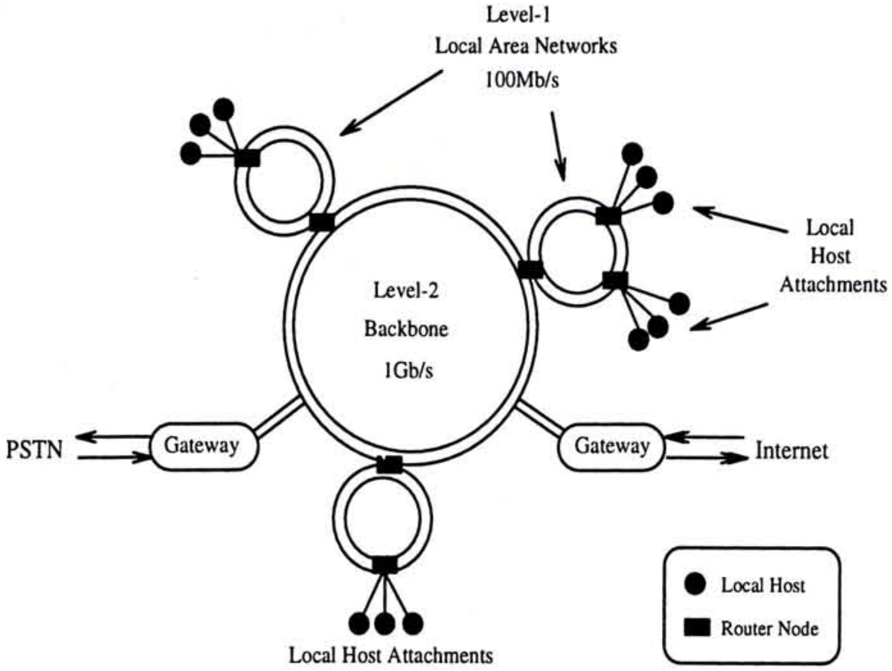


Figure 2.1: CUM LAUDE NET network architecture

gigabit/sec bandwidth at each input and output port. Under uniform traffic, the final backbone dual-ring network will have an aggregate capacity of approximately 8 gigabit/sec.

Each level-1 hierarchy is a 100-Mb/s dual ring network, aims at providing the same services for a more local area environment. Different hierarchies are connected by bridges/routers, whose function is to route packets from one hierarchy to another.

Each node in the level-1 hierarchy is either a local host(user terminal) or a hub(which serves as a concentrator/distributor to a number of local hosts). Each local host is equipped with a network interface unit(NIU) to process packets addressed between the network and the host. Due to the uniform design of different hierarchies, the hardware and software design of the bridges/routers are much simplified.

The CUM LAUDE NET prototype is designed around a fault-tolerant dual-ring topology because the ring topology has several unique advantages over a centralized switching hub topology [25] [26]. The linear topology allows reserved service guarantee and fair sharing of bandwidth among all nodes. The distributive, sequential arrangement of the nodes also facilitates real-time protocol implementation. Distributive packet routing simplifies packet processing and introduces little packet delay. Another advantage of the linear topology is that it reduces the problems due to network congestion and complexities in control and management.

Even though many bus/ring network protocols have been proposed in the past [23] [32] [36], they are not suitable for high-speed multimedia integrated networking either because of the limited throughput, inability to guarantee real-time services, confinement to local area service, or heavy overhead for supporting multimedia services.

2.2.2 Router-Node Design

The router-node (or simply called node) used in CUM LAUDE NET are specially designed to facilitate fault-tolerance. As shown in Figure 2.2, each router-node consists of three routers: Ring-A router, Ring-B router and Local router. Every router is equipped with a DSP (Digital Signal Processor) and 2 FPGA (Field Programmable Gate Array) ICs to control the on-board hardware, IP address resolution, I/O ports, buffers and the communications between different routers. The I/O FIFOs of the routers in a router-node are interconnected by data bus, thus packets can be easily routed among different routers. This design enables dynamic constructions of logical paths inside the router-node. The FIFOs also

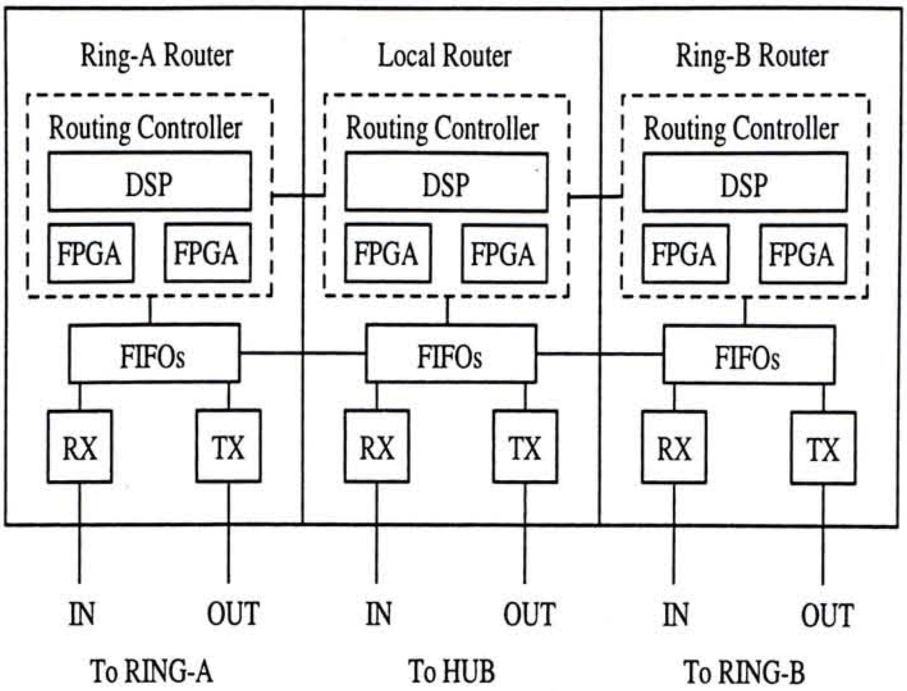


Figure 2.2: Block Diagram of the CUM LAUDE NET Router Node

act as buffers for the incoming packets and as a temporary storage for packets during restoration. Their size is being chosen to be 8K bytes each because this size is found to be suitable experimentally.

The communications between the routers in a router-node are handled by serial communication ports installed between each pair of routers. The chipset used for the transceiver interface is the new high-speed TAXI Am7968/69-125DC [2] from the Advanced Micro Devices (AMD). They have an maximum operation speed of 125 Mbaud on a serial link and use the robust 4B/5B coding scheme to detect transmission errors.

2.2.3 Buffer Allocation

There are four packet buffers installed in each router in a router-node, one is for receiving and the other three are for transmitting. These packet buffers

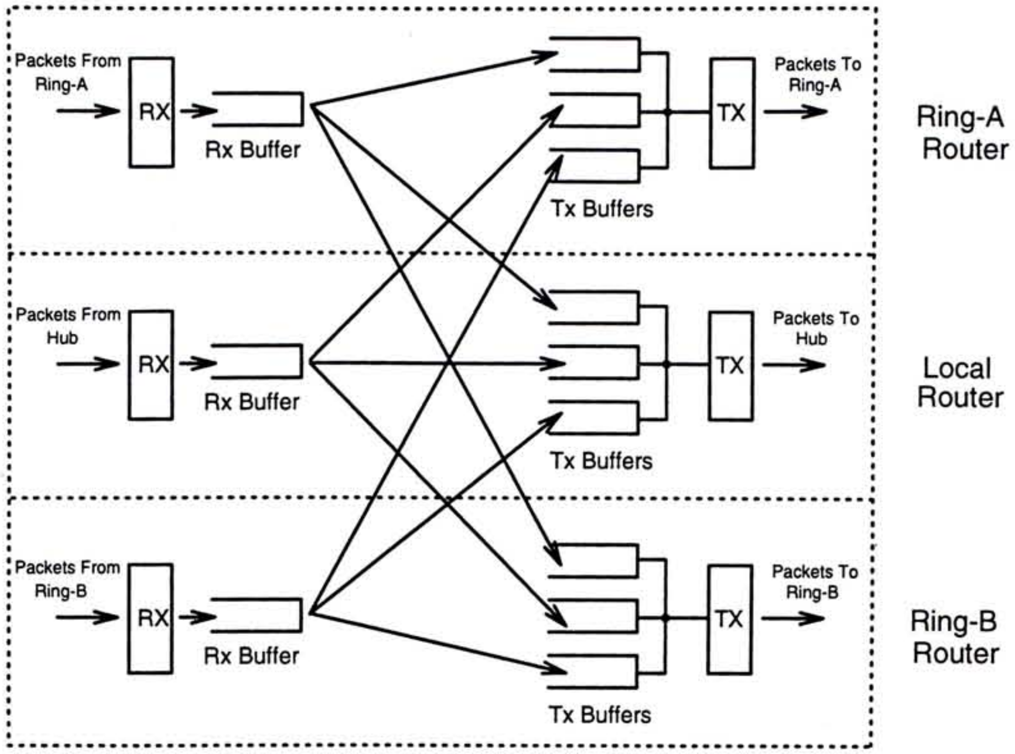


Figure 2.3: Buffer allocation in a CUM LAUDE router-node.

are actually FIFOs with a size of 8 Kbytes. Figure 2.3 shows the allocation of these transmission buffers in a router-node used in CUM LAUDE NET. The receive buffer is used to store packets coming from the rings or the hubs. After examined the headers of these packets, the router will know where these packets should be routed. Suppose a packet coming from Ring-A is being stored in the receive buffer, it will be routed to different transmission buffers under different situations: Ring-A (normal case), Local (if the packet is destined to a local host connected to the router-node) or Ring-B (in case the outgoing link of Ring-A has failed). The situations are similar for Local and Ring-B routers.

Buffer Fullness Flag	meaning
Full Flag(FF)	The capacity of FIFO is used up
Half Full Flag(HF)	Half of the capcity of FIFO is used
Empty Flag(EF)	The FIFO is empty

Table 2.1: Fullness Flags of the transmission buffers.

2.2.4 Buffer Transmission Priority

As depicted in Figure 2.3, all three transmit buffers of a router share a common transmission link. Therefore some kind of medium access control mechanism is required to ensure no contention of resources. In CUM LAUDE NET, the transmission priorities of the transmit buffers are controlled under a priority transmission policy. The policy utilizes the fullness condition of each transmit buffer as an indicator to assign a suitable priority to it. The fullness condition of a transmit buffer is represented by three flags - *Empty*, *Half-Full* and *Full*, as shown in Table 2.1. These flags are memory mapped to the controlling DSP's memory space. Thus they can be easily accessed and monitored.

The highest transmission priority is assigned to a buffer with the *Full Flag* on, followed by *Half-Full* and *Non-empty*. A buffer with its empty flag on will be ignored. For buffers having the same priority, transmission access will be granted in a round-robin fashion. By assigning a higher priority to buffers with more packets queued inside, the scheme successfully reduces the risk of packet loss due to buffer overflow. Moreover, it regulates the network traffic and prevents transmission resources contention.

2.2.5 Congestion Control

Congestion control is provided at each router-hub interface of the CUM LAUDE NET to prevent overflow of transmit buffers (note that the dual-ring itself is congestion free under the ACTA protocol). The idea is schematically shown in Figure 2.4. As mentioned previously, there are three transmit and one receive buffers in Ring-A, Ring-B and Local router of a router-node. Each of these buffers is used to hold packets awaiting processing. Overflow will occur at the transmit buffers under any one of the following three situations. First, if packets are continuously coming out from the receive buffers at a rate faster than the maximum rate that the transmitter can transmit. Second, the packet quota (defined by the ACTA protocol) for a router-node is used up but the hub is not aware of this and keeps forwarding packets. Third, there is no empty slots available on the ring and thus the packets are piled up in the transmit buffers and can not be transmitted.

In order to avoid these undesirable situations, we have used the fullness condition of the transmit buffers as a control to switch the packet flow from the hub 'on' and 'off'. When any one of the transmit buffers in Ring-A or Ring-B is found to be half-full, a *STOP_POLL* command is sent to the Local router. On receiving this command, the Local router will stop polling the host machines connected to the hub and hence new packets are blocked from entering the Local router. The transmit buffers in Ring-A and Ring-B routers are then able to empty their contents. Once the transmit buffers have emptied their contents, a *START_POLL* command is sent to the Local router to re-enable the polling process. Host machines connected to the hub will then be able to send packets again.

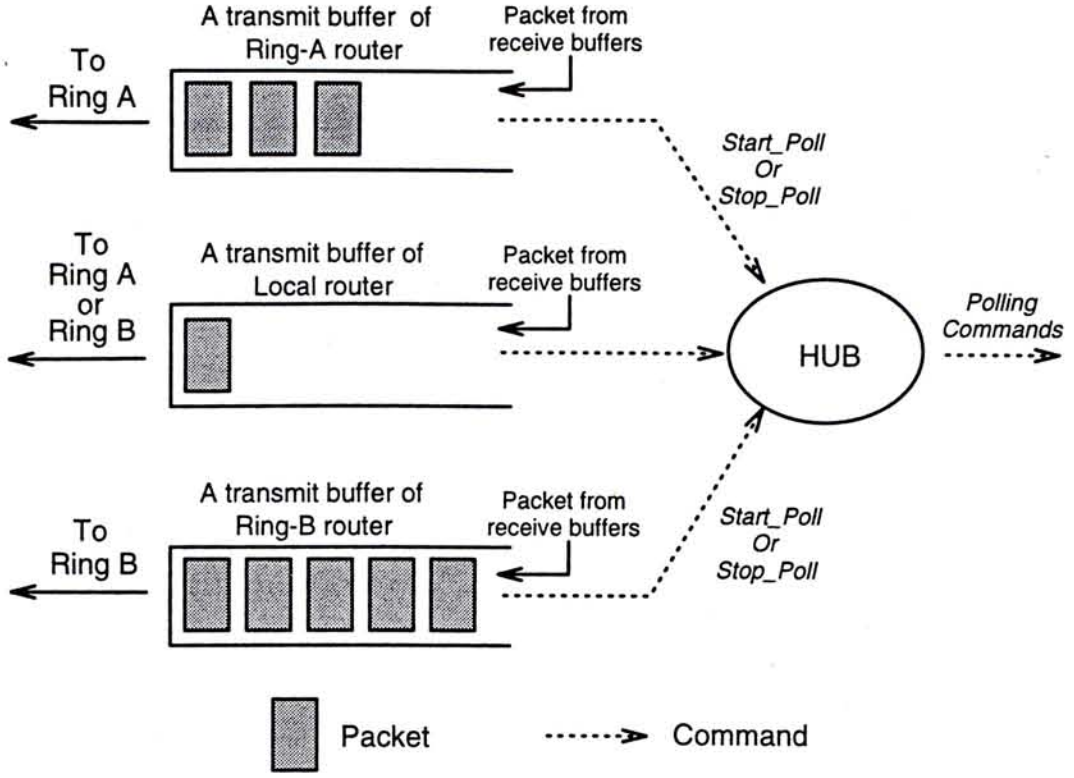

A transmit buffer of Ring-A router
Packet from receive buffers
To Ring A
Start_Poll Or Stop_Poll
To Ring A or Ring B
A transmit buffer of Local router
Packet from receive buffers
HUB
Polling Commands
A transmit buffer of Ring-B router
Packet from receive buffers
To Ring B
Start_Poll Or Stop_Poll
Packet
Command

Figure 2.4: A schematic diagram of the congestion control algorithm used in CUM LAUDE NET. In the figure, only one transmit buffer of each router in a router-node is shown for the sake of simplicity, as the other two are performing exactly the same functions.

2.3 Protocols

2.3.1 Design Overview

CUM LAUDE NET is designed to support high-speed, real-time multimedia services with maximum compatibility to IP-based networks and to provide an easy upgrade for these networks to support multimedia traffic. In order to achieve these goals, we have decided to use Fast Packet Routing (FPR) and integrated networking technology that employs:

- fixed size IP datagrams/FPR packets (576/582 octets)
- fast packet routing (FPR) in the MAC and network layer

- direct IP addressing in the transport and routing of IP datagrams
- connectionless delivery of packets

In the Fast Packet Routing (FPR) Layer, which combined the MAC layer and some of the network layer functions, the IP datagrams are encapsulated by a fixed-size header and trailer and all routing information is available in the header. This allows each router to perform fast packet routing efficiently and simplifies the gateway design between CUM LAUDE NET and Internet.

A novel network protocol ACTA (Adaptive Cycle Tunable Access) [25] is implemented in each Level-2 and Level-1 dual-ring hierarchy. Fair access is achieved by limiting the number of empty slots occupied by each router on each cycle. The cycle length is adjusted to reduce the packet latency and to increase the throughput.

2.3.2 ACTA - The MAC Protocol

The ACTA protocol is a simple bus/ring network protocol suitable for multichannel operations. It adopts a slotted cycle format to transmit packets, with adaptive variation of cycle lengths according to network loading to achieve optimal performance. The adaptation of the cycle length is governed by the equation below:

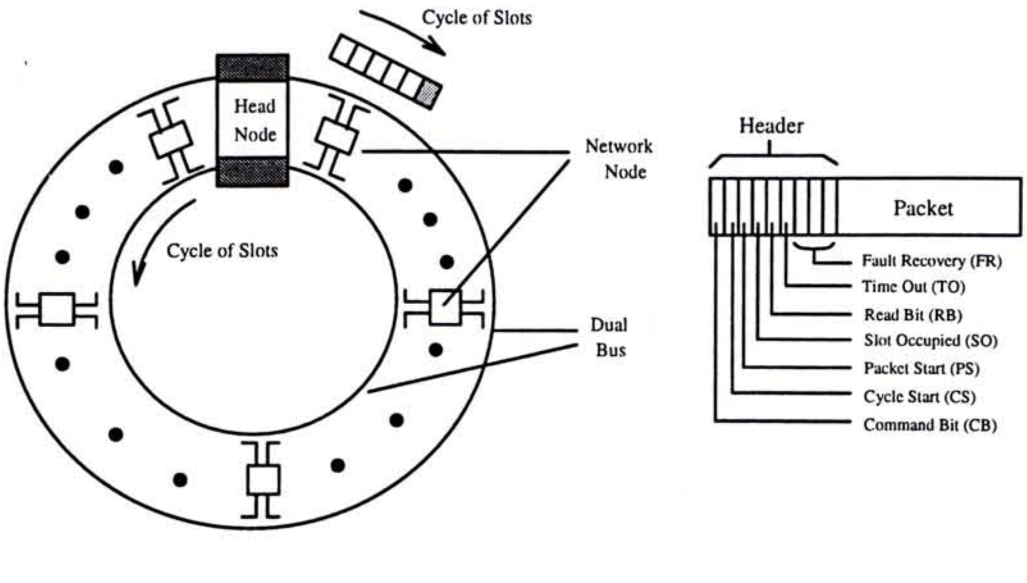
$$C_n = \frac{(C_c \times U_c)}{L_c} \tag{2.1}$$

where C_n is the new cycle length calculated through the above algorithm. C_c is the length of current cycle that the end node observed. U_c is the cycle utilization which is the number of slots used in the cycle divided by the cycle length, and the controlled load L_c is a parameter specifying the desired throughput under heavily overloaded condition. When the cycle utilization U_c is larger than the controlled load L_c, the new cycle length C_n will be increased. On the contrary, when the cycle utilization is smaller than the controlled load, the new cycle length will be decreased. From a statistical point of view, the maximum utilization is determined by the controlled load L_c.

In ACTA, only two control bits per-slot are required in the media access, thus it can be made slot compatible to ATM. The major advantages of ACTA protocol are its simple media access, reduced processing and its throughput performance, which could be ≥ 0.9(normalized) per channel with good fairness even under heavily overloaded conditions. Further details about the ACTA protocol can be found in [44]. The structure of the network with ACTA protocol being implemented is shown in Figure 2.5(a), each node consists of two pairs of receiving and transmitting modules, one for each ring. Empty slots are continuously generating from the Headnode in both rings, the cycle length of slots is varied according to the traffic conditions. In Figure 2.5(b), the structure of an ACTA slot is illustrated. It is a 512-byte packet with 1-byte control header.

2.3.3 Protocol Layering

The CUM LAUDE NET protocol layering is shown in Figure 2.6. The protocol is basically an extension of the Internet Protocol suite, and is designed to have maximum compatibility with IP. Since TCP is not suitable for real-time multimedia application, a new Video and Voice Transport Protocol, VVTP, which is more suitable for carrying real-time video and voice is designed. VVTP is similar to UDP, but it has a fixed size, and does not perform acknowledgment, error



(a) ACTA dual-ring network (b) ACTA slot structure

Figure 2.5: A dual-ring network with ACTA protocol.

detection/correction or error retransmission. The design decisions are based on the needs for fast packet routing and the fact that many TCP functions like acknowledgment and error detection/correction are too slow or unnecessary for real-time video and voice applications. The VVTP fragment size is chosen to be 552 bytes. The corresponding IP datagram encapsulation have a size of 576 bytes, which is the recommended size that can be handled by Internet networks and gateways without fragmentation.

The operating system for CUM LAUDE NET is a public domain system called Linux. The kernel of the operating system has been modified to support VVTP as well as TCP/UDP. VVTP has been given a higher priority than TCP/UDP to prevent non-real-time packets from blocking up the transmission queue.

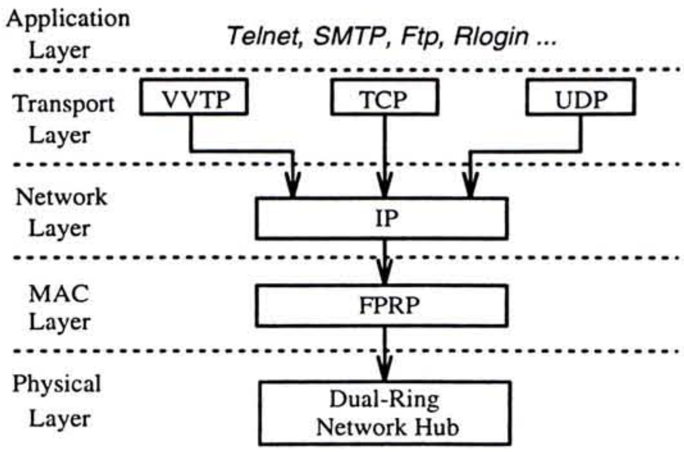


Figure 2.6: Protocol Layering of CUM LAUDE NET.

