



uOttawa

L'Université canadienne  
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES  
ET POSTDOCTORALES



FACULTY OF GRADUATE AND  
POSTDOCTORAL STUDIES

Nancy Samaan

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Computer Science)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

An Autonomous Network Management Architecture for Hybrid Communication Systems

TITRE DE LA THÈSE / TITLE OF THESIS

Ahmed Karmouch

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

M. Barbeau

A. Nayak (absent)

S. Chamberland

A. Miri

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

AN AUTONOMOUS NETWORK MANAGEMENT ARCHITECTURE FOR  
HYBRID COMMUNICATION SYSTEMS

by

Nancy A. Samaan

Thesis submitted to the  
Faculty of Graduate and Postdoctoral Studies  
In partial fulfillment of the requirements  
For the Doctor of Philosophy degree in  
Electrical and Computer Engineering

School of Information Technology and Engineering  
Faculty of Engineering  
University of Ottawa

©Copyright Nancy A. Samaan, Ottawa, Canada 2007



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-34146-9*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-34146-9*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

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

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

## Abstract

The relentless growth in wired/wireless communication technologies coupled with rapid advances in real-time applications renders the management of such technologies a major challenge. This dissertation investigates the *Quality of Service* (QoS) management problem and proposes a novel quadri-layer autonomous, self -adaptive and -reconfigurable management framework for hybrid wired/wireless communication systems. Necessary data pertaining to users, applications, domain and network components is collected at the first layer and utilized in the second layer to project future changes in the underlying network. These changes are fed to the third layer to reconfigure the underlying network components in order to best utilize the available resources while maintaining a smooth QoS delivery. Finally, the fourth layer provides a feedback mechanism based on network related measurements. This dissertation addresses the functionalities of these layers and describes novel schemes to realize those of the second and third layers.

To perform the functionalities of the second layer, a novel algorithm for mobility prediction through the utilization of Dempster-Shafer theory of evidence is developed. In contrast to existing approaches, the proposed scheme does not require a-priori training nor does it assume the availability of a history of previous users' movements.

The second component of the same layer presents a novel scheme for diagnosing probable network faults via statistical analysis and evidential reasoning aided with a training set of previously classified faults. The scheme also includes two new approaches to handle cases of imbalanced training sets.

Functionalities of the third layer are realized through the development of a new scheme for adaptive management of network-level QoS. The scheme addresses the management issue from a new perspective through posing it as a problem of learning from current system behavior while creating new policies at run-time. The scheme utilizes forecasting functions to estimate the impact of different adaptation decisions and to guide the decision-making process of adapting the behavior of network components.

Theoretical analysis and experimental studies using real data are presented to demonstrate the performance of the proposed schemes.

## **Dedication**

*To the spirit of my father for his eternal inspiration, to Joseph, my son, for his infinite love, to my mother for her endless support, and last but not least, to my husband for his incessant encouragement.*

## **Acknowledgements**

It is extremely difficult to dedicate a single page to mention and thank all those wonderful people who helped me during my Ph.D. journey at the University of Ottawa. It is even harder to try to harness words to convey my deep feelings of gratitude.

At the forefront of those whose support and mentorship presented the major driving force to bring this work into completion, comes professor Ahmed Karmouch. On both the technical and personal levels, professor Karmouch's support has been pivotal in shaping my goals and focus for the rest of my life. He has always supported me through all the ups and downs of my journey. I thank him for his kind supervision, constructive discussions and continuous motivation. I would like also to thank my colleagues at the Multimedia and Mobile Agents Research Laboratory for the fruitful discussions during our weekly meetings.

Words are not enough to thank my husband, for giving me courage to go on during my darkest moments, for teaching me how to be persistent in pursuing my dreams when they seem out of reach and for his patience during many busy weekends. My deepest gratitude to my mother for her unconditional patience, understanding, continuous support and encouragement. I want also to thank my brother for his supportive help and encouragement. I also wish to express my thanks to my officemate Jennifer for providing a cheerful and comfortable working environment.

Last, but not least, I would like to dedicate this dissertation to the spirit of my deceased father who always inspired me to continue in the research field following his footsteps. His words are always in my mind and I hope I made him proud.

# Contents

<b>List of Tables</b>	<b>x</b>
<b>List of Figures</b>	<b>xi</b>
<b>Abbreviations</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Network Management Challenges . . . . .	3
1.3 QoS Management Challenges . . . . .	4
1.4 Motivation . . . . .	6
1.5 Dissertation Overview . . . . .	8
1.6 Summary of Contributions . . . . .	10
1.7 Organization of the Dissertation . . . . .	11
<b>2 STATE-OF-THE-ART IN Network Management Schemes</b>	<b>13</b>
2.1 QoS Definitions . . . . .	14
2.2 QoS Abstraction Levels . . . . .	16
2.2.1 User-Level QoS . . . . .	16
2.2.2 Application-Level QoS . . . . .	17

2.2.3	System-Level QoS . . . . .	19
2.2.4	Network-Level QoS . . . . .	20
2.3	QoS Management . . . . .	21
2.3.1	QoS Specifications . . . . .	21
2.3.2	Management Functionalities . . . . .	22
2.3.3	QoS Monitoring . . . . .	26
2.3.4	Fault Diagnosis . . . . .	26
2.4	Standardized Models for QoS Management . . . . .	27
2.4.1	Open Systems Interconnection (OSI) QoS Model . . . . .	27
2.4.2	ITU-T Model . . . . .	28
2.4.3	Integrated Service Model . . . . .	28
2.4.4	Differentiated Service Model . . . . .	30
2.4.5	Multi-protocol Label Switching . . . . .	31
2.5	QoS Management Approaches in Wired Networks . . . . .	31
2.5.1	Application Level Approaches . . . . .	32
2.5.2	QoS Routing Mechanisms . . . . .	33
2.5.3	Network level Reservation Based Approaches . . . . .	34
2.5.4	Network Level Adaptive QoS Management . . . . .	35
2.5.5	Differentiated Services Extensions . . . . .	39
2.5.6	Policy-Based Network Management Approaches . . . . .	40
2.5.7	QoS Management Using Active Networks Technology . . . . .	42
2.5.8	Individual Efforts at Different Protocol Layers . . . . .	43
2.6	QoS Management Approaches for Wireless Networks . . . . .	44
2.6.1	QoS Management Challenges in Wireless Networks . . . . .	45
2.6.2	Application-Level Approaches . . . . .	46
2.6.3	Network-Level Reservation Based Approaches . . . . .	47

2.6.4	Network-level Based Adaptive Management Approaches . . . . .	49
2.6.5	Extensions for Differentiated Services . . . . .	50
2.7	End-to-End QoS Schemes for Heterogeneous Networks . . . . .	51
2.8	Automated Management of QoS-Enabled Networks . . . . .	52
2.9	Discussion . . . . .	53
<b>3</b>	<b>Autonomous Management of Hybrid Systems</b>	<b>56</b>
3.1	The Underlying Network Model . . . . .	57
3.1.1	Network Model . . . . .	57
3.1.2	Layered Model . . . . .	58
3.2	QoS Management System Requirements . . . . .	59
3.3	Proposed Management Architecture . . . . .	61
3.4	Information Gathering layer . . . . .	64
3.4.1	User Agent (UA) . . . . .	64
3.4.2	Application Agent (AA) . . . . .	66
3.4.3	Environment Agent (EA) . . . . .	66
3.4.4	Domain Agent (DA) . . . . .	66
3.5	Prediction Layer . . . . .	67
3.5.1	Mobility Prediction Agent (MPA) . . . . .	68
3.5.2	Network-status Prediction Agent (NPA) . . . . .	68
3.5.3	User-side Service Prediction Agent (USPA) . . . . .	69
3.5.4	Network-side Service Prediction Agent (NSPA) . . . . .	71
3.6	Adaptation Layer . . . . .	73
3.7	Monitoring Layer . . . . .	74
3.8	Agents Interactions for Service Negotiation . . . . .	75
3.9	Features of the Proposed Framework . . . . .	76

