MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
(ANSI and ISO TEST CHART No. 2)

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Quality of Microfilm

Qualité du microfilm

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

**CASE BASED REASONING
for
COMMUNICATION NETWORK
FAULT DIAGNOSIS**

by

Lyle Zary, B. Elec. Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements of the degree of

Master of Engineering

Ottawa-Carleton Institute for Electrical Engineering
Faculty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada, K1S 5B6
Jan 3, 1994

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-89905-0

Canada

Name Lyle Zary

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

Computer Science

SUBJECT TERM

0984

SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS

Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION

General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS

Language 0679
General 0289
Ancient 0290
Linguistics 0291
Modern 0401
Literature 0294
General 0295
Classical 0297
Comparative 0298
Medieval 0316
Modern 0591
African 0305
American 0552
Asian 0355
Canadian (English) 0593
Canadian (French) 0311
English 0312
Germanic 0315
Latin American 0313
Middle Eastern 0314
Romance 0501
Slavic and East European 0503

PHILOSOPHY, RELIGION AND THEOLOGY

Philosophy 0422
Religion 0318
General 0321
Biblical Studies 0319
Clergy 0320
History of 0322
Philosophy of 0469
Theology 0323

SOCIAL SCIENCES

American Studies 0324
Anthropology 0326
Archaeology 0327
Cultural 0310
Physical 0272
Business Administration 0770
General 0454
Accounting 0338
Banking 0385
Management 0501
Marketing 0503
Canadian Studies 0505
Economics 0508
General 0509
Agricultural 0510
Commerce-Business 0511
Finance 0358
History 0366
Labor 0351
Theory 0578
Folklore 0366
Geography 0351
Gerontology 0578
History 0578
General 0578

Ancient 0579
Medieval 0581
Modern 0582
Black 0328
African 0331
Asia, Australia and Oceania 0332
Canadian 0334
European 0335
Latin American 0336
Middle Eastern 0333
United States 0337
History of Science 0585
Law 0398
Political Science 0615
General 0616
International Law and Relations 0617
Public Administration 0814
Recreation 0452
Social Work 0626
Sociology 0627
General 0938
Criminology and Penology 0631
Demography 0628
Ethnic and Racial Studies 0629
Individual and Family Studies 0630
Industrial and Labor Relations 0700
Public and Social Welfare 0344
Social Structure and Development 0709
Theory and Methods 0999
Transportation 0453
Urban and Regional Planning 0453
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES

Agriculture 0473
General 0285
Agronomy 0475
Animal Culture and Nutrition 0476
Animal Pathology 0359
Food Science and Technology 0478
Forestry and Wildlife 0479
Plant Culture 0480
Plant Pathology 0817
Plant Physiology 0777
Range Management 0746
Wood Technology 0306
Biology 0287
General 0308
Anatomy 0309
Biostatistics 0379
Botany 0329
Cell 0353
Ecology 0369
Entomology 0793
Genetics 0410
Limnology 0307
Microbiology 0317
Molecular 0416
Neuroscience 0433
Oceanography 0821
Physiology 0778
Radiation 0472
Veterinary Science 0786
Zoology 0760
Biophysics 0425
General 0996
Medical 0996

EARTH SCIENCES

Biogeochemistry 0425
Geochemistry 0996

Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleontology 0345
Paleobotany 0426
Paleozoology 0418
Paleogeology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES

Environmental Sciences 0768
Health Sciences 0566
General 0300
Audiology 0992
Chemotherapy 0567
Dentistry 0350
Education 0769
Hospital Management 0758
Human Development 0982
Immunology 0564
Medicine and Surgery 0347
Mental Health 0569
Nursing 0570
Nutrition 0380
Obstetrics and Gynecology 0354
Occupational Health and Therapy 0381
Ophthalmology 0571
Pathology 0419
Pharmacology 0572
Pharmacy 0382
Physical Therapy 0573
Public Health 0574
Radiology 0575
Recreation 0575

Speech Pathology 0460
Toxicology 0383
Home Economics 0386

PHYSICAL SCIENCES

Pure Sciences 0485
Chemistry 0749
General 0486
Agricultural 0487
Analytical 0488
Biochemistry 0738
Inorganic 0490
Nuclear 0491
Organic 0494
Pharmaceutical 0495
Physical 0754
Polymer 0405
Radiation 0605
Mathematics 0986
Physics 0606
General 0608
Acoustics 0748
Astronomy and Astrophysics 0607
Atmospheric Science 0798
Atomic 0759
Electronics and Electricity 0609
Elementary Particles and High Energy 0610
Fluid and Plasma 0752
Molecular 0756
Nuclear 0611
Optics 0463
Radiation 0346
Solid State 0984
Statistics 0984

Applied Sciences

Applied Mechanics 0346
Computer Science 0984

Engineering 0537
General 0538
Aerospace 0539
Agricultural 0540
Automotive 0541
Biomedical 0542
Chemical 0543
Civil 0544
Electronics and Electrical 0348
Heat and Thermodynamics 0545
Hydraulic 0546
Industrial 0547
Marine 0794
Materials Science 0548
Mechanical 0743
Metallurgy 0551
Mining 0552
Nuclear 0549
Packaging 0765
Petroleum 0554
Sanitary and Municipal System Science 0790
Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY

General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

The undersigned recommend to the Faculty of Graduate Studies and Research
acceptance of the thesis

**CASE BASED REASONING for COMMUNICATION
NETWORK FAULT DIAGNOSIS**

submitted by Lyle Zary, B. Elec. Eng.
in partial fulfillment of the requirements for
the degree of Master of Engineering

Thesis Supervisor

Thesis Supervisor

Chair, Department of Systems and Computer Engineering

Carleton University
Jan 3, 1994

Abstract

Case Based Reasoning Systems use previously solved problems to provide solutions to new problems. As part of the Pegasus project, this thesis investigates and develops a Case Based Reasoning System for communication network fault diagnosis. Specifically, a network's status reports and traffic rates are utilized to solve three types of router problems occurring in a network.

Acknowledgments

I would like to thank Professor Bernie Pagurek, Professor Nick Dawes and Dr. Julian Craddock for their guidance and support during the course of my research. I also wish to thank my peers, Mr. Alan Farrell and Mr. Paul Kalab, for their helpful assistance in applying some of their research results to my thesis. I also wish to acknowledge the research funding from the Telecommunications Research Institute of Ontario.

Table of Contents

Abstract	iii
Acknowledgments	iv
Table of Contents	v
List of Figures and Graphs	viii
List of Tables and Equations	x
List of Acronyms	xi
Chapter 1: Introduction	12
1.1 Objectives	13
1.2 Overview	14
Chapter 2: A Communication Network	15
2.1 The BNR Wide Area Network	15
2.2 Network Management	18
2.2.1 The BNR Network Operations Center(NOC)	19
2.2.2 Status Messages	19
2.2.3 Traffic Collection	21
2.3 Network Faults	21
2.4 The Network Simulator	22
2.5 Discussion	23
Chapter 3: The Pegasus Project	24
3.1 Pegasus Blackboard and PegasusBase	24
3.2 Basic Nested Evidence Spaces(BANES) - KS#1	26
3.3 Expert Network Diagnostician(END) - KS#2	26
3.4 Line Element Anomaly Pattern Recognition(LEAPR) - KS#3	26
3.5 Detection of Abnormal Router Behavior - KS#4	27
3.6 Operational Research Tools - NPATH, MEDIAN - KS#5	27
3.7 Inductive Methods	28
3.8 Discussion	28
Chapter 4: Case Based Reasoning	29
4.1 Case Based Reasoning Overview	30
4.1.1 Case Representation and Indexing	30
4.1.1.1 Representation	30
4.1.1.2 Indexing	31
4.1.2 Retrieval and Matching	35

4.1.2.1	Nearest Neighbor Matching Algorithms	36
4.1.3	Solution Adaptation	39
4.1.4	Presentation and Storage	40
4.2	Discussion	40
Chapter 5:	The Problem Domain	42
5.1	The Router Problems	42
5.2	Case Representation	46
5.3	Simulation Assumptions	48
5.4	Discussion	49
Chapter 6:	Status Message Features	50
6.1	Formation of the Status Message Case Based System	51
6.2	Initial Case Representation, Matching and Adaptation	52
6.3	Initial Status Matching Results	54
6.4	Status Message Graph Representation, Indexing and Matching	56
6.5	Status Message Matching Tests	62
6.6	Status Message CBR Results	62
6.7	Discussion	65
Chapter 7:	Traffic Rate Features	66
7.1	Selecting a Measuring Technique	66
7.2	Windowing Method Tests	68
7.3	Trend Analysis	70
7.4	Trend Analysis Tests	71
7.5	Selecting Our Traffic Features	72
7.6	Traffic Observations	73
7.6.1	Router Table Corruption (RC) Tests	74
7.6.2	Large File Transfer Abnormalities (FT) Tests	79
7.6.3	Storm Conditions (ST) Tests	80
7.7	Conclusions from the Traffic Tests	86
7.8	Applying Trend Analysis to Traffic Rates	87
7.9	Formation of the Traffic Case Based System	89
7.10	Traffic Representation, Indexing, Matching and Adaptation	90
7.11	Case Based Testing	95
7.12	Discussion	97
Chapter 8:	The Pegasus Case Based System	98
8.1	Forming the Pegasus Case Based Reasoning System	99
8.2	Combining Case Representations and Indexing Phases	99
8.3	Combining Solution Adaptation Phases	100
8.4	Combining the New Case Explanation and Storage Phases	101

8.5	The Pegasus CBR	101
8.6	Testing of the Pegasus Case Based System Traffic Cases	103
8.7	Size of Zone	103
8.8	Feature Weights	106
8.9	Increasing the Case Library Size	112
8.10	K-NN Testing	116
8.11	Feature Independence	118
8.12	False Alarms	121
	8.12.1 Trend Analysis False Alarms	121
	8.12.2 Surrounding Router Reports	124
8.13	Discussion	125
Chapter 9: Conclusions, Contributions and Future Research		127
9.1	Contributions	129
9.2	Future Research	129
References		131

List of Figures and Graphs

FIGURES:

Chapter 2

Figure 2.1	BNR Network Workgroups	16
Figure 2.2	Router Connections	18
Figure 2.3	Status Message Format	20

Chapter 3

Figure 3.1	Pegasus System	25
------------	----------------	----

Chapter 4

Figure 4.1	A Case Based Reasoning System	31
Figure 4.2	MOP Discrimination Structure in CYRUS	33
Figure 4.3	Associative Memory Pattern for CYRUS	34

Chapter 6

Figure 6.1	Status Messages Collection during a Break Fault	53
Figure 6.2	Status Messages Matching	54
Figure 6.3	Graphical Representation of Network Node Connections	57
Figure 6.4	Status Message Case Structure	58
Figure 6.5	Graphical Representation of Status Messages	58
Figure 6.6	Search Tree for Graph Matching	59

Chapter 7

Figure 7.1	Windowing Example of 20 min ref, 6 mins current	70
Figure 7.2	Zone for Router A	73
Figure 7.3	Zone nDNDc1 and Surrounding Network Elements	74
Figure 7.4	File Transfer Through Zone for Router nDNDc1	80
Figure 7.5	Traffic Case Representation	91
Figure 7.6	Matching Between Traffic Cases	93
Figure 7.7	Similarity and Distance Example	94

Chapter 8

Figure 8.1	Pegasus Case Based Reasoning System	102
Figure 8.2	Reports as a Percentage of Zone Shown	125

GRAPHS:

Chapter 6

Graph 6.1	Match Values for Problem Case nRTPc1 with Case Library	56
-----------	--	----

Chapter 7

Graph 7.1	nDNDc1 Router Mean Bytes Out - No Windowing, 2 Min Intervals	69
Graph 7.2	nDNDc1 Router Mean Bytes Out - Windowing 30 Minutes, 2 Min Intervals	69
Graph 7.3	nDNDc1 window mean: corrupt at Int 100	75
Graph 7.4	nDNDc1 window std dev: corrupt at int 100	75

Graph 7.5	nDNDc1 variability in mean: corrupt at Int 100	76
Graph 7.6	nDNDc1 cost: corrupt at Int 100	76
Graph 7.7	nDNDc1 cost deviations: corrupt at Int 100	77
Graph 7.8	nDNDc1 cost variability: corrupt at Int 100	77
Graph 7.9	nDNDc1 zone deviations: corrupt at Int 100	78
Graph 7.10	nDNDc1 mean: file transfer at Int 100	81
Graph 7.11	nDNDc1 mean rms: file transfer at Int 100	81
Graph 7.12	nDNDc1 zone dev: file transfer at Int 100	82
Graph 7.13	nDNDc1 cost: file transfer at Int 100	82
Graph 7.14	nDNDc1 cost rms: file transfer at Int 100	83
Graph 7.15	nDNDc1 mean: storm at Int 100	84
Graph 7.16	nDNDc1 mean rms: storm at Int 100	84
Graph 7.17	nDNDc1 zone dev: storm at Int 100	85
Graph 7.18	nDNDc1 cost: storm at Int 100	85
Graph 7.19	nDNDc1 cost rms: storm at Int 100	86
Chapter 8		
Graph 8.1	Accuracy vs. Value of K	117
Graph 8.2	False Alarms and Detection Accuracy vs Window Size	123

List of Tables and Equations

TABLES

Table 4.1	K-Nearest Neighbor Example	38
Table 6.1	Status Message Matching	64
Table 7.1	nDNDc1 Router Corruption Trends	88
Table 7.2	Matching Tests for Twenty Traffic Cases	95
Table 8.1	Size of Zone Matches	105
Table 8.2	Distinguishing Feature Weights for 120 Cases	109
Table 8.3	Discriminating Feature Weights for 120 Cases	110
Table 8.4	Best Matches with Feature Weights	111
Table 8.5	Distinguishing Feature Weights with 240 Cases	113
Table 8.6	Discriminating Feature Weights with 240 Cases	113
Table 8.7	Accuracy Compared Between 120 and 240 Case Base: $k=10$	114
Table 8.8	Conditional Probability for Independence Measures	118
Table 8.9	Accuracy Before and After Removal of RMS Features	120

EQUATIONS

Equation 4.1	Similarity Metric	36
Equation 4.2	Distance Between Two Position Vectors	37
Equation 4.3	The Angle Between Two Position Vectors	37
Equation 8.1	Size of Zone Ratio	104
Equation 8.2	Weighted Distance Metric	106
Equation 8.3	Sum of Squared Conditional Probabilities for Feature Weights ...	107

List of Acronyms

		Page First Used	*
AI	Artificial Intelligence	24	
BNR	Bell Northern Research	15	
CBR	Case Based Reasoning	12	
EBL	Explanation Based Learning	32	
FT	File Transfer Abnormalities	43	
k-NN	K-Nearest Neighbor	38	
KS	Knowledge Sources	25	
LAN	Local Area Network	16	
MBR	Memory Based Reasoning	29	
MOP	Memory Organization Packet	33	
NN	Nearest Neighbor	36	
NOC	Network Operations Center	19	
PING	Packet Internet Grouper	19	
RC	Router Table Corruption	43	
RDA	Remote Diagnostic Access Tool	21	
SNMP	Simple Network Management Protocol	19	
ST	Storms	44	
UTP	Unshielded Twisted Pair	16	
WAN	Wide Area Network	15	

Chapter 1

Introduction

Communication networks are growing in complexity and size as more and more companies become dependant on communications for their business. Presently, network operators use a combination of experience and expert-like tools to diagnose network problems. However, as the network complexity and size increases, the operators may soon become overwhelmed by the number of faults that can occur at one time, the size of the network in locating the faults, and the number of diagnostic features¹ to use to distinguish the type of fault. To aide the network operators, the Pegasus project began research to find feasible solutions for diagnosing communication network faults in large networks.

This thesis shows our solution to diagnosing three types of router faults using a Case Based Reasoning (CBR) approach. A CBR system solves new problems by adapting solutions from previously solved problems. Solved problems are stored as cases (structures of problem features) that can be individually retrieved and matched with new input problems. The best match to the input problem is then used to provide a solution. Our motivation for using this CBR approach is to imitate the network operator's process for diagnosing a router fault. The network operator, for example, may diagnose a network fault based on the collection of features from the network during the fault and the reminding of similar features in previously solved network problems.

¹Diagnostic features refer to standard network features such as threshold level settings, status messages, and the traffic rates. To be discussed in chapter 2.

In this thesis, a set of network attributes, namely traffic data and status messages, are used to define a router problem in a network. Solved network problems with solutions are stored as library cases, and new problems are then matched to the library cases. Using a matching process, a best library match to the problem case is found and applied to diagnose the new case.

Our CBR system is a novel approach for diagnosing faults in a communication network. Our research in a CBR diagnostic system is based on some of the previous problem solving applications of CBR in such fields as legal reasoning and diagnosing heart failures (see [Riesbeck89]).

1.1 Objectives

The objectives of this thesis are in the two following areas:

1. Investigate the use of a CBR system for network router diagnostics: In this thesis, an investigation of diagnosing router problems in a communication network using a case based system is completed. Three specific router problems for solving in a CBR system will be investigated in this thesis: 1) router table corruptions, 2) large file transfer abnormalities and 3) storms². For this investigation a network simulator is used as the test network, and network features from this simulator are used to represent the three router problem types. Our investigation will initially cover a review of literature on the various aspects of CBR systems such as indexing and representing cases, retrieval and matching for best cases, adaptation to provide new solutions, and explanation methods for CBR users; and
 2. Implement a CBR system in the Pegasus Project to diagnose router faults: In this thesis, using the results from the investigation in objective 1, a CBR system is developed to monitor a simulated communication network to
2. These three types of router problems are discussed in Chapter 5.

diagnose the three previously discussed router problems.

1.2 Overview

This thesis is divided into 9 chapters. Chapter 2 provides an overview of a typical communication network and the Bell-Northern Research Wide Area Network simulator used in the Pegasus project. Chapter 3 discusses the Pegasus Project, previously developed communication diagnostic tools in Pegasus, and Pegasus' current diagnostic research. Chapter 4 provides an overview of Case Based Reasoning Systems. Chapter 5 introduces our three router problems, discusses our selection of traffic and status message features for our CBR system, and discusses our assumptions for simulating our network router problems. Chapters 6 and 7 account for our research in using status messages and traffic rates for features in our CBR system. Chapter 8 further discusses the development of our Pegasus Base CBR System and investigates some aspects of our traffic features. Chapter 9 concludes this thesis and discusses our contributions and future research areas.

Chapter 2

A Communication Network

This chapter will introduce the structure and operation of a typical communication network that will be used for the development of our CBR system. The actual structure of this network will be provided by Bell Northern Research's Wide Area Network. This chapter will also provide a summary of some of the typical faults that can be expected from a communication network and introduce some of network features that will be used in our CBR system. This chapter will further provide a summary of the development of the Pegasus' network simulator that was used to provide an accurate simulation of a Wide Area Network (WAN). This simulator will be extensively used in the feature extraction and testing for our CBR system.

2.1 The BNR Wide Area Network

A typical network is composed of many identifiable elements. Figure 2.1 shows part of the network connections used in the Bell Northern Research (BNR) WAN. Most of the items presented in this figure are common to all configurations of networks and are briefly described as follows:

1. Workstations: user terminals for the network;
2. FileServers: data storage devices that allow workstations the access to various applications and large amounts of data storage;
3. Mainframes: fast computer work-horses for network users;
4. Bridges: allow for the connection of multiple local area networks along paths

where minor routing decisions are required; and

5. Routers: data transfer devices that are responsible for the transfer of packets from one LAN to the next.

At BNR, (see [BNR90a]) the Local Area Network(LAN) is called a workgroup.

Figure 2.1 BNR Network Workgroups

Each workgroup is composed of workstations, primary hubs, secondary hubs, and fileservers. The workstations in a workgroup are connected to primary hub devices. The primary hubs are then connected to a secondary hub to form a connection between all users in the workgroup. The secondary hub also enables the connection of other primary hubs in the vicinity. The secondary hub may include the connection of a fileserver for use by the workstations in the workgroups. The workgroups use an ethernet protocol on unshielded twisted pair wiring(UTP). UTP ethernet is, however, limited to about 100 meters of distance, and it becomes necessary to use a large number of grouped LANs to

provide full coverage to all of BNR's employees. To provide connection of the LANs within a particular location a device called a bridge is used. The BNR bridge allows for minor routing decisions between LAN groups. External connections can then be made from the bridge to a device called a LAN fanout backbone. The fanout backbone also allows the connection of mainframe systems and servers for all LAN users. From the fanout backbone, a router serves the purpose of connecting other LAN fanout backbones located within the same building or to other remote locations. It is from these routers that the network becomes connected into the larger configurations of Metropolitan Area Networks and WANs.