2.3.4 Segment, Datagram and Packet Format

Fixed size segments (Transport Layer - 552 octets), datagrams (Network Layer - 576 octets), and packets (FPR Layer - 582 octets) are used in the CUM LAUDE NET. In the FPR layer, a CUM LAUDE packet is consisted of four fields (Figure 2.7):

- header (1 octet)
- destination address/VCI (4 octets)
- fixed size IP datagram (576 octets)
- trailer (1 octet)

The packet format is chosen to facilitate the hardware and software design of the FPR Layer. The control information required for fast packet routing is carried by the first 5 octets (header - destination address/VCI) of the CUM LAUDE packets. The header and trailer octets are used for frame synchronization as well as to provide information for routing. Since the CUM LAUDE NET employs the ACTA protocol in the fast packet routing layer, control information

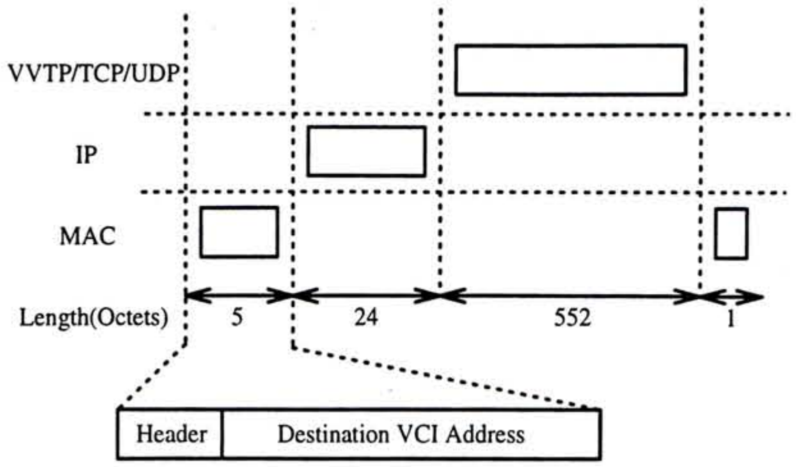


Figure 2.7: Packet encapsulation of CUM LAUDE NET.

SP/EP	CS	SO	IP/VCI	RD	NM	NM	NM

SP:Start of Packet
EP:End of Packet
CS:Cycle Start
SO:Slot Occupied
IP:Internet Protocol Packet
VCI:Virtual Circuit Identifier Packet
NM:Network Management Packet

(a)

R	H/R	C1	C0	A3	A2	A1	A0

R:Reserved for Future Use
H/R:Control Byte from Hub or Router
C1,C0:Hub Command
A0-A4:Hub Polling Address

(b)

Figure 2.8: The header structure used by (a) Router and (b) Hub.

for the implementation of the ACTA protocol like "cycle start" and "slot occupied" must be provided in the header. The details are given in Figure 2.8(a). The bits marked with 'NM' are used for network management, i.e., to support the fault-tolerant and auto-healing functions.

2.3.5 Fast Packet Routing

As it has been mentioned above, the network protocols used in the CUM LAUDE NET are designed to provide efficient implementation of fast packet routing. The media access and routing algorithms use only the first 5 header octets of a FPR packet shown above. For Level-2 or Level-1 dual-ring hierarchies, the media access/routing control is based on the ACTA protocol. For the hubs, media access is based on a modified Binary Exponential Backoff Polling (BEBP) algorithm mentioned in [31].

Level-1/Level-2 Bridge/Router

Fast packet routing algorithms based on ACTA are implemented on all Level-1/Level-2 Bridges/Routers of the dual-ring networks. The data rates for Level-2 and Level-1 nodes are set to be 1 Gb/s and 100 Mb/s on each ring respectively.

For incoming packets, the router examines the FPR packet header to determine whether the destination of the packet belongs to a host that is served by the router. If the destination does not match, the original packet will be forwarded onwards as shown in Figure 2.9(a). The packets will be erased only at the erasure node according to the ACTA protocol. Thus, it is not necessary for the router to search for information inside the IP datagrams.

The details of address comparison are as follows. The destination in the FPR packet header can either be a direct IP address or an indirect address called virtual connection identifier (VCI) which is suitable for multicasting. The two cases are indicated by a single bit in the header. If the destination is an IP address, a direct comparison is made to the address field. If it is a VCI address, the first three bytes in the address field will be used to compare with all VCI

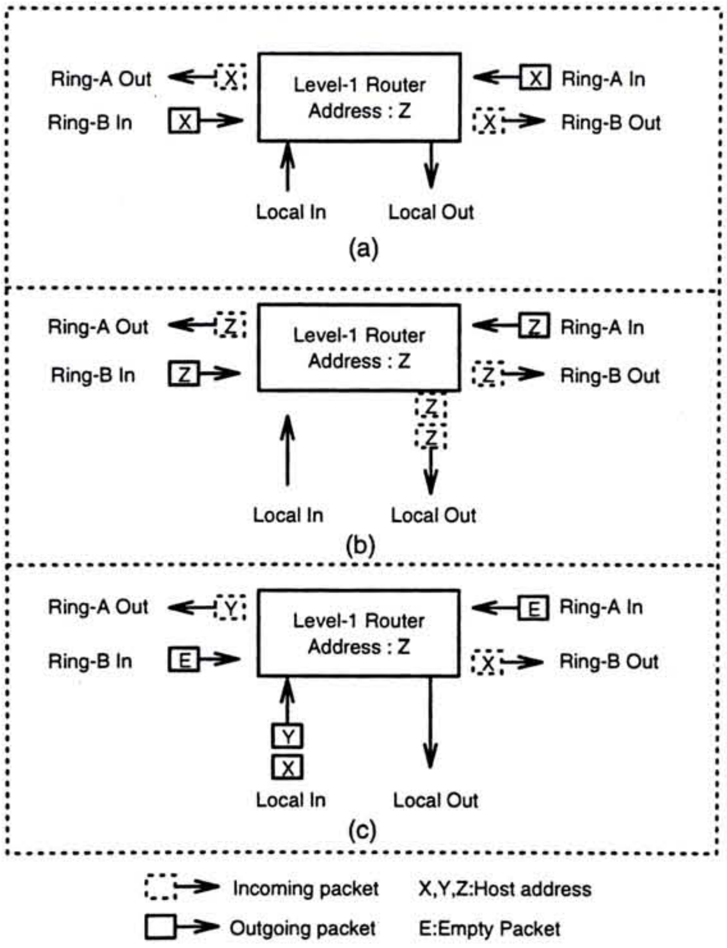


Figure 2.9: A schematic diagram showing how will a router respond to (a) an occupied packet on ring not addressed to router, (b) an occupied packet on ring addressed to router, (c) an empty packet on ring, under the Fast Packet Routing algorithm.

addresses in a VCI table dynamically stored in the router. Whenever a match is found, the router will copy the packet to the local buffer for forwarding to the next hierarchy or the local host and the original packet will also be forwarded onwards (Figure 2.9(b)). When the packet reaches the local host, the header, address and trailer fields of the FPR packet will be discarded, thus retrieving the original IP datagram.

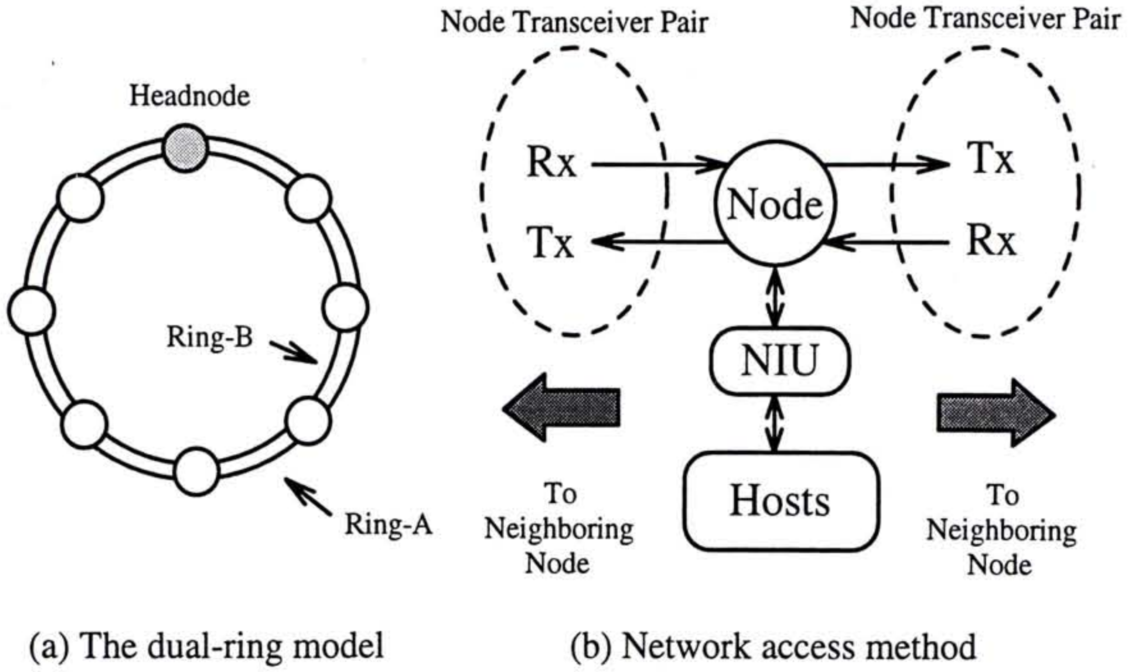
For outgoing packets, the router first determines whether there is any incoming empty slot. When there is an empty slot available, the router can fill up the empty slot with a packet in the queue according to the ACTA protocol (Figure 2.9(c)).

Level-1 Hub

Each hub connected to a router-node is used to serve as a concentrator/distributor to a number of local hosts. Any packets received by the hub will be broadcasted to all local hosts connected. The hubs also poll individual local hosts periodically to collect packets that are to be sent into the network. A BEBP based polling algorithm is used in order to provide orderly packet transmission from local hosts sharing a common broadcast link. The header format of the polling packet is shown in Figure 2.8(b).

2.3.6 Local Host NIU

The local host network interface unit (NIU) is directly connected to the host whose function is to process packets going between the network and the host. The NIU interrupts the operating system periodically to make sure that real-time packets can be served timely.



(a) The dual-ring model (b) Network access method

Figure 2.10: (a) The generalized dual-ring network model. (b) The network access scheme.

2.4 The Network Restoration Strategy

To facilitate fast and efficient network restoration in dual-ring networks (includes the CUM LAUDE NET), a distributed revertive restoration scheme designed on a generalized dual-ring model has been implemented. It restores the network by exchanging network information and control signals through the adjacent router-nodes. The scheme is distributive in nature and works on every router-node in parallel. This high level of parallelism allows faster response to failures and hence results faster restoration in general.

2.4.1 The Dual-Ring Model and Assumptions

2.4.2 Scenarios of Network Failure and Remedies

According to the model described above, two basic types of failures can occur in the network, i.e., link and node failures. Different combinations of link and node failures will lead to different failure scenarios. Some typical ones and their remedies are illustrated in Figure 2.11. As depicted in Figure 2.11 (a) and (b), single link and node failure can be recovered by wrapping up the links around the fault. In the case that the Headnode is failed, a new one can be regenerated among the operative nodes to keep the network operative, as depicted in Figure 2.11 (c). For multiple failures as shown in Figure 2.11 (d), isolated subnets will be formed, the local communications within the subnets can be maintained by regenerate a Headnode in each subnet.

2.4.3 Distributed Fault-Tolerant Algorithm

Phase I - Fault Detection Phase

In real practice, node failures can be interpreted as a subset of link failures. Therefore, our FT algorithm focuses only on the adaptation of link failures. Currently, two means are being used proposed to perform link failure detection. One is the out-of-synchronization detection in links. In this approach, the transmission quality of the link is constantly monitored and a violation signal is returned if the link is failed. This approach is generally applicable in all dual-ring networks, irrespective of the medium access protocols(random, slot, token, etc.,) used. However, it may require extra hardware to perform link synchronization detection.

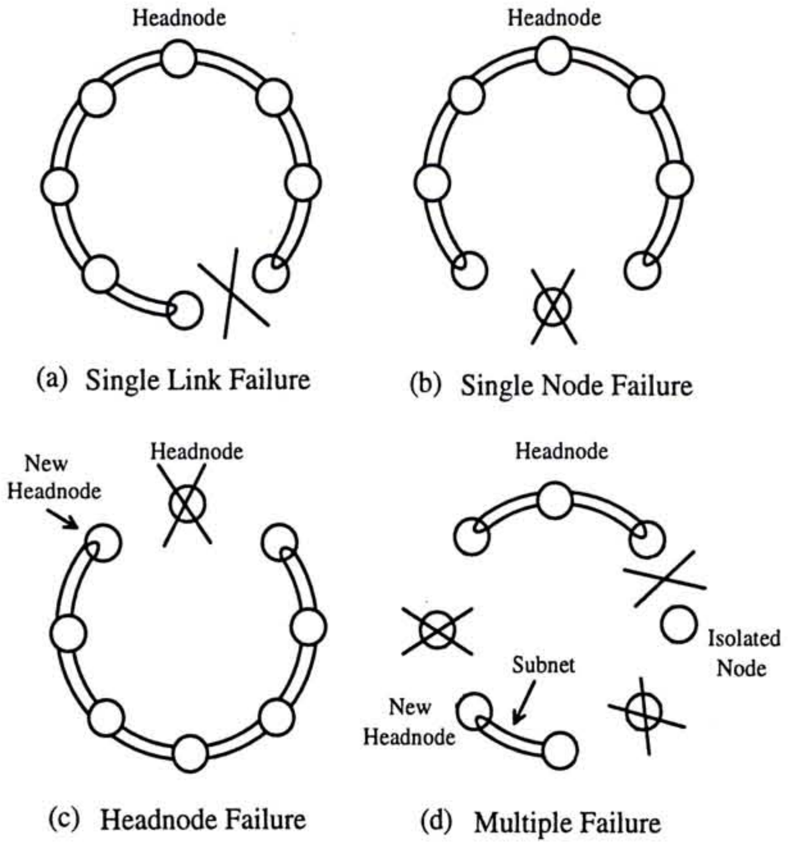


Figure 2.11: Typical failure scenarios in dual ring networks

Another means is the packet non-arrival detection. When a node receives no valid packet in a prescribed period, a *Test_Link* message will be sent to its neighbors to check whether the links in between of them are failed. Normally, a *Link_OK* message will be echoed back to indicate the links are healthy. However, If an instance of failure has occurred in between, no response will be received and the neighbor is assumed to be failed. For the case that no packet arrives a node in both rings, the network must be segmented by failures and hence the *Claim_Head()* process is invoked. This approach is particular suitable for networks using slot-access protocols and requires no extra hardware to detect link status.

Phase II - Fault Recovery Phase

Once the faults are located by the detection algorithm, the recovery process is initiated. It simply wraps the affected traffic in a ring to another. For the case of multiple failures as mentioned before, the network will be segmented into several subnets. Within each subnet, a new headnode will be generated according to the *Claim_Head()* process to maintain local communication. In the process, each node within the subnet will broadcast its registration number (a unique integer identifying a node, which is created during network initialization). Any node receives a registration number smaller than its own will generate an *Objection* message to the originating node. Finally, the node receives no objection after a pre-defined period of time becomes the Headnode. The FT algorithm is summarized in the 4-module pseudo code in Figure 2.12.

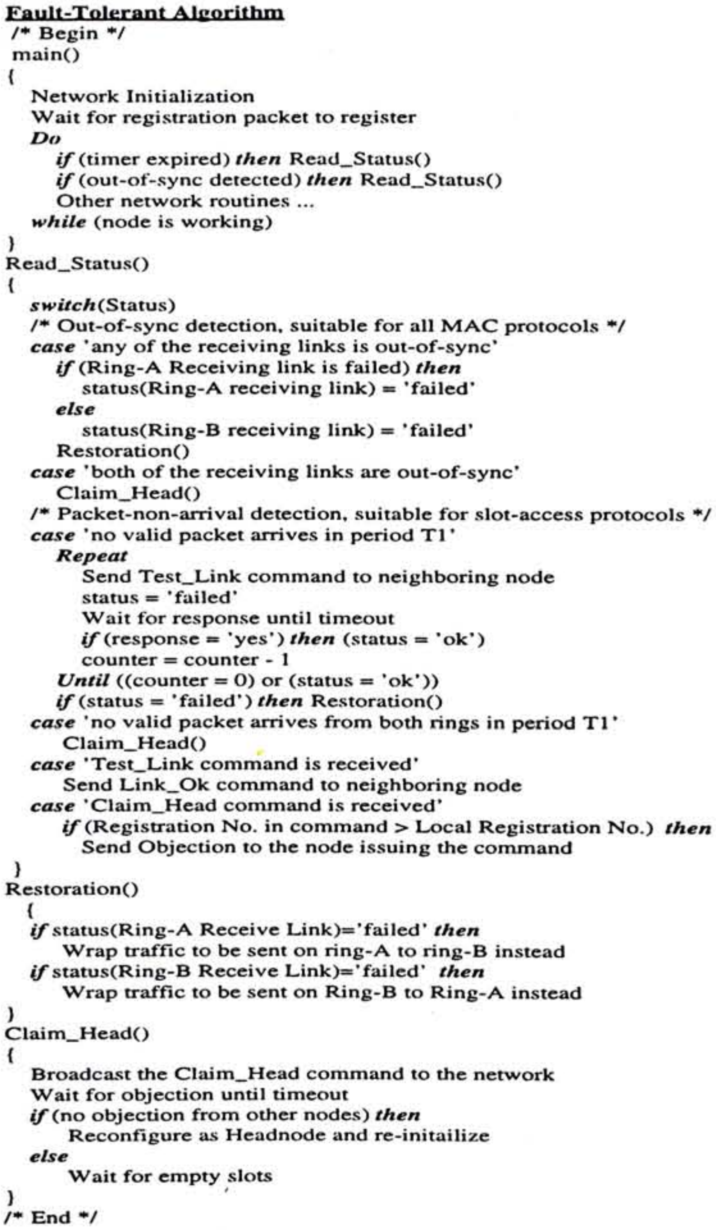
2.4.4 Distributed Auto-Healing Algorithm

Phase I - Sensing phase

Whenever any fault occurs in the system, the state of the faulty component will be continuously monitored by its healthy neighboring nodes. The neighbors of the faulty node will keep sending status-report requests (*Status_Request*) to the faulty node and wait for a response. If no response is received, the faulty node is considered as failed. The checking process is repeatedly performed until all components return to their normal states.

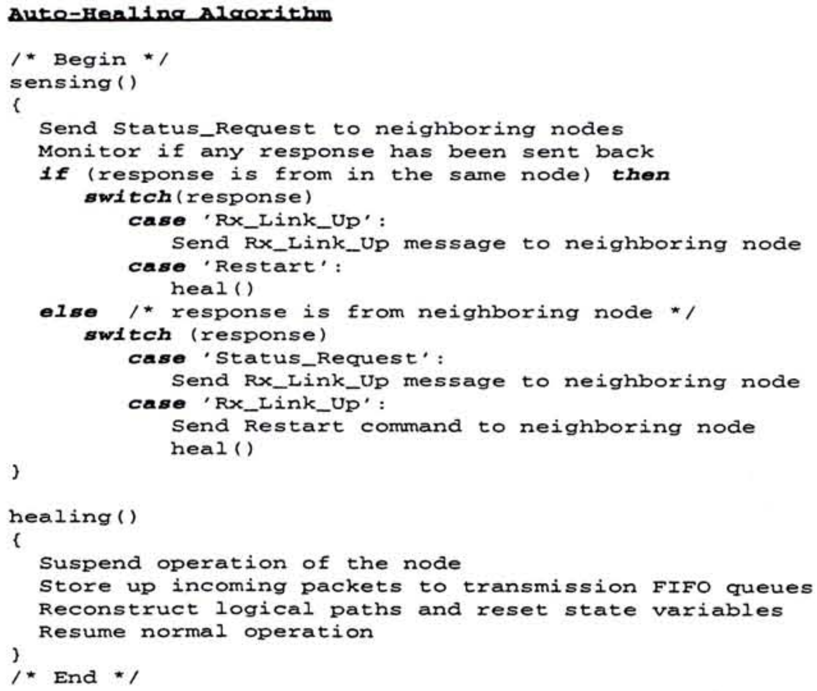
Phase II - Healing phase

After the faulty components have been repaired or replaced, the new healthy



```
Fault-Tolerant Algorithm
 /* Begin */
 main()
{
   Network Initialization
   Wait for registration packet to register
   Do
      if (timer expired) then Read_Status()
      if (out-of-sync detected) then Read_Status()
      Other network routines ...
   while (node is working)
}
Read_Status()
{
   switch(Status)
   /* Out-of-sync detection, suitable for all MAC protocols */
   case 'any of the receiving links is out-of-sync'
      if (Ring-A Receiving link is failed) then
         status(Ring-A receiving link) = 'failed'
      else
         status(Ring-B receiving link) = 'failed'
      Restoration()
   case 'both of the receiving links are out-of-sync'
      Claim_Head()
   /* Packet-non-arrival detection, suitable for slot-access protocols */
   case 'no valid packet arrives in period T1'
      Repeat
         Send Test_Link command to neighboring node
         status = 'failed'
         Wait for response until timeout
         if (response = 'yes') then (status = 'ok')
         counter = counter - 1
      Until ((counter = 0) or (status = 'ok'))
      if (status = 'failed') then Restoration()
   case 'no valid packet arrives from both rings in period T1'
      Claim_Head()
   case 'Test_Link command is received'
      Send Link_Ok command to neighboring node
   case 'Claim_Head command is received'
      if (Registration No. in command > Local Registration No.)  then
         Send Objection to the node issuing the command
 }
Restoration()
  {
   if status(Ring-A Receive Link)='failed' then
      Wrap traffic to be sent on ring-A to ring-B instead
   if status(Ring-B Receive Link)='failed'  then
      Wrap traffic to be sent on Ring-B to Ring-A instead
}
Claim_Head()
{
   Broadcast the Claim_Head command to the network
   Wait for objection until timeout
   if (no objection from other nodes) then
       Reconfigure as Headnode and re-initailize
   else
       Wait for empty slots
}
/* End */
```

Figure 2.12: The Fault-Tolerant Algorithm



```
Auto-Healing Algorithm

/* Begin */
sensing()
{
  Send Status_Request to neighboring nodes
  Monitor if any response has been sent back
  if (response is from in the same node) then
     switch(response)
        case 'Rx_Link_Up':
           Send Rx_Link_Up message to neighboring node
        case 'Restart':
           heal()
  else  /* response is from neighboring node */
     switch (response)
        case 'Status_Request':
           Send Rx_Link_Up message to neighboring node
        case 'Rx_Link_Up':
           Send Restart command to neighboring node
           heal()
}

healing()
{
  Suspend operation of the node
  Store up incoming packets to transmission FIFO queues
  Reconstruct logical paths and reset state variables
  Resume normal operation
}
/* End */
```

Figure 2.13: The Auto-Healing Algorithm.

node will immediately respond to the *Status_Request* command sent out by its neighbors. Hence its neighbors will be aware of that the fault has been removed and the healing process can be started. In the healing process, normal operations of the newly repaired node and its neighbors are temporarily suspended. All incoming packets during the restoration will temporarily be buffered until the healing process is completed. Logical paths are then reconstructed and the state variables are reset. After the reconfiguration of the network at the failure location is completed, the nodes are started up again and the buffered packets are released. The distributed auto-healing algorithm is summarized in Figure 2.13. It consists of the *sensing()* and *healing()* modules, which corresponds to the two phases described above.