<b>4</b>	<b>Mobility Prediction Scheme via Dempster-Shafer</b>	<b>78</b>
4.1	Introduction . . . . .	79
4.2	Related Work and Motivation . . . . .	80
4.3	Contextual Knowledge and Mobility Prediction . . . . .	83
4.4	Review of Dempster-Shafer Theory . . . . .	84
4.4.1	Mass belief assignment . . . . .	85
4.4.2	Evidence combination . . . . .	85
4.5	Proposed Mobility Prediction Scheme . . . . .	87
4.5.1	Information Gathering . . . . .	87
4.5.2	Evidence Extraction . . . . .	91
4.5.3	Decision Making . . . . .	96
4.5.4	Path Finding . . . . .	97
4.5.5	Illustrative Examples . . . . .	99
4.6	Simulation Details and Results . . . . .	100
4.6.1	Mobility Data and User Context Collection . . . . .	103
4.6.2	Mobile User Predictability . . . . .	105
4.6.3	Prediction Accuracy . . . . .	107
4.6.4	Accuracy Measurements . . . . .	109
4.6.5	Results of Trajectory Prediction . . . . .	111
4.6.6	Accuracy of the MPA vs. Number of Visited Locations . . . . .	113
4.6.7	Prediction Accuracy and the MPA Warm-Up Period . . . . .	113
4.6.8	Complexity Analysis . . . . .	114
4.6.9	Memory Requirements Analysis . . . . .	115
4.7	Summary . . . . .	115

<b>5</b>	<b>Anomaly Diagnosis via Statistical analysis and Evidential Reasoning</b>	<b>116</b>
5.1	Introduction . . . . .	117
5.2	Related work . . . . .	119
5.3	Outline of the Proposed diagnosis Scheme . . . . .	121
5.4	Step 1: Change detection . . . . .	122
5.5	Step 2: Dempster-Shafer for Anomaly diagnosis . . . . .	125
5.5.1	bpa for balanced training sets . . . . .	127
5.5.2	Anomaly diagnosis . . . . .	129
5.6	bpa for imbalanced training sets . . . . .	129
5.7	Adjustable bpa for imbalanced training sets . . . . .	132
5.8	Performance Analysis of the NPA . . . . .	135
5.8.1	Performance of Abrupt Change detection scheme . . . . .	135
5.8.2	Accuracy estimation of different <i>bpas</i> . . . . .	136
5.8.3	Experimental results . . . . .	146
5.8.4	Computational Complexity . . . . .	148
5.9	Summary . . . . .	149
<b>6</b>	<b>Policy-Based Adaptive Management</b>	<b>153</b>
6.1	Introduction . . . . .	154
6.2	Policies- Basic concepts . . . . .	154
6.3	Existing Policy Adaptation Schemes . . . . .	156
6.4	Proposed Policy Adaptation Approach . . . . .	159
6.4.1	Stage Setting: Information Gathering . . . . .	161
6.4.2	Candidate Actions Selection . . . . .	164
6.4.3	Policy Assembly . . . . .	165
6.4.4	Reassessment . . . . .	165

6.4.5	Illustrative Examples . . . . .	166
6.5	Simulation Details and Results . . . . .	167
6.5.1	Experiment 1: User-Based Policy Adaptation . . . . .	169
6.5.2	Experiment 2: Application-Based Policy Adaptation . . . . .	170
6.5.3	Experiment 3: Network Adaptation . . . . .	172
6.6	Summary . . . . .	175
<b>7</b>	<b>Conclusions and Future Research Directions</b>	<b>177</b>
7.1	Dissertation Contributions . . . . .	177
7.2	Future Research Work . . . . .	179
7.2.1	Case-Based Adaptive QoS Management . . . . .	179
7.2.2	Network Contextualization . . . . .	180
	<b>List of Publications</b>	<b>181</b>
	<b>Bibliography</b>	<b>184</b>

# List of Tables

- 3.1 Sample of policies as collected by different information gathering agents . . . . 65
  
- 5.1 Alarm rate over a one month period. . . . . 137
- 5.2 Anomaly detection accuracy. . . . . 140
- 5.3 Ratio of correctly to incorrectly classified DI vectors. . . . . 149
  
- 6.1 PAA Objectives . . . . . 167
- 6.2 Sample of PAA actions for an edge router . . . . . 168
- 6.3 Sample of PAA actions for a core router . . . . . 168

# List of Figures

3.1	The network reference model. . . . .	57
3.2	The layered reference model. . . . .	59
3.3	AUTONEMA layered model . . . . .	63
3.4	A schematic description of the adaptation process. . . . .	63
3.5	USPA MAUFs construction steps . . . . .	71
3.6	$\mathcal{U}_1(\cdot) = 0.5U_Q^{P_1^u}(\cdot) + 0.5U_R^{P_1^u}(\cdot)$ , $\mathcal{U}_2(\cdot) = 0.3U_Q^{P_2^u}(\cdot) + 0.7U_R^{P_2^u}(\cdot)$ . . . . .	72
4.1	Mobility Prediction Agent . . . . .	88
4.2	An Example of an SCM and MOAC representations of a spatial map. . . . .	90
4.3	An example of evidence-based mobility prediction . . . . .	98
4.4	Schematic representation of the simulation model. . . . .	100
4.5	Campus map . . . . .	101
4.6	User Context Model . . . . .	102
4.7	SCM representation of campus map . . . . .	103
4.8	Part of an XML file for a student . . . . .	104
4.9	Predictability degrees of different students . . . . .	107
4.10	Expected prediction accuracy vs. number of locations . . . . .	109
4.11	Error between points of actual and predicted trajectories vs. time . . . . .	111
4.12	Actual trajectory vs. predicted trajectory of a student with 0.8 predictability. . .	112

4.13	Actual trajectory vs. Predicted trajectory of a student with 0.6 predictability. . .	112
4.14	Actual trajectory vs. Predicted trajectory of a student with 0.4 predictability. . .	112
4.15	Hit ratio vs. number of visited locations . . . . .	113
5.1	A schematic description the NPA components . . . . .	121
5.2	Illustration of the performance of the signature-based scheme in two special cases. . . . .	131
5.3	Increments of <i>ifInOctets</i> (upper panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments). . . . .	137
5.4	Increments of <i>ifOutOctets</i> (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments). . . . .	138
5.5	Increments of <i>ipInReceives</i> (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments). . . . .	139
5.6	Increments of <i>ipInDelivers</i> (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments). . . . .	140
5.7	Increments of <i>ipOutRequest</i> (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments). . . . .	141
5.8	Probability Expectation of $\hat{F}$ for different <i>bpas</i> . . . . .	145
5.9	Classification of root-cause failures using training set #1 ( $F_1$ as squares, $F_2$ as triangles, $F_3$ as circles, $F_4$ as diamonds, filled shapes indicate a misclassification). 150	
5.10	Classification of root-cause failures using training set #2 ( $F_1$ as squares, $F_2$ as triangles, $F_3$ as circles, $F_4$ as diamonds, filled shapes indicate a misclassification). 151	
5.11	Classification of root-cause failures using training set #3 ( $F_1$ as squares, $F_2$ as triangles, $F_3$ as circles, $F_4$ as diamonds, filled shapes indicate a misclassification). 152	
6.1	Stages of adaptive policies assembly and modification . . . . .	161
6.2	Experimental testbed . . . . .	169