The transfer of data in the BNR network uses a connectionless datagram protocol at the network layer (i.e. the workstations communicate by transmitting information in data packets). In the transfer of data to external workgroups, the datapackets are passed by the bridges to the router, the routers then collectively channel the data along the best path to the destination. Figure 2.2 shows a typical connection between routers in a network. Routers are always interconnected to provide redundancy and quick routing of traffic. It is through the interconnections that the routers can provide a good 'quality of service' because the router can selectively choose the best paths for the packets that it needs to transmit.

The routers in the BNR network accomplish the task of selecting the best path to transfer packets by means of dynamic (adaptive) routing. Each router stores a router-table that provides an estimate of the cost in its attached lines; the cost for a line is based upon the line's current error rate, current activity and peak capacity. The router-table also contains specific addresses for all the various locations of the network. Given a packet to transmit, the router selects the line to transmit the packet on based upon the destination of the packet and the line's cost value. At set intervals, the routers transmit their own router-tables to the network so that other routers can become updated with new information.

Figure 2.2 Router Connections

Using this type of dynamic routing, the routers can account for broken lines or increased loads on particular paths. Dynamic routing is a critical factor in controlling the balance of traffic in a network; however, it adds to the network's traffic overhead because the routers must transmit the router tables and it adds to the complexity of diagnosing a faulty transmission because the paths of transmission are never fixed.

2.2 Network Management

One of the activities in network management is to effectively find faults in a network. A network operator uses network features such as node threshold levels and traffic rates, available network diagnostic tools, and personal experience to find and diagnose network faults. Also one of the first indications of a failure in a network is from network users complaining about being unable to perform certain network functions such as telnets, ftps, etc.....

By the late 1980s, network diagnostic systems grew to the point that the

management of networks required the incorporation of a standard protocol called the Simple Network Management Protocol (SNMP) and tools like the Packet Internet Grouper (PING). PING and SNMP allow for the monitoring of various network elements through direct interrogation. A device is interrogated with PING/SNMP, and the either the response or lack of response indicates the present status of the device. The device status is sent to the network operator as a status message report from the element. At BNR the monitoring of the network using the management protocols is integrated into a central office area called the Network Operations Center (NOC).

2.2.1 The BNR Network Operations Center (NOC)

The BNR NOC monitors the network using a system called MAGNet [BNR90b]. MAGNet probes selected network elements from network management workstations that are situated in the network; these management workstations are called data collectors. These data collectors are placed in the network to monitor dedicated network areas; a total of eight data collectors are used in the BNR network to provide total coverage. The data collectors use PING or SNMP to interrogate the network elements. Because the interrogation of these devices is accomplished through the network traffic lines (using in-band signalling), the overhead in the network is increased. To keep the overhead manageable, only devices that are considered more significant to the effective operation of the network are probed (i.e. routers, bridges and servers). Through SNMP, the network elements also have some ability to send reports to the data collectors stating a malfunction, but it mostly through the active interrogation of the network that faults are reported in the network.

2.2.2 Status Messages

The data collector's probe network devices about once every three minutes. If the data collector does not receive a response from an element, it will attempt to probe

the element again. After repeated attempts with no response, the data collector will then transmit a report to MAGNet that a particular device may be faulty. This report contains the time of the report, the name of the data collector reporting, the name of the device that is faulty and a numerical value that indicates the condition of the device. The numerical value of the status represents one of the five following conditions:

1. 10: the device has not responded to 9 successive attempts;
2. 11: the device has not responded to 20 successive attempts;
3. 12: the device has not responded to 30 successive attempts;
4. 1003: the device reported that it is broken; and
5. 2: the device is OK.

MAGNet displays the information from the data collector to a user at the NOC in the form of a status message as shown in Figure 2.3. MAGNet also has the capability to present the status message reports on a computerized graphical display of the network topology. As status reports arrive at MAGNet, the element that reported the fault is colored to attract the attention of the MAGNet user. As more status messages arrive, the color of the device changes to indicate the status value.

91/01/01 00:30:09:	car2.dc bCARh2	10.nANNc1.car.bnr	10
91/01/01 00:30:18:	car2.dcbCARh2	10.nBFLc1.car.bnr	10
91/01/01 00:30:33:	car2.dc bCARh2	nANNc1.car.bnr	2

Date and Time Stamp	Data Collector name	Node reporting problem	Status value

Figure 2.3 Status Message Format

As status values arrive, the NOC operator attempts to determine either if a fault is actually occurring or if the status message report is a false alarm. False alarms can occur for a number of reasons (for e.g. an element might be too busy to respond to an

interrogation), so the NOC operator usually determines a fault based upon a significant number of reports from a group of nodes in the same area. A more experienced operator may also use the collected network traffic rates as a further indication of a fault in an element.

2.2.3 Traffic Collection

The NOC Operator may use the Remote Diagnostic Access(RDA) Tool to localize a network fault within a LAN. This system allows the operator to dial-in to the secondary hubs at the workgroups to observe the traffic rates at the area. Through the RDA, the NOC operator may observe the congestion of traffic or mis-routing of packets to indicate a fault. The NOC center also collects the traffic rates from all of the routers located in the network (the routers at set interval transmit their traffic loads to the NOC). These traffic rates indicate the traffic load of the router's attached lines, and they can be used to observe a router's performance over a period of time. The traffic rates are also used to set threshold level indicators that will alert the network operator when a router's traffic load is excessive. For the most part, however, these traffic rates are not analyzed by the network maintainers and are simply stored away for future reference.

2.3 Network Faults

This section will provide an indication of some of the faults that an operator must deal with in trying to maintain a large communication network. In [Murphy92], some of the regularly occurring faults at BNR have been divided into four broad categories and include the following:

1. **Physical Faults:** The mechanical and electronic failure of components fall into this category of faults; these failures include such items as bad ethernet chips, faulty boards, and loose connectors. Some of the symptoms of these physical faults include large packets with large error rates, traffic storms, sessions dropping on

telnet services, workstations not being able to transmit, and only being able to send data to specific nodes.

2. **Communication Facility Faults:** These faults occur as a result of using another vendors services over which no network control can be maintained (ex. the leasing of phone lines from the telephone company). Some of faults observed at BNR include a poor transmit path and incorrect settings on the vendors multiplexer. Symptoms may include poor file transfer performance and severe errors occurring on transmit paths.
3. **Finger Problems:** These faults are human associated faults. They include faults such as the incorrect setting of network addresses by the network administrator to cause unstable routing or periodic shut-down of LANs.
4. **Protocol and Miscellaneous Issues:** Faults in this category include the problems associated with transferring data over different types of network protocols. These faults can also include severe traffic loads caused by the network users transferring large files during peak network use.

Presently, the methods in this thesis have been used to diagnose three types of router faults. Our router faults are included in the Physical and Protocol items of the above list. Future research may prove that our CBR methods will be useful for diagnosing a wider spectrum of faults.

2.4 The Network Simulator

With the information provided by BNR on their network operation, the DRAGON simulator[Dawes91b] was constructed; this simulator provides the same datagram operation, distributed routing and network monitoring as used in the BNR network.

The simulator uses a set of three text files to input the network configuration. It can simulate up to 32,000 active devices (workstations, routers, bridges, etc....) and can run up

to 50 times faster than real-time. The simulator also allows the simulation of various traffic environments (low, moderate, heavy) by allowing the user to set a number of repetitive file transfers while the simulation is running.

The simulator also allows for a variety of network faults to be induced as discussed in Section 2.3. Presently the simulator is configured to allow the introduction of element breaks, line breaks, packet loss rates, router corruptions, router line delays, large file transfers, and data storms. The simulator also allows the monitoring of network elements that the data collectors in the BNR network use, and it also provides status reports through simulated SNMP/PING protocols. Traffic rates can also be collected on the network routers during the simulation.

2.5 Discussion

Communication management is a difficult task considering the complexity of the network structure and the range of faults that can occur. The network operator has to rely on a collection of features during a fault, such as status messages and traffic rates, to deduce the location and type of fault that has occurred. In the Pegasus project, the simulator allows us to observe the status messages and traffic rates after a fault is produced to simulate the real-time occurrence of events in the NOC during a fault. The simulator further provides the Pegasus team with a very accurate model of a WAN for testing new diagnostic methods (as an example, BANES[Dawes91a] was developed with the simulator and was successfully integrated into the actual BNR network with little modifications).

Chapter 3

The Pegasus Project

As part of diagnosing a fault, a diagnostician will attempt to find a solution to a particular problem by collecting the problem features (i.e. forming a description of the problem) and applying his/her experience from solving previous problems and/or using a set of predefined solving rules. These diagnostic methods are an area in which Artificial Intelligence (AI) methods have been successfully applied. For example, Expert Systems use a set of rules taken from experts to solve problems. Inductive Systems attempt to create decision rules based on the collection of problem features; these inductive rules can then be applied to categorize new faults. Case Based Reasoners, Memory Based Systems, and Neural Networks are classification systems that require different types of learning sets to diagnose new problems. The Pegasus Project began in 1990 with a mandate to provide tools to diagnose communication network faults using various artificial intelligence methods. As a background for the development of a Case Based Reasoning System in network diagnostics, this chapter will present the current state of the Pegasus Project.

3.1 Pegasus Blackboard and PegasusBase

The central components in the Pegasus system are the Blackboard and PegasusBase. The Blackboard prototype which was built by [Iqneibi92] has a control structure to support the multiple paradigm diagnostic system of the Pegasus project. PegasusBase [Yuan92] is a dynamic database that provides for the storage of all

information that passes through the Blackboard. Figure 3.1 provides a general view of the Pegasus System.

Figure 3.1 Pegasus System

The Blackboard architecture is a central control structure that allows individual knowledge sources (diagnostic processes) to attach and detach themselves without affecting the other attached processes. The Blackboard system thus allows for a common interface for all of the individually developed systems. Through a set of *post* and *get* blackboard handling functions, attached knowledge sources (KSs) can pass messaging information through the blackboard for either the simulator or other KSs. Through a set of semaphore controls, the blackboard can also signal a KS that information has been posted that it may require. PegasusBase automatically stores all information that is posted to the blackboard to allow the KSs to retrieve any past data. Through the Blackboard and PegasusBase, the KS's can also post hypothesis on a network fault. A Master KS, although not presently implemented, will attempt to use the generated hypotheses from all the attached diagnostic tools and provide the NOC operator with best estimate for a fault.

3.2 Basic Nested Evidence Spaces(BANES) - KS#1

BANES[Dawes91a] was the first developed diagnostic KS in the Pegasus Project. It used status messages reports and the network topology to diagnose network faults. Using these two inputs, BANES constructed a network monitoring tree structure (called a nested evidence structure) on which status reports generated by devices were placed. BANES then reasoned on the best solution to a fault by considering the relationships between the path of the status messages reported and the total number of reports collected from the path's area. For example, if status reports existed from Node A, B, and C, and if node A and B propagated their status reports through node C, then a belief that node C was the faulty node was valid. Using the nested evidence structure, BANES obtained a 98% accuracy in diagnosing break faults.

3.3 Expert Network Diagnostician(END) - KS#2

END [Dussault92] was an expert diagnostic system written in the COSMIC CLIPS shell. It was divided into three modules that verified the BANES output, diagnosed broken elements and lines in the network, and traced traffic paths to confirm if excessive traffic was occurring in the network elements. As in all rule based systems, END's solutions to faults were accomplished by a series of rules that were deduced while solving for a fault. These rules were developed by interviewing the NOC operators, and the END rules tried duplicated the exact procedure by which the NOC operators used in diagnosing a fault. The END system achieved an accuracy equal to BANES for some faults, but it suffered from having to add traffic overhead to the network because it had to query the network during its problem solving.

3.4 Line Element Anomaly Pattern Recognition(LEAPR) - KS#3

LEAPR[Viens92] was a neural network approach to monitoring and classifying a network's router traffic pattern. LEAPR, through a period of training, adaptively learned the normal and abnormal state of a router's traffic performance. LEAPR's abnormal state

was set as a router packet congestion above a pre-defined threshold level. LEAPR's final output was a logical state that indicated whether the current router's traffic input was either a normal or abnormal pattern. LEAPR's output could either be used by other knowledge sources in classifying problems or presented to the network operator for analysis.

3.5 Detection of Abnormal Router Behavior - KS#4

[Farrell93] presented an algorithm that analyzed traffic data to detect the abnormal behavior of routers. This work paralleled research conducted by [Viens92], but used analytical methods to devise an appropriate model of a router's performance. [Farrell93] also presented a measure called the zone contribution. The zone contribution measured a router's performance in relation to the routers around it. In this respect, the zone contribution provided a measure of traffic correlation with neighboring nodes. [Farrell93] established threshold levels that were used to indicate the presence of a router delay fault, this threshold level was set based upon the differences between a set of base statistics and the current zone contribution and the root-mean-squared (rms) traffic bytes out values for router's being analyzed. A set of base statistics was collected by calculating the mean traffic of a router's mean bytes out for a twenty four hour interval, [Maxion90] used a similar measurement in analyzing network communications.

3.6 Operational Research Tools - NPATH, MEDIAN - KS#5

Although not fault diagnostic tools, NPATH and MEDIAN [Kalab93] were developed to aide researchers and network planners. NPATH finds the n shortest paths between two nodes in a network. In situations of dynamic routing, it aided a diagnostician by computing the most likely routes that a packet could travel. MEDIAN was developed to hierarchically compress a network topology into a simplified form. This simplified form could be used by researchers and planners in diagnostics. As an example, MEDIAN was used to find the optimal location of the data collectors in a network.

3.7 Inductive Methods

[Bo93] and [Shi93] are conducting research in implementing machine learning inductive algorithms for network diagnostics. Essentially, inductive methods use a set of positive and negative example states to form a set of class descriptions. These class descriptions attempt to account for all positive examples, while excluding all of the negative examples. Applied to network traffic rates, these class descriptions will form a set decision rules to cover specific events in a router's performance. As an example, a class description of a router experiencing a delay may take the form of the following:

```
router_delay_in_A true if
    line1_traffic_A ( > Valuej ) and line2_traffic_A ( < Valuei ) and
    time_of_day ( noon )
```

It is expected that a set of descriptions could be developed for a number of fault classes. These fault classes could form a set of rules for diagnostic reasoning systems.

3.8 Discussion

The Pegasus Project has successfully developed various methods to diagnose different types of faults in a network. END and BANES performed well to diagnose break faults. LEAPR and [Farrell93] diagnose router problems based upon congestion and packet delays respectively. This thesis will provide the diagnostics of three other types of router problems occurring in a network.

Chapter 4

Case Based Reasoning

In solving a new network problem, a network operator may recall a particular past experience in which either the same or similar symptoms for the fault occurred. Using this past experience, the network operator may apply the same solution or an adapted version of the past solution to solve the new fault. This type of problem solving is an experience which all of us tend to use at some point or another in solving problems. This methodology has also been somewhat 'standardized' into a field of AI called Case Based Reasoning (CBR). CBR methods use a set of attributes to provide a case (or frame) description of a problem. This problem and its solution are stored in a data structure called a case library; the case library can hold many possible problems and their solutions. When a new problem occurs, the CBR System takes as inputs a new problem's collected features and tries to provide a best match with a solved case from the case library. If the retrieved case is an exact match, the retrieved solution can be applied directly to solving the new problem. If the match is not exact, the retrieved solution may be adapted somewhat to fit the new problem. Systems which do not adapt solutions from retrieved cases are called Memory Based Reasoning Systems (MBR). After a solution to a problem is found, the newly solved problem is added to the case library for use in future situations.

This chapter will provide a background to previous work in the area of case based systems¹.

1. I will use the term case based to refer to a system using case libraries. It will refer to both CBR and MBR systems.

4.1 Case Based Reasoning Overview

In Case Based Reasoning, past stored information is used to help solve current situations. [Kolodner91] divided CBR into two styles: problem solving and interpretive. Problem solving CBR systems use old cases to derive new solutions. Old solutions are directly applied to new cases or through some adaptation rules, old solutions are adapted to fit into the context of the new problem. A mechanic, for example, will adapt to replacing a Honda Prelude alternator, even if he/she has only experienced replacing Ford Tempo alternators. The adaptation of the mechanic's stored Ford solution might be 'alternator is in a different location' and 'three bolts instead of four'. These types of problem solving systems are ideal for planning, design and diagnosis. Interpretive CBR Systems use old cases to justify a new case. They do not adapt solutions from old cases but instead use the same decision making process from the old case to justify the new case. As an example, a judge will use an old case to justify a decision in a new case. A typical CBR system is shown in Figure 4.1. CBR systems can be divided into four major phases: 1) Case Representation and Indexing, 2) Retrieval and Matching, 3) Solution Adaptation and 4) New Case Presentation and Storage.

4.1.1 Case Representation and Indexing

Representation and indexing the problem environment into a suitable problem case is the most important step in case based systems.

4.1.1.1 Representation

The representation of a problem must answer some of basic questions that would normally be asked when trying to remember similar situations: What happened? What did I notice at the time? What were the measurements? Answers to these questions usually take the form of a record of information in a data-structure and are placed into a database for retrieval. The selection of features for a case is a vital step to ensure that the case can be retrieved for use in similar problems. The occurrence of features and/or

Figure 4.1 A Case Based Reasoning System

the values of the features in the cases are used to distinguish among the different types of problems found in the case library. It is important also that the case should not contain any irrelevant information that would make the search of these features useless to case selections. For example, using the 'time of day' as a feature in cases that solve disputes may not be relevant (i.e. the time that previous disputes occurred are of no consequence to solving the new problem). In most cases though, the CBR systems are usually established with a goal to provide solutions to specific problems, so the CBR designer knows a-prior what features may be required.

4.1.1.2 Indexing

The indexing of a case is the method by which the case is stored in the case library; the indexing problem in CBR is to make the computer retrieve the appropriate case at

the appropriate time. The following are four guidelines that [Kolodner91] used in building indexing features:

- 1) Indexes should be predictive to the solutions outcome;
- 2) Indexes should be abstract enough to make a case useful in different situations;
- 3) Indexes should be concrete enough to be recognizable in the future; and
- 4) Indexes should be useful for indexing and retrieval purposes.

Various methods have been used to find the relevant indexing features to the problem case. [Barletta 88] uses some Explanation Based Learning (EBL) methods to represent and index a case; this EBL approach required some knowledge of the problem domain (taken from domain expert's). [Leake86][Barletta91] mention approaches in Inductive Learning to find relevant features to problem cases given a collection of all the features in the problem domain.

Most of the computer implementations of indexing systems in CBR has been accomplished by using indexing methods that attempt to model human memory indexing and retrieval mechanisms. These indexing methods include 1) discrimination nets and 2) associative retrieval systems:

1. **Discrimination Nets:** Many CBR systems are built on an indexing scheme using discrimination nets² to store episodic memory (information on events or episodes). Authors on CBR [Slade91][Barletta91] refer to [Schank82] as work that describes some of the first concepts of using a computer memory to store episodic knowledge. As an example of episodic information, "John went to the dentist" might be represented in episodic memory as an event in John's memory as 'going to the dentist'. While at the dentist, John also 'felt pain' and found out that he 'could not pay by cheque'. If John were to visit the dentist again, before grabbing his chequebook to pay, he might recall all the events of his previous visits, and he would be reminded that he can not pay by cheque. [Schank82] used the collection
2. Discrimination nets is a term applied to tree-like indexing schemes that are a combination of hierarchical and associative patterns. The discrimination in these nets is characterized by the various paths in the tree that can be chosen while searching.

of these events to form indexing schemes called Memory Organizational Packets (MOPs). These MOPs were combined to form a discrimination net of features. These discrimination nets are a type of hierarchical and associative indexing used to structure cases into a database. Figure 4.2 shows a particular MOP discrimination net from CYRUS [Kolodner86]. CYRUS kept track of the

Figure 4.2 MOP Discrimination Structure in CYRUS

events of Cyrus Vance when he was US Secretary of State. This particular figure shows a MOP structure of CYRUS' meetings with foreign diplomats. As an example of discrimination retrieval in Figure 4.2, a search from MOP1 for meetings with Begin on Camp David Accords could take two separate paths by specializing in either the participants or the topics; in both situations the correct MOP4 solution would be retrieved.