A state diagram illustrating the state transitions and communication signal involved in a node transceiver pair is shown in Figure 2.14. As described in our dual-ring network model, the transmitter (Tx) of a node in Ring-A/Ring-B and its receiver (Rx) in Ring-B/Ring-A will form a transceiver pair(Figure 2.14a). When a fault occurs, the transceiver pair will switch to one of the three possible failure states, i.e., (1) both transmitter and receiver are failed, (2) transmitter is failed but receiver is still operative or (3) transmitter is operative but receiver is failed. The healthy neighboring node will send the *Status_Request* command continuously to the faulty node and wait for a response. If both the receive and the transmit links between the faulty node and its healthy neighbor are repaired and become operative again, an *Rx_Link_Up* message will be echoed back from the faulty node. On receiving this message, the healthy neighbor will start to re-initialize itself and send a *Restart* command to re-initialize the repaired node as well. As the whole restoration process is completed within a milli-second, it is virtually transparent to most network users.

2.4.5 The Network Management Signals

As described in the above subsections, a few control signals are sufficient to perform the network fault-tolerant and auto-healing functions. Table 2.2 has summarized all the control signals involved in the proposed algorithms and the corresponding number of bits needed to represent them. It is found that a 2-byte restoration message overhead will be sufficient to carry all these signals. In failure recovery, a short restoration message is crucial as it can alleviate the additional network loading imposed by the control messages and can prevent the network from further congestion.

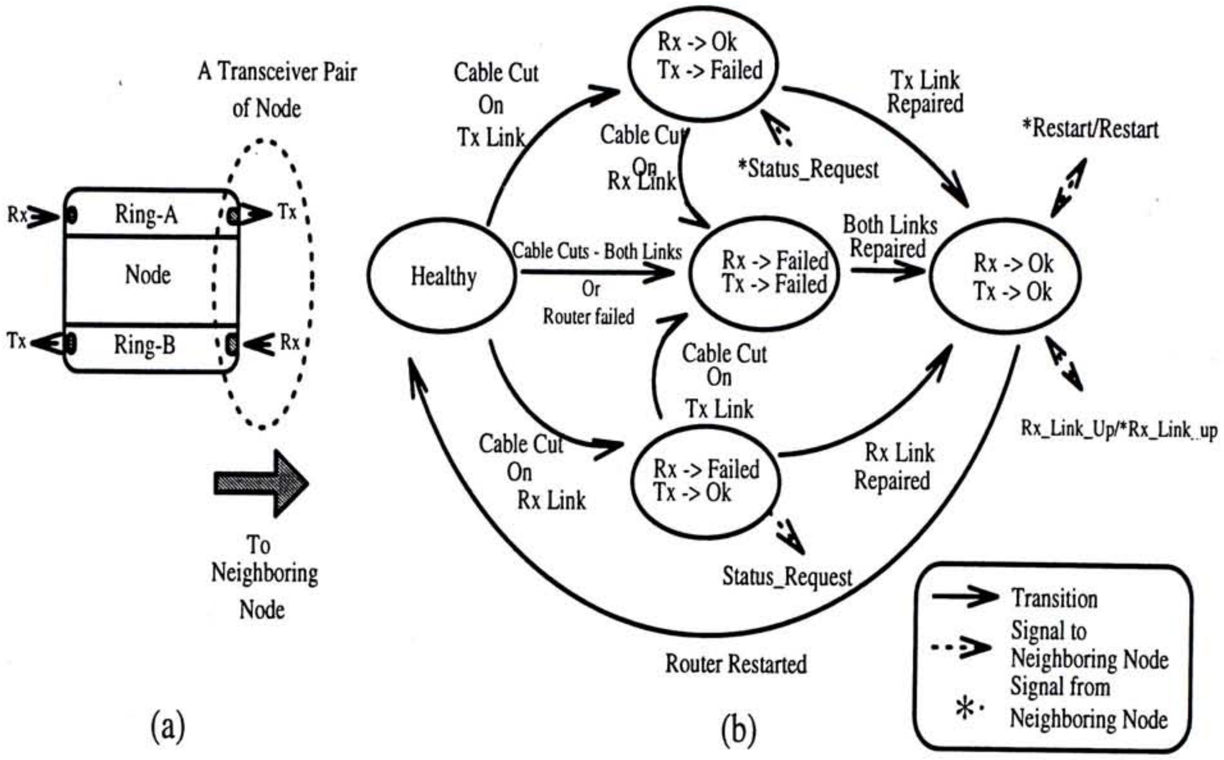


Figure 2.14: (a) A transceiver pair of a router-node. (b) The state diagram showing the state transitions and communication signals involved of a transceiver pair in the AH process.

2.5 Performance Evaluation

2.5.1 Restoration Time

The fault-tolerant and auto-healing time for node and link failures against different parameters are simulated. Various factors such as fault detection time, protocol handling time, signaling message transmission delay and FIFO delay are taken into account. In the simulation model, all router-nodes are connected in a fiber dual-ring. Control messages going in or out of a router-node are buffered by FIFOs at the transmitters and receivers. The amount and length of control messages generated in different types of failures are defined in the algorithms. The parameters used in the simulation are based on the measured

Control Signals	No. of bits needed
Test_Link	1
Link_Ok	1
Status_Request	1
Rx_Link_Up	1
Restart	1
Claim_Head	1
Objection	1
Node Registration No.	8

Table 2.2: A summary of the control signals used in the proposed fault-tolerant and auto-healing algorithms and the number of bits required to represent them.

Parameters	Value
packet non-arrival detection time	110.00 μs
out-sync detection time	55.00 μs
Message forwarding time of node	12 μs
FIFO delay	0.12 μs
Ring wrap time	3.00 μs
Node initialization time	6.10 μs
Router communication time	1.95 μs
Link capacity	100 Mb/s
Message traveling speed in links	1.99e8 m/s
Transmitter latency	0.74 μs
Receiver latency	1.22 μs

Table 2.3: The parameters used in the fault-tolerant and healing time model. They are obtained from the measured value in the network prototype and hardware specifications given by the network component manufacturers.

values in the network prototype we built earlier and component specifications provided by the component manufacturers. Their values are shown in Table 2.3.

The relation of fault-tolerant and auto-healing time for different node separation are shown in Figure 2.15. It is observed that, in general, more recovery time is needed when the node separation is increased. This reflects the fact that controlling messages will take longer to travel between neighboring nodes for larger node separation. The only exception occurs in the node failure recovery, this is because no communication is possible between the faulty node and its

Restoration Techniques		Type of Networks		
		FDDI	SONET	CUM LAUDE
Dedicated Facility Restoration	Te	Dual Homing	Diverse Prot.(APS)	–
	Eq	DCS	1:N APS	–
	Ti	second - minute	around 50ms	–
	Te	Protective Switching	Self Healing Rings	–
	Eq	Nodal Bypass Switch	ADM	–
	Ti	less than a second	around 50ms	–
Dynamic Facility Restoration	Te	Ring-Wrapping	Reconfig. Mesh	Ring-Wrapping
	Eq	DCS	DCS	Reconfig. Router
	Ti	second - minute	sec - min	less than a ms

Table 2.4: A comparison of restoration techniques for some common ring technologies based on the techniques used(Te), the equipment required(Eq) and the recovery time(Ti).

neighbor. Hence it is virtually independent of the node separation.

Among the existing fault-tolerant ring networks, FDDI and SONET are the most well known ones. Thus a simple comparison with the CUM LAUDE NET on the recovery techniques and times required is done. The result is tabulated in Table 2.4. It is found that dedicated facility generally provides faster restoration than dynamic facility and their restoration times can vary from milli-second to second order. In the above network technologies compared, the CUM LAUDE NET provides fastest restoration.

2.5.2 Reliability Measures

In order to characterize the reliability performance of a dual-ring network under our algorithms, several measures [3] [34] have been used. They included:

- Communicative Probability after failure, $P[C]$ - It is defined as the probability of all operative nodes that are still communicative after failure. It quantifies the ability of a network to isolate failures. A higher value of this measure means communications between nodes are less affected by failures.

- Survivability, s - It is defined as the fraction of a certain selected network feature remained after an instance of failure. Here we will use the fraction of nodes remains connected to the Headnode after failure as a measure of the network survivability. Usually, a survivability function is used instead of a single value survivability.
- Average Reachability, AR - it represents the average fraction of nodes that a node in the network can communicate after an instance of failure. A larger value of AR implies less impact brought by the failures will be experienced by the users as they can still communicate with most of the network nodes.

Each measure described above quantifies the reliability of a network only in a certain aspect. Therefore an integrated study of them is necessary for a better description of network reliability.

Communicative Probability after Failure

Generally, solving $P[C]$ for networks is an NP-hard problem and computer-aided analysis is necessary. However, the simple topological properties of dual ring networks allow closed form solutions. We have formulated the expression of $P[C]$ for dual-ring networks under our FT algorithm. The derivations have been put in the appendix and the results are summarized below:

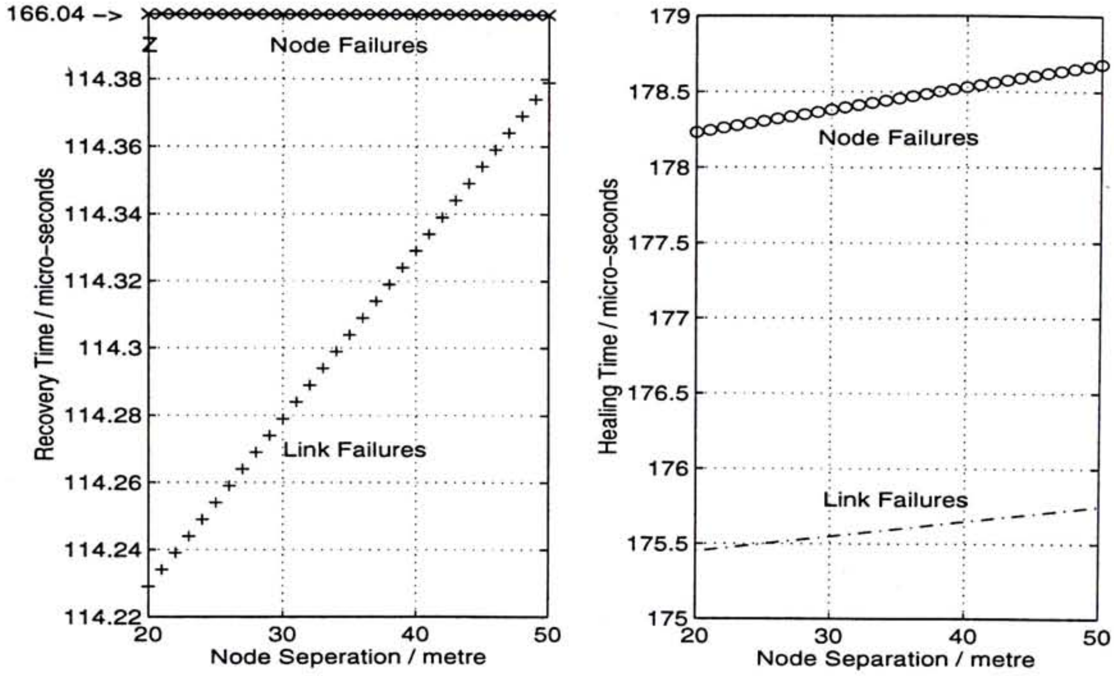


Figure 2.15: A plot of the fault-tolerant and auto-healing time for single node and link failures against the node separation.

Denotes:

N = Number of nodes in the network

l = Number of links failed

n = Number of nodes failed

$P[C|n,l]$ = Probability of all operative nodes are still communicative given that n nodes and l links are failed

<u>$P[C]$ of a dual-ring network with fault-tolerant algorithm</u>

$$P[C|n,l] = \begin{cases} \frac{(N-n)}{{}_NC_n} \times \frac{(N-n)({}_{2n+2}C_l - {}_{2n}C_l)}{{}_{2N}C_l} & n \neq 0, l \leq 2n+2 \\ 0 & n \neq 0, l > 2n+2 \\ \frac{2\,{}_NC_l}{{}_{2N}C_l} & n = 0, 2 \leq l \leq N \\ 1 & n = 0, l = 1 \end{cases} \tag{2.2}$$

Figure 2.16 shows the variation of $P[C]$ of a network with the FT algorithm implemented against different failure scales (i.e., fraction of nodes failed). In

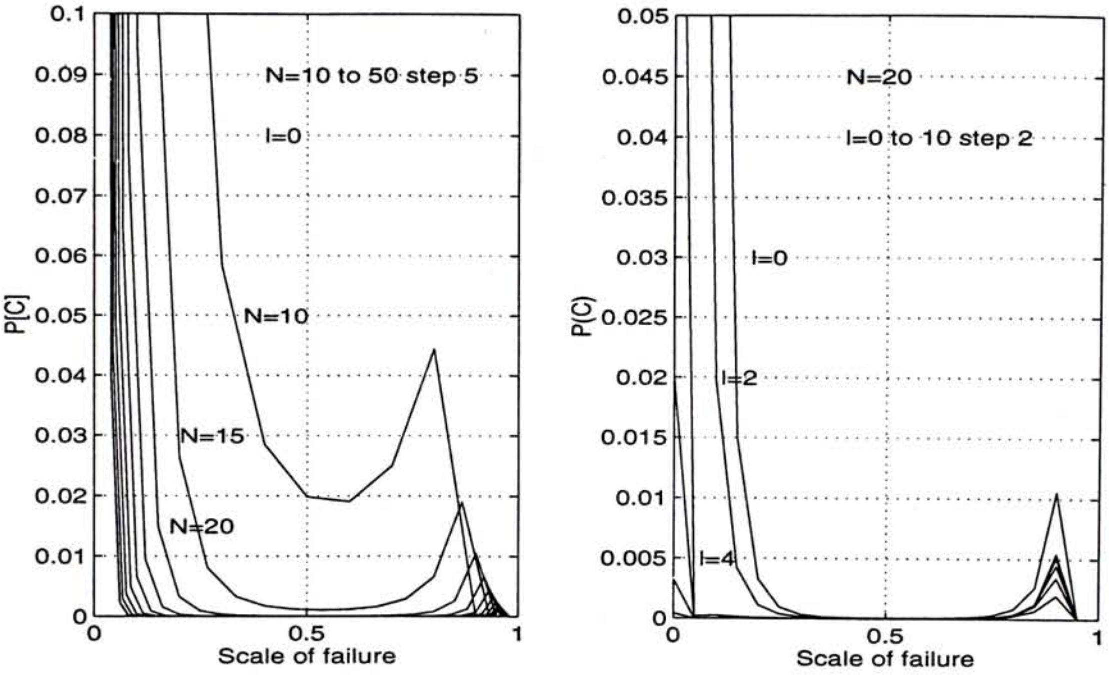


Figure 2.16: (a) $P[C]$ against the scale of failure for different network sizes. (b) $P[C]$ against the scale of failure for different numbers of link failures.

Figure 2.16(a), it is observed that the value of $P[C]$ decreases as the network size or failure scale grows larger. The reason behind is quite intuitive: As the network size or failure scale increase, there is a higher probability that isolated subnets are formed after the failure. Hence, a communicative network is less likely to be found, in other words, a lower value of $P[C]$ will be obtained. In Figure 2.16(b), different values of l are used to investigate the relation again, it is found that $P[C]$ decreases as number of link failures, l increases. Note that there is a peak on each $P[C]$ curve as the failure scale approaching unity. This simply reflects the fact that when most of the nodes are failed, the probability that only a single subnet remained is, on the contrary, becoming larger.

Network Survivability

In considering the survivability function of a dual-ring, only the cable cuts closest to the Headnode is important, other cuts will not affect the resulting survivability. There can be 1-4 cuts closest to the Headnode, but a single cut or a double cut on one side will not segment the network and hence does not contribute to the survivability function. Therefore only the cases involving 2(except double cut), 3 and 4 cuts are needed to be considered.

The number of possible network configurations, denoted by*No. of Config[]*, with 2, 3 and 4 closest cuts to Headnode that leads to the same survivability s (the fraction of nodes still connected with Headnode) is given by:

Case 1: 2 closest cuts

$$\begin{aligned} No.\ of\ Config[2\ cuts] &= 4Ns \times_{2N(1-s)-2} C_{l-2} \\ No.\ of\ Config[double\ cut] &= 2Ns \times_{N(1-s)-1} C_{l-2} \end{aligned}$$

Case 2: 3 closest cuts

$$No.\ of\ Config[3\ cuts] = 4Ns \times_{2N(1-s)-2} C_{l-3}$$

Case 3: 4 closest cuts

$$No.\ of\ Config[4\ cuts] = 4Ns \times_{2N(1-s)-2} C_{l-4}$$

The probability of obtaining a certain survivability s given that l links failed can be found by dividing the number of configurations that lead to s by the total number of configurations with l link failures$({}_{2N}C_l)$:

$$\begin{aligned} P(s|l) = \ &\{4Ns \times_{2N(1-s)-2} C_{l-2} - 2Ns \times_{N(1-s)-1} C_{l-2} + \\ &4Ns \times_{2N(1-s)-2} C_{l-3} + 4Ns \times_{2N(1-s)-2} C_{l-4}\}/_{2N}C_l \end{aligned} \quad (2.3)$$

If we also taking the special cases (e.g. l=0 or 1) into account, the final expression for survivability function of dual-ring networks with fault-tolerant algorithm

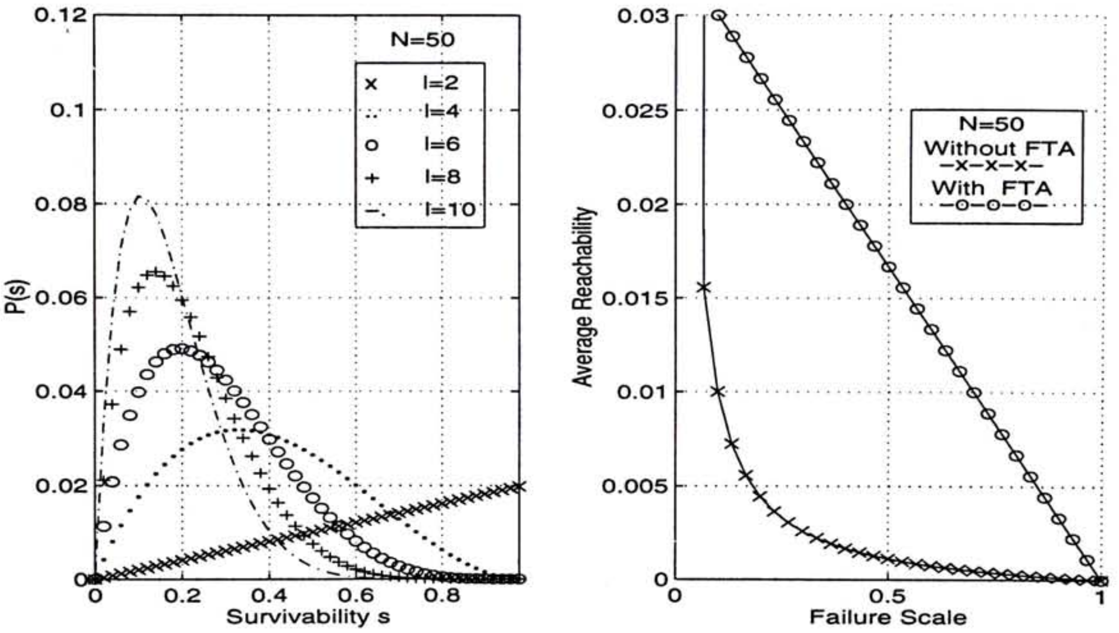


Figure 2.17: (a) Survivability functions of Dual-Ring networks with FT algorithms for different number of link failures. (b) Average Reachability of networks with/without FT algorithm

should be:

$$P(s|l) = \begin{cases} \{4Ns \times_{2N(1-s)-2} C_{l-2} - 2Ns \times_{N(1-s)-1} C_{l-2} + & \\ 4Ns \times_{2N(1-s)-2} C_{l-3} + 4Ns \times_{2N(1-s)-2} C_{l-4}\}/_{2N}C_l & l \geq 2 \\ 1 & l \leq 1 \end{cases} \quad (2.4)$$

Figure 2.17(a) shows the survivability function of a 50-node dual-ring network with the FT algorithm implemented against different number of link failures. It is found that, as the number of link failures grows, the survivability function of the network shifts towards the left and becomes sharpened. It means that the probability of less components survived is higher. The peaks on the curves correspond to the most probable fraction of components survived from the failure.