6.3	Adaptation based on user policy . . . . .	171
6.4	Adaptation based on Application policy . . . . .	172
6.5	Forecasted values for $A_1$ w.r.t. throughput of EF aggregate. . . . .	173
6.6	Forecast function for $A_2$ w.r.t. EF aggregates throughput and loss rate. . . . .	174
6.7	Throughput comparison between the statically configured network and the proposed scheme . . . . .	176

## Abbreviations

AA	Application agent
AF	Assured forwarding
AR	Auto regression
ARRSES	Adaptive-response-rate single exponential smoothing
AUTONEMA	Autonomous network architecture
<i>bpa</i>	basic probability assignment
BS	Base station
CBR	Case-based reasoning
CC/PP	Content capabilities/profile preferences
CLI	Command line interface
COPS	Common open policy service
CoS	Class of service
DA	Domain agent
DiffServ	Differentiated service model.
DI	Deviation Denial of service attacks
DST	Dempster-Shafer theory
EA	Environment agent
EF	Expedited forwarding
FF	Forecast function
FTP	File transfer protocol
IETF	Internet engineering task force
GDU	Gradual distribution of uncertainty scheme
GLR	Generalized likelihood ratio test
IntServ	Integrated service model

ISO	International standards organization
ISP	Internet service provider
ITU	Telecommunication union
LAN	Local area network
MA	Monitoring agent
MAUF	Multi-attribute utility function
MAUT	Multi-attribute utility theory
MIB	Management information base
MIPv6	Mobile IP version 6
MLS	Mobile location service
MOAC	Matrix of orientation, adjacency, and characterization
MPA	Mobility prediction agent
MPLS	Multi-protocol label switching
MT	Mobile terminal
MV	Majority-voting scheme
NDU	No distribution of uncertainty scheme
NPA	Network prediction agent
NSPA	Network-side service prediction agent
OWAMP	One-way active measurement protocol
PA	Prediction agent
PAA	Policy adaptation agent
PDP	Policy decision point
PEP	Policy enforcement point
PHB	Per hop behavior
QoS	Quality of service

RM-ODP	Reference model for open distributed processing
RMON	Remote network monitoring
RTCP	Real-time Transfer control Protocol
RTT	Round trip time
RSVP	Resource reservation protocol
SCM	Spatial conceptual map
SB	Signature based scheme
SLA	Service level agreement
SNMP	Simple network management protocol
SSH	Secure shell
UA	User agent
UMTS	Universal mobile telecommunications system
USPA	User-side service prediction agent
VoD	Video-on-demand
WEA	Way elementary area
WLAN	Wireless local area network
WWW	World wide web

# Chapter 1

## Introduction

### 1.1 Overview

Early communication systems were originally designed for simple data transfer applications such as file transfer (FTP) and remote access (Telnet) among different fixed computers connected together. These applications had limited resource demands and imposed no time-constrained requirements. At that time, the concept of best-effort services was sufficient to meet all users' demands and it was even favorable for its simplicity. The simplicity of the best-effort type of service at that time stemmed from its basic concept where all network traffic is treated equally and blindly.

In the early nineties, the introduction of the World Wide Web (WWW) shifted the development of earlier communication systems from being limited to scientists and military usage towards a widely recognized global phenomenon. Another factor that contributed to the wide spread of this phenomenon is the emergence of new applications, such as video conferencing, Web searching, electronic media, discussion boards, and Internet telephony; moreover, the development of these applications was taking place at an unprecedented speed. As a result of the increasing complexity of the underlying networks and the rapid growth of resource-intensive

applications, the simple best-effort service quickly fell short of providing the necessary sophisticated traffic engineering in order to differentiate between applications with strict demands and other tolerant ones. Since the already dominating IP protocol did not inherently support the preferential treatment of network traffic, the burden fell on network managers and service providers to design and implement mechanisms that are aware of the nature of different applications demands and can conform with various users' requirements. This has led to a continuous struggle of management system paradigms to keep pace with the continuously increasing demands and advancing technologies.

Another aspect that has contributed an additional dimension to the complexity of the management task is the phenomenon of wireless data communications. The advent of third generation wireless infrastructures and the rapid growth of wireless communication technologies enabled people with battery-powered mobile devices to access data *anytime* and *anywhere*. Nevertheless, in spite of the success and wide spread of these infrastructures, the issue of management of wireless networks remained an increasingly challenging issue. The difficulty therein stems from the existing dichotomy between the innate nature of wireless environments and their pre-existing wired infrastructure. Due to the unique characteristics of wireless networks, such as higher error losses, delay and bandwidth variation, existing management schemes designed for wired networks could not be directly applied to wireless systems.

In general, management of any communication system refers to the task of planning, allocating, configuring, deploying, administering, diagnosing and maximizing the utilization of the underlying network resources. Functionalities of a management system also include other tasks such as authorization, security management, reliability assurance, and performance guarantees. One aspect of management that best exemplifies the magnitude of the aforementioned challenges is typically encountered in *Quality of Service (QoS)* provisioning. QoS management refers to the activities of QoS specification, negotiation, monitoring and control of network limited resources to meet end-to-end users and applications requirements, business objectives and

resource availability while optimizing the overall system performance. To date, notwithstanding the significant progress that has been made in developing management components pertaining to different layers (e.g., application, transport, link or network layers), little progress has been made in addressing the issue of designing an overall autonomous management framework for hybrid communication systems that can be self-configurable and -adaptable by automating the management tasks while minimizing explicit operator or administrator involvement.

The work presented in this dissertation aims at developing an autonomous and self-adaptable management framework for hybrid wired/wireless communication systems that meets QoS guarantees. This chapter briefly discusses different aspects of autonomous network management and focuses on challenges of QoS management. It further identifies the motivations behind the proposed work. Subsequently, the proposed management architecture is briefly described and thesis contributions are outlined. Finally, the organization of the remainder of the dissertation is presented.

## 1.2 Network Management Challenges

In the past, as networks were limited in size and functionality, management tasks were performed manually. Experienced administrators relied on the analysis of a limited set of device variables to take management decisions. They opted to utilizing simple command line interfaces (CLI), such as telnet and Secure Shell (SSH), for direct configuration and management of network components. A simple system reboot was sufficient to handle network faults.

As networks continue to grow in size and complexity, the task of diagnosing network's performance increasingly involved the collection and analysis of an immensely growing amount of high dimensional data. Furthermore, with the introduction of new types of applications, management tasks become very time-sensitive, where a couple of minutes of experiencing service unavailability became intolerable. Fortunately, advances in network monitoring and capturing

tools, such as the simple network management protocol (SNMP) facilitated the capturing of the necessary measurements to describe the fine grained behavior of each networked equipment. Nonetheless, analysis of such data in order to diagnose network performance and conclude the necessary configurations to maintain a smooth operation is still considered an unmet challenge and remains to be the focus of numerous ongoing research efforts.

A particular difficulty in diagnosing a network's performance arises from the non-stationary and stochastic nature of data representing network-performance measurement variables in addition to the erroneous nature of the collected data due to possible glitches in the monitoring systems. Furthermore, selecting the appropriate system configurations in order to maintain an acceptable level of the quality of the delivered services is complicated by its reliance on several parameters, such as the degree of mobility of the serviced users, the severity of the detected anomaly, if any, the variations in the underlying traffic and so on. In the following section, we further elaborate on one particular important aspect in network management, namely QoS.