2. **Associative Retrieval:** An associative memory structure is a more common type

used in CBR systems (for their simplicity). Associative structures store features of a case independently of the other features. Associative structures are similar to relational data structures; however, in associative retrieval mechanisms the information stored is referenced by the entire contents of the data structures rather than to any one particular field in the data structure. In discrimination nets, memory patterns were stored as events; in associative structures, memory patterns are stored as phrases or single features. As an example, in an associative structure a question of “do you like to go to the dentist?” may be searched for all patterns of ‘dentist’ to find all related experiences with dentist. A common method of associative searching, found in [Gotlieb78], is accomplished by using a special mask register to find all of the stored cases that contain the same fields as the ones being searched. As a comparative example, Figure 4.3 shows an associative implementation of the CYRUS discrimination structure in Figure 4.2. In Figure 4.3, a search for meetings with Begin at Camp David would retrieve CASE 2. [Simoudis91], [Creedy92],[Lewis93] are examples of associative memory retrieval systems.

search_fields →	-	Begin	Camp David Accord	-	
MASK →	0	1	1	0	search results
	<i>meeting type:</i>	<i>participants:</i>	<i>topic:</i>	<i>goal:</i>	↓
CASE1	foreign diplomat	Dayan	Camp David Accord	resolve contract	0
CASE2	foreign diplomat	Begin	Camp David Accord	resolve contract	1
CASE3	foreign diplomat	Gromyko	SALT	resolve contract	0
CASE4	foreign diplomat	Begin	Jerusalem	resolve contract	0

Figure 4.3 Associative Memory Pattern for CYRUS

The discrimination net and associative memory structures have separate benefits. Discrimination nets offer quick retrieval because they use the features as guides in

searching for matching cases; discrimination nets however, have more difficult indexing schemes to implement and may not allow extraction of cases on certain features alone. Associative indexing is very simple to implement and ensures a best match, but it suffers from having to retrieve all the cases in the case library to find the best match.

4.1.2 Retrieval and Matching

As cases are retrieved from the Case Based Reasoner, a matching phase must determine which case is the best match. As an example, in associative retrieval many cases may be retrieved from a set of search fields. A matching algorithm must then further match the case features to decide which is the best match. Weighting on the features can also be used to account for some features in a particular problem case being of greater value than others. Chapter 8 of this thesis will show how the weighting of features in our case based system is accomplished.

Problems in matching occur when the features between cases are similar but not exact. For example, how would you match a problem case with a color feature of 'sky blue' to the library case colors of 'navy blue' and 'black'? Physically these three colors are different, but 'sky blue' is closer to 'navy blue' than to the feature 'black'. Some matching systems would match these attributes using string comparisons and would thus yield no best match cases. In most cases, however, a *fuzzy* type of matching between features is required to provide the best matching results. Fuzzy matching involves the use of a normalization table where all features are converted into suitable numerical values. A value-table for colors in this example might be 'sky blue =1', 'navy blue =1.5' and 'black=5'. A matching algorithm would then match the colors accordingly to their numerical value and assess the closest similarity to 'sky blue' as being 'navy blue'. However, in using these types of matching algorithms, a knowledge of the domain attributes is required; the normalization tables must be designed to suit the specific

problem at hand. Nearest Neighbor algorithms are often used in case based systems to match case features based on feature similarities.

4.1.2.1 Nearest Neighbor Matching Algorithms

Nearest Neighbor (NN) Algorithms are very simple pattern classification techniques. A numerical match-value is calculated for each matching attempt between a problem case and library case to decide the best match (or nearest-neighbor) for the input problem. Types of matching methods for nearest neighbor approaches include 1) similarity metrics and 2) vector approaches.

1. Similarity metrics sum the number of features that match between cases to compute a match-value. Equation 4.1 shows a similarity metric used by [Cain91] [Creecy92]. Similarity metrics can also incorporate the use of fuzzy matching tests to account for feature similarities.

$$\frac{\sum_{i=1}^n \text{sim}(\text{feature}_i, \text{cue}_i)}{n} = \text{match-value}$$

where cue_i is the feature of the problem case and feature_i is a feature from a retrieved case. $\text{sim}(x,y)$ is defined as:

$$\text{sim}(x,y) = \begin{cases} 1 & \text{if } x=y \\ 0 & \text{if } x \neq y \end{cases}$$

Equation 4.1 Similarity Metric

2. In vector analysis, the case features are represented as a multi-dimensional vector co-ordinates. Using methods as those found in [Pettofrezzo66], the features in a problem case are represented as position vectors and are matched to the vectors

from the case library. Equation 4.2 shows a calculation of a match-value by

$$d = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2 + (z_b - z_a)^2}$$

as d approaches zero, the vectors are closer matched.

Equation 4.2 Distance Between Two Position Vectors

measuring the distance between the two vectors points (often termed the Euclidean distance). Instead of measuring the distance, Equation 4.3 calculates a case match by calculating the angle between the two vectors (used in [Kitano92]).

$$\cos\theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| \cdot |\vec{B}|}$$

as cosθ approaches zero, the vectors are closer matched

Equation 4.3 The Angle Between Two Position Vectors

Each type of the NN matching algorithms has separate benefits. Vector approaches are useful because they automatically include fuzzy matching of the features in the calculation of the match value. They can include, for example, the differences in features by matching the 'blue color' as a close match to a 'sky_blue color'. Similarity metrics are useful if the case features cannot be easily divided into separate categories. As an example, similarity metrics would work well if the case features are a question

of being either a match or no match between a collection of unordered data.

An often used extension of the NN algorithm is k-Nearest Neighbor (k-NN) matching. In k-NN matching, the match-values calculated for each problem case to library case comparison are used to sort the library cases in a numerical order. The k-top cases are then used to decide what class the input problem belongs to. In many cases, k-NN matching and associative retrieval systems are combined to provide match values for every attempted match in the case library.

K-Nearest Neighbor can provide a better prediction of which class the input problem belongs to because it can make a prediction based on matching to a group of cases instead of only the top best match. As an example, Table 4.1 shows an example of match of the five top library cases to a problem case. If value of k in this table was

Number	Match_value	Library Case	Solution
1	5.4	34	'Do A'
2	4.5	12	'Do B'
3	4.0	5	'Do B'
4	3.7	8	'Do B'
5	2.1	65	'Do C'

Table 4.1 K-Nearest Neighbor Example

set to 1, then the best match would be library case 34 and the solution to 'Do A' would be applied (note: the simple nearest neighbor rule is $k=1$). With $k=5$, however, the best solution to the problem case would be to 'Do B'. In this case, there are 3 of the top 5 cases in which the same class of solution was found applicable to the input problem. More confidence in the 'Do B' solution is valid since many cases with 'Do B' solutions were retrieved that resembled the input problem. If 'Do A' was correct, then we should have retrieved more cases in the top 5 with a 'Do A' solution. K-NN is also very valuable in accounting for any inherent errors in the case features. As an

example, In Table 4.1, case 34 may have been selected as the top case because one of its features was incorrectly set. There is more confidence in the 'Do B' solution even if one of the three cases had incorrect features because the other 'Do B' cases would be valid retrievals. The value of k to use in selecting the best class from the top- k cases is an important consideration. The value of k is dependant on the number of cases that are available in the case library, the number of different classes that are selectable, and on any errors that the cases may contain. The best method to select the value of k is to conduct tests that measure the accuracy of the matching for various values of k .

4.1.3 Solution Adaptation

When the best matched case (or k -NN cases) is found, the solution retrieved from the best-matched case can then be applied to the new problem. Depending on the type of match, the retrieved solution can either be directly applied or it can be adapted to account for differences between the problem and best-matched case. Two types of adaptation are expressed in [Riesbeck89]: 1) structural adaptation and 2) derivational adaptation.

1. **Structural Adaptation:** In this type of adaptation the old and new problem cases are compared along each feature. Differences in the features are used to modify the solution. The solutions are usually adapted along an applied set of rules that are externally contained from the case library. As an example, JUDGE[Riesbeck89] is a CBR system that modified retrieved criminal sentences based on the serious of the crime. In JUDGE, if an assault and battery charge that was being assessed involved a knife cut to the victim, and if the best matched case to the assault and battery charge was one that involved a knife but no cut to the victim, then the retrieved case's sentence would be increased by one year to account for the contact to the victim. The structural adaptation is the common type of solution adaptation in CBR systems.
2. **Derivational Adaptation:** In derivational adaptation, the cases themselves contain

the methods by which the solution was derived. A new problem case then might use those same methods to arrive at a new solution. A simple example is a case containing a formula to get a solution; the problem case would then use its own features in the retrieved formula to arrive at a new solution. Another form of derivational adaptation is through the use of a causal model, as in [Koton89].

4.1.4 Presentation and Storage

After adaptation, newly solved cases are visually presented to the CBR user for acceptance. If the case is accepted, it will then be indexed and stored in the case library for future use. The presentation and storage phase is an important step in the CBR process because the storage of 'wrong' cases will result in the retrieval and matching of 'wrong' cases to new problems. Some CBR systems mentioned in [Riesbeck89] contain procedures that attempt to explain a failure in retrieval and attempt to repair the indexing system to find a better solution. The use of the discrimination net indexing scheme offers an easy repair strategy because links between MOPs can be easily removed and inserted.

4.2 Discussion

This chapter presented an overview of Case Based Reasoning Systems. A typical CBR system contains four separate phases: 1) case representation and indexing, 2) retrieval and matching, 3) solution adaptation and 4) presentation and storage. Memory Based Reasoning systems follow the same principles but do not contain an adaptation phase. This thesis shows an application of some of the CBR methods shown in this chapter to diagnosing communication network faults.

[Riesbeck89] provides a summary of some CBR applications such as CYRUS (story-understanding), JUDGE (legal reasoning), CHEF (meal-planning), and CASEY (heart diagnosis). Authors such as [Barletta91][Slade91] mention some aspects of CBR systems that provide advantages over Rule Based Systems (expert systems). Firstly rule-based systems suffer from being memoryless. In other words, they cannot easily learn as

they are being used. The memory in CBR systems comes from newly solved cases being stored for use in future situations. Secondly, expert systems are only as good as their rule-base. If they are given a problem to solve that no rules can cover, then the system will fail. In CBR systems, problems are related by matching features, so library cases can usually be retrieved that best resemble novel problems. Thirdly, rule based systems are difficult to use in situations where rules are not readily available. As an example, it is possible that an expert may not understand why a particular event occurs or how particular events relate. In CBR systems, features are used to represent situations, so it is not critical to know why the features occur.

Chapter 5

The Problem Domain

This chapter will introduce the three types of router problems that we will be solving in our CBR system. This chapter will then discuss our initial analysis of the network during the three network faults. The purpose of this analysis is to find network features that we can use to represent the router problems for our case based system. This discussion will show our decision to use status messages and traffic rates for the case features. From these features, two separate CBR systems will be formed. The first system will be based on status messages feature cases, and the second system will be based on router traffic rate features. Each CBR system will be developed with its own representation, retrieval, matching and adaptation phases; Chapters 6 and 7 will further discuss the development of these two systems.

This chapter will also discuss our assumptions that we will use to simulate our network faults and collect our network features from the simulator.

5.1 The Router Problems

Chapter 3 mentioned the analysis of network break faults and router packet delays that were solved in the past Pegasus diagnostic paradigms. In our CBR system, we will expand the scope of the Pegasus diagnostics in router faults by solving for three other types of router problems: 1) Router Table Corruptions, 2) Large File Transfer

Abnormalities, and 3) Storms. A brief description on the symptoms and an indication of what a network operator uses for diagnosis of these router problems is provided as follows:¹

1. **Router Table Corruptions(RC):** Chapter 2 mentioned how router-tables are used in deciding the best path to send a packet. A class of faults that can occur when these router tables contain errors are called a router-table corruptions. These router-table corruptions may cause the packets to be transmitted along the incorrect lines. Mis-transmitted packets will either reach the destination later or become discarded after their allowable time in the network has expired. Symptoms of router corruptions can include the a loss of connection or an increase in file transfer time. These types of faults are very difficult to diagnose in a dynamic routing network because the route at which the packets take to a destination is never a constant; a network diagnostician will have a difficult task in trying to trace the path of a file through a network to see which router may be mis-transmitting packets. The router table corruptions are caused by hardware and/or software problems, or they may be caused by a corrupt transmission of router tables from the other routers. For this thesis, our router corruption tests will be simulated by having the network simulator operate normally for a 100 minute simulated time period before the corruption. This time period will allow the simulator to settle after start up conditions. At the 100 minute mark, a router table corruption will be started on a specifically chosen router by using the simulator's user interface. Our investigations on the router will occur before and during the router table corruption.
2. **Large File Transfer Abnormalities(FT):** Large file transfers abnormalities are problems that plague all types of networks. These problems are human induced faults that begin by network users sending/retrieving large files through a busy

1. Our assumptions for these three types of router problems are discussed in Section 5.3.

network. A large file transfer during a busy network period will add more packets to the already loaded requirements of network, thus these large file transfers will slow the network even further. The symptoms of these types of problems include an increased time to do simple network transactions; networks users will also complain to the network operator on the poor network performance. The classification of a large file transfer abnormality in a network is dependant on the speed of transmission in the network. As an example, a file of 5 Mbyte at a Tx speed of 64K bits-per-sec will take about 10 minutes. The same file at 9.6K bits-per-sec will take about 70 minutes. The longer the transfer time, the greater is the effect on the network's performance. The network diagnostician looks for increased traffic on particular lines and routers to find the source of the large file transfer. To alleviate the problem, network operators ask users to do their large file transfer during non-peak network hours(i.e. at night). For this thesis, our large file transfer tests will be simulated by having the network simulator operate normally for a 100 minute simulated time period before the large file transfer. This time period will allow the simulator to settle after start up conditions. A large file transfer will be caused on a specifically chosen router by using the simulator's user interface; the file transfer will be set so that the transfer occurs between two workstations in the network that are in the path of the router that we want to investigate. Our investigations on the router will occur before, during and after the file transfer.

3. Storms(ST): Storms are a constant or repetitive transfer of large amounts of data through a particular part of a network. A router, for example, will be operating normally and then suddenly be required to route a large amount of data. The data then stops and after a short time re-appears. The network elements then become so overwhelmed by the traffic that their performance suffers. Users of the network may loose connections, data transfers may be

lost, and the router-table transfers may become corrupted. Storms are usually caused by hardware faults or improper network wiring that will cause data to be continuously sent in these repetitive streams. Network diagnosticians can apply some of the same techniques in finding the source of the large file transfers to find the source of the storms. For this thesis, our storm tests will be simulated by having the network simulator operate normally for a 100 minute simulated period before the storm is started. This time period will allow the simulator to settle after start up conditions. A storm will be caused on a specifically chosen router by using the simulator's user interface; the storm will be set so that the storm occurs between two workstations in the network that are in the path of the router that we want to investigate. Our investigations on a router will occur before and during the storm.

The above three problems explain the scope of our three router faults. These three types of problems occur in network systems, and their occurrence can cause a severe degradation on the network's performance. A diagnostic system to help solve these three router problems would be a valuable tool for the network operators. This diagnostic tool should also provide the network operator with indications that a problem in a network may be developing; these type of indications will allow the network operator to alleviate the problem before the network's performance becomes affected. One of the largest tasks in our CBR system will be to select the proper features to represent these three types of problems in a case based system. It is essential that these features be chosen so that they provide a quick and accurate diagnose for the network operator to use. Our selection of features should also provide some warning indications that a problem may be occurring in the network. Our feature selection for case representation will be the first step in developing our CBR system.

5.2 Case Representation

In our situation, we know a priori what problems our CBR system is going to solve. Therefore, in developing our CBR system, we can concentrate on collecting available network features on these problems to decide what features are best to use in our case representation. The representation will then dictate suitable retrieval and matching methods.

For proper case representation, the features (as discussed in Chapter 4.1.1) must be abstract enough to cover the problem domain and be predictable in distinguishing the different problems. As an example, the choice of using values to represent the amount of traffic flowing through a router may be a good choice for a case feature. Depending on the fault that occurs in a router, it may be noticeable that the traffic rates in a router corruption are different than the traffic rates during a storm problem. A library case could then use a collection of the traffic rates to represent a router fault. A new input problem would consist of the collection of traffic rates collected during an unknown fault. The best case to the new input problem would be the library case that has the closest values to the problem cases traffic rates. These traffic rates are abstract enough because they are available during a router corruption and storm problem, and they are predictable because their values can determine the type of fault that is occurring.

For our problem domain, the choice of case representation depends on what features are available from the network during its operation. In this thesis, we researched possible features from previous Pegasus diagnostic tools and the DRAGON simulator that we could use to represent our router problems. Our research in features concluded that from the available network features, the status messages and the traffic rates from the network could provide our feature abstraction and predictability. Status messages are produced during most network problems and our analysis may indicate that the status messages can diagnose a particular failure in the network. Traffic rates are available in

every line in a network. These traffic rates are sent on a regular basis (every one or two minutes) to collection sites and are stored in a database for future reference. Our analysis of the traffic rates may find that the traffic rates provide some predictions on the type of problem occurring in a network; the traffic rates may also indicate the degradation of a router before a failure does occur.

From our goal of feature abstraction, the status messages and traffic rates appear abstract enough to be included in various types of faults that can occur in a network. In order to study the predictability of status messages and traffic rates in diagnosing our three router problems, it will be necessary to collect and observe these features during our router faults. To do our analysis we will analyze the two features separately. The features will be separately placed into a suitable case representation to form two separate cases: status message cases and traffic cases. A matching, retrieval and adaptation phase will then be implemented to measure how useful the features will be to diagnosing new input problems. As an example, the status messages for a router corruption, large file transfer and storm will be collected and formed into a case. Library cases will consist of status messages collected during known router faults. A problem case will be the collection of messages during a new fault. A matching phase will take the problem case and one of the library cases and a match value will be computed based on the similarities between the two cases. The accuracy of the diagnosis will be found by observing if the best matched library cases can be matched with the same type of router fault in the problem case.

Our tests will actually form two separate case-based systems: a Status Message CBR and a Traffic Rate CBR. As a final solution for solving our three router problems, these two CBR systems will be merged to form single Pegasus Case Based Reasoner. Chapter 6 and 7 will discuss the development of our Status Message CBR and Traffic Rate CBR respectively. Chapter 8 will discuss the joining of these two systems to form the Pegasus Case Based Reasoner.

5.3 Simulation Assumptions

Before continuing in the analysis of the three problems faults, four assumptions on the network performance using the simulator will be used:

1. All tests will be assumed to be in heavy network operating condition: The heavy network conditions simulate a large network operating at its peak level; this level is a worst case condition at which the three types of faults could occur. For our analysis, a heavy network condition is assumed to be 3000 workstations transferring 1500 small files through the network;
2. The packet loss rate = 2: The packet loss rate is a simulator parameter that would allow packets to be dropped after a certain period. A loss rate of 2 is considered to be a good network performer. Packet loss rates of five or more are considered to be severe network degradations;
3. Traffic in the network elements follow a normal distribution pattern: Our traffic analysis involves computing mean and variance values of the traffic rates over an interval of time. Network models, as discussed in [Filipiak91], have used normal distribution patterns for modelling traffic rates given that traffic arrival times to network nodes were Poisson processes. We will assume that the simulator as set in 1) will provide these Poisson arrivals; and
4. Our method of router corruption, file transfer and storm induced faults is assumed to be an accurate model of real network situations: For router corruption faults, the simulator will directly corrupt the router tables that it uses in simulating the transfer of packets. This corruption is assumed to produce results common in real network router table corruption situations. For our file transfer abnormality tests, we will assume a file-transfer sizes of 5 to 10 megabytes at a 512 byte packet size. The file transfer will be set to occur

through the path of the router being investigated (this file size was actually observed to be large enough to produce a good representation of a heavy network load for a long period of time). Storms will be induced using a repetitive file stream of 5000 bytes at 60 second intervals. The storm path will also be set to occur through the router being investigated. Large repetitive data transfers through routers will be our common storm fault.

5.4 Discussion

Router Corruptions, Large File Transfers and Storms are three router problems that we will diagnose using a CBR system. A CBR system that can provide a quick and accurate diagnose of these router problems will be of benefit to the network operators. Based on our analysis of the selection of network features for our case representation, our case based system will be divided into two separate CBR solving systems using status messages and traffic rates respectively. Chapter 6 will discuss the status message features, and Chapter 7 will discuss the traffic rate features. These two systems will be combined to form our final Pegasus CBR system. Our assumptions for the operation of the network simulator as discussed in this chapter will be used for the remainder of our CBR tests.