Average Reachability

Practically, finding out the average reachability of a network subject to failures is equivalent to determining its average subnet size after failures. The average reachability of networks with or without the FT algorithm are formulated as follows:

Without Fault-tolerant Algorithm

When a network without FT algorithm is segmented into subnets by faults, only the subnet containing the headnode can maintain communications between survived nodes. Hence its average reachability, subjected to n node failures, is given by:

$$AR \;=\; \frac{Average\ size\ of\ subnet\ containing\ headnode}{Total\ number\ of\ subnets\ created} = \frac{\sum_{s=0}^{1} sP(s|n)}{n}$$

Together with the special cases of n=0 and 1, its average reachability becomes:

$$AR = \begin{cases} 1 & \text{if n=0} \\ \frac{(N-1)^2}{N^2} & \text{if n=1} \\ \frac{\sum_{s=0}^{1} sP(s|n)}{n} & \text{if n} \geq 2 \end{cases} \tag{2.5}$$

With Fault-tolerant Algorithm

For a network with FT algorithm, each subnet is able to maintain local communications. Therefore, its average reachability will be the sum of the average sizes of all subnets divided by the number of subnets generated:

$$AR \;=\; \frac{Sum\ of\ average\ sizes\ of\ all\ subnets}{Total\ number\ of\ subnets\ created} = \frac{\sum_{allsubnets} \sum_{s=0}^{1} sP(s|n)}{n}$$

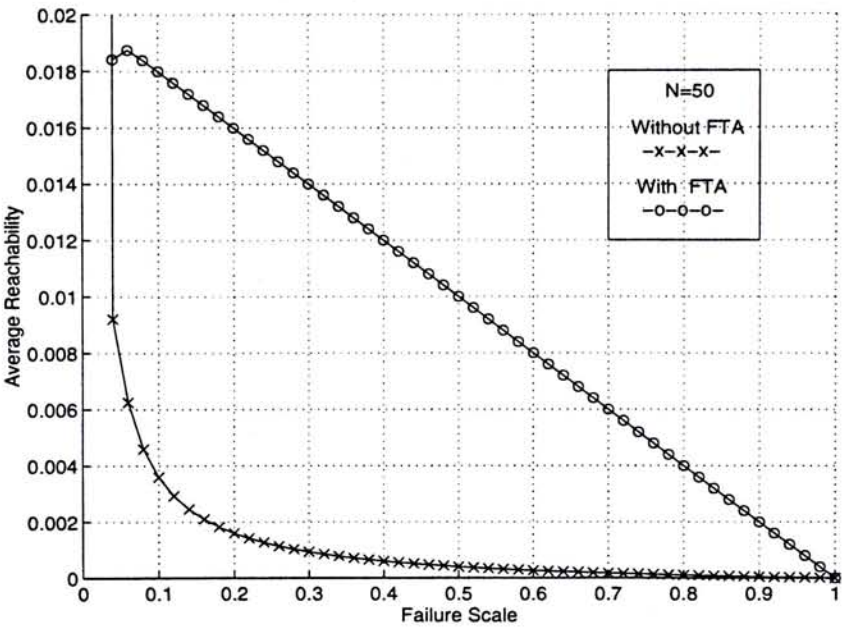


Figure 2.18: Average Reachability of networks with different topologies.

Taking the special cases n=0 and 1 into account, it becomes:

$$AR = \begin{cases} 1 & \text{if n=0} \\ \frac{N-1}{N} & \text{if n=1} \\ \frac{\sum_{allsubnets} \sum_{s=0}^{1} sP(s|n)}{n} & \text{if n} \geq 2 \end{cases} \tag{2.6}$$

A graph comparing the average reachability of a dual ring network with and without the FT algorithm is shown in Figure 2.17b. Networks with FT algorithm show a graceful degradation of average reachability against failures while the degradation in networks without the algorithm is much more abrupt.

2.5.3 Network Availability During Restoration

Beside survivability, maintaining high availability of networks during restoration is also crucial. This not only protects important data from being lost, but also causes less disturbance to the network users. According to the FT and AH algorithms, buffering memories are used to provide temporary storage for data during restoration. Together with the milli-second ordered failure restoration time, basically no data will be lost because of buffering memories overflow. As a result, the impact brought by the failures will almost be transparent to most network users and thus the network seems to be 'available' even failures have occurred. Figure 2.19 shows how incoming traffic is buffered in case of a link failure has occurred. In the figure, we assume a link failure has occurred in the outgoiug link of the Ring-A router. This fault will quickly be detected and triggers the restoration process. Before the whole restoration process is completed, the forwarding action of receive buffers of a router-node will be temporarily be suspended. Within this period, the incoming packets from Ring-A are stored in the receive buffer of Ring-A router. It is estimated the buffer will be full in about 15 packet's time (buffer size divided by packet size), which is sufficient for the whole restoration process to complete. When the restoration is done, the buffered packets are sent along the switched path.

55 μs, whether valid packets have arrived in this period. If no valid packet arrives in this period, a fault-detection routine will be invoked since some kind of failures may occur in the network.

2.6 The Prototype

A schematic diagram of the three-node dual-ring network prototype which had been deployed in the Chinese University of Hong Kong is shown in Figure 2.20. Each node is connected to a hub which acts as a concentrator/distributor of the traffic flowing between the network and the host machines. Network Interface

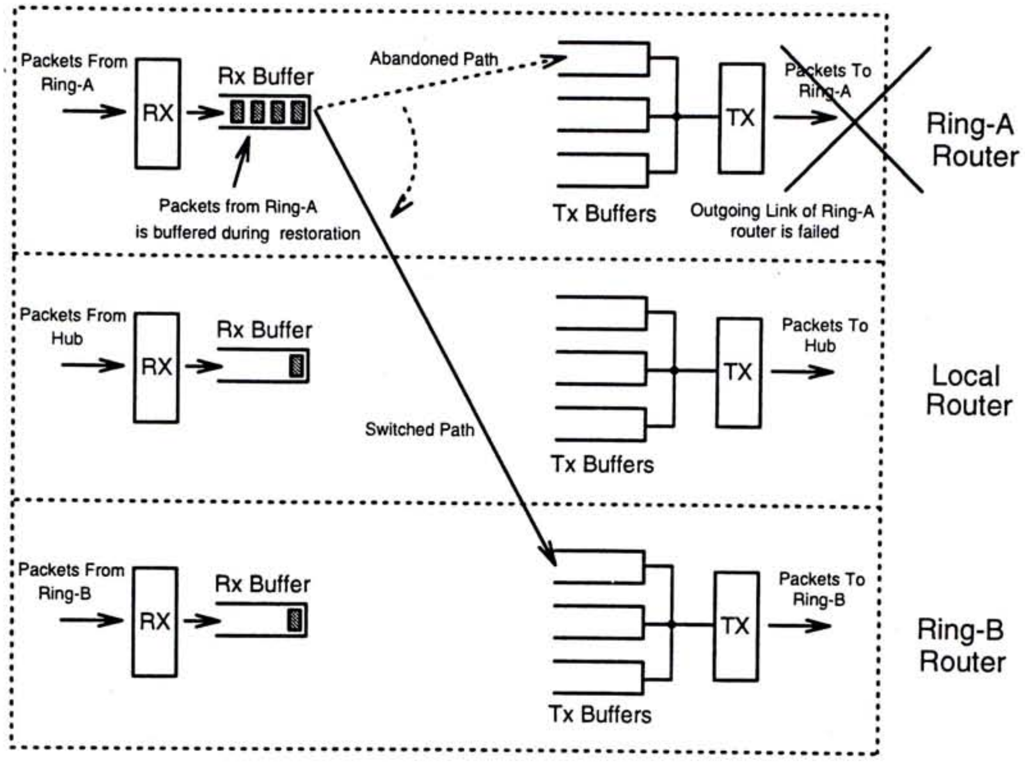


Figure 2.19: This figure shows how incoming traffic is buffered during restoration.

Cards (NIC) on 16-bit AT bus are used to interface the PCs to the network. Other than the NIC unit, every station is equipped with a video digitizer card and a full duplex sound card to support multimedia applications such as videoconferencing. The sound card, network interface card and all device drivers are designed and prototyped by our research group at the Chinese University of Hong Kong. In the field testing, the performance of the network is found to be satisfactory and the service outage time is less than a few percents of the total running time.

Also, CUM LAUDE NET has been connected to Internet, and all standard Internet services like electronic mail (SMTP), remote access (TELNET), and file transfer (FTP) are supported. The user interface is the industry-standard X-windows. The network has also been connected to the public telephone network

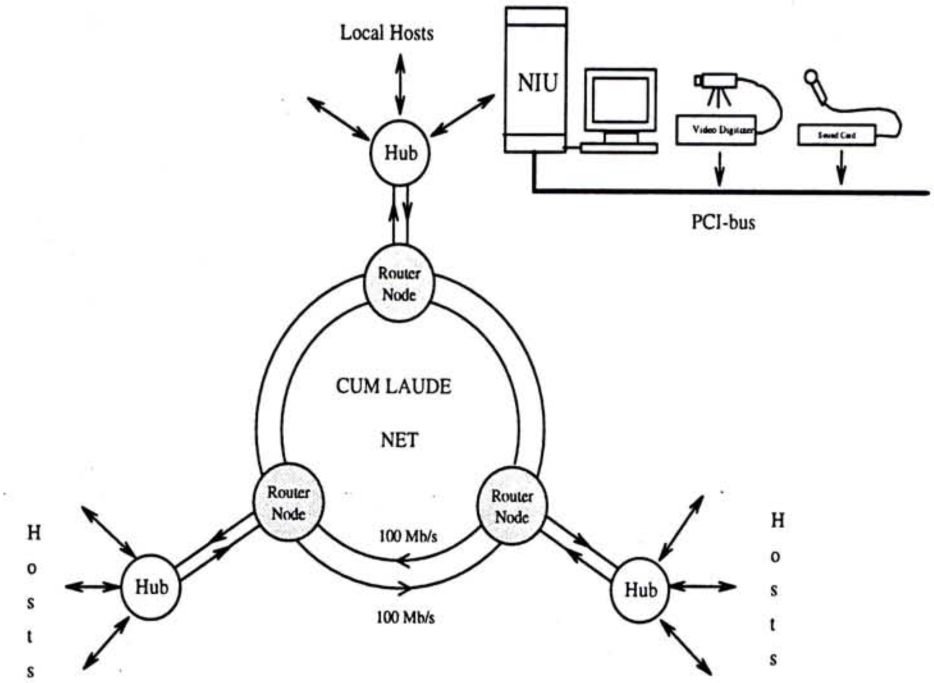


Figure 2.20: Configuration of network prototype under deployment in the Chinese University of Hong Kong.

through a T-1 gateway, thus allowing CUM LAUDE NET users to call up any telephone users and to send/receive voice mails or other customized services over a computer network.

The stability of the revertive restoration scheme, i.e., the FT and AH algorithms are tested by making artificial failures (e.g. disconnect the links or turn down the power of the node) on the network. It is found that various types of traffic, including video, audio and data being transmitted across the network is not much affected. When the failures are removed, the network heals itself and returns to normal working state.

2.7 Technical Problems Encountered

In the development of the fault-tolerant and auto-healing algorithms for the CUM LAUDE NET, a number of technical problems have been encountered. Some are solved and some are still under investigation. For example, an annoying problem is encountered in link failure detection: when a cable cut has occurred in a fiber link, stray lights will go into the fiber and cause random signals to be received by the receiver. These random signals have a finite probability of getting a bit pattern matched with one of the control signal we used in the fault management and restoration protocols. Errors thus may occur. Two remedies have been devised to solve the problem. One uses the method of repetition to distinguish a real control signal from a random signal, i.e., the signal is considered valid if it is received consecutively for a certain number of times (we use this approach instead of using a longer bit pattern in the packet header because the hardware being built limits the header length). However, this approach significantly increases the delay in protocol handling and slows down the restoration time. Another means to cope with the problem is: de-activate the receiver for a certain period of time when unrecognized signal is continuously received. This approach may be a bit better than the previous one, but it may cause significant delay for the detection of the control signal indicating that the link has been recovered. This in turn slows down the auto-healing time.

Another challenging problem we have encountered is the node synchronization problem. As mentioned previously, the last step of the auto-healing algorithm is to re-initialize the affected nodes simultaneously. Both early or late initialization of a certain node will cause false-triggering of failures by the other nodes. But this is virtually impossible since there exist non-zero time delay in the node signalling process. Hence, the algorithm has to provide some temporal tolerance in the 'simultaneous' re-initialization. Under-estimation of this time-tolerance will lead to asynchronized re-initialization and over-estimation of the time-tolerance will lead to slow recovery time. It has to be very careful in choosing an optimal value for this parameter. Moreover, the optimal value of time-tolerance depends on the component response time, fiber link transmission delay, buffering delay at node and many other parameters. Thus it takes us quite a lot of time to tune this parameter to an optimal value

2.8 Chapter Summary and Future Development

We have described the design and prototyping effort of the CUM LAUDE NET. Since we are still in the learning phase, many design decisions may still be changed or revised. In fact, the current network architecture, MAC protocols, restoration scheme, and network interface design have been revised many times.

After the completion of first phase work, we are now moving to the second phase construction of the CUM LAUDE NET. Researches on the topics related to the high-speed network interface, multimedia operating system design, integrated network management, high-speed transceiver and network protocols have been carried out already. We hope that a prototype for Level-2 Gb/s backbone will be available soon.

Also included in this chapter is the distributed fault-tolerant and auto-healing algorithms. The former allows fast recovery from failures while the latter permits hot replacement of faulty components. They together provide a revertible restoration scheme for dual-ring networks. The performance of the algorithms are studied by both simulation and some prelimenary measurements. It is found that dual-ring networks under the algorithms exhibit satisfactory performance against different reliability measures(communicative probability, survivability and average reachability). Also, it is measured in the prototype that the algorithms can be completed in milli-seconds, which is transparent to most existing network applications. In Chapter 4, a restoration algorithm built from the experience we learnt here will be introduced. It extends the range of application from dual-ring networks to arbitrary topology networks.

Chapter 3

A Simple Experimental Network Management Software - NETMAN

3.1 Introduction to NETMAN

NETMAN is an experimental network management software written for the purpose of monitoring and controlling our high-speed multimedia network prototype - CUM LAUDE NET. A Motif-like graphics-based management interface is provided for easier manipulation. In addition, a variety of detailed views and window displays are available for real-time visibility of network status. The graphical user interface (GUI) of the software is written primarily in a powerful scripting language called Tcl, together with its supporting X-Window System Toolkit (TK). The main body of the management software is written in both C and Tcl/Tk for better run-time efficiency. A snap shot of NETMAN's menu page is being shown in Figure 3.1.

In order to extend the functions of NETMAN to all TCP/IP-based networks and to support the Simple Network Management Protocol (SNMP), another Tcl-based network management language called Scotty is used. A number of features and functions such as auto-topology (a function that enables a node to scan and build a detail map of an IP network) is written in this language. These functions largely extend the range of application of NETMAN and enable centralized monitoring of TCP/IP networks from a single network management station.

In designing the NETMAN, we focus mainly on its commercial value rather than anything else. It is because we want to package the NETMAN and CUM LAUDE NET into a single product that can provide a complete network solution for both LAN and WAN environment. In order to achieve the goal, we have put our main efforts in optimizing their compatability and interoperability, the coverage of the NETMAN's function and its user interface design.

3.2 Network Management Basics

3.2.1 The Level of Management Protocols

Originally, many WANs included management protocols as a part of their link level protocols. If a packet switch began misbehaving, the network manager could instruct a neighboring packet switch to send it a special control packet. This control packet caused the switch to suspend normal operation and respond to commands from the manager. The manager could interrogate the packet

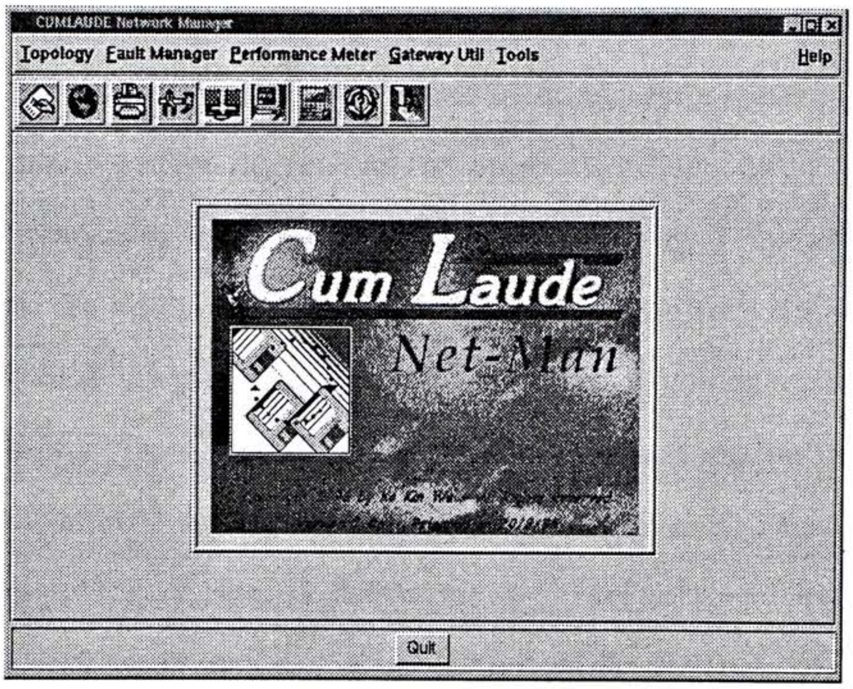


Figure 3.1: A snap shot of the NETMAN main menu page

switch to identify problems, examine or change routes, test one of the communication interfaces, or reboot the switch. Once managers repaired the problem, they could instruct the switch to resume normal operations. Because management tools were part of the lowest level protocol, managers were often able to control switches even if higher level protocols failed.

Unlike a homogeneous wide area network, a TCP/IP internet does not have a single link level protocol. Instead, the internet consists of multiple physical networks interconnected by IP routers. As a result, Internet management differs from traditional network management. First, a single manager can control heterogeneous routers. Second, the controlled entities may not share a common link level protocol. Third, the set of machines a manager controls may lie at arbitrary points in an internet. In particular, a manager may need to control one or more machines that do not attach to the same physical network as the manager's computer. Thus, it may not be possible for a manager to communicate with machines being controlled unless the management software uses protocols that provide end-to-end connectivity across an Internet. As a consequence, protocols for internet management operate at the application level and communicate using TCP/IP transport-level protocols. Designing internet management software to operate at the application level can mask out hardware differences between different networks. It also allows a single set of management protocol to run on all network elements, which means a uniform protocol can be used. Of course, building management software at the application level also has disadvantages. Unless the operating system, IP Software, and transport protocol software work correctly, the manager may not be able to contact the network elements(e.g., routers, hosts).

3.2.2 Architecture Model

Despite the potential disadvantages, having TCP/IP management software operate at the application level has worked well in practice. The most significant advantage of placing network management protocols at a high level becomes apparent when one considers a large internet, where a manager's computer does not need to attach directly all physical networks that contain managed entities. Figure 3.2 shows an example of the architecture.

As the figure shows, each participating host or router runs a server program. Technically, the server is called a *management agent.* A manager invokes client software on the local host computer and specifies an agent with which it communicates. After the client contacts the agent, it sends queries to obtain information or it sends commands to change conditions in the router. Of course, not all routers in a large internet fall under a single manager. Most managers

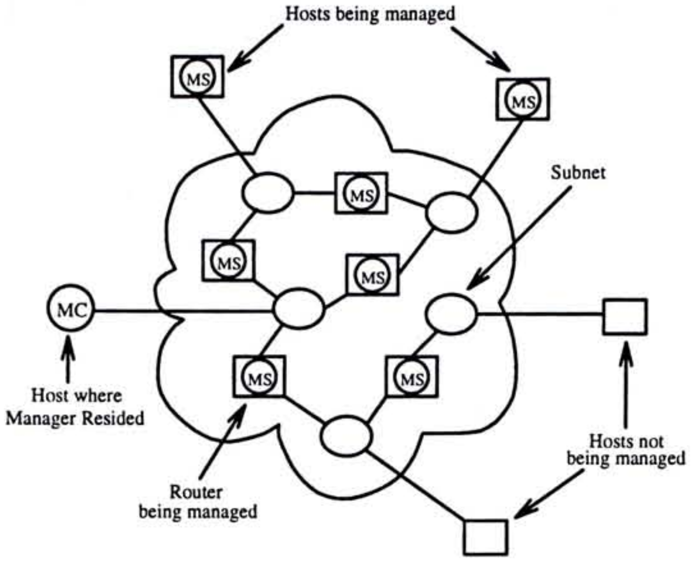


Figure 3.2: An example of a common network management model. A network manager invokes a network management client (MC) software that contacts management server (MS) software on routers or other hosts throughout the network.

only control a few routers at their local sites.

Internet management software uses an authentication mechanism to ensure only authorized managers can access or control a particular router. Some management protocols support multiple levels of authorization, allowing a manager specific privileges on each router. For example, a specific router could be configured to allow several managers to obtain information while only allowing a selected subset of them to change information or control the router.

3.2.3 TCP/IP Network Management Protocol Architecture

TCP/IP network [1] management protocols divide the management problem into two parts and specify separate standards for each part. The first part concerns communication of information. A protocol specifies how client software running on a manager's host communicates with an agent. The protocol defines the format and meaning of messages that the clients and servers exchange as well as the form of names and addresses. The second part concerns the data being managed. A protocol specifies which data items a router must keep as well as the name of each data item and the syntax used to express the name.

As such, the current network management framework for TCP/IP-based internets consists of: Structure and Identification of Management Information for TCP/IP-based internets, RFC 1155 [30], which describes how managed objects contained in the MIB are defined; Management Information Base for Network Management of TCP/IP-based internets: MIB-II, which describes the managed objects contained in the MIB (and supersedes RFC 1165); and, the Simple Network Management Protocol, RFC 1098 [17], which defines the protocol used to manage these objects.

[1] Technically, there is a distinction between internet management protocols and network management protocols. Historically, however, TCP/IP internet management protocols are known as network management protocols; we will follow the accepted terminology.