### **1.3 QoS Management Challenges**

QoS management is an end-to-end functionality which is concerned with providing users with their required services given some degree of quality guarantees. One tempting, yet self-defeating, solution is to over-provision network resources. However, over-provisioning fails in two aspects; firstly, peaks in network traffic and the bursty nature of most multimedia applications make it virtually impossible to have enough bandwidth for all possible occasions or even to predict which routers will experience high traffic demands and during which periods. Secondly, over-provisioning results in a waste of network resources; hence, it is extremely cost inefficient and unfavorable to network operators. Furthermore, it represents a poor management strategy, since network providers are not interested in blindly satisfying all demanded QoS, even for aggressive flows, without any preferential treatment.

One issue that contributes to complicating the QoS management task is that QoS is generally perceived from two almost contradicting angles; users' perspective and service providers' perspective. The users' perspective of the QoS management is fundamentally limited to the ability to access a set of services that are customized to their needs and delivered in a timely manner. The main goal of users is to be able to get the best service quality with the least possible cost.

From the administrators and providers' perspective, QoS management deals with the satisfaction of two objectives. While the major objective of service providers is to provide users with the desired services in a timely manner with performance guarantees, they are concerned with another goal of utilizing resources as efficiently as possible and according to a predefined set of business objectives in order to maximize their total revenue. According to their view, the fundamental idea behind any QoS management solution should be that the network handles traffic differently depending on the application, the kind of data being sent and the classification of involved users. However, since any system is characterized with a finite set of resources, it is clear that these two goals are contradicting.

In order to allot resources preferentially to certain traffic, it is necessary to identify different traffic and to associate it with certain resources. During loaded periods, the management system may not have sufficient resources to deliver the required quality to every running application. Hence, decisions must be made to be able to dimension the available resources among these applications such that a global objective is maximized. The management architecture has to take different decisions, such as delaying or dropping packets, shaping traffic or even denying service to certain flows. Given the complexity of the underlying network components and the bursty nature of the serviced traffic, the selection of the right strategy is not a trivial task. Even if configurations were estimated to best optimize network resources at a certain time, these static configurations may not present an equally optimal solution at all periods of high loads.

Another important challenge for QoS management is the ability to sustain an acceptable service level with the existence of network faults. A major issue with network faults is that they are mostly unpreventable. Nonetheless, the earlier they are detected and diagnosed, the faster for the management system to make the necessary system reconfigurations in order to maintain as much acceptable system performance as can be achieved.

The management task is further complicated by the introduction of mobility concepts where users roam along different domains and change their point of attachment to the network as they move. They may even be serviced by different providers during one running session. Nevertheless, they expect that their applications will always have the same QoS delivered regardless of their location. Besides the stringent nature of the wireless links, such as fading, terminal limitations and bandwidth scarcity, handoff operations represent another unavoidable source of increasing latency to the delivered QoS.

## 1.4 Motivation

Much progress in existing research work has been made on providing the ability to manage QoS requirements, and manage situations when the required QoS is not available. Furthermore, formal notations, standards, and models have been lately introduced in literature.

Adaptive management systems represent an obvious solution to the problem of QoS management. A closer examination of existing adaptive management techniques shows that they can be classified into two distinct approaches for adaptation; adaptation with respect to the network and the operating system components and adaptation at the application level. Application-based adaptation solutions provide the means to co-exist with heterogeneous QoS-unaware end-to-end best-effort computing environments. Adaptive applications can accept and tolerate resource scarcity by dynamically changing demands within a specific range of QoS requirements based on availability of existing resources. Apparently, applications which have strict

hard real-time requirements do not fit in this category. On the other hand, network level adaptation solutions provide flexible means for the management of the underlying variable resources. Nevertheless, existing adaptation frameworks still have certain limitations. These techniques usually lack an essential degree of flexibility. On one hand, they are heavily dependent on decisions taken by human operators. On the other hand, they fail to build upon past experiences gained from the impact of previously pursued adaptation strategies on system behavior.

A major limitation of most of the existing frameworks arises from their static configurations that are built a-priori by administrators into network devices. These frameworks usually lack the flexibility required by wired/wireless communication environments and may not be sufficient to handle different changes in the underlying environments. Another direct consequence of that static configuration nature is that network administrators recourse to using estimates of network traffic and users' requirements in configuring network components. These estimates can be a major source of inflexibility. For example, an initial distribution of network resources to different classes of network services may not come close to optimal utilization of these resources without factoring in past and future forecasted traffic loads. Furthermore, with the current highly competitive market of service providers, besides service quality, service cost becomes an important factor. Nevertheless, the reliance on human operators is a major contributor to the current cost of services.

Finally, in current management systems, network reconfiguration in response to users requests for service customization can only be performed manually by a network operator. This results in significant delays ranging from minutes to days. Existing frameworks must be extended to support dynamic QoS selections so that customers are able to precisely tailor individual connections to their particular requirements. It is usually disadvantageous to limit the QoS negotiation at the time of connection establishment. Specified QoS levels do not often remain valid for the lifetime of the entire session.

The aforementioned limitations of current management frameworks represent strong mo-

tivations for the development of a novel autonomous self-adaptable management architecture with inherent dynamic capabilities. This framework must manage, customize and extend the underlying complex infrastructure of communication systems resources in response to the continuously changing business objectives and users' requirements. By making the management systems more autonomous, the need for direct and continuous involvement of human operators is reduced.

## **1.5 Dissertation Overview**

This thesis approaches the issue of QoS management from two different, albeit related, levels; the first level is concerned with predicting future changes in the underlying network and in users' demands. In the second level, an adaptive management of available resources is developed to manage available network resources based on these changes.

The predictive phase deals with two challenging issues, namely, predicting future locations of mobile users and detecting, and if possible predicting, network components' faults. QoS management is particularly challenging in the case of wireless systems since the service quality depends on the geographical location of the mobile user. This work addresses this issue through the utilization of Dempster-Shafer theory of evidence along with available environment context in order to accurately predict the user's location. On the other hand, predicting, if possible, or detecting and diagnosing probable network faults, allows the management system to reliably deliver the required services in a timely manner. This is achieved through the development of an efficient network anomaly diagnosis scheme that relies on statistical analysis of changes in the monitoring variables to detect possible performance deterioration. A measure of the network deviation from its normal behavior is calculated and fed into a novel classification algorithm in order to identify the most probable root-cause fault in the network.

The adaptive phase is approached as a dynamic process where network components are

configured at run-time, rather than statically by network administrators, in response to the predicted changes gathered during the first phase. To facilitate this task, policies are utilized as tools to continually guide the behavior of the underlying network. The adaptation process is carried-out by assembling, modifying or deleting these policies at run time. The proposed adaptation scheme is posed as a problem of learning from previously applied adaptation strategies. Using the results of this learning process, the appropriate management strategies, described as policies, are created and applied at run-time.

Incorporating the above two phases is carried out through a quadri-layer autonomous framework. In the first layer, the necessary data pertaining to users, applications and network components is collected. The mobile users' future travelling trajectories are predicted along with probable network failures in the second layer. This information is reported to the third layer in which the underlying network components are dynamically reconfigured to best utilize available resources while maintaining a smooth service delivery to mobile users. The fourth layer is equipped with monitoring components that are dedicated to provide a feedback mechanism based on QoS measurements. The objectives of the proposed research work can be summarized as follows,

- *Autonomous Management:* The management system has to be self-adaptable and self-reconfigurable in response to changes in the surrounding environment at different layers.
- *Simplify human management tasks:* By introducing autonomicity into management systems administrators are shielded from unnecessary details of management and freed up to other design and development tasks.
- *Scalability:* The performance of the management system has to be maintained regardless of the number of managed network components or the amount of experienced traffic.
- *Proactive Management:* Rather than waiting for QoS violations as a trigger for QoS

adaptation, the system has to pro-actively detect possible violations and react accordingly.