Chapter 6

Status Message Features

The status messages for use in a case based system will be analyzed in this chapter. These status message reports are one of the first indications that a network operator receives on a possible fault in the network. At first glance at the messages, the network operator may not realize the type of fault that has occurred, but only that it has occurred in a particular area of the network. For our analysis of the status messages for use in a case based system, it will be necessary to measure the predictability of the messages to diagnosing faults. This predictability will be a measure of how well the status messages can be used in predicting the type of fault that has occurred. If the status messages cannot be used to predict the type of fault, then their use in the case based system will be irrelevant to finding a solution.

To test the predictability of the messages for diagnosis, a Status Message CBR will be formed. This CBR will use a status message case library. The library cases will consist of a collection of status messages and a solution for each of the three types of router problems. A problem case will consist of status messages collected during a new router fault. The library cases will then be retrieved and compared to the new problem fault. The comparison will consist of calculating a match value to measure the number of features that match between the problem and library case. The library case that has the best match value will be considered the best solution for the problem case. The accuracy of the status

messages diagnosis will be found by measuring how accurate the best retrieved solution is to the input problem's actual solution.

6.1 Formation of the Status Message Case Based System

To begin the development of our Status Message Case Based System, a computer program will be developed with three phases of a CBR system as outlined in Chapter 4. These phases are: 1) a case representation and indexing phase, 2) a matching and retrieval phase and 3) an adaptation phase. The phases of this system will include the following design and will be further explained in the following sections in this chapter:

1. **Case Representation and Indexing:** The status messages from the network will be automatically collected and placed into a suitable text based case representation. It was decided to use a text representation so the cases can be easily readable and editable. The library cases will be stored individually, and will be completely retrieved for the matching process with the problem cases. This type of retrieval is similar to the indexing scheme of an associative memory pattern, but no specific mask will be used to retrieve the cases i.e. the cases will not be searched on specific fields for retrieval, but rather each entire case will be retrieved;
2. **Case Retrieval and Matching:** The retrieval phase of our case based system will involve the complete reading of a stored text-based case in the library. The matching between a new case and a library case will involve the matching of each field in the case representation using the nearest neighbor approach; and
3. **Solution Adaptation:** The adaptation phase of our system will involve the direct retrieval and application of the solution from the best retrieved case. As an example, the best matched library case solution of 'Router_Corruption' will be shown to the CBR user as a solution to the new input problem case.

6.2 Initial Case Representation, Matching and Adaptation

Initially, we will analyze the network status messages during simulated network break faults. We will initially study router break faults because they produce fewer status message reports than some of our router problems; the fewer messages will allow for an easier analysis of the status messages. Our research results in the small collection of messages during break faults will then be applied in our status message case based system for solving our three router faults.

For our break fault tests, network elements will be broken in the simulator and the status messages from the breaks will be collected. The collection of messages from the network will commence from the beginning of the fault for approximately 30 minutes of real-time. The 30 minute time-limit is considered a suitable time at which the status messages would appear from the network (in fact, a period of 20 minutes was later observed to be suitable for collecting initial reports). Using our designed CBR system, a case representation format for the collected messages will be chosen as shown in Figure 6.1; this format will form our storage of the status reports into a text-formatted cases representation. As an example, Figure 6.1 shows the collection of messages during a router break fault for router 'nRTPc1'. In this example, eight status messages were collected during the fault in a thirty minute period from the time the router was broken.

For our break fault tests, 14 breaks will be collected: 2 workstation broken faults, 1 bridge broken fault, and 11 router broken faults. These break fault tests will be stored as separate library cases of four fields of data as shown in the format of Figure 6.1. These fields include the node_name, the port number, the node_type and the status value of the status message. It should be noted that in a real network environment, the status report is generated with the node name, port number and status value. Using a network topology file, the network operator can also find the node_type of node is being reported (for example, the fact that node nKTPc1 is a router is available from the network topology). In

	node_name	port_no	node_type	status value
(1)	nRTPc1	4	RT-LINE	11
(2)	nRTPc1	-1	RT	11
(3)	nRTPc1	3	RT-LINE	11
(4)	nRTPc1	4	RT-LINE	10
(5)	nRTPc1	-1	RT	10
(6)	nRTPc1	3	RT-LINE	10
(7)	nNSHc2	3	RT-LINE	1003
(8)	nANNc2	3	RT-LINE	1003

Solution → Router 'nRTPc1' is down
 Note: Port number reporting fault(-1 means the node itself)

Figure 6.1 Status Messages Collection during a Break Fault

our system, our status message collection process will be designed to automatically search the topology file for the node name and report the node type with the status messages.

Our status message matching process will be designed to match on the four fields as shown in Figure 6.1. To test the system, a collection of messages during a new break fault will be taken and formed into a problem case. This problem case will then be matched with the 14 status message library cases. For our initial design, the matching procedure will involve matching every input status message from the new problem case with all of the messages in a retrieved library case. A match value for each status message will be generated by using the similarity metric discussed in Chapter 4. Figure 6.1 shows an example calculation of the similarity match between a single status messages in a problem case and library case. To match the entire collection of messages in a case, the problem and the library case messages that have the best match will be removed from the case matching process, and the matching process will continue until all the problem status messages are matched with the retrieved library case. A case similarity match will be found by adding up all of the similarity values for the problem and library status messages that are matched (see Figure 6.1). The best library case from all the library cases matched will be chosen as the case that has the largest case similarity match with the problem case. The adaptation phase for our initial design will consist of the direct application of the library

Figure 6.2 Status Messages Matching

cases solution to the problem case.

6.3 Initial Status Matching Results

Tests using a designed case based system of Section 6.2 were conducted using 14 status message break fault library cases and a collection of input problem break faults.

These test results are not provided in this thesis, but our conclusions are summarized

below.

One of the conclusions from the test results is that the best predictable features for diagnosing break faults are the node type and status values. For example, these features offer some distinguishability between either a router being broken or a bridge being broken. This distinguishability is seen in the collection of messages during a simulated 'broken router' where we expect reports of status value 10 from the broken router, status value 10 from the lines of the broken router, and broken line reports of value 1003 from the routers around the broken router. If a bridge element is broken, we can expect reports from only the broken bridge and no reports from a router; a bridge fault will also produce no status report values of 1003.

Other conclusions are that the node name and the line number features do not provide any benefit in predicting the type of fault that is occurring. This is expected since the node name and the line number are very specific features i.e. the matching on the node name does not allow any prediction for the same fault occurring on another node with a different node name. Another conclusion from our tests is that the number of status messages reported during a fault varies with the router being tested. A router in a heavy traffic area tends to produce more status reports than a router in a lower traffic area. A heavy traffic router is usually based on the number of lines that the router is servicing; the more attached lines that a router has the more traffic on the router. Although not used in our initial tests, it may be necessary to better match routers based on the number of lines that a router has attached. As an example, it may be better to match router A that has six lines to router C with five lines than to router B that has only three lines.

Our results from our initial tests show that the status messages can provide some predictability to diagnosing the type of fault. This predictability is shown in Graph 6.1 which shows a bar chart of computed match values using our similarity metric. This graph shows the calculated match values for matching a broken router fault in node nRTPc1 with

Graph 6.1 Match Values for Problem Case nRTPc1 with Case Library

the case library of 14 cases; case 13 is the expected perfect match to the reported problem and the cases in workstations(case 1,2,3) and bridges(case 4) are the expected lowest case matches. The similarity metric matching works well for finding the best matched case but the major problem that emerged in testing is that this type of matching technique cannot account for multiple faults in the same collection of status messages. If more than two router faults occur in the same interval, the similarity metric will treat the messages as one problem. Based on these problems in the initial case based systems, we decided to further evolve the representation and matching of status reports using graph theory.

6.4 Status Message Graph Representation, Indexing and Matching

Graph theory is a technique that represents elements and paths from elements as a group of nodes and arcs respectively. The paths from the elements can also have a length and direction associated with them. Graph theory is not new to network communications, and successful application of graphs was used by [Kalab93] to find shortest distance paths

in a network. Figure 6.3 shows a graph representation of the connection between nodes in a network. Using graph methods, the shortest distance between the nodes A and F in Figure 6.3 would be calculated as the path A-C-F with a distance of six.

Figure 6.3 Graphical Representation of Network Node Connections

Our novel graph matching system will apply the same graphical representation to status messages. Given a collection of status messages, graphs will be drawn to show the interconnection of the reporting of the messages. The goal of this method will be to group messages into specific clusters, so that each cluster may be analyzed separately for fault reports. This method can thus provide some detection of multiple faults in a group of messages because status reports from different parts of the network will not become connected together. To use a graphical method, our status message case fields will be further expanded to include two more fields. These fields will be called the connection node and the connection type. These fields are available from the network topology file and tell the network operator what node and node type is connected to the status message report. In our case based system, these two fields will be added to our status message collection process. To store these graph structures, the representation of the status messages in the cases will take the form as shown in Figure 6.4 (this particular collection is from a router table corruption). Our previously developed status case based system will be updated to include this new case representation.

Applying our graph methods to the status messages of Figure 6.4 yields the graph

Figure 6.4 Status Message Case Structure

structure of Figure 6.5. Figure 6.5 also explains our criterion for developing the graph method for the status messages. The depth of the status message graph will be set to 1; this

Figure 6.5 Graphical Representation of Status Messages

depth will include only the problem nodes and their immediate neighbors. A suitable matching scheme will also be required for these graph forms. As discovered in our initial similarity test for break faults, our matching algorithm should take into account the status

value, the reporting node and the number of nodes reporting. Using the graph representation, these three values will be shown as the length of the arc, direction of the arcs, and the number of arcs respectively. Our new representation will also include the nodes and type of nodes that the status messages reports are connected to.

To perform a match between a library and problem graph, we will use methods researched in [Bunke83]. Bunke developed methods for in-exact graph matching for structural pattern recognition. He used a search tree to search for all possible combinations of matches between two graphs. Using [Bunke83]'s search tree method, our case based matching process will be expanded to construct a search tree of possible graph matches between a problem graph case and the library graph cases. An example of our graph

Figure 6.6 Search Tree for Graph Matching

matching process for a match between two status messages graphs is shown in Figure 6.6.

In this figure, the problem case and library case for the match process are also shown.

Note that no node names will be required in the graph matching process because the node names were discovered to be non-predictive features for diagnosis. The procedure we will use for our graph matching will be divided into eight steps and is described as follows:

1. Collect status reports from the network simulator for a simulated twenty minute period. From the collection of the messages, use the network topology file to include the type of node reporting, the node that is connected to it, and the type of node connected to it. If the node reporting a problem is not a port problem, then include all of the nodes ports in the collection of status messages. Form the status messages into a text file as shown in Figure 6.4;
2. From the first status message in the status message collection, begin forming a graph of the node connections. Start the graph by taking the first status message as the first node. Continue the process by checking if the next status message will connect with the graph. If there is no connection possible, then check any of the other graphs or start a new graph. At the end of this process, the status messages will be in groups of graphs. For our system, each graph is a collection of status messages that is a possible problem report from the network. The problem graphs will be of a text form of the graph shown in Figure 6.5;
3. Starting with each problem graph, try to match the problem graph with the library cases. Every library graph is matched with the input problem graph;
4. The graph tree matching process starts by taking the nodes with the most connections in the problem and library graph. The node with the most connections in either the library or problem graph is used as the starting node (note: for our description, we will assume that the starting node is in the problem case. The roles of the problem and library cases would be reversed in descriptions 5 and 6 if the starting node was in the library case);
5. From the starting node in the problem case, to find the node that is most similar to the nodes in the library case. Search all of the nodes in the library case for a match. The matching process involves finding a match to a node based on the node_type, the status value and the connection type. The matching process also tries to match on the nodes with the same number of connections. For every match with a node between the problem and library case, a branch in the search

tree is added. In Figure 6.6, the starting node would be node 3 of the problem case; the best matched library nodes are nodes 1 and 4 so two branches are formed for the starting node;

6. Starting with the first branch that is formed, begin the process again by matching the next nodes that are found in the branch nodes. The branch nodes now become the new starting nodes. The same matching criteria are used, and new branches in the search tree will be formed for all matches with the starting nodes. In Figure 6.6, the first branch is problem node 3 and library node 1. A match on problem node 3 with library node 1 finds that problem node 3 has no status values of 11 coming in from another router and no status values of 1003 leaving to another router. There is no match at this node and the branch is stopped. The matching process then continues with problem node 3 and library node 4;
7. The match process continues for all branches of the tree. The matching stops in any branch of the tree that has no matches; and
8. In completion of the process, the search tree will contain a configuration similar to Figure 6.6. This search tree is then read for the longest tree branch that was formed in the matching process. The number of best matched nodes are used for computing the match value for the problem and library case.

For Figure 6.6, the matching of the two graphs shows that there are twelve combinations of best-matches in which four nodes can be matched at one time. Using this tree, a match value for the match between the problem and library case will be computed by using a relation between the number of nodes that matched between the problem and library cases (Equation 6.1). This match-value will also include a factor to favor matching graphs which contain the same number of nodes; this factor will be called the *number of nodes ratio*. This number of nodes ratio will be added because of our observations in the router break fault tests which discovered that nodes in heavier traffic behave differently than lower traffic routers. This number of nodes ratio will also account for the fact that storms, in general, produce more status reports than router corruptions, so graphs of the same density should be closer matched. For Figure 6.6, the match-value of the two graphs

will be calculated as 3.33 - also shown in Equation 6.1.

$$\text{number_of_nodes_ratio} = \frac{\text{minnodes}(\text{problem}, \text{library})}{\text{maxnodes}(\text{problem}, \text{library})}$$

$$\text{match-value} = \text{number_of_nodes_matching} \times \text{number_of_nodes_ratio}$$

Example Calculation of Figure 6.6

number of nodes in the problem case = 5	← minnodes = 5
number of nodes in the library case = 6	← maxnodes = 6
number of nodes matching = 4	

$$\text{match-value} = 4 \times \frac{5}{6} = 3.33$$

Equation 6.1 Match-Value Calculation for Graph Matching

6.5 Status Message Matching Tests

Using our graphical representation of Section 6.4, status message collections from the simulator for our three types of router problems were performed. For these tests, specific router's in the network were located to represent various levels of network loads and router performance. The routers were then given separately administered router corruptions, file transfer problems and storms using the simulator, and the status messages resulting from these faults were collected and formed into graph library cases with solutions.

Using our representation of Figure 6.4, a previously solved case library was formed that consisted of 15 cases of graph structures: cases 0-7 were router corruption graphs and cases 8-14 were storm graphs. Our tests used four unsolved problem cases to calculate the accuracy of the graph matching algorithm: 2 router table corruption(RC) and 2 storm(ST) problems. Our matching used k-nearest neighbor graph matching with k=5, and all library graphs were retrieved to find the best matches.

6.6 Status Message CBR Results

In the collection of the messages for our status message graph tests it is noticed that only router corruptions and storms produced status messages. It appears that the large

file transfers do not degrade the network performance to a degree for $t^h \geq$ routers to unacknowledge a management request. For router corruptions, it is also noticed that the node that is corrupted never appears in the status reports; however, it does appear as a neighbor to the nodes that report a problem. An explanation for this is that a router table corruption causes a router to re-route packets along the wrong data path. The data collector traffic that is passing through the corrupted router is also affected and may not reach either the data collector or the router that is being interrogated. Since the management packets don't reach the intended destination, the interrogated router is reported as failed by the data collector. The corrupted router, however, still can handle its own interrogation without being reported as faulty. Storm problems have a different effect on status reports. It is noticed that the status messages reported from the network during a storm form a graph which closely represents the path of the storm. An explanation for this is that the storm path affects most of the routers that it passes through. As a result, the routers in the storm path cannot properly respond to their interrogation and will produce status reports. It is also noticed that the storm produces status reports from the neighboring routers to the storm path because the interrogation of these routers is also affected. Since router corruptions and storms have separate status reporting patterns, the graphs of the status messages are expected to be good predictors for diagnosing storms and router corruptions.

The test results for the matching tests are shown in Table 6.1.

Table 6.1 shows an accuracy of 3 out of 4 cases, or 75%, correctly diagnosed based upon a $k=5$ value. The problem storm case through nBASc1 provides an error in the matching because the storm set through this node is through a short path and the small number of collected messages provides a more similar match to router table corruption cases.

Problem Case	Actual Fault	Graph match value	Library Case no	Library Type
nCLDc1	Router Table Corrupt	4.29	1	RC
		4	0	RC
		4	3	RC
		3.33	4	RC
		3.33	10	ST
bMERc3	Router Table Corrupt	6	3	RC
		4.29	5	RC
		4.17	10	ST
		4	0	RC
		3.4	1	RC
nBASc1	Storm	5	10	ST
		4.17	3	RC
		4	4	RC
		3.33	0	RC
		3.2	2	RC
nCSWc1	Storm	6	13	ST
		5.13	8	ST
		5	14	ST
		2.41	12	ST
		2.4	0	RC

Table 6.1 Status Message Matching

During graph matching, we realized that the use of status message will not be good predictors for our three router fault types. Three problems are as follows:

1. Of the 121 routers tested in the network, only 23 router corruptions produce status reports. Of these 23 router corruptions, only 10 reports include the node that is corrupted. If these reports are to be used as a feature of prediction then the node that is corrupted must be included in the graph. Some investigation was completed on the depth of the status messages required for all router corruption nodes to be included. If the status reports are graphically handled to include the nodes around it to a depth of 4, then most router nodes that cause the corruption will be included. However, the search time and complexity of the search tree in trying to match such a large graph structures would be very slow. Also the large graph structure would start to resemble the matching of the storm graphs, so the accuracy of matching would be expected to decrease;

2. Status message reports do not appear in file transfer problems, so file transfer abnormalities cannot be predicted using status reports. The status messages matching works well for input problem cases of storms and router corruptions, but the matching of status reports may not be useful if our case based system expands to include other network problems; and
3. As mentioned previously, the path of the storm in a network can be somewhat characterized by the status reports; however, not all status reports from storms provide this observation. In some cases, the storms produce such a large amount of status messages that matching on these structures would be time consuming using a graph tree matching method.

6.7 Discussion

In this chapter, status message features are used in a case based system to diagnose router table corruptions and storms. Large file transfer problems are not diagnosed with status reports because no status messages are produced during this type of fault. For our status message cases, a graph method is used to find best matches. The graph method is successful for the table corruption and storm problems because the structure of the status messages that are reported during a network fault varies between the router and storm problems. The accuracy of our system is 75% with a k value of 5. The status message cases; however, are not very good predictors for the three router faults because of the lack of status reports during the large file transfer problems and the small collection of faults in router corruptions. The next chapter will discuss the development of a case based system for diagnosing our three router problems based on traffic rates features.

Chapter 7

Traffic Rate Features

The selection of traffic rate features and their use in a case based system will be discussed in this chapter. To find our traffic rate features, we will collect various traffic rate measurements on selected routers in the network. From these measurements we will select the features that observably change when a router fault occurs in the network. To test our selected traffic features, this chapter will further discuss our formation of a CBR system that uses traffic cases. The goal of this traffic feature CBR system will be to store a collection of traffic rates from a router while it is experiencing one of our three router faults. This storage of traffic rates will form a library case, and the type of fault that occurs during the traffic collection will also be stored in the library case as the solution. When a new problem occurs with a router in the network, the traffic features will be collected and represented as a problem case. The problem case will then be matched with the available library cases, and the best library matches to the problem case will be used to diagnose the new fault. The accuracy of the traffic rate diagnosis will be found by measuring how accurate the best retrieved solution is to the problem case's actual solution.

7.1 Selecting a Measuring Technique

Before collecting the network traffic features for our cases, a suitable technique will be required for measuring the traffic rates from the network routers, and in this thesis we will use a measuring technique called trend analysis. This trend analysis will be a

measure of a routers traffic performance over a specified time interval. It will be hypothesized that this trend analysis may reveal a router fault by observing the steady change of a router's traffic performance over a period of time.

In a real network operations center, emphasis is usually placed in using traffic threshold level indicators to flag a router has having a traffic flow above a set limit. The threshold levels, however, are very difficult to set because they are specific to the device being monitored and the load of the network at the time of monitoring. Using our trend analysis technique, a network operator could be warned of a possible router failure because the changes in the router's performance may show as an increasing or decreasing trend before a set limit is reached. As an example, the increasing traffic trends in a router may show that the router is experiencing a large file transfer abnormality. Router trend analysis will also not be specific to the type of device being monitored or the load condition of router because the trends measure changes in rates and not actual levels.