3.2.4 A Standard Network Management Protocol On Internet - SNMP

The current standard TCP/IP network management protocol is the Simple Network Management Protocol (SNMP). A second version known has SNMPv2 has been approved by the Internet Architecture Board (IAB). This new version adds new capabilities, including stronger security.

As reported in RFC 1052, IAB [2] Recommendations for the Development of Internet Network Management Standards [19], a two-prong strategy for network management of TCP/IP-based internets was undertaken. In the short-term, the SNMP was to be used to manage nodes in the Internet community. In the long-term, the use of OSI network management framework was to be examined. This strategy was quite successful in the short-term: Internet-based network management technology was fielded, by both the research and commercial communities, within a few months. As a result of this, portions of the Internet community became network manageable in a timely fashion.

As reported in RFC 1052, IAB Recommendations for the Development of Internet Network Management Standards [19], the Internet Activities Board has directed the Internet Engineering Task Force (IETF) to create two new working groups in the area of network management. One group was charged with the further specification and definition of elements to be included in the Management Information Base (MIB). The other was charged with defining the modifications to the Simple Network Management Protocol (SNMP) to accommodate the

[2] IAB stands for Internet Architecture Board. It provides the focus and coordination for much of the research and development and development underlying the TCP/IP protocols, and guides the evolution of the Internet.

short-term needs of the network vendor and operations communities, and to align with the output of the MIB working group. The output of the SNMP Extensions working group is RFC1157, which incorporates changes to the initial SNMP definition [18] required to attain alignment with the output of the MIB working group.

3.2.5 A Standard For Managed Information

A network element being managed must keep control and status information that the manager can access. For example, a router keeps statistics on the status of its network interfaces, incoming and outgoing traffic, dropped datagrams, and error messages generated. Although it allows a manager to access these statistics, SNMP does not specify exactly which data can be accessed. Instead, a separate standard specifies the details. Known as a Management Information Base (MIB), the standard specifies the data items a host or router must keep and the operations allowed on each. For example, the MIB specifies that IP software must keep a count of all octets that arrive over each network interface, and it specifies that network management software can only read those values. The MIB for TCP/IP divides management information into eight categories as Table 3.1 shows. The choice of categories is important because identifiers used to specify items include a code for the category.

The definition of MIB is always kept independent of the network management protocol. This provides advantages for both vendors and users. A vendor can include SNMP agent software in a product such as a router, with the guarantee that the software will continue to adhere to the standard after new MIB items are defined. A customer can use the same network management client software

MIB Category	Includes Information About
system	The host or router operating system
interfaces	Individual network interfaces
addr. trans.	Address Translation (e.g., ARP mappings)
ip	Internet Protocol software
icmp	Internet Control Message Protocol software
tcp	Transmission Control Protocol software
udp	User Datagram Protocol software
egp	Exterior Gateway Protocol software

Table 3.1: Categories of information in the MIB. The category is encoded in the identifier used to specify an object.

to manage multiple routers that have different versions of a MIB. Of course, a router that does not have new MIB items cannot provide the information in those items. However, because all routers use the same language for communications, they can all parse a query and either provide the requested information or send an error message explaining that they do not have the requested item.

3.3 The CUM LAUDE Network Management Protocol Suite (CNMPS)

Early network management protocols use a large number of commands to perform management functions. For example, they may use commands to: reboot the system, add or delete routes, disable or enable a particular network interface. The main disadvantages of building management protocols around commands arise from the resulting complexity and synchronization problem(e.g., to reboot several machines at the same time). The set of network management protocol proposed for the CUM LAUDE NET are called CUM LAUDE Network Management Protocol Suite (CNMPS). It is a trimmed down version of SNMP specially customized to adapt the CUM LAUDE NET architecture. Instead of defining a large set of commands, CNMPS casts all operations in a *fetch-store* paradigm. Conceptually, the CNMPS contains only two commands that allow a manager to fetch a value from a data item or store a value into a data item. All other operations are defined as side-effects of these two operations. For example, although CNMPS does not have an explicit reboot operation, an equivalent operation can be defined by declaring a data item that gives the time until the next reboot and allowing the manager to assign the item a value (including zero). This approach brings in a number of advantages such as stability, simplicity and flexibility. CNMPS is stable because its definition remains fixed, even though new data items are added to the MIB and new operations are defined as side-effects of storing into those items. CNMPS is simple to implement, understand, and debug because it avoids the complexity of having special cases for each command. Finally, CNMPS is especially flexible because it can accommodate arbitrary commands in an elegant framework.

As Table 3.2 shows, CNMPS offers a total of four operations. These operations are designed based on the *fetch-store paradigm* we described above. Operations *fetch*, *store* and *reply* provide the basic fetch and store operations (as well as replies to those operations). The *trap* operation allows managers to program servers to send information when an event occurs. for example, an CNMPS server (or equivalently, a router-node) can be programmed to send a manager a *trap* message whenever one of the attached networks becomes unusable (i.e., an interface or link goes down). At the time of this writeup, a number of data items have been defined in our MIB and are tabulated in Table 3.3. They provide the most basic information and control of routers and hosts in the

Command	Meaning
fetch	Fetch a value from a specific variable
reply	Reply to a fetch operation
store	Store a value in a specific variable
trap	Reply triggered by an event

Table 3.2: The set of possible CNMPS operations.

CUM LAUDE MIB Variable	Category	Meaning
sysUpTime	system	Time since last reboot
ifNumber	interfaces	Number of network interfaces
ifMtu	interfaces	MTU for a particular interfaces
sysRebootTime	system	Time remaining before next reboot
nodeId	Router	The ID of the node assigned during initialization
ringaRx	Router	Status of the receiver in Ring-A
ringaTx	Router	Status of the transmitter in Ring-A
ringbRx	Router	Status of the receiver in Ring-B
ringbTx	Router	Status of the receiver in Ring-B
routerResetTime	Time remaining before next router reset	

Table 3.3: The set of MIB variables currently defined for the CNMPS.

CUM LAUDE NET. Implementation and testing of these network management functions are still in progress and modifications are very likely to be made for better and more complete control of network elements.

3.3.1 The Architecture

Implicit in the CNMPS architectural model is a collection of network management stations and network elements. A Network management station is defined as the station where the CNMPS client software resides. It executes management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, hubs, router-nodes and the like, which have management agents (CNMPS servers) responsible for performing the network management functions requested by the network management stations. Figure

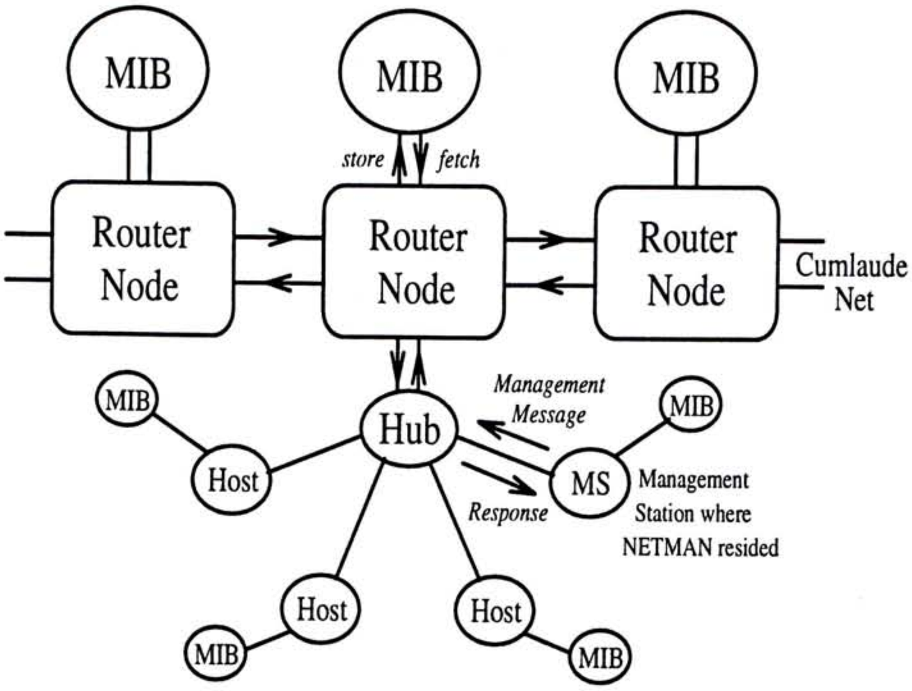


Figure 3.3: The architectural model used in the network management protocol for CUM LAUDE NET.

3.3 shows the architectural model described above. The CNMPS is used to communicate management information between the network management stations and the agents in the network elements.

3.3.2 Goals of the CNMPS

The primary goal of CNMPS is to explicitly minimizes the number and complexity of management functions realized by the management agent itself. This goal is attractive in at least three respects:

- The development cost for management agent software necessary to support the protocol is accordingly reduced.
- The degree of management function that is remotely supported is accordingly increased, thereby admitting fullest use of network resources in the

management task and imposing the fewest possible restrictions on the form and sophistication of management tools.

- Simplified sets of management functions are easily understood and used by developers of network management tools.

A second goal of the protocol is that the functional paradigm for monitoring and control be sufficiently extensible to accommodate additional, possibly unanticipated aspects of network operation and management.

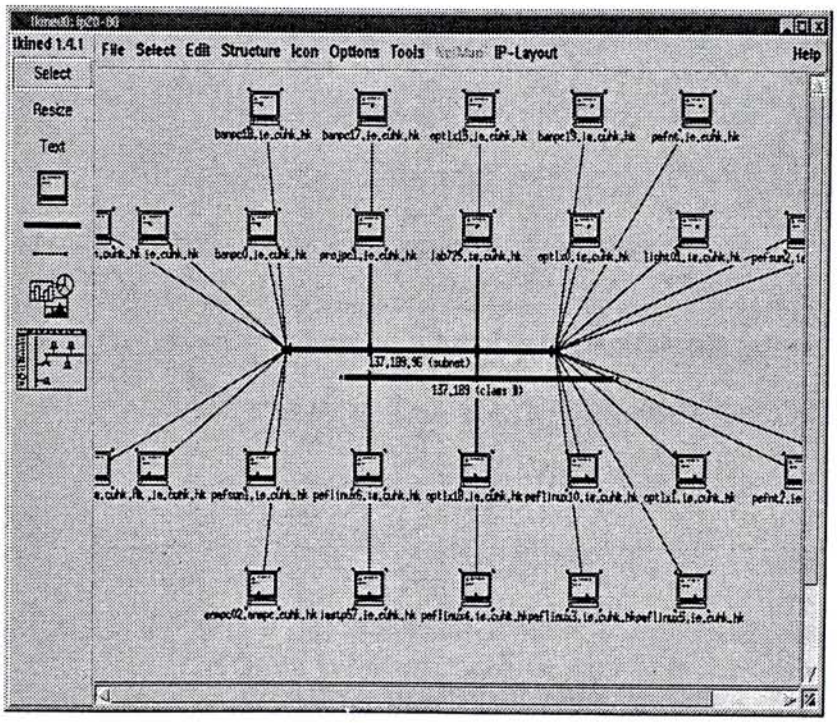


Figure 3.4: A segment of IP network with address ranging from 137.189.97.20 to 137.189.97.80 being explored by the Autotopology function of NETMAN.

A third goal is that the protocol suite should be, as much as possible, independent of the architecture and mechanisms of particular hosts or particular routers.

3.4 Highlights of NETMAN

Some enhancements have been made on NETMAN to increase its network management power. These new features include:

- **Autotopology** - This feature automatically "discovers" the logical and topological relations of all major network devices such as routers, gateways, end stations and displays them in a real-time map of the routed network. Arbitrary-sized segments of TCP/IP network can be explored easily by this function. A snap shot of a certain IP network being explored is shown in Figure 3.4. In the figure, A segment of IP network with address ranging from 137.189.97.20 to 137.189.97.80 has been explored. The result is being displayed graphically in a compact window. The topological relations between hosts, routers and subnets are also shown.

- **Real-time Traffic Analysis** - This feature offers a dynamic, real-time analysis of the network traffic. The total traffic exists in the network are broken down according to the type (e.g., IP, UDP, TCP or ICMP) or protocols (e.g., IEEE 802.3, Novell, DEC, IEEE 802.5). Their contributions to the network utilization are plotted in charts. A snap shot showing the traffic breakdown of a departmental local area network is shown in Figure 3.5.

- **Fault Alarming and Configuration Management** - Basic fault and configuration monitoring functions are implemented in NETMAN by the use of a Tcl network management extension known as Scotty and its toolkits. From this network vantage point, we can perform the following basic fault monitoring and configuration functions.

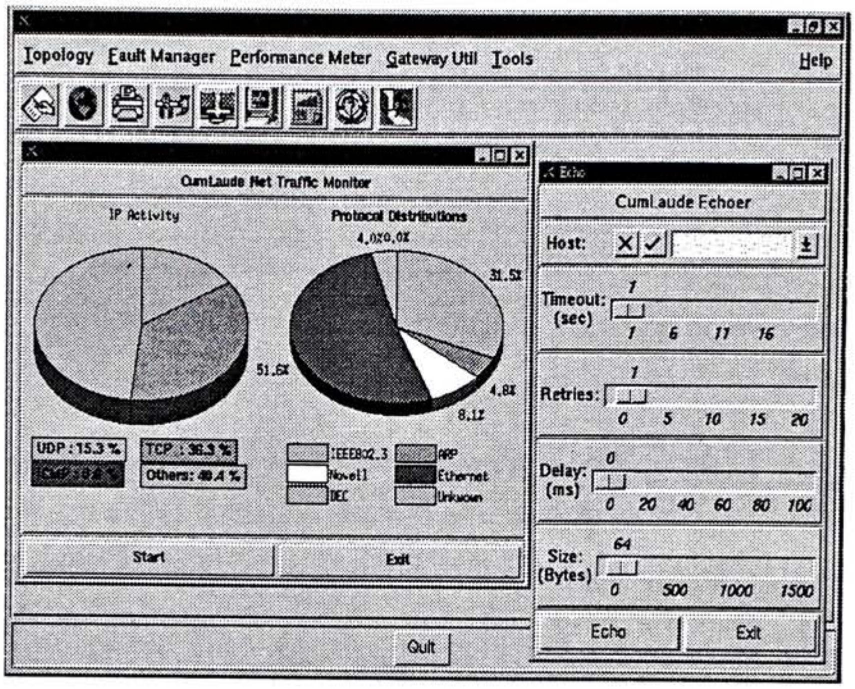


Figure 3.5: The traffic breakdown of a departmental local area network (A Ethernet).

  - Report faults within the network under surveillance.
  - Group a number of host machines or other network components to form a subnet and monitors it as a single entity.
  - Reconfiguration of network by the Autotopology function described earlier to reflect latest change.

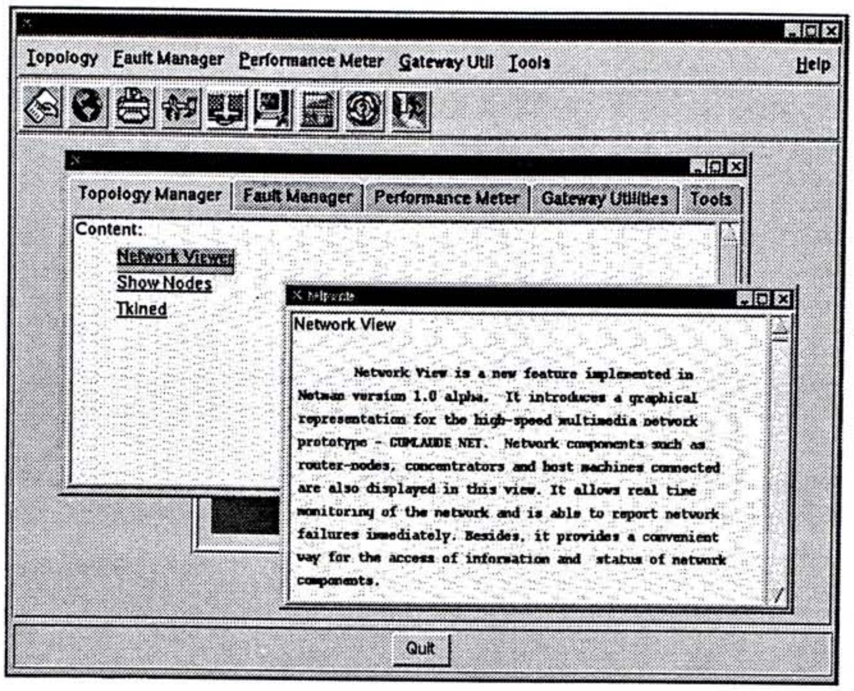
- **Complete On-line Help** - A complete set of on-line help menu is designed to address majority of users' questions. Explanations to all commands, functions and trouble-shooting techniques are well-documented in this menu. A snap shot of the help screen is shown in Figure 3.6.

Menu Item	Functional Descriptions
Topology	Network topology map displaying
Fault Manager	Fault alarming and management
Performance Meter	Performance metering
Gateway Utility	Gateway monitoring and utilities
Tools	Common network tools
Help	On-line help

Table 3.4: A brief description of the functions of the items in the main menu.

3.5 Functional Descriptions of NETMAN

The functions and features provided by NETMAN are divided, according to their uses and purposes, into six categories. Each category of functions is represented by a menu item in the main menu bar (Figure 3.1). A complete list of all available menu items and their functional descriptions are summarized in Table 3.4. As described in the table, functions grouped in the menu item 'Topology' are used to provide network topology information such as network connectivity and status of components (for example, routers, gateways and host machines) to the system administrators. Functions grouped in the menu item 'Fault manager' provides fault alarming and real-time network surveillance. 'Performance meter' contains tools to collect network statistics such as network capacity utilization and to provide network traffic breakdown and analysis. 'Gateway Utility' provides utilities to retrieve data from our gateway software and conducts real-time gateway traffic and performance analysis. 'Tools' collected some handy network tools which provides convenient functions such as host reachability test, screen capture/printout and host information gathering. 'Help' provides on-line help and explanations to different items and options in the menu bar.



Network View

Network View is a new feature implemented in Netman version 1.0 alpha. It introduces a graphical representation for the high-speed multimedia network prototype - CUMLAUDE NET. Network components such as router-nodes, concentrators and host machines connected are also displayed in this view. It allows real time monitoring of the network and is able to report network failures immediately. Besides, it provides a convenient way for the access of information and status of network components.

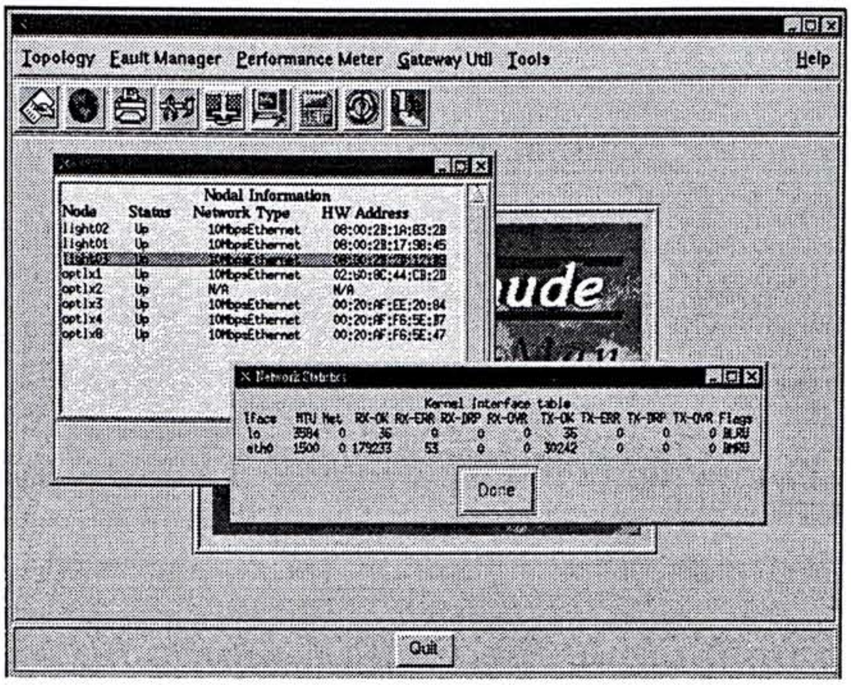
Figure 3.6: The NETMAN help screen.

3.5.1 Topology Menu

- **Network View** graphically displays the network topology map. It provides a simple and visual way for network administrators to monitor the network. Faults can easily be located as the faulty components will be displayed in red and blink.

- **Nodal View** provides information that may be useful for network administrators who need to know on a node-by-node basis what is happening on a certain segment of the overall network. This information can also be obtained by double-clicking the desired object in the network view. A snap shot is being shown in Figure 3.7.

- **Tkined Network Editor** invokes an interactive editor, as shown in Figure 3.8, for creating and maintaining network maps. A number our network monitoring functions are run under this environment.



Nodal Information

Node	Status	Network Type	HW Address
light02	Up	10MbpsEthernet	08:00:2B:1A:83:2B
light01	Up	10MbpsEthernet	08:00:2B:17:9B:45
light03	Up	10MbpsEthernet	[illegible]
optlx1	Up	10MbpsEthernet	02:60:8C:44:CB:2B
optlx2	Up	N/A	N/A
optlx3	Up	10MbpsEthernet	00:20:AF:EE:20:84
optlx4	Up	10MbpsEthernet	00:20:AF:F6:5E:B7
optlx8	Up	10MbpsEthernet	00:20:AF:F6:5E:47

Kernel Interface table

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flags
lo	3584	0	36	0	0	0	36	0	0	0	BLRU
eth0	1500	0	179233	53	0	0	30242	0	0	0	BMRU

Figure 3.7: The NETMAN nodal view

3.5.2 Fault Manager Menu

- **Network Interface Status** gives the latest report of the interface status of the network. A list of all the network interfaces detected will be shown.

- **Diagnostics** provides a failure report of the network under investigation. A list of failed components and the possible causes will be shown in the diagnostics screen.