- *Maximize resource utilization:* Similar to all management systems, the key goal of the proposed framework is to maximize the utilization of the underlying network resources.
- *Accommodation of effects of mobility related operations:* The management framework must minimize service disruption during mobility management operations such as mobile terminal handoff.
- *Accommodation of effects of network faults:* The management tasks should be performed in order to achieve the best possible services even with the existence of internal anomalies (e.g., congested routers and failed servers) and external ones (e.g., Denial of service attacks).
- *Learning:* The management system must possess the ability to gain knowledge from past experiences of previously applied decisions.
- *Flexibility:* The management system must provide users with tools for flexible QoS specifications, accommodate changes in applications requirements, and provide administrators with flexible tools for the specification of business objectives.

## 1.6 Summary of Contributions

The major contributions of this dissertation can be summarized as follows,

1. A novel algorithm for mobility prediction through the utilization of Dempster-Shafer theory of evidence is developed. The proposed approach does not require prior training nor does it assume the availability of a history of previous movements [1], [2], [3], [4].

2. A robust scheme for network anomaly detection and classification via statistical analysis and evidential reasoning. To handle cases of imbalanced training sets, the scheme includes two new approaches to assign mass belief values to the classes under investigation. The latter contribution can be of independent interest to research related to the Dempster-Shafer theory and may be applied in contexts other than the present one [5].
3. An adaptive policy-based QoS management scheme is developed through utilizing forecasting functions to estimate the impact of different adaptation decision and to guide the decision-making process of adapting the behavior of network components. Forecasting functions generally feature desirable advantages in that recent history can be captured efficiently with negligible storage requirements [6], [7], [8].
4. A novel hierarchical framework for autonomous network management of wired/wireless differentiated communication systems is proposed. It identifies the necessary components in order to automate the management tasks [9], [10], [11], [12].

## 1.7 Organization of the Dissertation

The remainder of the dissertation is organized into the following chapters.

Chapter 2 presents related work and discusses various approaches adopted by the research community. It identifies different limitations of various research efforts for QoS management over wired/wireless networks.

Chapter 3 outlines the design of the proposed adaptive management framework. A hierarchical approach is employed to facilitate adding needed functionalities to the framework in a phased manner. Responsibilities of the different layers along with their interactions are specified.

Chapter 4 presents a novel scheme for a mobility prediction based on the use of contextual

information and spatial conceptual maps. Simulation results are also presented to demonstrate the performance of the proposed scheme.

Chapter 5 presents a robust scheme for network anomaly diagnosis via statistical analysis and evidential reasoning. Theoretical analysis along with experimental evaluation using real-data are presented to demonstrate the accuracy of the proposed scheme.

Chapter 6 first introduces concepts of policy-based management systems and provides the necessary background for policy configurations. Subsequently, the proposed scheme for adaptive network level management based on the utilization of forecasting functions is discussed.

Finally, Chapter 7 summarizes the presented contributions and discusses directions of future research work.

## **Chapter 2**

# **STATE-OF-THE-ART IN Network**

## **Management Schemes**

Before embarking on describing the proposed architecture to resolve problems posed by QoS management requirements for hybrid systems, we start by characterizing the QoS problem and management difficulties. This chapter presents a survey of current research efforts related to the QoS management problem. We aim to address two questions; what are the requirements of any QoS management system? and why is it difficult to satisfy these requirements with current approaches? This chapter is organized as follows. Various definitions of QoS used in literature are first presented in Section 2.1. QoS concepts, terminologies and parameters at different levels of abstraction are subsequently given in Section 2.2. Section 2.3 discusses a formalization of the QoS problem and the challenges faced by the management systems. Some of the QoS models and standardization efforts that have been proposed are then highlighted in Section 2.4. Existing research work that has been carried-out in the area of QoS management in wired, wireless and hybrid communication systems is reviewed in Sections 2.5, 2.6 and 2.7, respectively. Current efforts for the development of automated management systems are presented in Section 2.8. Section 2.9 concludes the chapter with a discussion which summarizes

existing contributions and identifies some open issues.

## 2.1 QoS Definitions

The term QoS by itself is a universal term that can be applied to various areas (e.g., software quality, data quality). In that context, it generally reflects the degree of satisfaction of a consumer of a particular service. In communication systems, it is related to different aspects of the delivered network services such as end-user quality perception, application-level quality of service as well as network and system performance. Albeit broadly used in the area of network management, there is no common or formal definition of QoS in this field of research. Nevertheless, there are a number of definitions found in standards and in literature.

- In [13], quality is defined by the ISO/OSI as *"The totality of features and characteristics of a product or a service that bear on its ability to satisfy stated or implied needs"*. In network systems, the terms *product or a service* may refer to a movie when delivered to end users, the available bandwidth for an application, or even the preferential treatment of traffic from one network domain while passing through the other. In spite of the simplicity of the ISO definition, a closer examination identifies two major issues. Firstly, as will be discussed later in this chapter, the formalization of the features and characteristics of any of these services is a vital step in the QoS management process, yet it is not a straightforward task. Secondly, the problem of QoS is not mainly concerned with achieving the *best quality*, but rather to ensure that the service is always delivered with certain pre-stated or implied qualities.
- The International Telecommunication Union (ITU) [14] refers to QoS as *"the collective effect of service performances which determine the degree of satisfaction of a user of the service"*. Similar to the ISO definition of QoS, this definition relates the QoS to the

users' contentment with the delivered service.

- ISO's definition in [15] refers to the QoS as "*a set of qualities related to the collective behavior of one or more objects*". Here, an object refers to one of the components of the distributed system in the Reference Model for Open Distributed Processing (RM-ODP). This concept can be applied to network level QoS by considering network components as sets of related objects.
- The most used network related definition of QoS is the one given by the Internet Engineering Task Force (IETF) working group. QoS is understood by IETF [16] as "*a set of service requirements to be met by the network while transporting a flow*". This definition focuses more on network behavior rather than on user related requirements.
- Another QoS definition, specifically tailored for application-to-network QoS relation, is presented in [17]. It indicates that "*Quality of Service is a concept based on the statement that not all applications need the same performance from the network over which they run. Thus, applications may indicate their specific requirements to the network, before they actually start transmitting data*". Here, the definition views QoS as a contract between a consumer, i.e., the application, and a producer, i.e., the network.
- Vogel et al. [18] describe QoS as "*the set of quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application*".

In general, based on the definitions given by the standardization bodies (e.g., IETF, ITU-T and ISO), QoS can be viewed as either a notion that is focused on user-perceivable effects (e.g., ITU definition) or a set of services related to network functionalities required to provide a certain service to the users (e.g., IETF definition). In this dissertation, we adopt a more general approach which perceives QoS as *a set of characteristics described at different layers, namely,*

*user, application, operating system and network layers. Furthermore, the satisfaction of these characteristics guides the behavior of components functionalities at these different layers.* More precisely, QoS is viewed as an integrated relation between different specifications as well as functionalities of various components at these different layers.

Based on the aforementioned definition, we further define *QoS management* as *the necessary supervision and control of functionalities at different interrelated layers to ensure that the desired quality of service properties at those layers are maintained.*

## 2.2 QoS Abstraction Levels

Based on the adopted definition of QoS in this dissertation, QoS characteristics are described at different hierarchical layers of abstractions. Specifically, four main abstraction levels are identified in systems subject to QoS characterization; the user level, the application level, the system level and finally the network level. Although they can differ strongly in meaning, some of the characteristics specified at these different levels are usually interdependent and can be derived or mapped along these layers in both directions of the layered hierarchy. The following sections briefly discuss QoS specifications at these layers.