Our procedure for calculating the trends in the traffic rates will be based on the use of a running window smoothing technique. The running window method will involve finding the average of a collection of traffic values from a router over a specific time interval. The average will be calculated by taking a current interval of traffic values and adding it to the sum of a previous amount of traffic values. After the average is calculated, the measurements will be shifted forward in time by one interval for the next set of calculations. From the running window definition, the time interval for the average calculation is called the 'window' and the 'running' term implies the shifting of the intervals forward for every calculation.

The purpose of using a running window method will be to absorb sudden fluctuations in a routers traffic that occur in the normal operation of a router. These fluctuations occur because the router's traffic consists of various groups of packets entering and leaving the router at unspecific intervals i.e. an example would be the

network users sending large and small files to various locations at different times. For our trend analysis to work properly, the natural fluctuations in the router's performance should be filtered out so that only the basic trend pattern of the router is used for our traffic measurements. Using a running window technique, the fluctuations will be absorbed because of the averaging of the traffic values in the window. A basis for our trend analysis technique actually comes from [Maxion90] who shows how a running window technique is able to absorb sudden fluctuations in the network and yet still provide fault observations of the traffic data.

7.2 Windowing Method Tests

To show the smoothing technique of the windowing method, Graphs 7.1 and 7.2 show the mean bytes out of router nDNDc1 before and after a window method was applied. The values for Graph 7.1 were calculated by designing a traffic collecting program to automatically poll router nDNDc1 on two minute interval periods. The traffic program provided nDNDc1's total mean bytes out for each two minute interval. For Graph 7.2, the traffic program calculated the traffic mean bytes out of nDNDc1 using a 30 minute running window period. This 30 minute window was defined as the router's traffic mean using the most recent interval and the previous 14 intervals of traffic collections. The running window mean was the sum of these 15 intervals divided by 15. The traffic samples for both samples were taken over the same 200 minute time period. Graph 7.2 shows how the windowing method smoothed fluctuations in the traffic rates. It should be noted that the window calculations can only start after the first 15 intervals of simulation time to allow the window to initialize.

From our tests on the running window, it is observed that the size of the running window is an important consideration for collecting a router's traffic. If the running window size is too large, the resulting traffic pattern will be very 'flat' and may not show any trends in the traffic data. If the running window is too small, the resulting traffic

Graph 7.1 nDNDc1 Router Mean Bytes Out -No Windowing, 2 Min Intervals

Graph 7.2 nDNDc1 Router Mean Bytes Out - Windowing 30 Minutes, 2 Min Intervals

pattern may contain too many fluctuations which would make the trend observations of the traffic very difficult. The running window method offers a good method for smoothing the traffic patterns.

7.3 Trend Analysis

Using the running window technique as a basis, our trend analysis technique will involve taking a current sample of mean bytes out from a router and comparing it to a set of previously sampled rates. For this trend analysis, we will require a window pattern that will reflect accurate changes in the network performance over a number of samples i.e. a running window that will not be too small or too large.

In our trend analysis we will define the set of previously sampled rates as a reference window. Figure 7.1 shows an example running windowing method using a 20 minute reference window and 6 minutes of current samples. Our measures for the trends

Figure 7.1 Windowing Example of 10 min ref, 6 mins current

will consist of calculating the percent change in the mean of the traffic measured from the current interval to the mean of the traffic in the reference window. After every calculation, the running window will be moved ahead the length of the current intervals to perform the next set of calculations. The trend of the traffic over a set of intervals will be found by comparing a set of the stored percent change calculations. The number of intervals that are compared for trends will be called the *trend window size*. As an example calculation, if a

trend is upward in a router over a trend window size of 5, it will mean that the mean of the current interval traffic measurements is larger than the mean of the reference window for 5 consecutive comparison tests.

7.4 Trend Analysis Tests

To find the best trend measuring technique, tests were conducted on router nDNDc1's mean bytes out. The purpose of these tests was to find the best windowing method to calculate the trends in the routers. To find the trends in a router, our traffic collecting system for the windowing tests of router nDNDc1 was expanded to calculate the trends of the traffic over various sizes of current and reference window periods. Using this system, five windowing tests were conducted on the traffic rates of the bytes out of a router before and after a router-table corruption. The five windowing intervals that we studied included the following tests:

1. Window of 18 minutes reference, 2 mins current;
2. Window of 30 minutes reference, 2 mins of current;
3. Window of 30 minutes reference, 6 mins current averaged;
4. Window of 32 minutes reference, 4 mins current averaged; and
5. Window of 60 minutes reference, 6 mins of current averaged.

For brevity reasons the results of the five windowing tests are not shown in this thesis. Our results from these tests conclude that the best trend windowing method to use for our future trend tests is 30 minute reference window compared with 2 minutes of current traffic rates. The smaller windowing periods do not show trend lines for a very long period. The larger reference window sizes work well to observe trends, but are less accurate in following the normal changes of traffic loads in the network. The size of the current interval that is compared to the reference window is also an important consideration. As an example, the larger current intervals produce smaller periods of trends during a router corruption fault.

7.5 Selecting Our Traffic Features

Now that our router measuring technique is finalized, the best method of finding traffic features for our case based system will be to observe a group of traffic measurements from a router before and after our three faults are simulated. Based on these tests we should be able to select what features from our initial group are the best measurements for representing our three router faults. For this thesis, our initial group of features will consist of seven traffic features available for all of the routers in our network: 1) the router's traffic mean bytes out, 2) the mean bytes out root-mean-square (rms) value, 3) the variability of a router's traffic, 4) the cost of the router 5) the cost's rms value, 6) the variability of the cost and 7) the zone deviations of a router.

The variability of a router is calculated from the mean and rms values of the features collected (variability is the mean divided by the variance). The cost of the router is discussed in Section 2.1 and is available through the simulator. The zone deviation of a router is a measurement successfully used by [Farrell93] to detect router traffic delay abnormalities. It is derived from measuring the changes in intervals of a zone's contribution levels. The zone contribution level measures a router's performance in relation to its neighboring routers. A router's zone is defined as the router plus all direct routers connected to it at a depth of one link. As an example, in Figure 7.2 if the router of interest is Node A, then the router's zone is defined as Node A plus the nearest routers to it at a depth of one router away. Thus zone A is composed of center-router A and routers B, C, and D; B and C are included because they can be reached directly through the LANs connected to the center-router A. The center router's zone contribution is defined as a percentage of the zone's total bytes out. As an example, in Figure 7.2, if router A's total bytes out for all of its lines for a particular interval in time is 10000 bytes, and routers B, C, and D have a total of 20,000, 30,000 and 40,000 respectively, then router A's contribution to its zone will be 10%. The zone contribution measurement offers a

Figure 7.2 Zone for Router A

correlation measure of a node and the nodes around it. The zone deviation levels are a measure of the change in the zone's contribution from one interval in time to the next. For our measurements, a positive change in the zone deviation signifies a decrease in the zone contribution level.

7.6 Traffic Observations

To observe our seven traffic statistics, two routers will be analyzed before and after our three types of router faults are simulated. The purpose of these tests will be to observe our seven types of traffic measurements in predicting a router fault. As an example, if it is observed that a router's mean bytes out does not behave any differently between a router corruption, a large file transfer or a storm, then the use of the mean bytes out traffic feature would be not be valuable for diagnosing our types of router problems. The two routers for our observations will be nDNDc1 and nANNc1. Figure 7.3 shows router nDNDc1 and some its surrounding network elements. Also shown is the zone by which this node encompasses - routers nEDMc1 and bMERc3. For brevity, only router nDNDc1 figures will be shown in this thesis. For our tests, routers nDNDc1 and nANNc1 are chosen because they are situated in different network areas and are configured with a different

Figure 7.3 Zone nDNDc1 and Surrounding Network Elements

number of network connections. Because of these differences, the traffic pattern of these two routers should not be similar and our analysis of these routers should be representable of the other configurations of routers in the network.

7.6.1 Router Table Corruption (RC) Tests

For these tests, routers nDNDc1 and nANNc1 were monitored for a period before and after a router corruption. The traffic and cost values of the router were collected at 2 minutes intervals and averaged with a running window of 30 minutes. The router corruption was set to begin at interval 100 (or 200 minutes into the simulation) after the running window was initialized.

The following Graphs (Graph 7.3 to Graph 7.9) show the seven types of measurements that we collected for node nDNDc1 and its neighboring nodes. These Graphs show how the mean bytes out, the rms value of the mean, the variability of the

Graph 7.3 nDNDc1 window mean: corrupt at Int 100

Graph 7.4 nDNDc1 window std dev: corrupt at int 100

Graph 7.5 nDNDc1 variability in mean: corrupt at Int 100

Graph 7.6 nDNDc1 cost: corrupt at Int 100

Graph 7.7 nDNDc1 cost deviations: corrupt at Int 100

Graph 7.8 nDNDc1 cost variability: corrupt at Int 100

Graph 7.9 nDNDc1 zone deviations: corrupt at Int 100

mean, and the zone deviation values change when a router is corrupted. The cost values, however, do not show any type of abnormality after the router corruption. It is also noticeable that the rms values and the variability values that are measured from the traffic rates (for all three types of router faults) are very similar comparisons. The variability values do not offer any added benefit to the measurements, so we decided to not include them in future trend analysis measures of the router faults.

These graphs show us that a corruption in nDNDc1 produces a significant change in the router's traffic pattern such that a trend analysis of these patterns should offer some type of prediction that a fault is occurring. These Graphs also show a change in the traffic rates of the neighboring nodes of nDNDc1 after nDNDc1 is given a router corruption - the affect that a router had on its neighboring routers was also noticed by [Farrell93].

The router corruption tests for nANNc1 show similar results except that router nANNc1 has an increase in the mean bytes out during a corrupt' . . The other router's in its zone, as in nDNDc1, are not greatly affected by the router corruption. The cost of nANNc1 does not change during the corruption, but one of the neighboring router's cost values is noticeably affected.

From our simulations with nDNDc1 and nANNc1, it appears that the corruption of a router affects the router's mean bytes out, mean bytes out rms values, and the zone contribution values. The cost of the router is mostly unaffected except for some changes in the cost in neighboring nodes. These simulations show the change in traffic rates for a router corruption, further simulations are required to see if either these traffic changes are predictable only to router corruptions or if the same patterns are observable during large file transfers and storms.

7.6.2 Large File Transfer Abnormalities (FT) Tests

For these simulations, a file transfer was set to occur through the routers being analyzed at interval 100 into the simulation. For nDNDc1, the router file transfer path was set as shown in Figure 7.4.

Graphs 7.10 to 7.14 are the collected traffic measurements for these tests. From these Graphs it is observed that the traffic measurements are predictive to the path of the large file transfer. As an example, the path of the file transfer is through nodes bMERc3 and nDNDc1 and Graph 7.10 shows the mean of the router's bytes out for bMERc3 and nDNDc1 increasing for the transfer time. Further observed is that the traffic bytes out of router nEDMc1 remains unaffected during the file transfer. After the file transfer is complete, the traffic values return to the same level before the transfer. Other observations are that the zone deviation levels for nDNDc1 and bMERc3 show a decrease in the beginning of the file-transfer, whereas the zone deviation for nEDMc1 shows an increase. This is expected since a decrease in the

Figure 7.4 File Transfer Through Zone for Router nDNDc1

zone deviation percentage level signifies an increase in the traffic levels for the zone i.e. nDNDc1 and bMERc3 have an increase in their traffic bytes out so the zone deviation for their zones should decrease. An observation between file transfer and router corruption problems shows that the cost mean and rms values increase for nDNDc1 and nEDMc1 in a file transfer but do not change in the router corruption.

The results for zone nANNc1 are the same. The routers on the path of the transfer show an expected increase in the mean and rms values of the traffic rates. The zone deviation levels are affected among all the routers because of the transfer, and the cost levels are affected in three of the six routers in nANNc1's zone.

7.6.3 Storm Conditions (ST) Tests

For these simulations, the simulator was set to introduce a data storm through the router zone being analyzed at 200 minutes (the 100th interval) into the simulation. The storm was sent through the same path as the file transfer. The same statistics were

Graph 7.10 nDNDc1 mean: file transfer at Int 100

Graph 7.11 nDNDc1 mean rms: file transfer at Int 100

Graph 7.12 nDNDc1 zone dev: file transfer at Int 100

Graph 7.13 nDNDc1 cost: file transfer at Int 100

Graph 7.14 nDNDc1 cost rms: file transfer at Int 100

calculated for nDNDc1 and nANNc2.

The results for these tests for nDNDc1 are shown in Graphs 7.15 to 7.19. The results with the data storms produce similar patterns as those found in the large file transfers except in certain areas. The mean bytes out in the path of the storm shows an increase, but the other routers attached to the node which experience the storm show a decrease in the traffic. In the file transfer problem, it is observed that the attached routers which do not transfer the file will show no change. The storm problem also seems to greatly affect the routers in the vicinity of the storm.

Of the features, the zone deviation values are the most distinguishable features between a file transfer and a storm. The deviations of the zones in the path of the storm show similar results to the file transfer, but the attached routers show a rapid fluctuation in traffic rates. The cost values also show a distinguishable change in patterns. The cost of the routers in nDNDc1's zone all experience a decrease in the cost. The

Graph 7.15 nDNDc1 mean: storm at Int 100

Graph 7.16 nDNDc1 mean rms: storm at Int 100

Graph 7.17 nDNDc1 zone dev: storm at Int 100

Graph 7.18 nDNDc1 cost: storm at Int 100

Graph 7.19 nDNDc1 cost rms: storm at Int 100

cost rms values also reflect this situation. For zone nANNc1, the results experienced for the storm are similar.

7.7 Conclusions from the Traffic Tests

From our three router faults, it is observable that the different router faults produced different traffic features. Router corruptions, for example, produce a decrease in mean bytes out in nDNDc1, whereas a file transfer in nDNDc1 produces an increase. File transfers increase the cost function of a router while a storm decreases the costs. It is also observable that the affects that the other routers in the zone are experiencing during a fault is important to consider in diagnosing the type of fault. As an example, the mean bytes out increase in nANNc1 during a router corruption and a storm. To distinguish on what type of fault is occurring, an investigation of the other routers in the zone reveals that in a router corruption most of the other routers do not have any change in performance; however, in a storm the other routers show either a significant increase or decrease in their mean bytes

out.

It is in the various types of distinctions in the traffic measurements that make the traffic trend values predictive features to distinguish among the different types of router problems. Our conclusion from the router corruptions, file transfer abnormalities, and storm problems tests is that the traffic features do provide an indication of the type of router fault that is occurring. The chosen traffic measurements that we will use for our case feature analysis will be 1) the router's traffic mean bytes out, 2) the mean bytes out root-mean-square (rms) value, 3) the cost of the router 4) the cost's rms value, and 5) the zone deviations of a router.

7.8 Applying Trend Analysis to Traffic Rates

By using of our trend analysis techniques, we will be able to measure the change in traffic levels that we noticed for our three problem tests.

An important aspect of our trend measuring process will be to find the best trend window size to collect features. *Our trend window* will be defined as the number of trend intervals that will be collected to predict whether the router actually has a change in trend occurring. If the trend measurements for a router are upward or downward for the trend window size, then the router will be reported as acting abnormal. The size of the trend window is important to consider. A small trend window size will cause a large number of false alarms to be reported (called the false alarm rate). As an example, if the trend window of the mean bytes out is set to two intervals, then routers will be constantly reported as acting abnormal because the occurrence of two consecutive upward or downward trends occurs often in a routers normal performance. On the other hand, a large trend window size will decrease our accuracy of finding a fault in the network. As an example, if the trend window is set to ten intervals, then a router that exhibits only six positive trend periods during a router corruption fault will not be reported as abnormal.

To accomplish our traffic trend analysis, the traffic collection system that we used to find the traffic rates for routers nDNDc1 and nANNc1 was modified. The modified program allowed the user to select any router in the network for monitoring, and the trend window size was adjustable at the programs initialization. An example of the calculated trends using our trend analysis program is shown in Table 7.1. This table shows the trends of nDNDc1 in router corruption from interval 95 to interval 120 - taken from traffic rates used for Graphs 7.3 to 7.9 using a trend window size of five.

Int	mean bytes out trend	mean rms trend	zone deviation trend	cost trend	cost rms trend
95	-0.354	-2.795	0.03	-0.009	-0.001
96	-0.775	0.668	0.61	-0.033	-0.395
97	-0.650	-0.810	0.23	-0.117	3.073
98	-0.562	-8.148	0.19	-0.088	1.714
99	0.456	-0.371	0.20	-0.110	2.494
100	-4.360	27.752	-3.58	-0.000	-7.214
101	-10.213	82.346	-4.35	-0.070	-0.783
102	-9.136	24.138	-4.51	0.023	-49.771
103	-7.945	13.105	-2.33	-0.105	5.017
104	-7.595	6.426	-3.68	0.033	0.137
105	-7.708	7.443	-3.63	-0.040	-9.717
106	-6.769	4.556	-1.79	-0.008	0.025
107	-5.564	-0.226	-1.99	-0.010	-2.404
108	-5.315	0.616	-1.94	-0.038	-16.563
109	-3.314	-1.870	-0.59	-0.000	-0.480
110	-3.387	-7.317	-4.47	-0.024	-1.945
111	-3.968	-5.279	-0.58	0.006	-6.790
112	-2.142	-8.340	-0.23	-0.056	2.399
113	-2.480	-12.306	-0.32	0.045	0.728
114	-1.729	-20.263	-0.09	-0.067	7.074
115	-0.488	-44.716	0.52	-0.041	-1.391

Table 7.1 nDNDc1 Router Corruption Trends

Int	mean bytes out trend	mean rms trend	zone deviation trend	cost trend	cost rms trend
116	1.776	-57.591	1.08	-0.035	-6.501
117	-1.130	4.639	-0.43	0.073	9.385
118	-0.262	0.027	0.60	0.001	-6.117
119	0.030	-1.932	1.10	-0.006	-7.641
120	-0.474	1.051	-0.47	-0.017	-11.155

Table 7.1 nDNDc1 Router Corruption Trends

Using our trend tables, we are able to further study the details of our collected problem graphs. Further observations using the trends in Table 7.1 are that the change in measured traffic values are most noticeable in the traffic mean, the mean rms and the zone deviation values (notice at interval 100 how these trends suddenly change and either remain upward or downward for a number of intervals). One of our major observations from these trend tables is that a faulty router can be recognized as acting abnormal by the observation of a trend in either the mean bytes out or zone deviation features. A closer observation of the other features can then help reveal the type of fault that is occurring.

For our future collection of trends in the network, we will to use a trend window size of five intervals. A trend window of five is observed in our trend tests on the problem graphs to be a good width to discover the presence of a trends in the faulty routers. Tests will be conducted in Chapter 8 to find the optimum setting of the trend window size.

7.9 Formation of the Traffic Case Based System

To measure the importance of the traffic trends for use in a case based system, a small case based system will be required to store traffic cases with their solutions. These traffic cases will then be retrieved and compared to a new traffic problem case. The best traffic cases will then be used to provide a solution to the problem case. Our results from this case based system will indicate if the traffic trends are good features for diagnosing our three router faults

Our case based system will be developed using traffic cases with three phases of a CBR system as outlined in Chapter 4. These phases will use some of the same programmed methods of the Status Message Case System and will include the following: 1) a case representation and indexing phase, 2) a matching and retrieval phase and 3) an adaptation phase. These phases are briefly introduced below and will be further explained in the next section.

1. **Case Representation and Indexing:** The traffic cases will be stored as a text file so they can be easily readable and editable. The cases will be stored individually, and will be completely retrieved for the matching process with other cases. This type of retrieval will be similar to the indexing scheme of an associative memory pattern;
2. **Case Retrieval and Matching:** The retrieval phase of our case based system will involve the complete reading of a stored text-based case in the library. The matching between a new case and a library case involves the matching of each of the fields in the traffic case representation using the nearest neighbor approach; and
3. **Solution Adaptation:** The adaptation phase of our system involves the direct application of the solution from the best retrieved case. It will then be necessary to keep the solution sets for the traffic messages to simple sentence explanations i.e. 'Router is experiencing a storm'. These simple solutions will provide the required diagnosis for the three types of faults. These solutions can be expanded to include further explanations as our traffic case system develops.

7.10 Traffic Representation, Indexing, Matching and Adaptation

Tests on the network have concluded that the five trend features of traffic mean

bytes out, mean rms, cost, cost rms, and zone deviations offer a good representation of the router problems we are investigating. Since the trend of the window period that we are using can take one of three directions (increasing trend, decreasing trend, and no trend), we will normalize these three directions to numerical values such that '1' represents an increasing trend, '-1' represents a decreasing trend and '0' signifies no trend. From our research in the three router faults, we will represent a problem traffic case as a collection of trends from the zone of a faulty router. This zone problem case will contain the trends of the faulty router and the faulty router's neighboring routers - our earlier research concluded that it was important to include the neighboring routers to predict the type of fault that was occurring. An example case representation for the trends in nDNDc1 during a router corruption is shown in Figure 7.5. The trends consists of number vectors of five

Figure 7.5 Traffic Case Representation

fields of trends as shown in Figure 7.5. These vectors will also allow us to apply the vector analysis techniques of nearest neighbor matching as discussed in Chapter 4.