3.5.3 Performance Meter Menu

- **Network Activity** offers a dynamic, real-time analysis of the network traffic under the surveillance of NETMAN. The total traffic exists in the network are broken down according to the type (e.g., IP, UDP, TCP or ICMP) or protocols (e.g., IEEE 802.3, Novell, DEC, IEEE 802.5). Their contributions to the network utilization are plotted in charts. A snap

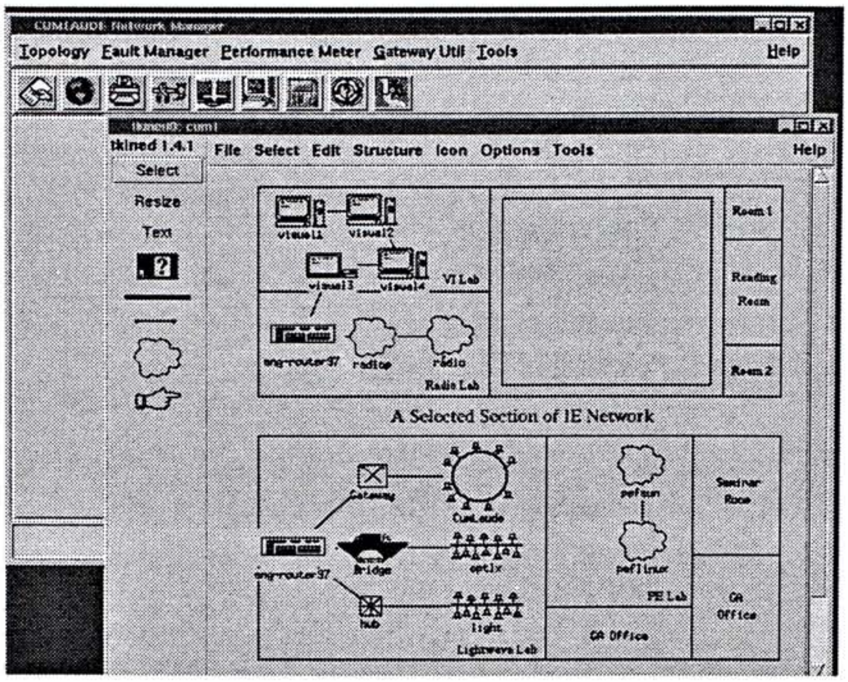
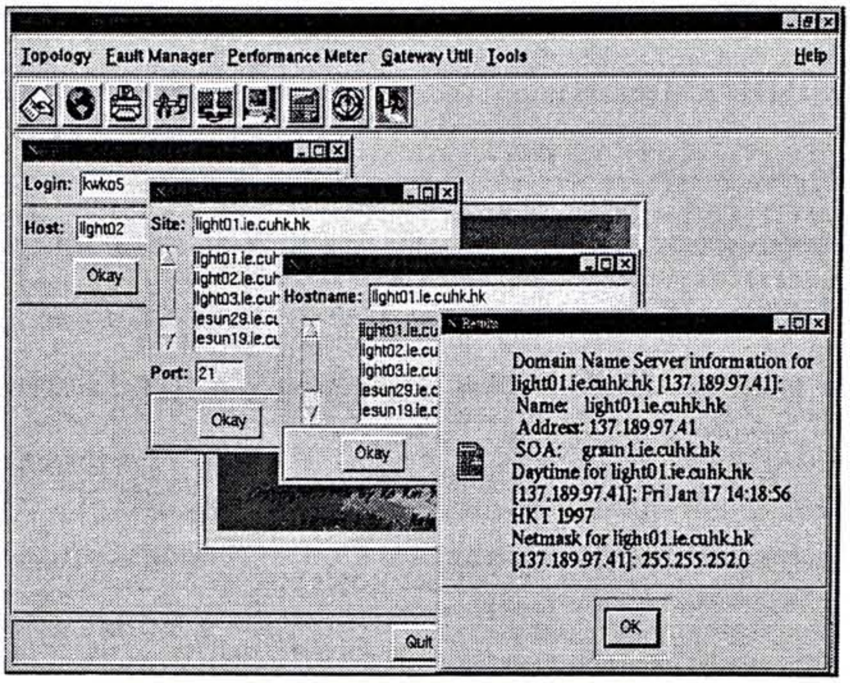


Figure 3.8: The interactive network editor invoked from NETMAN.

shot showing the traffic breakdown of a departmental local area network is shown in Figure 3.5.

- **Node Activity** displays the cumulative packet and frame statistics for the network. Measures such as the number of errors detected, number of packets dropped and the number of collisions will be shown. It also shows the number of multicast packets and broadcast packets that have been received on the network.

- **Recorder** displays a graph that represents the number of successful and unsuccessful bytes that have been received on the network since you selected the Recorder menu selection. A number of statistic measures based on them will also be displayed.



Login: kwko5

Host: light02

Site: light01.ie.cuhk.hk

light01.ie.cuh
light02.ie.cuh
light03.ie.cuh
iesun29.ie.cu
iesun19.ie.cu

Port: 21

Hostname: light01.ie.cuhk.hk

light01.ie.cu
light02.ie.cu
light03.ie.cu
iesun29.ie.c
iesun19.ie.c

```
Domain Name Server information for
light01.ie.cuhk.hk [137.189.97.41]:
 Name:    light01.ie.cuhk.hk
 Address: 137.189.97.41
 SOA:     grsun1.ie.cuhk.hk
Daytime for light01.ie.cuhk.hk
[137.189.97.41]: Fri Jan 17 14:18:56
HKT 1997
Netmask for light01.ie.cuhk.hk
[137.189.97.41]: 255.255.252.0
```

Figure 3.9: A number of convenient tools is provided in NETMAN, some sample results are being shown in this figure.

3.5.4 Gateway Utility Menu

- **Traffic Monitor** gives real-time information on the statistics gathered from the gateway. One can monitor the gateway performance by a graphical means.

3.5.5 Tools Menu

The tools menu, as displayed in Figure 3.9, allows you to use some small and convenient tools to manage the network or to do other network administrative works.

- **Host Info** summarizes the information such as daytime, SOA, netmask for a certain host and displays them in a window.

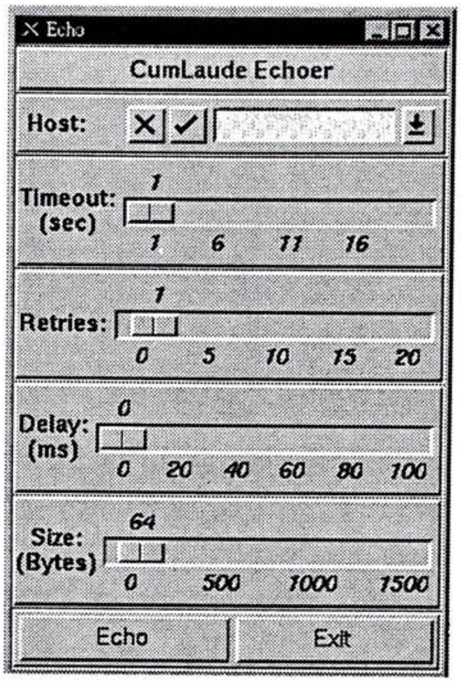
- **Echo Test** tests a network component to ascertain if it can be reached on the network. When selected, a box appears with a field to input the IP address or the name of the network element. Various parameters used in the echo test such as time-out and retries can be adjusted by horizontal scales. The profile of the box is being shown in Figure 3.10. A successful echo causes a result screen appears, with the message 'host responded' and other accounting statistics.
- **Reset node** allows the network administrator to reset a certain malfunctioning router.
- **Telnet, Ftp and Rlogin** are familiar functions. However, they are still included in NETMAN as they provide convenient means for the users to access other hosts.
- **Print screen** captures the screen currently displayed and allows the users to print it on a destined printer or save it in a file.

3.5.6 Help Menu

The help menu, as Figure 3.6 shows, provides on-line help for the users. Detail descriptions for the commands and functions, as well as the trouble-shooting techniques for problems are included.

3.6 Chapter Summary

This chapter presented an experimental network management software called NETMAN for the monitoring and control of the CUM LAUDE NET. Network



Timeout: (sec) 1

Retries: 1

Delay: (ms) 0

Size: (Bytes) 64

Figure 3.10: The CUM LAUDE echo test tool.

management basics including protocol architecture, SNMP and MIB concepts have been discussed. After that, the proposed network management protocol suite to be used in NETMAN, which is designed based on the Internet network management model, is introduced. Finally, detail descriptions of functions provided in NETMAN are documented.

Chapter 4

A Local Map Based (LMB) Self-Healing Scheme for Arbitrary Topology Networks

4.1 Introduction

With the extensive deployment of fiber optic transmission systems, there is an increasingly urgent need for strategies that can protect the data from various kinds of network failures. A number of interesting techniques and strategies have been adopted to provide better restoration time and ratio. These techniques can be roughly divided into three main categories, namely, dedicated facility restoration, dynamic facility restoration and integrated restoration. Generally, dedicated facility restoration such as Automatic Protection Switching (APS) offers the fastest restoration, but it requires the largest dedicated capacity. Dynamic facility restoration provides slower, but more flexible and spare-capacity efficient restoration than the former one. It may depend on different control schemes (centralized [11], distributed [7] [28] [42] or hybrid [5]), routing plans (pre-planned [5] [21] or dynamic [7] [42]) or restoration levels (path [15] or line [7] [8] [42]), and may further be divided into different categories. Other than these two main restoration approaches, there is an integrated approach which combines both dedicated and dynamic facility restoration. It protects high priority traffic by dedicated switching facility and low priority traffic by Digital Cross-connect Systems (DCS) based alternative routing.

The advent of high-speed DCS has had a radical impact on the design of self-healing algorithms. It allows rapid network reconfiguration to adapt failures. Many algorithms utilizing this flexibility offered by DCS have been proposed [5] [8] [16] [21] [28] [42]. They all led to drastic improvement in both restoration time and ratio compared to the conventional protection technique based on diversity routing. Restorations in sub-second range were reported in [5] [7].

In this chapter, a Local Map Based (LMB) Self-Healing Scheme utilizing the DCS is proposed. It is a path-level distributed restoration scheme that can provide fast and efficient network restoration. The scheme, as its name implies, facilitates restoration based on the information available on a local map in each individual DCS node. The local map is a small-sized table containing vital network information such as node connectivity and working/spare capacity of links of the region surrounding the node. It is built during network initialization, and is updated after each occurrence of failure. The size or scope of a local map is measured by the number of hops it spans, or the term 'level' which we will use in the rest of the chapter. With the local map stored in each DCS node, a simple searching and sorting on the information available in the local map is sufficient to provide a set of locally optimized paths for restoration. These processes can be easily handled by a DCS controller and thus allow the algorithm to be executed at the hardware level.

In real practice, it is intuitive for restoration algorithms to utilize spare resources around the failure for restoration. This leads to faster recovery and better utilization of spare resources. Besides, unnecessary traffic loading due to poor alternate path is reduced. Therefore, in most cases, a local map of reasonable size is sufficient to provide sub-optimal restoration for failures. From a study of some existing Telecom networks, we found that a local map of level 2-3 is sufficient to provide restoration for link failures and a local map of level 3-4 is sufficient to provide restoration for node failures. Simulations also reveal that the LMB scheme has a restoration efficiency approaching centralized methods while restoration time is even faster than fully distributed methods.

4.2 An Overview of Existing DCS-Based Restoration Algorithms

As stated in previous section, DCS-based restoration algorithms can roughly be classified into 3 categories, namely, centralized, distributed and hybrid algorithm. In centralized restoration, only one central controller is responsible for the whole network to decide how re-routing should be done. It can provide optimal solution for recovery but restoration time is rather slow as the signaling delays between the central controller and other nodes are large. There is also the danger of catastrophic failure should the central controller fail. On contrary, distributed restoration allows each node to decide how re-rerouting should be done according to the failure situations. The earliest distributed algorithm is Grover's self-healing algorithm - SHN [42], which first introduced the sender-chooser restoration concepts. Later, a number of different distributed restoration algorithm has been proposed for DCS-based network. They included FITNESS [8], Komine's algorithm [15], Two Prong [7], Double Search-Self-Healing [16], RREACT [9], and MRS[33]. SHN, FITNESS and RREACT all use an sender-chooser approach similar to Grover's, i.e., sender broadcasts help messages to the whole network. When these messages reach chooser, a technique known as reverse-linking is used to establish the restoration path. Komine's algorithm employed a path-level sender-chooser approach with multi-destination flooding and is capable of handling all single/multiple link and node failures.

The Two-Prong and Double Search Self-Healing algorithm differ from the above in the sense that they have used bi-directional searching technique in establishing new restoration path. MRS is an integrated restoration algorithm. It consists of two parallel processes for both pre-planned and real time restoration and is capable of handling multiple-link and node failures.

NETSPAR [5] from Bellcore is the first hybrid restoration algorithm proposed. It restores the network by a distributive protocol to determine the failure type and uses a pre-computed plan for restoration. If the network configuration after failure cannot be handled by the pre-computed plans, a centralized restoration algorithm is invoked. NRNN [10] is another example of hybrid algorithm which uses the concept of neural network to adapt failures.

4.3 The Network Model and Assumptions

Figure 4.1 shows the network model being used by the LMB Self-Healing Scheme. In the model, a network consists of nodes and links only. A node is uniquely identified by an ID assigned during network initialization and has a Digital Cross-Connect System (DCS) that is capable of switching traffic between different channels connected to it. A link is identified by its originating and destined nodes and has two numbers associated with it, i.e., the number of working channels (or bandwidth used if the bandwidth/channel is non-uniform) and the number of spare channels assigned for restoration. The bandwidth contained in each channel can be DS3 or STS-1, which depends on the type of network in use (Asynchronous or Synchronous). It is assumed that each node can detect the failure of links connected directly to it and has all necessary information (i.e., working capacity, spare capacity, originating node and destined node) about its adjacent links.

If a link failure occurs, all the channels contained in the failed link are lost. When a node failure occurs, all node directly connected to the failure node will detect a link failure at the same time. The scheme also assumes some failure detection scheme such as APS is already in use. Therefore it will be activated after the failure detection is completed.

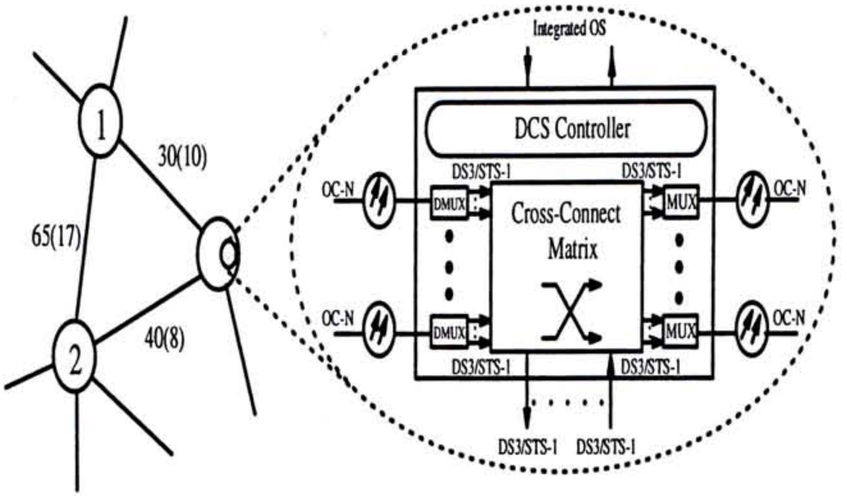


Figure 4.1: A generalized network model for the LMB network restoration scheme.

4.4 Basics of the LMB Scheme

4.4.1 Restoration Concepts

The idea behind the LMB scheme is based on two common characteristics observed from existing restoration algorithms. First, all algorithms tend to use spare resources nearest to the location of failure to achieve restoration because this would lead to faster restoration and shorter restoration paths. Second, alternate path finding process can usually be completed in a localized region given that sufficient spare resources is provided, or the size of the region is sufficiently large. These two observations directly lead to the conclusion that a good restoration should be, or can be made localized. Therefore, if one possesses correct network information within the affected localized region, restoration with efficiency approaching centralized method is obtainable. It should also be noted that the restoration time can be very fast as the time-consuming alternate path seeking process (as in most distributed restoration algorithms) is replaced by a simple path searching algorithm over the network information available for the region.

4.4.2 Terminology

As mentioned in the introductory section, a *local map* is a small-sized table stored in a node which contains vital network information such as node connectivity and working/spare capacity of links of the surrounding region. The size or scope of a local map is measured by the number of hops it spans, called *level.* Figure 4.2 shows a local map with a maximum level of three, i.e., a level-3 local map. In the figure, each node in the local map is marked with the level it belongs to. This number actually represents the minimum number of hops a node needed to reach the reference node i inside the region. Those nodes farthest apart from the reference node i in the local map are denoted as *boundary nodes.* The basic construction unit of a local map is an entity known as *node-map.* It is a small block of data containing all information associated with a node. This includes the node ID, the working/spare capacity and the source/destination associated with each link terminating at that node. One can also interpret a node-map as a level-1 local map. A collection of node-maps in a restricted region of network forms a local map.

In the LMB restoration scheme, nodes detecting failures are called *activated nodes* and the one finally initiates a LMB restoration process is called the *Master.* The arbitration rules to determine which one among the activated nodes will become the Master are as follows: For link failures, the node with a smaller node ID in the two affected nodes is regarded as the Master and will be responsible for the restoration. For node failures, the node with the smallest node ID in the

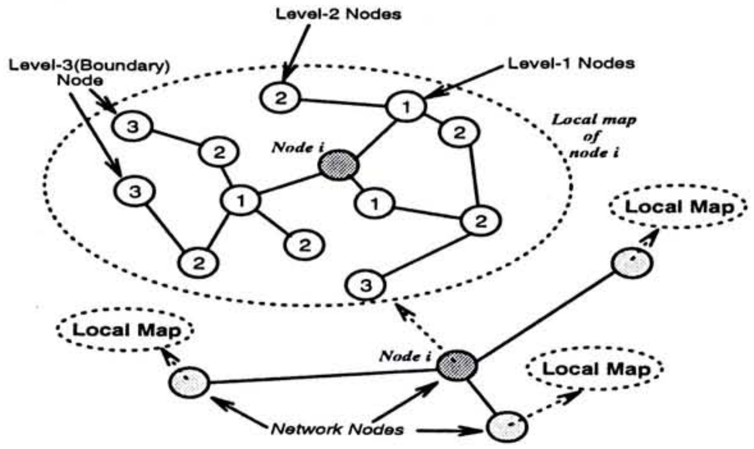


Figure 4.2: An example illustrating the leveling concept of the LMB scheme. In the figure, a level-3 local map of a certain network node i is being shown. The number marked inside each node represents the level it belongs to.

affected group of nodes will serve as the Master. For all possible combinations of link/node failures that will happen in the network, only two types of scenario will occur, namely, non-overlapping and overlapping LMB restoration. Figure 4.3(a) shows a network under two single link failures which leads to two non-overlapping LMB restorations. In this case, each Master restores the failure according to its own local map and suffers no contention of spare resources. Figure 4.3(b) shows a network under a single link and node failure which leads to two overlapping LMB restoration. This leads to contention of spare resources between two Master nodes, which in turn reduces the restoration ratio if the spare resources are insufficient.

4.4.3 Algorithm Parameters

There are two important parameters in the LMB restoration scheme, namely, the *local map level* and *search depth* within the local map. The former represents the scope or the size of the local map. The latter represents the maximum hop

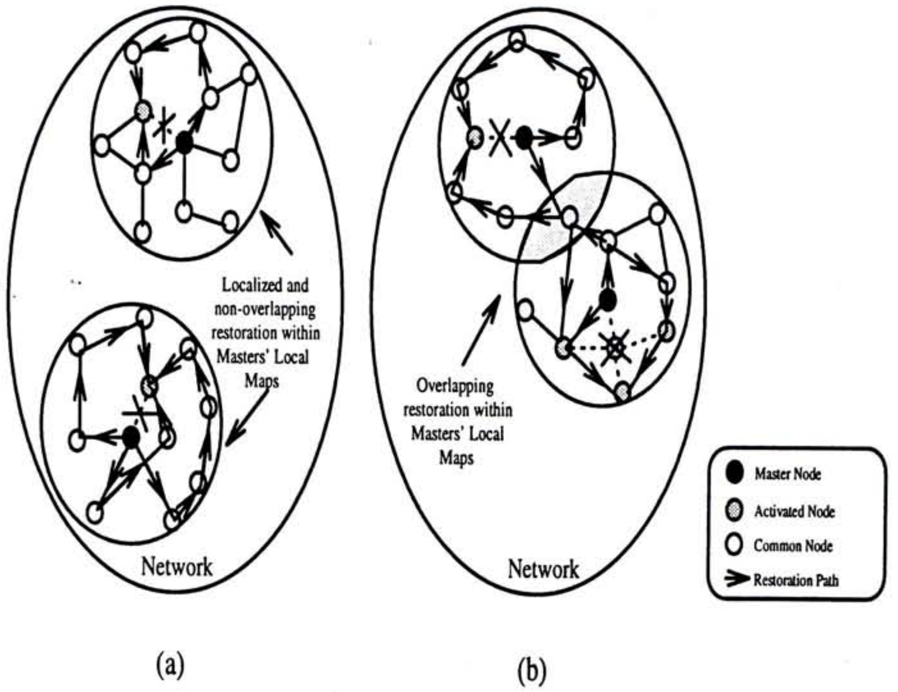


Figure 4.3: (a) Two single link failures occurred apart in the network and cause no overlapping of local maps between Master nodes. (b) A link and node failure occurred in network and cause overlapping of local maps between Master nodes, which in turn lead to contention of spare resources.

of alternate paths to be explored inside the local map. A more detail analysis of their effects and meaning in the restoration will be presented later.

4.5 Performance Assessments

In order to characterize the performance of the LMB scheme and facilitate comparison with other restoration algorithms, some measures have been defined. They include:

- Average Restoration time (ART): The time required by a restoration algorithm to achieve maximum level of restoration, averages over a sample or all possible combinations of a certain failure type. Theoretically, the further the ART below the 2-second restoration objective [20], the lower the probability a call or connection will be dropped.

- Average Restoration Ratio (ARR): The proportion of restored channels or virtual circuits relative to the number of failed ones, averages over a sample or all possible combinations of a failure type. The ideal value of ARR is 1, which corresponds to the situation of all lost channels or virtual circuits are restored. ARR, together with ART, are the most important performance metric in characterizing a restoration algorithm.