### 2.2.1 User-Level QoS

The user-level represents the highest level of abstraction for QoS specification yet it is the hardest to capture. QoS at this level is related to the subjective user experience at the access point of the service. In general, the user's view of QoS is usually expressed by parameters that are relevant to user-perceivable effects, independent of the network design and measures. A user's perceived QoS can be expressed in term of attributes describing the received media (e.g., video smoothness, color depth, audio quality), attributes related to the entire session

(e.g., establishment time, connectivity), or attributes related to the quality of service obtained from the service provider (e.g., accessibility, availability, reliability, performance). The main concern of the work in this dissertation is the QoS related to the received media.

In general, the task of evaluating user's QoS requirements for a received media is complicated by many factors; the perception of the user is a function of different factors such as the purpose of the interaction, the surrounding environment and the interaction between different concurrently running applications. Recent studies, e.g., [19], [20], [21], have developed different qualitative and quantitative methods appropriate for capturing users QoS specifications into both application- [22] and network- level parameters [21]. These techniques range from simple systematic mechanisms such as a range to value mapping and utility functions to more complicated frameworks that attempt to model the users' behavior in order to capture these requirements.

In order to guarantee a quantifiable performance, Service Level Agreements (SLAs) [23] are used as a formalization of the users' QoS requirements. SLAs formally describe the relationship that exists between a service provider and users. An SLA is a contract between a service provider and customers that specifies in quantitative values the QoS provided by the former to the latter and the penalties the former is expected to pay if the committed agreement is not satisfied. In this work, the existence of a dynamic SLA is assumed, i.e., users can dynamically reconfigure their QoS requirements at run-time.

### **2.2.2 Application-Level QoS**

Within the application-level, QoS is described in application oriented parameters of the media. Some of these parameters are obtained by mapping users QoS specifications. Such parameters are, for instance, video frame rate, picture size, color depth or the requested quality in terms of coding formats or bit depth.

Broadly speaking, an application parameter is a measure of a specific feature related to the application itself. For example, the number of frames of video shown per second in a video broadcast application is meaningless for an audio application while a frequency range used for an audio application can not be applied to a file transfer application. However, to simplify the presentation of QoS at the application level, one can abstract QoS specifications of most applications into a common QoS metric that includes the following parameters:

- *Throughput*: This is the effective share of bandwidth the application is getting from the network.
- *Latency*: This is the end-to-end delay that the application experiences. The application level end-to-end latency is the sum of all delays accumulated through the entire flow pipeline. This includes sender coding and packetization, network transmission, reception and decoding. Latency directly affects the user satisfaction, for example a high latency in a voice stream can result in an overlapping conversation.
- *Information loss*: It is worth noting that at the application level, information loss does not necessarily coincide with data packet loss at the network level (e.g., loss of fidelity due to encoding).
- *Delay variation*: is caused either by the dynamic changes of queues in the network or by process scheduling irregularities in end hosts and network routers. While certain levels of jitter can be absorbed by the application through the use of buffering, excessive variations of end-to-end latency may not be tolerated.

Based on this metric, applications can be broadly classified according to their QoS requirements as either real-time or non real-time media types [24].

- Non-real time traffic can be characterized by stringent bounds on loss but flexible latency and jitter constraints. This includes asynchronous applications like email and interactive

applications like Telnet and ftp. Two types of applications can be classified under the non real-time category, the interactive and background classes. Because they have looser delay requirements than the conversational and streaming classes, they can provide better error rates using channel coding and retransmission. The main difference between the interactive and background classes is that the interactive class is mainly used by interactive applications, e.g., interactive e-mail or interactive Web browsing, while the background class is meant for background traffic, e.g., background download of e-mail or other files.

- Real-time traffic is characterized by stringent bounds on delay and jitter. However, it is typically more tolerant to packet losses. Examples of real-time applications are real-time streaming voice and video. Real-time traffic can be further classified into conversational and streaming types. The conversational class supports real-time services like video telephony that have particularly very strict-end-to-end QoS requirements, especially with respect to end-to-end delay and jitter. Nevertheless, they have less demanding bandwidth requirements. The streaming class supports one-way flows, and therefore is somewhat less delay sensitive.

### 2.2.3 System-Level QoS

System-level QoS encompasses end-system resources required for execution of the processes. Such resources are the CPU, access to devices, main memory, system bus bandwidth, processing time to compress and decompress information, access to specialized hardware and Input/output resources. In order to give adequate support to the user, higher-level requirements have to be mapped according to the system QoS parameters. For example, a frame rate specification at the application level is mapped in an operating system into a period and a computation time.

### 2.2.4 Network-Level QoS

Network-level QoS parameters represent the lowest level of abstraction for QoS specification as they are easier to quantify and to measure. The main QoS parameters are packet bandwidth, delay, jitter and packet loss rate.

- *Bandwidth*: The maximal data transfer rate that can be sustained between two end points of the network is defined as the bandwidth of the network link. Bandwidth is not only limited by the physical infrastructure of the traffic path within the transit networks, which provides an upper bound to the available bandwidth, but is also limited by the number of other flows sharing common resources on this end-to-end path. The term bandwidth is used as an upper bound of the data transfer rate, whereas the expression throughput is used as an instant measurement of the actual exchanged data rate between two entities.
- *Delay*: Packet delay is the time required by a packet to travel from the sender to the receiver. It consists mainly of two components: propagation delay and congestion delay. Propagation delay is the time taken by a packet to propagate from the sender to the receiver and is dependent on the length of the data path over which the packet travels. The congestion delay is the time spent by a packet in a queue at different routers along the path.
- *Jitter*: The variation of the delay between the arrivals of successive packets is known as the delay variation or jitter.
- *Packet loss rate*: The fraction of undelivered packets per flow is known as the packet loss rate. In wireline networks, packet loss is mainly attributed to dropped packets in full queues due to congestion. On the other hand, in wireless networks, packet losses are caused by bit errors at the link layer, handoff operations as well as due to congestion.

## 2.3 QoS Management

If infinite network resources were available, then all application traffic could be carried at the required bandwidth, with zero latency, zero jitter and zero loss. However, network resources are not infinite. As a result, there are parts of the network in which resources are unable to meet demand. QoS management work by controlling the allocation of network resources to application traffic in a manner that meets predetermined service requirements. The QoS management problem can be further simplified into several main steps. The first step to provide service quality is the specification of offered services. These specifications are derived from requirements supplied by the service users or by the application itself and controlled by the service providers' business objectives. The second step includes mapping of the obtained requirements into parameters that can be used at the system and the network levels. Accordingly, network administrators utilize knowledge of service specifications, network topology, link properties (e.g., capacity, propagation delay) and availability of resources to perform different management operations such as routing, access control, admission control, network dimensioning, resource reservations and traffic provisioning. Finally, to ensure that the quality of the delivered services meets original specifications, QoS monitoring is performed to compare between the actual service and the previously specified requirements. In order to provide a deeper understanding of the network management task, the following sections present an overview of the main QoS management functionalities. It should be noted, however, that not all functionalities have to be applied to all network domains.

### 2.3.1 QoS Specifications

QoS specification entails the creation of an agreement between service providers and users that states the specification of the offered QoS. In general, there are four models of QoS specifications. These models vary in the degree of guarantees of the offered QoS.