The collection of the library cases will be conducted using a modified version of our traffic collecting program used to find the trends of the routers. This program will continuously monitor the network and collect trends for all of the routers in the simulator's network. From the results in Section 7.8, this trend measuring process will be designed to produce traffic reports when either the mean bytes out or zone deviation trends

are upward or downward for the trend window size. These traffic reports will take the form of a case as shown in Figure 7.5. This consists of the node reporting a trend and the nodes attached routers. For our tests in this chapter, the trend window size will be set to a value of 5. The collection of reports during known router problems will form our library cases. As an example, if a router corruption is placed on router bKANc1, the resulting report of cases from this fault will be collected and the solution of 'Router is experiencing a router corruption' will be manually added to the case to form a library case. To set up our initial case library, a total of 20 cases will be collected from the network; these cases will be based on a selected number of routers and will consist of a total of 7 router corruption (RC) cases, 7 file transfer (FT) problem cases and 6 storm (ST) cases. To test our matching and retrieval mechanisms, we will use six problem test cases: 2 RC, 2 FT, and 2 ST.

Given the collection of library cases collected for our three types of router faults, the retrieval and matching of these library cases to new input problem cases will be required. For these phases, we will use a retrieval structure with k-NN matching to find best matched cases. Using the k-NN approach will allow us to study the effects of comparing the top k solutions to the problem case. If the top k cases contain a majority of library cases from the same category (e.g. all from a Router Corruption) then our use of the traffic trends for predicting the type of fault will increase our confidence in using these features in our router diagnosis.

For our matching process, the problem cases center-node trend features will be used as the search fields to find the k -library cases with the most similar center-node trend fields. This initial retrieval will form the first search for the best k -matched library cases. A second matching test between the problem case and the best k -matched library cases will then be conducted to find which of the k -library cases from the first retrieval are closer on the attached router trend fields. A final match-value will be found by adding the match-values from the center-node matches and the attached router matches. Figure 7.6

further expresses this approach.

Figure 7.6 Matching Between Traffic Cases

Our matching of the cases into two separate mechanisms will be similar to retrieval mechanisms used in [Simoudis91]. [Simoudis91] divided cases into distinguishable and discriminating features. The distinguishable features were used to retrieve initial cases in a mechanism called surface retrieval. The discriminating features were then used to further discriminate between the surface retrieved cases. Simoudis noticed that the retrieval of the cases on the distinguishing features identified the initial group of cases that were relevant to the solution. The discriminating features were then used to further match the cases in order of best matching. In our situation, using Figure 7.6, match-value1 is calculated using the distinguishing features of the center-node vector and match-value2 is calculated using the discriminating features of the attached router vectors.

As mentioned, the matching tests in our traffic matching will consider the trends in the cases as vectors of five dimensions. This format will allow us to easily apply the matching techniques of either similarity metrics or vector approaches as discussed in

Chapter 4. The selection of the best NN metric to use in our matching algorithm will depend on the importance of the matching of the features. The vector approach of measuring the cosine of the angle between two input vectors is useful when the size of the vectors are not important in the matching process (i.e. if one of the vectors was increased in length by a factor of two, the vector angle would still be the same). The vector angle matching is considered inappropriate for our tests because the size of the vectors may become important in the matching process (note: our current selection of vectors in the cases are suitable for angle measurements but future added features in our case system may not be). Our next choices are the similarity metric and the distance metric. To choose among these two types of matching, it will be necessary to conduct tests to see if the differences in the features are important to consider. Tests for differences in the features will conclude if, for example, it is important to match on the mean trend in the problem case of '1' being closer to a library case mean trend of '0' than to a library case mean trend of '-1'. Figure 7.7 further explains this concept by showing the matching of vectors using a distance metric and a similarity metric. Notice that the similarity metric provides the same match value for the chosen vectors but the distance metric shows that the first vector is a better match to the problem vector. The distance metric shows that the first vector

Figure 7.7 Similarity and Distance Example

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS
STANDARD REFERENCE MATERIAL 1010a
1963-1964 TEST CHART No. 21

point is closer to the problem vector.

7.11 Case Based Testing

A case based system using our designs of Section 7.10 was programmed. The goal of this system was to test traffic cases for use in diagnosing router faults. The case based system was initially developed to use two methods to calculate a match value. These two methods were the similarity and distance metrics. The purpose of the two matching metrics was to find the best type of matching algorithm to use to match the problem and library cases. Our initial traffic case tests used 20 library cases and a k value of 5. The large k value was chosen to provide an indication of the grouping of the best matchings to the problem case.

Our results for the matching tests with distance and similarity matching with 20 traffic cases are shown in Table 7.2. This table shows that our traffic cases are very accurate in diagnosing a router fault. This table also shows that the similarity and distance metric provide the same accuracy in finding the best match cases.

node_name	Fault induced	no.	Best Match Value with similarity	Case #	Case Solution	Best Match Value with distance	Case #	Case Solution
rDNDc1	Router Table Corrupt	1	10	0	RC	0	0	RC
		2	7	4	RC	2.4142	4	RC
		3	7	5	RC	2.91421	2	RC
		4	6.5	2	RC	2.94317	5	RC
		5	6.5	6	RC	3.44317	6	RC
bKANc1	Router Table Corrupt	1	9	1	RC	.80474	1	RC
		2	7.667	3	RC	1.91068	3	RC
		3	7.667	11	FT	1.91068	13	FT
		4	7.667	13	FT	2.1381	11	FT
		5	7	6	RC	2.3570	5	RC
nATLc2	File Transfer	1	10	10	FT	0	10	FT
		2	8.5	7	FT	1.20711	7	FT
		3	8.5	12	FT	1.80474	8	FT
		4	8	8	FT	2.0215	12	FT
		5	7	9	FT	2.6212	9	FT

Table 7.2 Matching Tests for Twenty Traffic Cases

node_name	Fault induced	no.	Best Match Value with similarity	Case #	Case Solution	Best Match Value with distance	Case #	Case Solution
bRTPc1	File Transfer	1	7.8	1	RC	1.83	1	RC
		2	7.8	6	RC	2.162	11	FT
		3	7.5	5	RC	2.183	5	RC
		4	7.111	11	FT	2.219	3	RC
		5	7	2	RC	2.276	6	RC
nFFMc1	Storm	1	8.5	14	ST	1.70711	16	ST
		2	8.5	15	ST	1.70711	17	ST
		3	8.5	16	ST	2	14	ST
		4	8.5	17	ST	2	15	ST
		5	7.5	19	ST	2.41421	18	ST
nCLDc1	Storm	1	8.5	14	ST	1.81421	16	ST
		2	8.4	15	ST	2.5	14	ST
		3	8.2	17	ST	2.689	17	ST
		4	7.6	16	ST	2.76569	15	ST
		5	7.333	19	ST	3.24402	19	ST

Table 7.2 Matching Tests for Twenty Traffic Cases

Based on the small number of library cases that we are using, it appears that the traffic trends are very predictive to the type of router fault. If the trends were not predictive we should have noticed a larger variation in the top five solutions to the problem case. For the most part, Table 7.2 shows the top five cases as being in the same class of solution as the problem case. All problem cases are correctly diagnosed except for bRTPc1's file transfer problem case; both measures incorrectly diagnose a router table corruption problem based on the top five cases. A closer look at bRTPc1's case reveals that the distance metric provides a better matching in retrieving a file transfer case (the similarity metric matches one in 4th position, the distance metric matches one in the 2nd position). All other matches between the two metrics are generally the same; noticeable is that the position of the best cases retrieved is sometimes reversed. The accuracy of these tests reveals an accuracy of matching best cases of 5 out of 6 problem cases or 83%.

A second set of tests was conducted using a case library of 120 cases: 40 RC, 40 FT and 40 ST. Our results from these tests are not shown in this thesis.

Our results from the second test finds that the distance matching tests provides a

somewhat better matching accuracy to diagnosing a router fault. In the case library of 120 cases, the similarity metric obtains an accuracy of 83%, the distance metric obtains 100% using a k-value of 10. This increase in the accuracy for the distance metric indicates that the distance between the feature values should be considered in the matching process.

Our results from the above tests provides very good results in matching the problem and library cases. An accuracy of 83% given only a library of 20 cases shows that our traffic case representation is very good for diagnosing the router faults. If our case representation and matching were inadequate, we would have expected a very small matching accuracy. Our test results prove our original hypothesis of Section 7.1 that the trends are predictive to the type of router fault occurring.

7.12 Discussion

This chapter discussed the formation of our traffic case based system for diagnosing our three router problems. In the formation of this system we found that a trend analysis technique on the traffic rates offers an accurate method for diagnosing router faults. The five trend fields which we use to diagnose the type of router fault using traffic trends are: 1) the mean bytes out, 2) the mean bytes out standard deviation, 3) the cost, 4) the cost standard deviation, and 5) the zone deviation. Our designed Traffic Trend Case Based System monitors the communication network and collects problem cases when either the trend of the mean bytes out or zone deviation is found to be increasing or decreasing for the trend window size. These problem cases are then sent to the case based system for diagnosis. The distance metric matching is used to match the problem and library cases.

Chapter 8

The Pegasus Case Based System

In the previous two chapters, two CBR systems were developed to diagnose three types of router problems. In the development of these systems, case libraries were formed that contained solutions to the router problems. The final set of case libraries consisted of 14 status message graph cases and 120 traffic trend cases. The two CBR systems were developed with separate case representation and indexing, case retrieval and case matching phases. The case adaptation of each CBR system involved the direct application of the solutions from the retrieved cases to the input problem.

In this chapter, the development of our Pegasus CBR system will be concluded by bringing the two separate CBR systems into a single process. In addition, this chapter will present our investigation in improving certain aspects of the traffic trend CBR system. The status message portion of our CBR system will not be further analyzed in this thesis.

To analyze the traffic CBR system, this chapter will show the implementation of a 'size of zone' measure and feature weights in the traffic matching phase. A comparison of the matching accuracy from using a case library of 120 to 240 traffic cases will be conducted, and feature dependency factors will be used to test the relations of some of the traffic features. Finally, the false alarm rate in collecting traffic cases will be discussed.

8.1 Forming the Pegasus Case Based Reasoning System

A goal in this thesis, as outlined in chapter one, was to form a Pegasus CBR system to diagnose router problems. To reach this goal, we will develop a Pegasus CBR by combining the four phases of the Traffic Feature and Status Message CBR systems developed in the previous two chapters. The combining of these four phases will be described in Sections 8.2 to 8.4.

8.2 Combining Case Representations and Indexing Phases

The combined representation phase of our Pegasus CBR system will consist of two separate storage facilities for the cases. The case representations for the status message cases will take the form of Figure 6.4 and the traffic case representations will take the form of Figure 7.5. The indexing phase for our system will use the separate associative retrieval mechanisms as discussed in Chapter 6 and 7.

From our Traffic CBR development, we concluded that our traffic case based system was very accurate in diagnosing router faults. The status message CBR system did not provide such an accurate diagnosis system as outlined in Section 6.6. In forming a Pegasus CBR, we will keep the traffic and status message cases stored independently. This type of storage will increase our confidence in a diagnosis by allowing for a diagnosis of a fault using two types of cases. As an example, we will be more confident that a fault is correctly diagnosed if the traffic and status message cases provide the same solution. It should be noted that our division of cases into two independent features will be similar to research conducted by [Redmond90]. In [Redmond90], cases were separated into separate pieces of information which were called snippets. These snippets were a feature or group of features that were available for the matching phase a CBR to use. These snippets of information increased the accuracy of Redmond's CBR system because they allowed for the retrieval of multiple parts of previous cases to provide solutions to a single new problem.

8.3 Combining Solution Adaptation Phases

Adaptation phases, as discussed in Section 4.1.3, are concerned with modifying retrieved solutions using differences in the problem and library features. In our final CBR system, we will use an adaptation phase that will directly apply the retrieved solutions to the input problem. We will not implement a very detailed adaptation phase, such as the structural or derivational phases, in our combined CBR system for two reasons:

1. Our accuracy of 100% shows that our traffic representation is very accurate in retrieving best solutions. An adaptation phase of the structural or derivational type will not provide any better results for providing solutions to our three router problems; and
2. Our solutions to router problems will involve only a general diagnosis of the router faults. These solutions are not specifically designed for adaptation rules. Our solutions set consist of simple sentences to describe the type of problem i.e. such as 'Node is experiencing a Router Table Corruption'. Detailed adaptation rules are usually applied for solutions that either require mathematical results or that have many features that could be used to bias the possible solution to the input problem.

In future requirements for our Pegasus CBR system, an adaptation phase could be used to offer further information from a newly solved case such as the size of the trends or the number of status messages being reported. We will not implement these types of adaptation in our system because we are presently more concerned with testing our retrieval and matching mechanisms. If our CBR system expands to include more types of router problems and more features, the solutions may become more detailed such that an adaptation phase would be required. As an example, further investigation may reveal that an adaptation phase could be employed to distinguish on the different types of router

corruptions such as 'router table corrupt because of corrupt update from Node X' or 'router table corrupt because of internal router error'.

8.4 Combining the New Case Presentation and Storage Phases

In our case matching and retrieval systems of Chapter 6 and 7, the top k-cases solutions that were selected from the matching processes were presented to the user. Based on these k-solutions, the user was able to view the cases and decide what was the best solution. Based on these systems, we will have our Pegasus CBR system show the CBR user the same viewing procedure. In this procedure, the best k' traffic solutions and the best k'' status message solutions will be presented to the user for every input problem. The value of k' for the traffic matching process will be set as a user input value when the CBR process is initialized. The k'' value for status message matching will be set for a value of 5 based on the results we experienced in Chapter 6. Further in this chapter, the best value of k for k-NN matching in the traffic case matching process will be tested.

Based on the k-nn best traffic and status message solutions, the user will be able to decide on the correct diagnosis of the router problem. The diagnosis in our CBR system for a problem node for both the traffic and status message cases will be one of the three following types:

1. Node is experiencing a Router-Table Corruption;
2. Node is experiencing a Large File Transfer; and
3. Node is experiencing a Storm.

Our CBR system will only allow the storage of the accepted traffic cases - the status case storage after acceptance of the solution will not be implemented. Future effort will be required to have the cases properly indexed and stored into the case library.

8.5 The Pegasus CBR

A Pegasus Case Based System using the above descriptions was programmed and

developed using C++. This Pegasus CBR is shown in Figure 8.1.

Figure 8.1 Pegasus Case Based Reasoning System

In the operation of our Pegasus CBR system, the user will be able to set the CBR to act in either a monitoring or user activated mode. In the user activated mode, the CBR user will be able to start the collection of trends and status messages into problem cases on the user's request. In the monitoring mode, the CBR system will be set to continuously monitor the network router traffic and it will create problem cases when a router's zone deviation or mean bytes out trend changes for the trend window size (our previous research in trend cases showed that the mean bytes out and zone deviations were the best trend parameters to observe in a faulty router). The trend window size for the traffic collection will be set as a user defined parameter at start-up. In the monitoring mode, traffic measurements from all the network routers will be taken at two minute intervals.

If a problem case is found in the monitoring of the network, the traffic trends and status messages will be collected. The status messages will be collected for the previous twenty minutes of simulator time. For every traffic problem case that is collected, the center router name from the collected traffic case will be searched in the status messages. If a match is found, the status messages with that name will be formed into a status message graph case. The traffic case and its status message case will then be used as input to the matching and retrieval phase of the Pegasus CBR system. The best k-NN solutions to the traffic cases and status message case will then be outputted to the user for the explanation and storage phase. The user will be able to either accept or reject the new solutions.

8.6 Testing of the Pegasus Case Based System Traffic Cases

The above sections formed our Pegasus CBR System. From the formation of this system, we will further test portions of the traffic case matching and retrieval mechanisms. We will use the traffic cases because of our earlier success in chapter 7 with the accuracy of the traffic cases in diagnosis. Also, our tests will require a large number of cases to study which will be more obtainable with traffic cases than with status messages cases i.e. Chapter 6 discussed the absence of status messages in large file transfer problems and how very few status cases are available in a router table corruption.

The study of the traffic trend cases will cover the following areas: accounting for the size of the zone in a case, calculating feature weights, increasing the case library size, varying the value of k in nearest neighbor testing, checking traffic feature dependency, and discussing traffic false alarm reports.

8.7 Size of Zone

In earlier stages of the development of our CBR system it was noticed that routers which had a minor role in routing traffic in the network behaved differently during a fault than the routers which were considered essential to the routing of traffic. As an example,

the behavioral differences were noticed in the collected traffic trends of routers nDNDc1 and nANNc1 during router corruptions and storms. Router nDNDc1 operates on the edge of the network, and is considered a minor router that routes traffic to two attached routers. Router nANNc1, however, operates in a heavier environment as it must route traffic to five attached routers.

In this section, our traffic matching algorithm will account for these behavioral differences by incorporating a *size of zone* ratio when calculating the match-value between a problem case and library case. The size of zone ratio will be implemented by calculating the ratio between the number of attached routers between the problem and library case. The distance match-value between the matched cases will then be multiplied by this ratio to provide a new match-value. This size of zone ratio will thus give more weight to matching cases that had an equal number of routers in their zones. Equation 8.1 shows our method for incorporating the *size of zone* in the matching process.

$$\text{new_match_value} = \text{old_match_value} \times \frac{\min(\text{problem}, \text{library})}{\max(\text{problem}, \text{library})}$$

$\min(\text{problem}, \text{library})$: means the minimum number of attached routers in either the problem or the library case.

$\max(\text{problem}, \text{library})$: means the maximum number of attached routers in either the problem or library case.

Equation 8.1 Size of Zone Ratio

Using our developed Pegasus Case Based System, the size of zone ratio was added to the matching phase for the traffic cases. Tests were conducted with six problem cases with and without the 'size of zone' ratio weight included in the matching phase. Results for two of these problem cases are shown in Table 8.1 (the smaller match values indicate the better match). The case library consisted of 120 cases, and k was set to 10. The large k value was used to provide some indication of how our matching performed to matching to

a group of cases with the same solution.

node_name	Fault induced	no.	Best match	Case #	Case Solution	Best Match with ratio weight	Case #	Case Solution
nFFMc1	Storm	1	1.36603	16	RC	2	80	ST
		2	1.70711	82	ST	2	115	ST
		3	1.70711	83	ST	2.12132	99	ST
		4	1.70711	116	ST	3.20711	111	ST
		5	1.86603	100	ST	3.43677	16	RC
		6	1.86603	113	ST	3.46081	98	ST
		7	2	80	ST	3.49071	89	ST
		8	2	81	ST	3.5694	85	ST
		9	2	90	ST	3.66908	100	FT
		10	2	96	ST	3.73607	90	ST
bRTPc1	File Transfer	1	.889	65	FT	.889	65	FT
		2	1.824	105	ST	2.5346	44	FT
		3	1.823	1	RC	3.001	105	ST
		4	1.943	86	FT	3.174	15	RC
		5	2.0177	24	RC	3.448	1	RC
		6	2.024	15	RC	3.558	13	RC
		7	2.104	66	FT	3.568	48	FT
		8	2.104	69	FT	3.587	12	RC
		9	2.1622	44	FT	3.663	11	RC
		10	2.183	5	RC	3.697	6	RC

Table 8.1 Size of Zone Matches

In Table 8.1, we see that the matching is dependant on our included size of zone ratio. As an example, node nFFMc1 matching with size of zone ratio retrieves different cases than the matching without the size of zone. Also noticed is that the size of zone matching favors a better retrieval of matching in the nFFMc1 storm problem (i.e. without 'size of zone' the first best matched case is a RC match, with 'size of zone' the first best matched case is a ST match). A decrease in accuracy, however, is seen in the case of the file transfer through bRTPc1. For this case, the best matched cases are router corruptions. An investigation shows that this decrease in accuracy is due to the inadequacy of the case library to contain cases of file transfers which have the same number of nodes as bRTPc1 (i.e. there are more router corruption cases of the same size of bRTPc1 to favor the size of zone ratio matching). For our six test cases, the accuracy at k=10 with 'size of zone' matching is 67%. This accuracy is a decrease from our results obtained in Chapter 7 with

120 cases and an accuracy of 100%.