- Cumulative Restoration Ratio (CRR): It is defined as the ratio of cumulative number of channels (or bandwidth) restored to the total number of channels lost as a function of time. The accumulation is done for all possible failures belong to the failure type interested. The gradient of the CRR curve for a restoration algorithm reflects its rate of restoration.

- Average Message Volume (AMV): It refers to the number of restoration messages generated in the restoration process, averages over a sample or all possible combinations of a failure type. It is desirable that AMV be as small as possible. This can alleviate the queuing delay experienced in the nodes and lead to faster restoration.

- Average Message Complexity (AMC): It is the average length of the restoration messages, averages over a sample or all possible combinations of a failure type. Algorithms having larger AMC suffer longer transmission queuing and node processing delays.

- Spare Resource Utilization (SRU): It is the ratio of spare channels reserved to spare channels used for restoration. Higher value of this metric means better utilization of spare resource, which in turn reduces the spare resource required in networks.

- Range of Application (RA): It refers to the types of failures a certain restoration algorithm can manage. A number of proposed algorithms concentrate only on single link failures while some can handle multiple link/node failures as well. Definitely, algorithms provide a wider range of protection is more attractive.

Among the metrics described above, some provide performance evaluation on a discrete basis and some on a continuous basis. Discrete estimators have the merit of easy interpretation while continuous estimator may provide more information. It is difficult for an algorithm to outperform another in each aspect described above. Rather, an algorithm may trade in some of its ability in a certain aspect in exchange of better performance in another.

4.6 The LMB Network Restoration Scheme

4.6.1 Initialization - Local Map Building

In the construction of a level-n local map, each node in the network will selective broadcast an initialization message containing information of its node-map to the network. The hop limit of this message will be set to the maximum local map level n. To effectively suppress the number of messages generated in the broadcast, the following selective re-broadcast mechanism is used: Each initialization message contains a field holding the IDs of the nodes that have received the broadcast message in the last two broadcasts. Any node receiving an initialization message will re-broadcast it to the surrounding nodes that have not received the message in the last two broadcasts. When an initialization message is found to be duplicated, it will be discarded instead of re-broadcasted. The above message re-broadcasting mechanism reduces the message volume from (C^n) to approximately (nC), where C is the average connectivity of the network and n is the maximum level of the local map. When a node receives a new initialization message, it will update its local map.

4.6.2 The LMB Restoration Messages Set

There are totally four phases in the LMB scheme, namely, the local map update phase, update acknowledgment phase, restoration/confirmation phase and cancellation phase. One unique type of messages will be used in each phase. The information contained in the different types of messages and their corresponding storage capacity required are shown in Table 4.1. Some fields mentioned in the table are not commonly seen and may need further explanations. The field boundary-nodes holds the IDs of the boundary nodes of an activated node's local map. The broadcasted nodes are used to distinguish which of the neighboring nodes should be broadcasted.

4.6.3 Phase I - Local Map Update Phase

In this phase, the nodes which detect failures (the activated nodes) will broadcast a local map update message to other nodes within its local map.

Fields	Length/(byte)	Use By (Message Type)
Message Type	1	1 2 3 4
Source Node ID	A	1 2 3 4
Destination Node ID	A	- 2 - -
Failed Link ID	2 x A	1 2 3 4
Hop Count	1	1 2 3 4
Boundary Node IDs	A x M	1 - - -
Broadcasted Node IDs	A x N	1 - - -
Requested Bandwidth	1	- - 3 -
Time Information	2	1 - - -
Route	A x R	- - 3 -
Released Bandwidth	1	- - - 4

Table 4.1: A table showing the information contained in different types of messages used by the LMB scheme. Type 1 for Map Update Message, Type 2 for Map Update Acknowledgment, Type 3 for Confirmation Message, Type 4 for Cancellation Message. The variables A, M, N and R stand for the address size to represent a node, number of boundary nodes, number of nodes broadcasted and number of nodes in a restoration path respectively. In most literature, A is assumed to be 1 or 2. However, if the mapping of network address and node ID is also taken into considerations, this value will be quite different. For example, A will be assigned to 4 in the Internet Protocol.

When a node receives a new local map update message, the message is rebroadcasted to all neighboring nodes that have not yet receive it (those nodes not specified in the field broadcasted nodes). A local map update message will be discarded when it reaches one of the boundary nodes or when its hop count reaches zero. In this way, the local maps of the activated nodes are updated.

4.6.4 Phase II - Update Acknowledgment Phase

When a new map update message reaches one of the boundary nodes as specified in the message, its content will be recorded in a table and a map update acknowledgment will be sent back to the message's source (the activated node sending out the message). After all boundary nodes have responded or the waiting time has exceeded a predetermined time-out, the activated node will decide, according to the arbitration rules mentioned previously, whether to initiate the restoration phase by itself (becomes the Master).

4.6.5 Phase III - Restoration and Confirmation Phase

After the local map update and update acknowledgment phases, the Master will obtain an updated local map which correctly reflects the real situation of the network. It will then start searching for the best alternate paths to restore the lost channels in its updated local map. The paths found are then sorted, in descending order of priority, according to two criteria: (1) Number of hops and (2) Spare bandwidth available. A list of possible alternate paths is hence constructed. Confirmation messages based on this set of possible paths are sent until all lost channels are restored. This phase actually consists of two separate phases. However, as no message generation is involved in the restoration process (alternate-path seeking), the restoration phase is combined with the confirmation phase.

4.6.6 Phase IV - Cancellation Phase

In the case of multiple failures, the Masters' local maps may overlap with each other. This may lead to contention of spare capacity in links and hence some requested bandwidth in the confirmation messages cannot be accommodated. Under this situation, a cancellation message specifying that the bandwidth cannot be accommodated is sent back to the corresponding Master node. Cross-connection is made along the back-track to release the reserved bandwidth for later trials. When a Master receives this message, it will pick an untried path(s) with bandwidth equal or larger than the failed bandwidth in the alternate path list and send a new confirmation message.

4.6.7 Re-Initialization

After the whole restoration process is completed, the information of local map stored in the nodes within the affected areas may be outdated. Therefore a re-initialization process (similar to the initialization, but only nodes within the activated nodes' local maps will be involved) is invoked to update them.

4.6.8 Path Route Monitoring

In order to facilitate path level restoration, each node must have route information on the paths passing through it. Here we use a similar approach as proposed by Komine[15]. Some space is reserved in the path overhead of packets to hold the route information, which is a record of node IDs of last two (or more, which depends on the scale of failure to be protected) nodes in their paths. Every packet passing through a node will have its oldest entry of path record being replaced by the new node ID. Thus, every node receives continuous and real-time route information which is crucial in node failure restoration.

4.7 Performance Evaluation

4.7.1 The Testbeds

We have built a network simulation system to verify the validity of the LMB scheme and observe its performance. It is written in an event driven simulation language - SIMSCRIPT II.5. In the system, each node has an common message buffer for incoming links and a transmission buffer for each outgoing link. The scheme has been tested on two mesh network models, as shown in Figure 4.4. Figure 4.4(a) shows the first testbed - Testbed-A we used (often it is referred to as the New Jersey LATA network [7] [8] [42]). It is a sample network based on a real LATA Network in the United States and is being used as a testbed in Yang's FITNESS algorithm in 1988. A number of restoration algorithms published later also use it as a testbed. Testbed-A consists of 11 nodes and 23 links. The parameters and assumptions used in this network are as follows: (1) node processing time for each incoming message is 10 ms, (2) message transmission time for each outgoing message is 10 ms, (3) link propagation delay is 0.5 ms and, (4) all message received by a node are queued at a FIFO before being processed.

Testbed-B, as shown in Figure 4.4(b), is a testbed used in [14]. It consists of 15 nodes and 28 links and is designed to guarantee 100% restoration for any single link failure. The number of working and spare STS-1s on each link are represented by the figures associated with the link. A different set of parameters and assumptions has been used in this network model. They are: (1) a dedicated 64 kb/s signalling channel is used for communication between the nodes, (2) node processing time for each incoming message is 3 ms, (3) transmission delay between interface port and the DCS controller is 1 ms, (4) there is no cross-connection delay and propagation delay is well below 1 ms.

Different parameters and assumptions are used in the above testbeds because

Statistics	Testbed-A	Testbed-B
Number of Nodes	11	15
Number of Links	23	28
Average Connectivity	4.18	3.73
Spare/Working Ratio	0.50	0.57

Table 4.2: A table showing the statistics collected from the testbeds.

we want to evaluate the performance of the LMB restoration shceme under different physical constraints, and to simulate realistic situations for it. For example, the dedicated 64 kb/s channel assumed in Testbed-B has simulated the situation in SONET, which usually uses fixed-rate dedicated channels to convey network management signals. The absence of this constraint in Testbed-A reflects the relaxed bandwidth limitation for transmitting OAM signals in ATM networks. Also, the parameters used in the two testbeds have their own unique significance. For instance, different node processing times mean different computaional powers of the DCSs and different propagation delays mean different geometric dimensions of the networks.

Some topological statistics of these two testbeds are tabulated in Table 4.2. Testbed-A provides higher average connectivity while Testbed-B has larger spare/working capacity ratio. These statistics is important when the performance of restoration is compared across different networks.

4.7.2 Simulation Results

The effects of changing algorithm parameters including local map level and search depth have been studied based on the testbeds described in the last subsection. Figure 4.5(a) & (b) show plots of the Cumulative Restoration Ratio

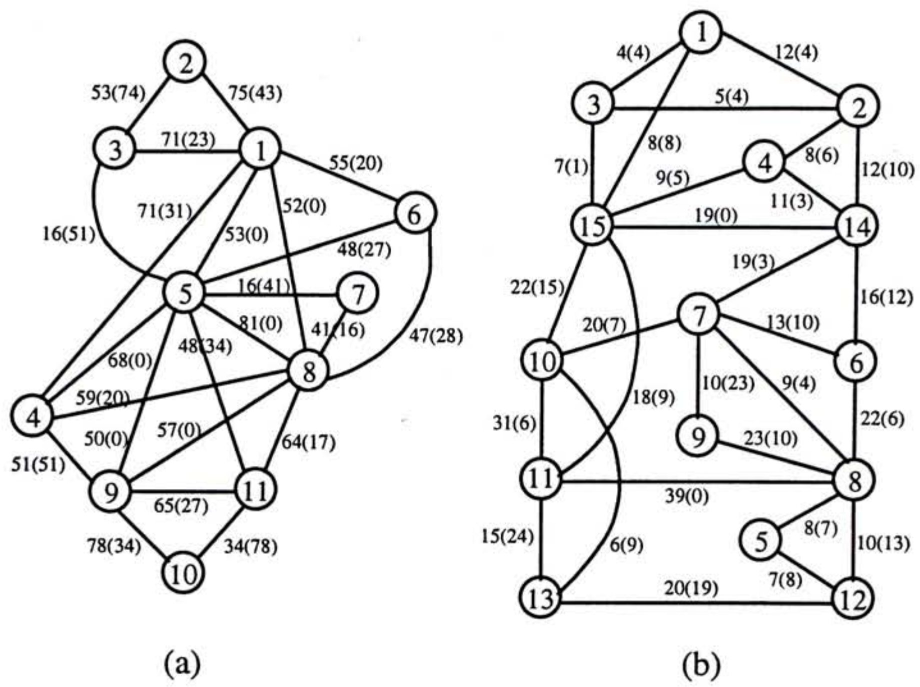


Figure 4.4: (a) Testbed-A, also known as New Jersey LATA Network, consists of 11 nodes and 23 links. (b) Testbed-B consists of 15 nodes and 28 links.

(CRR) for different levels of local map. It is observed that the final restoration ratio increases as the level of local map increases, while the rate of restoration is approximately the same. The result is intuitive because a local map of higher level allows the algorithm to explore more feasible paths and hence improve the ultimate restoration ratio. In general, a local-map with lower level (or equivalently, smaller size) results in earlier restoration as less update time is required. The only exception happens when the local map level equals one. This is because the node on the other side of the failed link is always unable to receive the map update message delivered by the Master and thus no update acknowledgment will be sent back. Consequently, the Master has to wait until timeout before it can start executing the LMB scheme as not all boundary nodes will acknowledge its map update message. This explains the late initiation of the LMB scheme for level-1 local maps.

Figure 4.6(a) & (b) illustrated the effects of changing search depth in the LMB scheme. Similar to an increase in the local map level, an increase in search depth results in better restoration ratio. This is because a larger search depth allows the algorithm to seek out more possible paths for restoration. Although the effect of increasing local map level and search depth is similar, there are still some conceptual differences between them. The former represents an increase in scope for path searching while the latter resembles a more thorough search in a constrained scope. Also, an increase in search depth does not require extra memory to store the local maps.

In order to justify the previous performance claims of the LMB scheme, a comparison with Two Prong, FITNESS and SHN based on various measures has been made. The comparison is done on the Testbed-A and the result is shown in Table 4.3. For all measures including average restoration time (ART), average restoration ratio (ARR), spare resource utilization (SRU) and average message volume (AMV), the LMB scheme exhibits significant improvements. Another comparison of the LMB scheme with some common restoration algorithms is presented in Figure 4.7. In the figure, the performance of the algorithms in the Testbed-B is investigated by plotting their CRRs as a function of time. SHN, Komine's algorithm and the Level-4 LMB scheme fully restored lost channels for all single link failures. NETSPAR, NETRATS and Level-3 LMB scheme can restore 98%, 98% and 97% of all lost channels in the 28 possible single link failures respectively. Among the algorithms compared, LMB scheme and NETSPAR provide the fastest rate of restoration (lost channel restored/unit time), followed by SHN, NETRATS and Komine's. The completion times of Level-2, Level-3, Level-4 LMB scheme, NETSPAR, SHN, NETRATS and Komine's are 160, 220,

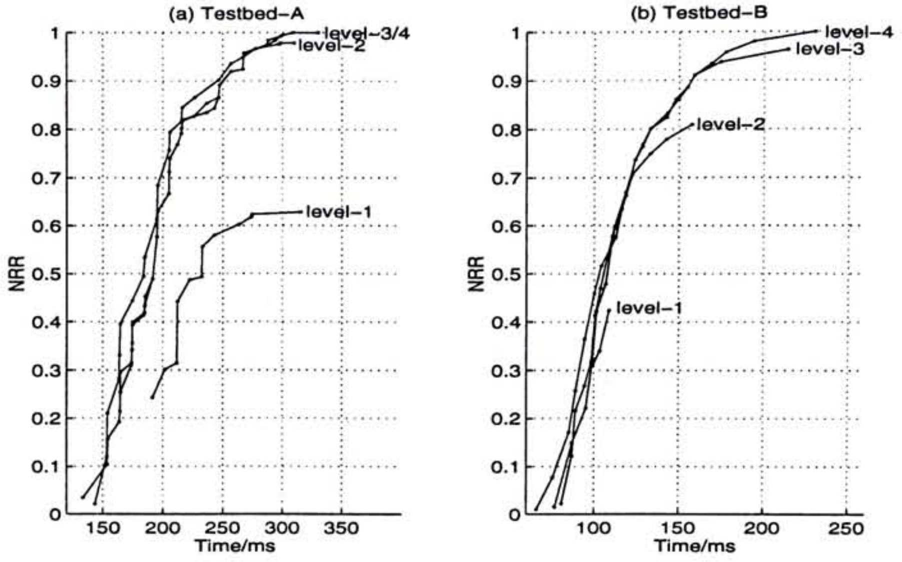


Figure 4.5: The Effect of changing the level of local map on (a) New Jersey LATA Network. (b) The testbed Network.

Restoration Algorithms	Performance Metrics			
	ART/ms	ARR	SRU	AMV
FINESS	963	100%	33.8%	107
RREACT	435	100%	27.7%	93
Two Prong	267	100%	28.5%	116
LMB(lvl-2)	221	98%	8.7%	38
LMB(lvl-3)	231	100%	8.8%	40

Table 4.3: A performance comparison for FITNESS, RREACT, Two Prong and LMB Self-Healing Scheme on the New Jersey Network

230, 370, 850, 900 and 1300 ms respectively, but the scale in the figure shows up to 500 ms only.

4.7.3 Storage Requirements

The amount of space required for the storage of local map in a node is estimated by the following model. First we assumed each node is represented by a node ID, which takes only one byte(suitable for network with no more than 255 nodes). Each link is uniquely specified by its source and destination node

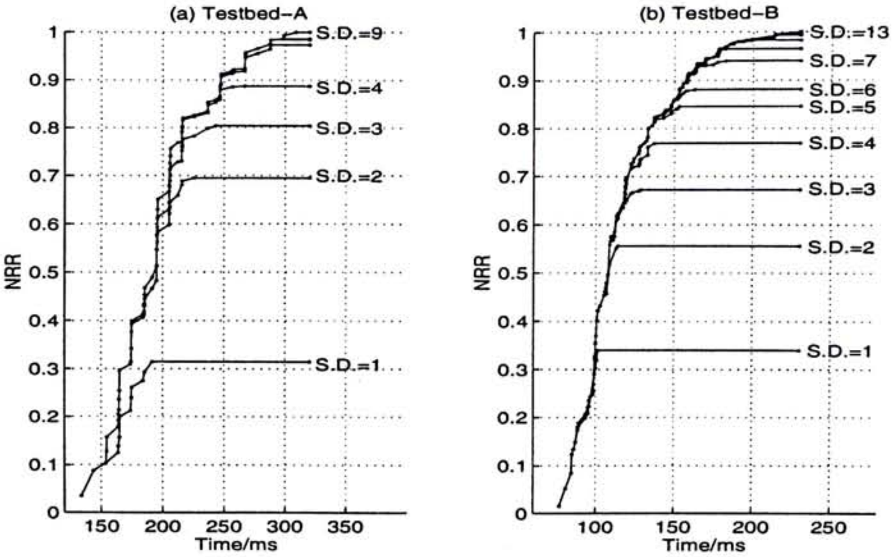


Figure 4.6: Effect of changing search depth on restoration time and ratio. (a) Testbed-A with local map level equals 3. (b) Testbed-B with local map level equals 4.

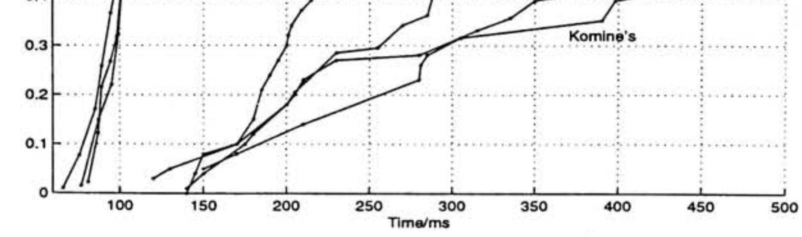


Figure 4.7: CRR, as a function of time, illustrating the restoration performance for different algorithms under single-link failure in Testbed-B.

Level of local map	Maximum memory required / bytes
1	85
2	289
3	901
4	2.74K
5	10.95K

Table 4.4: A performance comparison for FITNESS, RREACT, Two Prong and LMB Self-Healing Scheme on Testbed-A

ID, thus it takes 2-byte to represent a link. It is further assume that the spare and working capacity of link is measured in unit of STS-1, therefore a 1-byte integer is sufficient to represent a capacity of up to 13 Gb/s. The estimation of memory required by a level-n local map, M_n, can be easily obtained by multiplying the number of node-maps inside with the memory required to store a single node-map ($M_{node-map}$). Mathematically, it is given by:

$$M_n = \begin{cases} (1+C) \times M_{node-map} & n = 1 \\ C \times M_{n-1} & n > 1 \end{cases} \tag{4.1}$$

where n is the local map level and C is the average connectivity of network. For example, by assuming the average connectivity is 4, the amount of memory required at each DCS node to hold the local map for different levels is given in Table 4.4. It is found that only 11 Kbytes of memory is needed for a local map with size as large as 5. It is small compared with NETSPAR, which may need a few hundred kilobytes to several meagbytes of memory to provide similar level of failure protection.

4.8 The LMB Scheme on ATM and SONET environment

ATM/SONET has been well recognized as the most promising technology for backbone and switching network. Thus considerations have been made since the designing phase of the LMB scheme to efficiently adapt to the environment. Referring to Figure 4.8, a LMB restoration message is passed to the ATM adaptation layer and gets segmented into ATM cells. These cells are then mapped onto the SONET payload. In most cases, a restoration message can be transmitted in one or two frame times. The transmission delay is expected to be less than one millisecond. This allows effective transmission of the restoration message over the ATM/SONET environment.

The LMB restoration message transfer can also be done in SONET/SDH environment. The approach uses the overheads provided in SONET for path, section and line embedded operation channels (EOC) to provide a means for the communication between nodes. Data Communications Channel (DCC) of bandwidth 192 kb/s, 576 kb/s and 64kb/s is available for section (D1-D3), line (D4-D12) and path (F2) level equipment to exchange network management message and maintenance information [1]. The transfer mechanism is shown in Figure 4.9, in which some specific bytes are used to carry the LMB restoration message. This approach suffers from a larger message transfer delay as the transmission rate of these dedicated channels is relatively slow. However, it is not required to handle the ATM Adaptation Layer (AAL) protocol in this case. In order to accelerate the message transfer process, the SONET overhead bytes (Z-bytes) reserved can also be used.

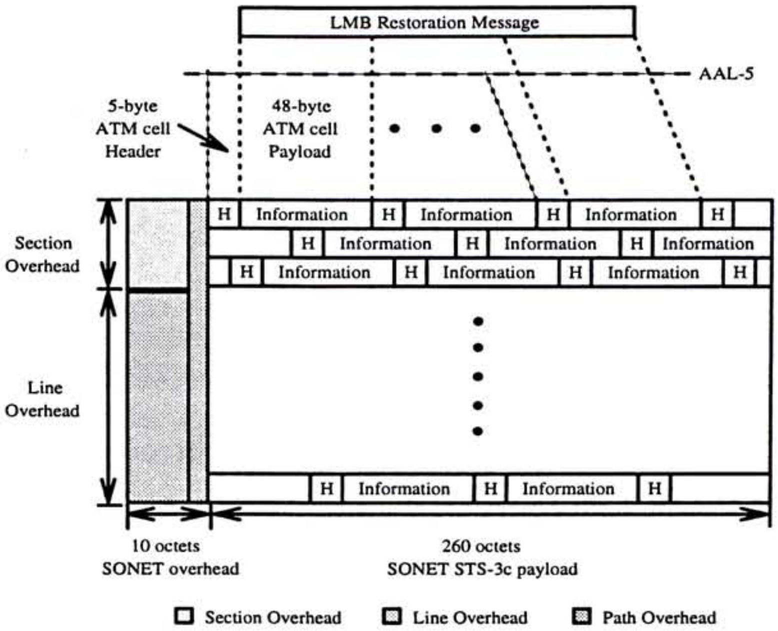


Figure 4.8: The mapping of restoration message to ATM cells on SONET STS-3c.