- **No QoS guarantees:** This type of guarantee represents the well known best efforts service. The same QoS is offered for all traffic types. While it offers the cheapest types of service no traffic guarantees are conferred to end users.
- **Statistical (stochastic) QoS guarantees:** These guarantees are specified in terms of a predetermined percentage of time or the ratio of packets where the offered QoS may not be granted. An example of statistical guarantees is *network delay is guaranteed to be no more than 100 ms for 95% of the packets*.
- **Relative QoS guarantees:** In this case, the management system offers service quality to a certain number of predetermined classes. These classes are ordered based on their packet forwarding quality parameters. While the actual quality level offered to each class might vary with class loads, the ratio between classes remains fixed and controlled by the service providers. In other words, flows belonging to a higher service class are guaranteed to receive better service than a lower class.
- **Absolute QoS guarantees:** The strictest QoS specification model. Offered guarantees must be met independent of all other factors such as different concurrent network loads. For example, a user is given a network connection with a guaranteed 10Mbps bandwidth all the time. This type of guarantees is usually associated with penalty rates and service compensation agreement that must be performed by providers if the offered guarantees were not met.

### 2.3.2 Management Functionalities

The above QoS specifications are used in important network functions such as admission control, resource allocation, network dimensioning, traffic provisioning and monitoring. The purpose of these functionalities is to predict, maintain and monitor the underlying network behav-

ior to ensure that the contracted QoS is sustained. They can operate on different time scales and on different levels of traffic aggregations.

### Levels of traffic aggregations

In general, the functionalities of network management are carried out at different levels of granularity: packet-level, per-flow level and aggregates-level.

- **Per-packet operations:** Example of per-packet operations are packet classification and dropping.
- **Per-flow operations:** According to Aurrecochea et al., [25], “ *A Flow is defined to characterize the production, transmission and consumption of the data associated with a single media (e.g., audio, video, data)*”. In other words, flows represent streams of IP packets between end-hosts and applications which have the same source and destination addresses, the same TCP/UDP port numbers, and the same protocol field. In order to perform per-flow operations each flow state must be maintained at each router in the path. Examples of per flow operations are resource reservations and admission control.
- **Aggregate operations:** Since maintaining and processing millions of per-flow states inside a router is significantly difficult, flows with similar service requirements are grouped together to form flow aggregates. Operations such as buffering and scheduling for these flows can then be applied to each aggregate without having to maintain a per-flow state.

To achieve scalability, the goal of any management system is to minimize maintenance and processing operations for individual packets and even flows and rather focus on management of traffic aggregates.

### **Admission Control**

The role of any admission control algorithm is to ensure that admittance of a new flow into a resource constrained network does not violate service commitments made by the network to admitted flows. Admission control techniques are used in deciding if a new connection across the network is to be allowed or rejected. Such decisions must take into account the current state of the network nodes that will be servicing the requested traffic. The connection through these nodes should be allowed if the required resources are available and if giving them over to the new connection will not degrade the quality of the delivery of other active guaranteed services. In general, there are two basic approaches to admission control: a parameter-based approach, which computes the amount of network resources required to support a set of flows given a priori flow characteristics. The second approach, a measurement-based, which relies on measurement of actual traffic load in making admission decisions.

### **Network Dimensioning**

Network dimensioning is process of dimensioning the network bandwidth to best meet the expected demands. It involves the determination of the amount of bandwidth to be allocated for each class of service across each network links. Unfortunately, the proportion of bandwidth to dimension is a complex decision due to the interaction of a variety of factors, such as the traffic characteristics and the level of QoS required. One approach to simplify network dimensioning is to leave the amount of allocated bandwidth for each class static regardless of the traffic dynamics, while controlling the amount of incoming traffic through admission control. A more flexible approach is to dynamically adapt the amount of bandwidth allocated based on the dynamics of the incoming traffic. The effectiveness of a dimensioning scheme, just like any other QoS mechanism, is determined by its ability to ensure QoS levels contracted in SLAs are not breached while maximizing the utilization of available resources.

### Traffic Provisioning

As mentioned earlier, a serviced user is entitled to services for which he has made an agreement with a service provider through an SLA. In order to ensure that the user's traffic receives, and does not exceed, this service, the received traffic has to be conditioned at different parts of the network. The following functionalities comprise possible traffic provisioning mechanisms.

- **Classification:** When a packet arrives at a network router, a classifier is used to classify the packet according to the bilateral service level agreement. Based on its classification, the classifier forwards the packet to the appropriate traffic conditioner for metering, marking or shaping.
- **Metering:** Meters measure the temporal properties of the stream of packets selected by a classifier against a pre-specified traffic profile. A meter passes state information to other provisioning functions to trigger a particular action for each packet.
- **Marking:** Markers mark a packet with a particular code point, adding the marked packet to a particular class of service.
- **Shaping:** Shapers delay some or all of the packets in a traffic stream in order to shape the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is no sufficient buffer space to hold the delayed packets.
- **Dropping:** Droppers discard some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile.
- **Scheduling:** In order to provide differentiated service levels to different classes of traffic, different scheduling mechanisms are used. Schedulers are means of resource partitioning, such as the division of link bandwidth.

- **Buffering:** Due to scheduling, traffic in queues that were not serviced could be lost as the queues are filled and overflowed; hence, buffers are used to control the delays and loss rate experienced by traffic belonging to different classes.

In most networking environments, the configuration of different provisioning mechanisms is usually static. On the other hand, current increased changes in traffic requirements have forced this fixed resource allocation systems to require updating or to waste the resources of link bandwidth or buffer capacity. One of the goals of this dissertation is to develop predictive methods for the automated configurations of these mechanisms.

### 2.3.3 QoS Monitoring

Network monitoring refers to collecting raw data about network status such as bandwidth, latency and loss rate. Monitoring functionalities can be achieved either through network level or application level protocols. An Example of an application level monitoring protocol is the Real-time Transfer control Protocol (RTCP). Network level monitoring is achieved through simple probing services, e.g., standard ping or by piggy-backing network status information on packets traversing the network.

### 2.3.4 Fault Diagnosis

Once raw data about the network's performance is collected via monitoring functionalities, it is important to analyze it, in a timely manner, in order to actively isolate and identify any anomalous behavior of any of the network components. This process is referred to as network fault, or anomaly, diagnosis. In general, network anomalies can be attributed to various causes; in addition to hard failures due to hardware problems (e.g., equipment failure and power outage), soft failures can result from inappropriate use of network resources, protocol failures, temporary congestion, mischievous users and denial of service attacks. While anomalies in communica-

tion networks are usually unavoidable, early detection and classification of such faults is crucial to providing networking services with a high level of availability and reliability.

## **2.4 Standardized Models for QoS Management**

This section provides an overview of most established QoS management models defined by three international organizations; the International Standards Organization (ISO), the International Telecommunications Union (ITU) and the Internet Engineering Task Force (IETF).

### **2.4.1 Open Systems Interconnection (OSI) QoS Model**

The OSI QoS Framework [15] is one of the earliest efforts towards a formal model for QoS management. The model is built on the concepts of the OSI basic reference model and hence it adopts the same layered approach. For each layer, a set of qualities related to provisioning a certain service to the next layer are defined and used to exchange the QoS information between layers. The requirements of the QoS management functionalities are performed through two types of management entities; layer specific and system-wide entities. Control entities at each layer determine the policies which apply at that layer. This can include security, time-critical communication, resource control issues, selection and management of layer-specific protocols based on the required QoS. System wide entities interact with each layer-specific control entities to provide an overall selection of QoS functions and facilities.

Although the model provides useful QoS concepts and management functionalities, the framework is intended for existing or planned OSI standards and may be applicable to other fields. Furthermore, the OSI model was intended originally for non-time critical applications. The model relies on the implementation of QoS as a set of predefined set of parameters that are mainly concerned with the transport and session layers. Applications are responsible for

translating their requirements into a predefined set of statically defined parameters. Finally, a major limitation of the ISO/OSI QoS framework is that it does not include any specifications for different QoS functionalities such as negotiation, mapping, resource allocation, or QoS monitoring.