To increase the case retrieval accuracy with the additional 'size of zone' ratio added to the matching process, feature weighting will be explored in the next section.

8.8 Feature Weights

A potential problem with the nearest neighbor metrics is that they treat all the case features as having the same predictability to finding a matched case. In our situation, however, it may be necessary to account for the fact that certain features are more prevalent to certain types of problems than others; so if these certain features occur, then the matching should further favor the cases that contain these features. In k-NN retrieval mechanisms, a set of weight calculations can be included in the distance metric to account for predictability of the individual case features. To include these weights in our feature matching process, our distance metric will take the form of Equation 8.2, which will be similar to weighted euclidean distance shown in [Hartigan75].

$$d = \sqrt{(W_x \cdot (x_b - x_a))^2 + (W_y \cdot (y_b - y_a))^2 + (W_z \cdot (z_b - z_a))^2}$$

Equation 8.2 Weighted Distance Metric

A set of calculations will be required to find feature weights of our case features, and using Equation 8.2, tests will be required to determine if the weights can increase the traffic case matching accuracy.

Methods for finding feature weights vary as mentioned in [Hartigan75], [Creecy92], and [Stanfill86]. Our method of weighting will use a sum of the squared conditional probabilities shown in Equation 8.3 to calculate weights. [Creecy92] used this formula to calculate a weighting factor that favored the selection of the library cases given that a feature of a certain value was present in the input problem case. As an example, in a

$$Weight(f) = \sqrt{\sum_{i=1}^n P(C_i|f)^2}$$

f = feature
C = category
n = number of cases

Equation 8.3 Sum of Squared Conditional Probabilities for Feature Weights

case library of ten categories, if a feature of a certain value was present in only two categories then the corresponding weight for the feature for selecting a category would be calculated as follows:

max number of categories = 10 = n
note: an example of a category would be 'router table corruption problems'
number of categories containing feature 'f' of value = 2 and the probability of selecting the correct category given the feature is present would be 0.707

$$\text{weight for feature of value 2} = \sqrt{\sum_{i=1}^n P(C_i|f)^2} = \sqrt{\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2} = 0.707$$

Example 8.1 Feature Weights Calculation 1

[Stanfill86] used a different approach with this equation to calculate feature weights that accounted for all possible values of the features in selecting a case. [Stanfill86]'s method was a measure of the variance of the values for a feature, and it is similar to the variance measure of weights used in [Hartigan75]. Stanfill's measure was based on the selection of a feature given the group of cases from a particular category i.e. the probability measure for Equation 8.3 was reversed to read the probability of a feature occurring given the selection of a category. In Stanfill's approach, if a feature was present in 100 cases in a certain category, and if the feature had two possible values of 10 cases with value A and 90 cases with value B, the corresponding feature weight would be calculated as shown in Example 8.2.

The two different methods as shown above are useful in different situations:

$$\text{weight of a feature} = \sqrt{\sum_{i=1}^n \langle f_i | C \rangle^2} = \sqrt{\left(\frac{10}{100}\right)^2 + \left(\frac{90}{100}\right)^2} = 0.9055$$

Example 8.2 Feature Weights Calculation 2

[Creecy92]'s method is used in selecting the best category given the presence of a feature, whereas [Stanfill86]'s method is used in selecting the best cases based on the value of the individual features for a given category.

In Chapter 7, the division of the traffic cases into distinguishing and discriminating matching was discussed. The distinguishing matching used the center-node to find the best k-NN cases. Our distinguishing features thus provided a categorical search from the case library to find the center-node that best resembled one of the three types of problems (or categories).

To find our distinguishing feature weights, the methods of [Creecy92] will be applicable because these features are categories for the three problem types, and we want the weights to reflect the choosing of the proper category given that a certain feature is a certain value. To find our probability measures we will also use the frequency interpretation of probability, and we will assume that our cases consist of a large number of vector fields which provide a good collection for our three network problems (i.e. no error in collected features). Example 8.3 shows a calculation to find our distinguishing feature weights. This example shows the calculation on selecting a category given that the mean_trend feature of the problem case is equal to 1; the category is the mean trend being a value of '1' and the features are the three categories of Router Corrupt, File Transfer and Storm Problems. In the example, a total of 76 cases have a mean trend of '1' from the collection of 120 cases.

Table 8.2 shows the complete calculated distinguishing weights for the case library

$$P(\text{RouterCorrupt}|\text{mean_trend}=1) = \frac{10}{76}$$

$$P(\text{Storm}|\text{mean_trend}=1) = \frac{26}{76}$$

$$P(\text{FileTransfer}|\text{mean_trend}=1) = \frac{40}{76}$$

$$\text{weight of mean trend is } 1 = \sqrt{\left(\frac{10}{76}\right)^2 + \left(\frac{26}{76}\right)^2 + \left(\frac{40}{76}\right)^2} = 0.6414$$

Example 8.3 Distinguishable Weights Calculation

of 120 cases. The larger the feature weight, the better the feature will be in selecting the

Distinguishing Feature	Value of Feature		
	1	0	-1
mean_trend	.6414	.7454	.8339
mean_rms_trend	.5865	.6492	0
cost_trend	.9025	.6747	.8046
cost_rms_trend	.7156	.6770	.7454
zone_trend	.6224	.6431	.9235

Table 8.2 Distinguishing Feature Weights for 120 Cases

correct category. From Table 8.2 it would seem that the cost_trend of being equal to '1' and the zone_trend being equal to '-1' are good features for selecting categories. For our cases, a manual search of the 120 traffic cases verified that the cost_trend of '1' was more prevalent in the storm problems and the zone_trend of '-1' was more prevalent in the router corruption problem.

Our second set of calculations will involve finding the weights for the traffic cases' discriminating features. As mentioned in Chapter 7, these discriminating features are the set of attached router fields, and they are used to further sort the k-NN best matched cases after the categorical search is made with the distinguishing features. In these vector mappings we want to calculate the probability of the features occurring with a certain

value given that a particular category has been chosen. To find the feature weights in this situation requires the methods as used by [Stanfill86]. The discriminating weights will be calculated by finding the variance of the three different values for all the collected vectors from each of the three problem categories. Example 8.4 shows a calculation of the

$$\text{frequency of a '1' in the mean trend} = \frac{20}{186}$$

$$\text{frequency of a '-1' in the mean trend} = \frac{62}{186}$$

$$\text{frequency of a '0' in the mean trend} = \frac{104}{186}$$

$$\text{weight of mean trend} = \sqrt{\left(\frac{20}{186}\right)^2 + \left(\frac{62}{186}\right)^2 + \left(\frac{104}{186}\right)^2} = 0.6598$$

Example 8.4 Discriminating Weights Calculation

discriminating weights for the Router Table Corruptions. In this case the feature is the mean trend and the categories are the values of '1', '0' and '-1'.

Using our discrimination measures, Table 8.3 shows the complete discriminating weight calculations for our three case categories.

Discriminating Feature	Category		
	Router Corruption	Storm	Large File Transfer
mean_trend	.6598	.5867	.6682
mean_rms_trend	.7385	.7844	.6982
cost_trend	.8138	.8838	.6943
cost_rms_trend	.8562	.9067	.7461
zone_trend	.6055	.5877	.6174

Table 8.3 Discriminating Feature Weights for 120 Cases

From the calculated Tables 8.2 and 8.3, tests with/without our calculated feature weights included in the matching process were conducted on the case library of 120 cases

at $k=10$ and with the 'size of zone' ratios included. The results are shown in Table 8.4 (note: because we use the distance metric for matching, better case matches are shown by the smaller match values).

In Section 8.7 we obtained an accuracy of 63% without feature weights. With feature weights our accuracy is increased to 83%, a 20% improvement in our retrieval and matching accuracy. From these tests we conclude that feature weighting is an important consideration for our case feature matching.

Nodename	Fault induced	Best match	Case #	Case Solution	Best Match with feature weights	Case #	Case Solution
bKANc1	Router Corruption	1.48803	86	ST	1.00998	86	ST
		1.91068	3	RC	1.37134	68	FT
		1.91068	46	FT	1.48906	3	RC
		2.07869	68	FT	1.49188	66	FT
		2.1546	66	FT	1.67069	1	RC
		2.2717	1	RC	1.88738	64	FT
		2.78885	20	RC	1.91105	24	RC
		2.80105	24	RC	1.91201	79	FT
		2.80105	64	FT	1.97412	20	RC
		2.82514	79	FT	2.07362	15	RC
nDNDc1	Router Corruption	0	0	RC	0	0	RC
		1.36603	26	RC	.672	26	RC
		2.23607	27	RC	1.65615	27	RC
		2.48607	21	RC	1.82112	35	RC
		2.48607	35	RC	1.8596	21	RC
		2.73607	34	RC	1.9589	34	RC
		3.23607	22	RC	2.36498	29	RC
		3.48607	36	RC	2.39402	36	RC
		3.49071	29	RC	2.85348	9	RC
		3.73607	28	RC	3.15321	7	RC
nATLc2	Large File Transfer	.910684	59	FT	.629391	59	FT
		1.47159	60	FT	1.0012	60	FT
		1.72159	74	FT	1.18665	74	FT
		2.0215	45	FT	1.30138	45	FT
		2.15738	70	FT	1.47887	70	FT
		2.23607	52	FT	1.54827	52	FT
		2.29545	40	FT	1.56493	40	FT
		2.35454	55	FT	1.61763	55	FT
		2.98142	43	FT	2.04565	43	FT
		2.98142	54	FT	2.04565	54	FT

Table 8.4 Best Matches with Feature Weights

Nodename	Fault induced	Best match	Case #	Case Solution	Best Match with feature weights	Case #	Case Solution
hRTPc1	Large File Transfer	.889	65	FT	.662178	65	FT
		2.5346	44	FT	2.3116	15	RC
		3.001	105	ST	2.34757	105	ST
		3.174	15	RC	2.4426	1	RC
		3.448	1	RC	2.50423	13	RC
		3.558	13	RC	2.6602	6	RC
		3.568	48	FT	2.66977	69	FT
		3.587	12	RC	2.67873	66	FT
		3.663	11	RC	2.84473	5	RC
		3.697	6	RC	2.95898	24	RC
nFFMc1	Storm	2	80	ST	1.2359	80	ST
		2	115	ST	1.2359	115	ST
		2.12132	99	ST	1.24564	99	ST
		3.20711	111	ST	2.15457	111	ST
		3.43677	16	RC	2.37278	89	ST
		3.46081	98	ST	2.45268	85	ST
		3.49071	89	ST	2.46977	98	ST
		3.5694	85	ST	2.52475	16	RC
		3.66908	100	ST	2.64009	100	ST
		3.73607	90	ST	2.64786	90	ST
nCLDc1	Storm	2.13006	118	ST	1.3254	118	ST
		2.55009	117	ST	1.5997	119	ST
		2.5799	119	ST	1.7722	110	ST
		2.70864	110	ST	1.875	117	ST
		3.2717	108	ST	2.2703	108	ST
		3.35454	106	ST	2.34877	107	ST
		3.35454	107	ST	2.5603	106	ST
		3.85028	82	ST	2.86193	82	ST
		4.36612	102	ST	3.3492	103	ST
		4.36612	103	ST	3.3666	102	ST

Table 8.4 Best Matches with Feature Weights

8.9 Increasing the Case Library Size

Our frequency interpretation for calculating the probability relies on the frequency of occurrence of particular features in a group of cases. The Law of Large Numbers states that a large set of cases should be used to provide an accurate value for the probabilities. To test our accuracy of our feature weights calculated in Section 8.8, this section will analyze the accuracy of our matching process as the case library size is increased. For these tests, the case library will be increased to 240 cases and the feature weights will be re-calculated. The 240 library cases will be equally divided among our three types of

router problems - 80 RC, 80 ST, 80 FT.

Using the same methods of Section 8.8, the weight calculations using 240 collected cases was conducted. The distinguishing and discriminating feature weights are shown in Table 8.5 and Table 8.6 respectively.

Distinguishing Feature	Feature		
	1	0	-1
mean_trend	.6400	.6797	.7477
mean_rms_trend	.5894	.6309	1
cost_trend	.8814	.6509	.8041
cost_rms_trend	.7371	.6867	.8246
zone_trend	.6017	.6248	.7906

Table 8.5 Distinguishing Feature Weights with 240 Cases

Discriminating Feature	Category		
	Router Corruption	Storm	Large File Transfer
mean_trend	.6800	.6026	.6586
mean_rms_trend	.7457	.7841	.7152
cost_trend	.8321	.8314	.6825
cost_rms_trend	.8754	.9160	.7640
zone_trend	.6239	.5817	.5990

Table 8.6 Discriminating Feature Weights with 240 Cases

A comparison of the distinguishing and discriminating weights between the 120 and 240 cases shows only noticeable changes from the distinguishing feature weights of the '-1' feature, in this case the extra 120 cases collected from the network have more frequent occurrences of a '-1' feature to produce the smaller results. The overall small changes in weighting factors shows some confidence in our initial 120 case selection as

being a good representation of the three network problems (i.e. if large changes in weights were found between the 120 and 240 cases, it would have showed that our initial set of 120 cases offered a poor collection of the network problems). A comparison of the accuracy in retrieval between 120 and 240 cases is shown in Table 8.7. This Table shows three separate tests: 1) the accuracy of 120 cases with weights from Section 8.8, 2) the accuracy using 240 library cases with no weights, and 3) the accuracy with 240 cases with the set of weights calculated in this section.

			Case #	Case Solution	Case#	Case Solution	Case #	Case Solution
Nodename	Fault induced	#	120 cases with Section 8.8 weights		240 cases without weights		240 cases with Section 8.9 weights	
hKANc1	Router Corruption	1	86	ST	74	RC	74	RC
		2	68	FT	44	RC	44	RC
		3	3	RC	166	ST	166	ST
		4	66	FT	3	RC	1	RC
		5	1	RC	86	FT	106	FT
		6	64	FT	145	FT	127	FT
		7	24	RC	156	FT	15	RC
		8	79	FT	108	FT	159	FT
		9	20	RC	54	RC	185	ST
		10	15	RC	106	FT	105	FT
nDNDc1	Router Corruption	1	0	RC	0	RC	0	RC
		2	26	RC	63	RC	63	RC
		3	27	RC	72	RC	72	RC
		4	35	RC	69	RC	69	RC
		5	21	RC	26	RC	26	RC
		6	34	RC	27	RC	27	RC
		7	29	RC	21	RC	35	RC
		8	36	RC	35	RC	21	RC
		9	9	RC	76	RC	76	RC
		10	7	RC	34	RC	34	RC
nATLc2	Large File Transfer	1	59	FT	155	FT	155	FT
		2	60	FT	99	FT	99	FT
		3	74	FT	123	FT	123	FT
		4	45	FT	100	FT	100	FT
		5	70	FT	114	FT	114	FT
		6	52	FT	85	FT	85	FT
		7	40	FT	147	FT	147	FT
		8	55	FT	110	FT	110	FT
		9	43	FT	92	FT	80	FT
		10	54	FT	80	FT	92	FT

Table 8.7 Accuracy Compared Between 120 and 240 Case Base: k = 10

			Case #	Case Solution	Case #	Case Solution	Case #	Case Solution
bRTPc1	Large File Transfer	1	65	FT	152	FT	152	FT
		2	15	RC	105	FT	105	FT
		3	105	ST	139	FT	127	FT
		4	1	RC	127	FT	185	ST
		5	13	RC	185	ST	159	FT
		6	6	RC	159	FT	15	RC
		7	69	FT	15	RC	1	RC
		8	65	FT	1	RC	106	FT
		9	5	RC	6	RC	166	ST
		10	24	RC	106	FT	44	RC
nFFMc1	Storm	1	80	ST	204	ST	204	ST
		2	115	ST	160	ST	160	ST
		3	99	ST	195	ST	195	ST
		4	111	ST	170	ST	179	ST
		5	89	ST	239	ST	201	ST
		6	85	ST	201	ST	234	ST
		7	98	ST	234	ST	239	ST
		8	16	RC	191	ST	238	ST
		9	100	ST	229	ST	169	ST
		10	90	ST	238	ST	165	ST
nCLDc1	Storm	1	118	ST	213	ST	213	ST
		2	119	ST	228	ST	217	ST
		3	110	ST	217	ST	228	ST
		4	117	ST	203	ST	198	ST
		5	108	ST	198	ST	203	ST
		6	107	ST	231	ST	231	ST
		7	106	ST	197	ST	197	ST
		8	82	ST	215	ST	215	ST
		9	103	ST	188	ST	188	ST
		10	102	ST	186	ST	187	ST

Table 8.7 Accuracy Compared Between 120 and 240 Case Base: k = 10

We should expect the accuracy of our retrieval and matching system using 240 cases and feature weights to increase for two reasons. Firstly, the case library is increased in size to offer a larger selection of cases for a problem case to match with (i.e. a problem case is able to select from more cases of the same problem type to use in k-NN matching). This is shown in Table 8.7 by the 240 cases without weights producing the same accuracy as the 120 cases with weights (83%). Secondly, our feature weights should be more accurate because we are using 240 cases to find the feature weights (i.e. the larger the sample size the more confident our frequency interpretation is).

With weighting factors and 240 cases, our traffic matching system provides a 100% accuracy in diagnosing an input problem case. The next section will show that the value of k is also an important aspect on the accuracy of our matching system.

8.10 K-NN Testing

In our system, the k -NN algorithm provides a diagnostic solution based on the frequency of a number of library cases from a particular class being matched to problem case; this frequency raises our confidence that our problem case is closer to a certain category (see Section 4.1.2.1). As an example, for our traffic case based system in Chapter 7, we based our diagnosis on the matching of the problem case to the top ten matches in the case library. In the status message case based system, we used a k value of five in selecting our best matched cases. The k values for our initial case based systems of Chapter 6 and 7 were chosen to provide a good percentage of matches with the library case sizes that were used i.e. the top matches for 5 out of the 14 status cases (~35%) and 10 out of the 120 (~8%) traffic cases were shown. The initial value of $k = 10$ for the traffic case matching was less than 10% of the total number of library cases, but this value did provide us with some very good results for the best matching.

In using the k -NN algorithm, the best method of selecting k is to conduct tests. In this section we will conduct a simple test for finding the best k value for our traffic case based system. This test will involve varying the value of k over a set number of traffic library case matches. The best values of k will be found by calculating the accuracy of the matches as the k value is changed. For our k -NN tests, we used 9 problem cases and 240 library cases with feature weights and the 'size of zone' ratio matching. We will conduct 13 separate accuracy tests with 9 problem cases at k values of 1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25 and 30.

The results for these tests are shown in Graph 8.1.

Graph 8.1 Accuracy vs. Value of K

This figure shows that the k-value below 12 for a case library of 240 cases provides an accuracy of 100% in matching the nine problem cases, an accuracy of 89% is accomplished for values of k including and above 12. This graph shows some very interesting aspects of our traffic cases. Normally, we would expect that our matching accuracy to vary as the k value is changed. For our system, however, the accuracy of our matching is 100% for values of k from 0 to 10. A strong conclusion from these results is that our traffic cases are very distinguishable for the three types of router problems that we are investigating. If the cases were not so distinct, we would have expected our results to show a very variable accuracy as the k value was changed. Our results from our k tests, show that we could have used a $k = 1$ value in our tests to obtain the same accuracy of 100%. Other tests not conducted in this thesis should involve studying the matching accuracy by varying the size of the case library. If the case library decreases in size, our accuracy should decrease because the k-nn algorithm will have fewer cases in a particular category to select from for every problem case match. However, because of our very distinguishable cases, it may be found that the accuracy does not vary for the changing

case library sizes.

8.11 Feature Independence

Measures of independence on our case features will provide an indication of how our chosen case features are related. In a CBR system, independence measures for features can be used to remove unnecessary features in the cases. As an example, if the occurrence of one feature is highly dependant with the occurrence of another feature in all the library cases, then the removal of either feature should not decrease the accuracy of the system. As a benefit of removing unnecessary features, our CBR system may obtain a quicker matching phase.

From our cases, we will examine the independence of the mean and cost trends with their respective rms trends (these features will be chosen because the change in mean or cost trend should affect the calculation of the rms values). To calculate the independence factors between two variables, we will use measures of conditional probabilities of a feature appearing with another feature. Our conditional measures, as an example, will calculate the probability of the mean rms trend = '1' and mean trend = '1' appearing together given the mean trend = '1'. High probability values should be found if the mean rms trend and mean trend values are dependant.