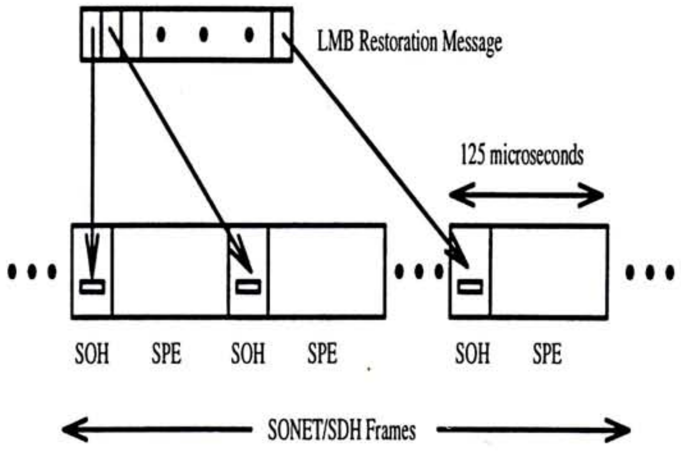


Figure 4.9: LMB restoration message transfer on SONET/SDH environment.

4.9 Future Work

The LMB restoration scheme, when implemented in SONET environment, can protect hard failures caused by cable cut or soft failures caused by signal degradation (BER exceeding predefined threshold). However, faults originating from software failure or malfunctioning hardware cannot be protected. The limitation arose because there is no reliable method to detect these errors. Thus, our next step is to find a reliable way to identify these failures and combines this with the LMB scheme to provide general protection for both software and hardware failures in networks.

Another important issue in restoration algorithm design is the algorithm robustness. Intermittent failures that last for a short time interval and consecutive failures inside a localized region are failures that cannot be covered by the LMB algorithm. Two revertive restoration and failure locking techniques that can be used to manage these failures are under investigation.

4.10 Chapter Summary

This chapter described a Local Map Based (LMB) Self-Healing Scheme for fast and efficient restoration of networks with arbitrary topology. It has the advantages of fast recovery time, high spare resource utilization ratio, suppressed restoration message volume and localized restoration. These claims are justified by comparing the performance of the LMB scheme with most existing DCS-based restoration algorithms. The LMB scheme also provides a wide range of failure coverage including link/node and multiple failures and is able to work on SONET and ATM environment. Besides, the scheme is flexible as it can control the level of restoration and restoration time to meet practical needs by adjusting the algorithm parameters (local map level and search depth).

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Network fault-tolerance is one of the most important issues in high-speed fiber network design. In this thesis, various topics in this area have been addressed. In Chapter 1, we overview the recent development and the related issues in survivable network planning. We point out that the survivability planning of a network can be separated into fault prevention, fault detection and fault adaptation phases. Different concerns are raised in different phases. Customer needs and rapid technologically advances are the main driving forces of the development of survivable networks.

In Chapter 2, we have presented the design and implementation of a scalable fault-tolerant multimedia network - CUMALAUDE NET. It is a hierarchical dual-ring network designed to serve both LANs and WANs. Level-1 CUM LAUDE NET has been implemented and is able to transmit multimedia data at a rate of 100 Mbps. The packet format currently being used is compatible with TCP/IP. It has been connected to public telephone network through a T-1 gateway and can support voice and video data. To minimize the probability of packet loss in bursty traffic and provide efficient medium access, a priority transmission control scheme has been implemented. Besides, congestion control is used to avoid traffic from the hub overflowing the receive and transmit buffers in the router-nodes. The network protocol used in CUM LAUDE is called ACTA (Adaptive Cycle Tunable Access). Fair access is achieved by limiting the number of empty slots occupied in each cycle. The cycle length is adjusted to reduce the packet latency and to increase the overall throughput. To increase the efficiency of the protocol implementation, the ACTA protocol has been slightly modified. Instead of generating a continuous stream of empty slots, a single access control packet containing the number of empty slots for different priority levels is used. This arrangement saves significant packet processing time in the router-nodes. Finally, a revertive restoration scheme allowing fast recovery from major types of hardware failures has been devised and implemented. Restoration time on the order of milli-second is achieved. Also, the availability of network during restoration is maximized by the method of packet buffering. The performance has been tested with different network loading and the result is found to be very satisfactory.

In Chapter 3, a generalization of the revertive restoration scheme used in CUM LAUDE NET has been proposed. The scheme consists of two distributed fault-tolerant and auto-healing algorithms. They are based on the inter-communications and hand-shaking processes of adjacent nodes to exchange network status information in case of network failure. The fault-tolerant algorithm allows fast recovery from failures while the auto-healing algorithm permits hot replacement of faulty components. It also explained why high network availability can be provided to users by the algorithm during restoration. Besides, different reliability measures including communicative probability, survivability and average reachability have been defined to describe the performance of the algorithms. A three-node network prototype has been built to verify the above claims. Simulations have shown that the recovery and auto-healing processes can be completed in milli-seconds.

In Chapter 4, an experimental network management software (NETMAN) prototyped to control and monitor the CUM LAUDE NET is demonstrated. In NETMAN, a Motif-like graphics-based management interface is provided for easier manipulation. Moreover, a variety of detailed views and window displays are available for real-time visibility of network status. In order to extend the functions of NETMAN to all TCP/IP-based networks and to support the Simple Network Management Protocol (SNMP), a Tcl-based network management language called Scotty is used. New features and functions that largely extend the range of application of NETMAN have been built.

In Chapter 5, a Local Map Based (LMB) Self-Healing Scheme for fast and efficient restoration of arbitrary topology networks is proposed. It is based on the use of information available in small-sized data tables known as local maps stored in each node to restore disrupted traffic. The scheme provides a wide range of failure coverage including link/node and multiple failures and is well-adapted to SONET and ATM environment. The size of local map is scalable according to the type of failure to be protected. A study of existing Telecom networks reveals that local maps span 2 to 3 hops of network offer good link-failure coverage while local maps span 3 to 4 hops are needed for node failures.

Since the most time-consuming alternate-path seeking process in distributed restoration can be replaced by simple searching and sorting algorithms executed in individual DCS node. Couples of advantages such as fast failure recovery, high spare resource utilization, suppressed restoration message volume and localized restoration are achieved. Simulations with different sets of parameters have shown that the restoration time is well below the 2-second service restoration objective.

5.2 Future Work

As indicated in the thesis, the importance of fault-tolerance and survivability planning in high-speed network design is well recognized. Network vendors and engineers have put in large efforts to design highly survivable networks. This atmosphere gives room for researchers to develop better solutions to the problem. In general, the following research topics on network fault-tolerance and survivability planning requires further investigations.

- So far most research on network fault-tolerance is in the direction of hardware failure protection. It is desirable to devise an algorithm that can provide complete coverage for both hardware and software failures. However, software failure detection is not as easy as its hardware counterpart. It requires a more complex failure detection and identification algorithm and hence it is unlikely to be implemented in the physical or data link layers. In view of this, it is desirable to develop a path-level restoration scheme at the network layer to cover both hardware and software failures. One of the most challenging problems is to design a failure detection and identification algorithm to detect and distinguish various types of fault caused by software failures.

- Another issue that worth paying attention is the porting of the network restoration schemes from networks based on Digital Cross-connect Systems (DCSs) to Optical Cross-connect (OXC) Systems. For network with OXC switches, packets are processed optically. This largely increases the node throughput as the electrical processing bottleneck no longer exists. Therefore, it is reasonable to predict, in the foreseeable future, the role now played by the DCSs will soon be taken over by the OXCs. As a result, there is a large potential market for the OXC-based restoration algorithms. One of the implementation problems is to port the DCS-based restoration algorithms to OXC-based networks and to decide which network control scheme should be used. Currently, there are two approaches available: First, a dedicated channel is used to carry the network management information is used. Second, the OAM information is embedded in packets. The first approach has the advantage of simple implementation but the bandwidth that can be used to carry the OAM information is limited. The second approach basically imposes no limit on the bandwidth available for the OAM information, but it requires fast add/drop of the OAM information from the high-speed data stream. Todate, key technologies such as wavelength routing [24] [29] required for implementing a OXC-based survivable network are mature, therefore we believe that it is an appropriate time to drill into this research area.

Bibliography

[1] CCITT Recommendations G707, G708, G709 - Synchronous Digital Hierarchy, in Blue Book. 1989.

[2] Advanced Micro Devices, Inc. Am7968/Am7969 TAXI TM - 125 Technical Manual. 1993.

[3] Ali Zolfaghari and Fred J. Kaudel. Framework for Network Survivability Performance. *IEEE Journal on Selected Areas in Communications*, 12(1):46–51, January 1994.

[4] AT&T Technical Ref. 43801. Layer 1 In Service Digital Transmission Performance Monitoring. Technical report. Also proposed ANSI standard T1M1.3/89-091.

[5] B.A. Coan et al. Using Distributed Topology Updates and Preplanned Configurations to Achieve Trunk Network Survivability. *IEEE Trans. on Reliability*, 40:404–416, 1991.

[6] Bellcore. The role of digital cross-connect systems in transport network survivability. January 1993. Bellcore Special report SR-NWT-002514.

[7] C.E. Chow, J. Bicknell, S. McCaughey and S. Syed. A Fast Distributed Network Restoration Algorithm. *Proc. of 12th Int. Phoenix Conference on Computers and Communications*, March 1993.

[8] C.H. Yang and S. Hasegawa. FITNESS : Failure Immunisation Technology for Network Service Survivability. *IEEE Globecom 88*, pages 1549–4554, 1988. Hollywood, USA.

[9] C.H.E. Chow, S. McCaughey and S. Syed. RREACT:A Distributed Protocol for Rapid Restoration of Active Communication Trunks. *UCCS Tech Report EAS-CS-92-18*, 1992.

[10] Chia-Jiu Wang, Hong Ying Zhou and Ching-Hua Chow. Automatic Network Restoration using a Two-Level Associative Memory. *IEEE International Conference on Neural Networks*, pages 3565–3570, 1994. NRNN.

[11] D.K. Doherty, W.D. Hutcheson and K.K. Raychaudhuri. High Capacity Digital Network Management and Control. *IEEE Globecom'90*, pages 60–64, 1990.

[12] Forensic Technologies International Corporation. Hinsdale Center Office Fire Final Report: Executive Summary, Annapolis, MD. Technical report, March 1989.

[13] G.P. Agrawal. *Fiber-Optic Communication Systems.* John Wiley and Sons, 1993.

[14] H. Kobrinski and M. Azuma. Distributed Control Algorithms for Dynamic Restoration in DCS Mesh Networks: Performance Evaluation. *IEEE Globecom'93*, pages 1584–1588, 1993.

[15] H. Komine, T. Chujo et al. A Distributed Restoration Algorithm for Multiple-Link and Node Failures of Transport Networks. *IEEE Globecom 90*, pages 459–463, 1990. San Diego, USA.

[16] Hiroyuki Fujii and Noriaki Yoshikai. Restoration Message Transfer Mechanism and Restoration Characteristics of Double-Search Self-Healing ATM Network. *IEEE Journal on Selected Areas in Communications*, 12:149–158, 1994.

[17] J. Case, M. Fedor et al. Simple Network Management Protocol (SNMP). *RFC 1098*, April 1989.

[18] J. Case, M. Fedor, M. Schoffstall and J. Davin. A Simple Network Management Protocol. *Internet Engineering Task Force working note*, March 1988. Network Information Center, SRI International.

[19] J. Davin, J. Case, M. Fedor and M. Schoffstall. A Simple Gateway Monitoring Protocol. *RFC 1028*, November 1987.

[20] J. Sosnosky. Service applications for SONET DCS distributed restoration. *IEEE Journal of Selected Areas in Communications*, pages 59–68.

[21] J.E. Baker. A Distributed Link Restoration Algorithm with Robust Preplanning. *IEEE Globecom'91*, pages 306–311, 1991.

[22] Jiahnsheng Yin and Charles B. Silio, Jr. Reliability of FDDI's Dual Homing Network Architecture. *Proceedings IEEE INFOCOM'94*, 3:1382–89, June 1994.

[23] J.O. Limb and C. Flores. Description of Fasnet - A Unidirectional Local-Area Communications Network. *Bell Syst. Tech. J.*, (61):1413–1440, 1982.

[24] K.W. Cheung. Scalable, Fault-Tolerant 1-Hop Wavelength Routing Networks. *IEEE Globecom '91*, pages 34.7.1–34.7.6, 1991.

[25] K.W. Cheung. Adaptive-Cycle Tunable-Access (ACTA) Protocol: A Simple, High-Performance Protocol for Tunable-Channel Multi-Access (TCMA) Networks. *ICC'93*, pages 166–171, May 1993.

[26] K.W. Cheung and L.K. Chen and C. Su and C.T. Yeung and P.T. To. Tunable-Channel Multi-Access (TCMA) Networks: A New Class Of High-Speed Networks Suitable For Multimedia Integrated Networking. *SPIE'93 - Multigigabit Fiber Communication Systems, San Diego, CA*, July 1993.

[27] K.W. Cheung, C.T. Yeung, W.K. Lam and et al. CUM LAUDE NET - A High -Speed Multimedia Integrated Network Prototype. *1st ISMM International Conference Multimedia And Distributed Applications, Hawaii*, 1994.

[28] K.W. Ko, S.F. Lam, K.W.Cheung et al. Distributed Fault-Tolerant and Auto-Healing Algorithms on Dual-Ring Networks. *IEEE SICON'97*, April 1997.

[29] M. Ajmone Marsan, Andrea Bianco, Emilio Leonardi and Fabio Neri. Topologies for Wavelength-Routing All-Optical Networks. *IEEE/ACM Transactions on Networking*, 1(5):534–546, October 1993.

[30] M. Rose and K. McCloghrie. Structure and Identification of Management Information for TCP/IP-based internets. *RFC 1155*, May 1990.

[31] Lam Wing Kwan Ringo. An integrated broadbrand concentration/distribution network for multimedia application compatible with the hybrid fiber-coax (hfc) architecture. Master's thesis, The Chinese University of Hong Kong, June 1995.

[32] R.M. Metcalfe and D.R. Boggs. Ethernet: Distributed Packet Switching for Loacl Computer Networks. *Commun. ACM*, (19):395–404, 1976.

[33] R.S.K. CHNG, et al. A Multi-layer Restoration Strategy for Reconfigurable Networks. *IEEE Globecom'94*, pages 1872–1878, 1994. MRS.

[34] S. Liew and K. Lu. A framework for network survivability characterization. *Proc. IEEE Int. Conf. Communications (ICC)*, 1992.

[35] Shanzhi Chen, Shiduan Cheng and Junliang Chen. Survivability Strategies for the Future Chinese SDH Transport Network. *ICCT'96*, 2:26.10.1–26.10.5, May 1996.

[36] Standard IEEE 802.6. Distributed Queue Daul Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN). December 1990.

[37] T.H. Wu. A Novel Architecture for Optical Dual Homing Survivable Fiber Networks. *Proceedings of IEEE International Conferences on Communications (ICC)*, pages 309.3.1–309.3.6, April 1990. Atlanta, GA.

[38] T.H. Wu. *Introduction to Algorithms.* Artech House, May 1992.

[39] T.H. Wu, D.J. Kolar and R.H. Cardwell. High-Speed Self-Healing Ring Architecture for Future Interoffice Networks. *IEEE Globecom '89*, pages 23.1.1–23.1.7, November 1989. Atlanta, GA.

[40] Tsong-Ho Wu. Emerging Technologies for Fiber Network Survivability. *IEEE Communications Magazine*, pages 58–74, February 1995.

[41] W. Falconer. Services Assurance in Modern Telecommunications Networks. *IEEE Communications Magazine*, 28(6):32–39, June 1990.

[42] W.D. Grover. The Self-Healing Network : A Fast Distributed Restoration Technique for Network Using Digital Cross-connect Machines. *IEEE Globecom'87*, pages 1090–1095, 1987.

[43] T.H. Wu. Roles for Optical Components in Survivable Fiber Networks. *Digest of Optical Communications Conference (OFC'92)*, February 1992. CA.

[44] Y.H. Wang, L.K. Chen and K.W. Cheung. Performance of Integrated Services on A High-Speed Multimedia Network. *Globecom'95*, 2:780–784, 1995.

Appendix A

Derivation of Communicative Probability

This appendix derives the communicative probability of a dual-ring network with the FT algorithm. Consider a N-node dual-ring network with n node and l links failed, as shown in Figure A.1. The operative nodes remained can communicate if and only if they satisfy the following conditions: (1) All operative nodes(including the headnode) remained are consecutive. (2) There is no link failure in between. We may consider these two conditions separately to simplify the matter. The number of ways to put n failed nodes in a N-node dual-ring network is ${}_NC_n$. The number of ways to arrange the $(N - n)$ operative nodes as a consecutive segment containing the headnode is $(N - n)$, which can be obtained by shifting the headnode from the start of the segment, to the end of the segment. Hence, the probability that the network remained communicative even if n nodes failed is:

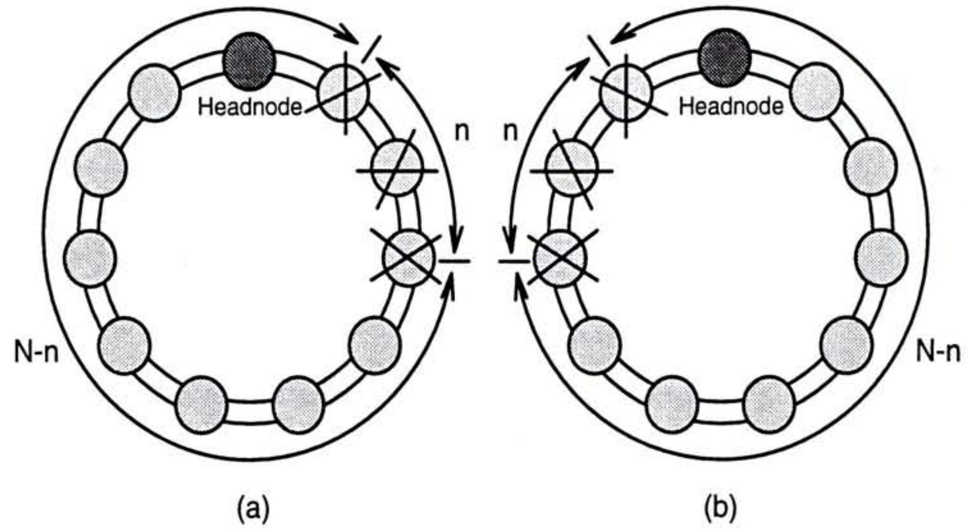


Figure A.1: A N-node dual-ring network with n nodes and l links failed. Figure (a) and (b) illustrate two typical scenarios that $N-n$ nodes remain operative after n nodes have failed.

$$P[C|n] = \frac{(N-n)}{{}_{N}C_{n}} \tag{A.1}$$

To satisfy the second condition that no link failure in between of the operative node segment, we just need to ensure that all the l-link failures are occurred within the failed-node segment. For a certain configuration of the failed nodes, there are ${}_{2n+2}C_l$ ways to put the l failed links within the n failed nodes. If we shift the whole set of failed nodes by one node, in either clockwise or counter-clockwise direction of the dual-ring, a new configuration is obtained. In this new configuration, there are still ${}_{2n+2}C_l$ ways to put the l failed links. However, ${}_{2n}C_l$ ways are repeated in these two adjacent configurations. Therefore the net number of ways to put the l failed links is $({}_{2n+2}C_l - {}_{2n}C_l)$. Multiplying this factor with the total possible configurations for the failed network segment $(N-n)$, the probability that the network remains communicative for an l-link failure becomes:

$$P[C|l] = \frac{(N-n)({}_{2n+2}C_l -{}_{2n} C_l)}{{}_{2N}C_l} \tag{A.2}$$

where the denominator ${}_{2n}C_l$ is the number of ways to have l link failures on the $2N$ links in the dual-ring. There are three special cases which cannot be described by the equations above, i.e., $(n, l) = (0, 1)$, $(0, 2 \leq l \leq N)$, $(n \neq 0, l > 2n + 2)$. Their communicative probabilities are 1, $2({}_NC_l)/({}_{2N}C_l)$, and 0 respectively.

Summarizing the results above, the communicative probability of a dual-ring, with the fault-tolerant algorithm implemented, subjected to a n-node and l-link failure is:

Dual-ring networks with fault-tolerant algorithm

$$P[C|n, l] = \begin{cases} \frac{(N-n)}{{}_NC_n} \times \frac{(N-n)({}_{2n+2}C_l - {}_{2n}C_l)}{{}_{2N}C_l} & n \neq 0, l \leq 2n+2 \\ 0 & n \neq 0, l > 2n+2 \\ \frac{2{}_NC_l}{{}_{2N}C_l} & n = 0, 2 \leq l \leq N \\ 1 & n = 0, l = 1 \end{cases} \tag{A.3}$$

Appendix B

List of Publications

1. K.W. Ko, S.F. Lam, K.W.Cheung et al. Distributed Fault-Tolerant and Auto-Healing Algorithms on Dual-Ring Networks. *IEEE SICON'97. April, 1997.*

2. Kin-Wa Ko, Sze-Fan Lam and Kwok-Wai Cheung. Fast Revertible Restoration Scheme for Dual-Ring Networks. *IEEE ISCC'97. July, 1997.*

3. Kin-Wa Ko and Kwok-Wai Cheung. A Local Map Based (LMB) Self-Healing Scheme for Arbitrary Topology Networks. *IEEE GLOBECOM'97. Nov, 1997.*