Using our conditional probability method, Table 8.8 shows our calculated probabilities using the distinguishing features in our 240 traffic library cases. From

Feature	CATEGORY		
	RC	ST	FT
$P(\text{mean_rms_trend}=1 \text{mean_trend}=1)$	0.75	0.75	0.87
$P(\text{mean_rms_trend}=1 \text{mean_trend}=-1)$	0.54	0.81	n/a
$P(\text{mean_rms_trend}=0 \text{mean_trend}=0)$	1	0.37	0.5

Table 8.8 Conditional Probability for Independence Measures

Feature	CATEGORY		
P(cost_rms_trend==1 cost_trend==1)	0	0	0.45
P(cost_rms_trend==1 cost_trend=-1)	0.17	0.98	0
P(cost_rms_trend==0 cost_trend==0)	0.81	0.13	0.91

Table 8.8 Conditional Probability for Independence Measures

Table 8.8, the rms values appear to have a some dependance with the appearance of a change in the mean or cost trend. As an example, there is a high probability value between the occurrence of the mean and its standard deviations (i.e. values of 0.75, 0.75 and 0.87 for RC, ST and FT respectively). The table also shows some dependance factors of 0.98 for a change in a storms cost trend to '-1', and values of 0.81 and 0.91 for a '0' change in the cost trends for a RC and FT respectively. The 'n/a' factor in Table 8.8 appears because the mean trend of '-1' in a FT did not appear in our 240 test cases. The zero dependance factors in the table are a result of a small number of cases that exhibited trends (as an example, the cost_trend of '1' in a storm appears only once in our 80 cases of storms).

Based on the above table, it appears that the mean and cost rms fields are somewhat dependant of changes in their respective mean and cost trend fields, and to test the dependency we will further conduct a set of matching tests with and without the rms case features using six problem test cases (using k=10, feature weights and 'size of zone').

Table 8.9 shows three of these six tests before and after the rms features of the mean and cost were removed from the case matching procedure.

With feature removal an accuracy of 83% is obtained, a decrease from the 100% accuracy previously found with all features. The only major change in the retrieval occurs in the matching of bKANc1's problem case in which more FT cases are incorrectly retrieved.

			with all features		with rms features removed	
Nodename	Fault induced	Match Number	Case Retrieved	Case Solution	Case Retrieved	Case Solution
hKANc1	Router Corruption	1.	74	RC	74	RC
		2.	44	RC	44	RC
		3.	166	ST	91	FT
		4.	1	RC	145	FT
		5.	106	FT	167	ST
		6.	127	FT	156	FT
		7.	15	RC	130	FT
		8.	159	FT	52	RC
		9.	185	ST	108	FT
		10.	105	FT	106	FT
nDNDc1	Router Corruption	1.	0	RC	0	RC
		2.	63	RC	63	RC
		3.	72	RC	26	RC
		4.	69	RC	72	RC
		5.	26	RC	69	RC
		6.	27	RC	222	ST
		7.	35	RC	27	RC
		8.	21	RC	40	RC
		9.	76	RC	218	ST
		10.	34	RC	23	RC
nCLDc1	Storm	1.	213	ST	198	ST
		2.	217	ST	213	ST
		3.	228	ST	217	ST
		4.	198	ST	199	ST
		5.	203	ST	190	ST
		6.	231	ST	231	ST
		7.	197	ST	220	ST
		8.	215	ST	215	ST
		9.	188	ST	187	ST
		10.	187	ST	188	ST

Table 8.9 Accuracy Before and After Removal of RMS Features

We can conclude from these tests that the rms features are redundant features in our case library. Our accuracy with feature removal shows a decrease, but not by a substantial amount (1 wrong solution in 6 problem cases tested). We would expect that the removal of essential features to decrease our accuracy by a much larger amount. A close look at Table 8.9 also shows that the cases that were matched after the removal of the rms case features were generally the same as the cases retrieved with all the features. If the features were an essential part of the retrieval process we would not have expected such a

similar case retrieval.

8.12 False Alarms

In the operation of our traffic CBR system in Chapter 7, we discovered that our trend detection system produced a large number of false reports. These false alarm reports were caused by two factors: 1) our method of trend analysis and 2) the attached routers to a faulty router. These false reports were collected as problem cases, and were presented to the user as actual problem faults. Lowering these false alarm reports is an important consideration for our system because too many false reports will cause a user of our CBR system to become disassociated with our system's ability in diagnosing real network problems. This section will discuss the two factors responsible for our false alarms and will provide our research in lowering the false alarm rates.

8.12.1 Trend Analysis False Alarms

A false alarm will be generated in our CBR system when a router's normal performance trend is upward or downward for length of the trend window period. As discussed in Chapter 7, faulty routers were signalled as being those routers which have a continuous upward or downward mean bytes out or zone deviation trend for a trend window period. The trend window size was defined as the number of intervals that were observed to signal an abnormal trend in a router.

Our initial tests for lowering the number of false alarms will involve changing the trend window size. It will be expected that an increase in the trend window size will lower our false alarm rate and decrease our detection of actual faults. The detection of actual faults depends on the length of the trend that a router exhibits during a fault. As an example, in our fault tests of Chapter 7, routers nDNDc1 and nANNc1 exhibited very lengthy mean byte and zone deviation trend lines that lasted from 10 to 12 intervals in length. However, some of our tests on other routers exhibited smaller trend intervals during a fault that lasted from only 5 to 7 intervals. As a result, if we increased our trend

window to a length of 10 intervals, some of the actual faults in the network were not detected. In our collection of problem and library traffic cases for our CBR tests we used a trend window size of 5 intervals; this trend window size ensured that we would receive all the true faulty router reports from the network.

In order to find the best setting for a trend window size, we will conduct tests with nine test problems(3 RC, 3 ST, 3 FT). The nine problems will be induced into the network and the traffic trends for a thirty minute interval will be collected. The traffic reports will then analyzed to find the probability of detecting a fault for various trend window sizes. Another set of tests on the entire network with no faults will be conducted to find a false alarm rate for various trend window sizes occurring in a thirty minute period.

By adding code to our trend window traffic system, the above tests were conducted. Graph 8.2 shows the false alarm rate and the probability of detection vs. the trend window size. This graph shows that the detection accuracy remains at 100% for trend window sizes of 5, 6 and 7, but then gradually decreases to 56% at a trend window size 15 intervals. At a trend window size of 5 intervals, 280 false alarms are reported in a 30 minute interval. The false alarm rate then rapidly decreases to provide only 4 false alarms at a trend size of 15. From these tests, we can conclude that a compromising setting in our trend methods might be to use a trend window size of 12 intervals to provide a detection accuracy of 78% and a false alarm rate of 9 reports in a 30 minute period.

Also to alleviate the false reports, the repetitiveness by which the nodes are reported will be considered. The repetitiveness of reports is defined as the number of reports from a zone over a number of trend interval periods. The more reports that are shown from a zone, the more likely the zone is experiencing a fault. As an example, a corruption in nDNDc1 will produce zone problem reports from nDNDc1 for about six consecutive intervals using a trend window of size 10. Usually, false alarm reports are only present for one to two intervals. A repetitiveness counter could be used to store the

Graph 8.2 False Alarms and Detection Accuracy vs Window Size

number of times that a router is reported as faulty. A router which reports being in a trend condition for a certain number of times could then be flagged as acting faulty. This repetitiveness counter method will not be tested in our CBR system, but it may prove useful in future tests to lower the false alarm rate.

Although not thoroughly tested in this thesis, another aspect to decreasing the false alarm rate using our trend analysis technique may be by changing the windowing interval methods. Our windowing measurements for the trend analysis uses 2 minutes of current traffic compared to 30 minutes of a reference window. Calculations of the trends based on 1 minute intervals over the same 30 minutes of reference may provide a better selection (these calculations of the trends will also depend on the availability of the traffic

measurements from the network in question i.e. are 1 or 2 minute measurements available?).

8.12.2 Surrounding Router Reports

Chapter 7 discussed how a fault will affect the surrounding routers to the faulty router. In our trend collection system, the surrounding routers that experience a trend change will become collected as actual problem cases. The CBR user will then become swamped by all of these problem reports, and he/she must decide which of the problem cases are the actual faults and which of the problem cases are from surrounding routers.

Our observations on these surrounding router false alarms is that the number of reports from the surrounding routers varies on the type of fault and the location of the actual faulty router. As an example, a storm problem will generate true problem cases from the routers that the storm passes through and also from a large number of surrounding routers at a depth of one to three away. A router corruption's reports, however, will be more localized to the faulty router, but on some occasions a router corruption may affect surrounding routers to a depth of two.

Some research in accounting for these surrounding router reports will be conducted in this Section. Our solution to these surrounding reports will be to present the CBR user the total number of collected cases and their zone percentages. This zone percentage will be calculated by taking each router report and finding how many routers in its zone are present in the collection of all the router reports. These percentages will show the faulty router as having the largest number of reports from the surrounding routers in the collection of all the traffic problem cases. In other words, the higher the percentage will show the higher probability that the router fault is from this node.

Our traffic cases collection program was modified to calculate the router percentages. Figure 8.2 shows an example calculation of the percentage of router reports

collected for a router corruption in nDNDc1 during one interval. Zone nDNDc1, nEDMc1

Figure 8.2 Reports as a Percentage of Zone Shown

and nEMRc1 have 100% of their routers reported in Figure 8.2 (e.g. nDNDc1's zone is composed of routers nDNDc1, nEDMc1 and bMERc3 and all three are found in the problem reports). Based on these percentages, the CBR user would make a confident guess that the fault was in either nDNDc1, nEDMc1 or nEMRc1.

8.13 Discussion

The Pegasus Case Based system was formed by combining the case based systems for the status messages and traffic trends as developed in Chapter 6 and 7 respectively. Based on this system, certain aspects of our CBR system were analyzed. The conclusions from this analysis include the following:

1. The size of zone measure can be used in the traffic matching to account for behavioral differences in faulty routers in different sizes of zones;
 2. Feature weights can be used to increase our matching accuracy;
 3. The value of k for k -nearest neighbor matching is an important value to consider to maintain a good accuracy;
 4. Feature correlation measurements can be used to measure feature dependency.
- In our CBR system, we show that the rms trend measurements for the mean bytes out and cost trend measurements are redundant features; and

5. The false alarm rate in our traffic reporting system can be reduced by increasing the trend window size. The increase in the window size, however, will decrease our detection accuracy. The false alarm rate can also be lowered by using a repetitiveness counter and by researching other measurements of trend windowing. The false alarm rate is also based upon the in-correct collection of the surrounding routers to a faulty router. To help find the correct solutions, a percentage based presentation of the routers that reported faults can be used.

Chapter 9

Conclusions, Contributions and Future Research

In diagnosing a network fault, a network operator may recall similar situations from past network problems to find a solution. These recalled situations are triggered by certain events or features that occur in the new network fault. The recall of similar situations is an area that Case Based Reasoning Systems imitate. Problems are represented by features that can be collected before, during and after a problem has occurred. These features are represented into a store collection of features called a case. These problem cases then proceed through a retrieval and matching phase to find best matches to solved cases that have been stored in a database. The solutions from the best matched cases can then be applied to the new problem or they may be adapted to fit the new problem.

In this thesis a novel approach to applying a CBR system to diagnose three types of network router faults is accomplished. Our developed CBR system is able to accurately diagnose router corruptions, large file transfer problems and storm problems using a collection of status messages and traffic trends from a simulated communication network. An accuracy of 75% using status messages to diagnose router corruptions and storms is measured. An accuracy of 100% using traffic trends in diagnosing router corruptions, large file transfer problems, and storms. This accuracies do show that a CBR system is a viable solution to solving network router faults.

This thesis covered two objectives. The first objective was to review past literature and to investigate the utilization of a CBR System for network router diagnostics; Chapter 4, 5, 6 and 7 covered this objective by explaining our research in the case representation, retrieval and matching, and adaptation phases of two CBR systems to diagnose three types of router problems. Chapter 8 covered the second objective of this thesis by showing our implementation of a Pegasus CBR system to diagnose three router problems. Chapter 8 also provided an account of our investigation in different matching methods for the traffic cases for our Pegasus CBR system.

This thesis shows how traffic trend cases to represent a network problem. Five traffic trend measurements are used to represent a router problem in a traffic case: 1) traffic mean bytes out, 2) traffic mean rms, 3) cost, 4) cost rms, and 5) zone deviations. In this thesis, we further observe that our trend analysis technique provide a a good method of predicting faults. Our trend analysis technique also produces high false alarm rates that can be reduced by changing the trend interval size and by using methods to account for the reports from the attached routers in a zone.

This thesis shows how we use status message cases in a graph method form to represent our router faults. The graph matching method is also successfully used to match the graph cases. The status messages cases, however, do not provide a good measure of predictability in diagnosing some of the router corruption faults. The status messages are also not good predictors for any of the file transfer problems because no status reports are produced.

Using the test results in Chapter 8, our traffic test cases produces a 100% matching accuracy using feature weighting, 'size of zone' ratio, and a k value of 10. This high accuracy shows that the traffic case are quite distinguishable between the three problem types. A measure of feature correlation also shows that not all the traffic features are required to maintain a high matching accuracy.

9.1 Contributions

This thesis provides the following contributions:

1. An investigation of CBR methods for communication network diagnostics is accomplished. Our high accuracy in retrieval and matching shows that a CBR system is feasible for network diagnostics;
2. A new technique for network status message representation and matching is accomplished. Status messages are placed into a graphical representation, and graphical structure matching is used to match status message cases; and
3. New techniques in traffic trend analysis and traffic representation are accomplished. Trends in traffic are used to predict faults in a network, and the collection of traffic trends in a zone of nodes is used to represent and match traffic cases. Methods to reduce the false alarm rates from the trend analysis is also researched.

9.2 Future Research

The following three aspects discuss future research that will be required if our CBR methods are to be implemented in real-network situations:

1. Increase coverage of the CBR cases: Our case based system provides solutions for three network router problems. In a real network, the number of router problems that can occur include such faults as broken or faulty router lines, congestion of packets, and other forms of router table corruptions. These types of faults should be investigated and implemented as problem cases in the case library. Further investigation of the case representation and the accuracy of retrieval will be required as these other cases are included;
2. Expand the adaptation phase: As our developed case based system begins to

include other types of problem cases as mentioned in 1), an improved adaptation phase may be required to adapt the best retrieved solutions. Improved adaptation phases could be implemented, as an example, in our current case based system to measure the size of the trends reported or to calculate the percentages of the zone that were found in the problem reports; and

3. **Improve the trend analysis technique:** Our current trend analysis technique suffers from a high false alarm rate. Our tests concluded that the false alarm rate can be reduced by increasing the trend window size and by incorporating methods to account for the reports from the attached routers. A further investigation, however, of trend analysis techniques may provide better methods of calculating the trends without producing high false alarms. As an example, if the traffic measuring intervals were one minute instead of the two minute intervals we used in our system, the trends of faulty routers might provide longer consistent trends over a shorter time period - an increased trend window size could then be used that would maintain the detection accuracy but lower the false alarms.

References

- [Barletta88] Explanation Based Indexing of Cases, Ralph Barletta. Lockheed AI Center. Proceedings of a Case Based Reasoning Workshop, May 1988. pages 51-60.
- [Barletta91] An Introduction to Case-Based Reasoning. Ralph Barletta. AI Expert, August 1991. pages 43-49.
- [BNR90a] BNR Local Area Network Environment: An Introduction for Network User, Ottawa, Canada., July 1990.
- [BNR90b] MAGNet: Network Management Tools, Release 3.0. Ottawa, Canada, June 1990.
- [Bo93] Master of Engineering Thesis. Carleton University. in-progress.
- [Bunke83] Inexact Graph Matching for Structural Pattern Recognition. H. Bunke. Pattern Recognition Letters 1, 1983, pages 245-253.
- [Cain91] Using Domain Knowledge to Influence Similarity Judgements, Timothy Cain. Proceedings of a Case Based Reasoning Workshop 1991, pages 191-199.
- [Creecy92] Trading MIPS and Memory for Knowledge Engineering. Robert H. Creecy. Communications of the ACM, August 1992, vol 35, Number 8, pages 48-63.
- [Dawes91a] Network Diagnosis by Reasoning in Uncertain Nested Evidence Spaces. Nick Dawes. Carleton University Technical Report, SCE-91-26, 1991.

- [Dawes91b] Dragon: A Very High Speed, Very Large Scale Network Simulator, N.W. Dawes. Carleton University Technical Report, SCE-91-31, 1991.
- [Dussault92] A Diagnostic Expert System for Wide Area Networks. Master of Engineering Thesis, Carleton University, March 1992.
- [Farrell93] Detection of Abnormal Router Behavior in a Wide Area Network. Masters of Engineering Thesis, Dept. of Systems and Computer Engineering, Carleton University, 1993.
- [Filipiak91] Real Time Network Management, Janusz Filipiak, Elsevier Science Publishers, Netherlands, 1991. Pages 175, 183.
- [Gotlieb78] Data Types and Structures, C.C. Gotlieb, University of Toronto, Prentice-Hall, Englewood Cliffs N.J., 1978.
- [Hartigan75] Clustering Algorithms, John A. Hartigan, Department of Statistics, Yale University. John Wiley and Sons Inc., 1975.
- [Iqneibi92] A Blackboard Architecture to Support Network Fault Diagnosis. Master of Engineering Thesis. Carleton University, 1992.
- [Kalab93] Operations Research Techniques Applied to Network Fault Management. Master of Engineering Thesis. Dept. of Systems and Computer Engineering, Carleton University, 1993.
- [Kitano92] Building Large-Scale and Corporate Wide Case-Based Systems: Integration of Organizational and Machine Executable Algorithms. Proceedings of the Tenth National Conference on Artificial Intelligence, pages 843 - 849.
- [Kolodner86] Towards a Memory Based Architecture that Supports Reminding. The Eighth Annual Conference of the Cognitive Science Society, 1986, pages 467-477.
- [Kolodner91] Improving Human Decision Making through Case-Based Decision Aiding. Janet L. Kolodner. AI Magazine, Summer 1991. pages 52-68.
- [Koton89] Using Experience in Learning and Problem Solving. Phyllis Koton, MIT Technical Report-441, March 1989.
- [Leake86] Organizing Memory for Explanation. The Eighth Annual Conference of the Cognitive Science Society, 1986, pages 710-715.

- [Lewis93] **A Case-Based Reasoning Approach to the Management of Faults in a Communication Network, Lundy Lewis. Integrated Network Management III, Elsevier Science Publishers, North-Holland, 1993, pages 114 - 120.**

- [Maxion90] **Anomaly Detection for Diagnosis. Roy A. Maxion. 20th IEEE International Conference on Fault Tolerant Computing, 1990. pages 20-27.**

- [Murphy92] **Classical Faults in Real Networks. Bernie Murphy. Oral Presentation. Information Technology Division, Bell-Northern Research, 1992.**

- [Pettofrezzo66] **Vectors and Their Applications. Anthony J. Pettofrezzo. Prentice Hall 1966, Englewood Cliffs, New Jersey.**

- [Redmond90] **Distributed Cases for Case-Based Reasoning: Facilitating the Use of Multiple Cases. Michael Redmond. Proceedings of the 8th National Conference in Artificial Intelligence. Vol1, 1990. pages 305 - 315.**

- [Riesbeck89] **Inside Case-Based Reasoning, Christopher K. Riesbeck. Lawrence Erlbaum Associates Publishers, Hillsdale NJ, 1989.**

- [Schank82] **Dynamic Memory: A Theory of Learning in Computers and People. Cambridge, Cambridge University Press, 1982.**

- [Shi93] **Master of Engineering Thesis. Dept. of Systems and Computer Engineering, Carleton University, in-progress.**

- [Simoudis91] **The Application of a CBR to Help Desk Applications. Evangelos Simoudis. Proceedings of a Case Based Reasoning Workshop 1991. pages 25-35.**

- [Slade91] **Case Based Reasoning: A Research Paradigm. Stephen Slade. AI Magazine, Spring 1991. pages 42-55.**

- [Stanfill86] **Toward Memory Based Reasoning. Craig Stanfill. Communications of the ACM, December 1986, vol 29. Number 12, pages 1213 - 1229.**

- [Viens92] **A Neural Network Approach to Detect Traffic Anomalies in a Communication Network. Master of Engineering Thesis, Dept. of Systems and Computer Engineering, Carleton University, 1992.**

[Yuan92]

PegasusBase-A Run-Time Schema Extendible System. Master of Engineering Thesis, Dept. of Systems and Computer Engineering, Carleton University, 1992.

END

'21-07-94

FIN