### **2.4.2 ITU-T Model**

The ITU-T model [26] can be described in terms of a set of generic building blocks, namely, a control plane, a data plane and a management plane. These blocks are responsible for controlling and delivering the network service response to a service request. The main control plane is concerned with routing, admission control and resource reservations. The data plane contains data traffic functionalities such as buffer management, traffic conditioning and scheduling. Finally, the management plane includes the mechanisms necessary for the administration aspects of the users' traffic as well as the management of service level agreements (SLAs). Different ITU-T groups have also provided recommendations for standard performance parameters, QoS classes, and network-interface-to-interface objectives. A major limitation of the current specifications of the ITU-T mechanisms for different planes is their static nature, that may in turn hinder these mechanisms from keeping pace with the evolution of the next-generation networks supporting dynamic QoS requirements. Nevertheless, the ITU-T model is, fundamentally, intended to identify a set of generic QoS network mechanisms at each plane and provide a structure for them. The ITU-T model can be regarded as a generic model that can fit and tie together different existing and newly developed QoS mechanisms.

### **2.4.3 Integrated Service Model**

The Internet Engineering Task Force (IETF) proposed the Integrated Service (IntServ) model [27] for QoS management based on resource reservation. For real-time applications, before

data are transmitted, the applications must first setup paths and reserve resources. The basic idea of the IntServ model is that the flow-specific states are kept in every IntServ-enabled router. A flow-specific state should include the information about bandwidth requirement, delay bound, and cost etc., of the flow. IntServ proposes two service classes in addition to best effort service. One is Guaranteed Service [28] and the other is Controlled Load Service [29]. The Guaranteed Service is provided for applications requiring fixed delay bound while the Controlled Load Service is for applications requiring reliable and enhanced best effort service. Because every router keeps the flow state information, the quantitative QoS provided by IntServ is for every individual flow. In an IntServ-enabled router, IntServ is implemented with four main components: the signaling protocol, the admission control routine, the classifier, and the packet scheduler. The Resource Reservation Protocol (RSVP) [30] is used as the signaling protocol to reserve resources in IntServ. Applications with Guaranteed Service or Controlled-Load Service requirements use RSVP to reserve resources before transmission. Admission control is used to decide whether to accept the resource requirement. It is invoked at each router to make a local accept/reject decision at the time that a host requests a real-time service along some paths through the Internet. Admission control notifies the application through RSVP if the QoS requirement can be granted or not. The application can transmit its data packets only after the QoS requirements are accepted. Recently, RSVP has been extended in several ways to reserve resources for aggregation of flows [31].

In general, the IntServ model represents a fundamental shift from the existing Internet architectures which were founded on the concept that all flow-related state information are kept only in end systems. Nevertheless, IntServ model suffers from many drawbacks. It is usually hard, if not impossible, to give precise estimates of the amount of required resources for the lifetime of a specific complex application. Hence, it is difficult to issue resource allocation estimates in a priori manner for reservation purposes. Another problem with this model is the amount of state information that must be kept at each IntServ enabled router along the reserved

path. This amount increases proportionally with the number of flows. Secondly, each router encounters an extra processing overhead of understanding the RSVP signals, admitting the requested traffic, and performing packet scheduling for each individual flow.

#### 2.4.4 Differentiated Service Model

While the Integrated Service model provides per-flow guarantees, the Differentiated Services (DiffServ) model [32] follows the philosophy of mapping multiple flows into few service levels, also referred to as Classes of Service (CoS). The DiffServ architecture is a simple approach to enhance QoS for data and multimedia applications. In contrast to IntServ, DiffServ complexity is pushed to the boundary routers of a network domain to keep core routers simple. The edge routers at the boundary of an administrative domain classify, shape, mark, and drop traffic if necessary. A packet arrives at the classifier and is classified according to SLAs. The classifier forwards the packet to the traffic conditioner. The traffic conditioner may include a meter, a marker, a shaper, and a dropper. At the core routers packets are inspected and those with the same marked code are treated with the same QoS level through a set of classifiers, buffers and schedulers. The IETF working group proposed two service classes in addition to Best Effort Service, the Expedited Forwarding and the Assured Forwarding classes.

The Expedited Forwarding (EF) class [33] can be viewed as a Virtual Leased Line service. Therefore, the bandwidth can not be exceeded but the user can leave it idle or use it to the full extent of its capacity. The holder of this pipe should not be affected by the presence or absence of other users. In order to actually provide this high service level, the amount of traffic injected into the EF class needs to be carefully policed. The Expedited Forwarding per hop behavior (PHB) has the following properties: peak bit rate, and low queuing delay.

The Assured Forwarding (AF) service class [34] does not provide a bandwidth guarantee but packets are given a higher priority. These packets have a higher probability to be transmit-

ted over the network than packets from the best-effort traffic. In congestion situations the user of the Assured Forwarding service should encounter less bandwidth decrease than Best-Effort users. Four Assured Forwarding Service classes are defined. Each of these classes has three levels of dropping precedence: low, medium, and high. Packets of a micro flow are mapped based on service on the classification to a single class. The different classes will be handled independently from each other. This means that within a domain, reordering of packets belonging to the same flow will not happen. DiffServ has an obvious scalability feature over the IntServ model.

#### **2.4.5 Multi-protocol Label Switching**

The Multi-Protocol Label Switching (MPLS) is another QoS management related technique [35]. Like ATM and frame relay, MPLS is defined in the second and third layers of the OSI/ISO model. MPLS was originally intended to simplify packet forwarding in routers rather than to address service quality. Currently, its main role is traffic engineering and virtual private network support. Some features of MPLS can facilitate QoS assurance. It can extend IntServ and DiffServ capabilities to a wider range of platforms beyond the IP environment. It has to be noted that, IntServ and DiffServ network models are different from the MPLS model in that they are not dependent on second OSI/ISO layer techniques and define general QoS architecture for IP networks, which can integrate different transmission techniques in one IP network.

### **2.5 QoS Management Approaches in Wired Networks**

In the context of communication systems, solutions to the problem of QoS management have been introduced at two complementary levels, the application and network levels. Approaches

for QoS management at the network level can be further classified into those that mainly investigate QoS enabled routing mechanisms and those that are concerned with the management and configurations of available network components. In addition to these approaches, there have been other individual efforts that tackle the management problem through the development of different technologies such as active networks and policies. The following sections further investigate significant contributions in each of these approaches.

### **2.5.1 Application Level Approaches**

The basic motivation behind the developments of application level QoS management mechanisms is their compatibility with the original best effort network model. Hence, they can function without incurring any changes to the intermediate routers traversed by serviced packets. Since the main concern of the proposed work is a general framework for QoS management, only a sample of existing research work for application level adaptation is presented. However, the developments of these mechanisms can be regarded as complementary works that can be built on top of the proposed framework.

In general, the key idea behind existing approaches at this level is to allow applications to continuously reconfigure their internal settings (e.g., video size, audio codec, video code) to adapt to the changing resources capacity of the underlying network. These adaptations are commonly driven by measurements of low-level QoS parameters such as the bandwidth, jitter, delays, packet losses, etc. A genetic algorithm is used in [36] to decide when to trigger the adaptation process based on the current network conditions. The selection of the new set of settings for the real-time media is then carried-out according to a set of rules which model the user-perceived QoS. These rules are learnt through a rule induction algorithm. Using mobile agents technology, the UbiQoS system [37] supports QoS tailoring and adaptation of video-on-demand (VoD) flows in response to user preferences and terminal properties. Based on the

user's request for a certain video content, the UbiQoS agent identifies a suitable server and establishes a server-to-client network path for the VoD flow. A set of dynamically installed UbiQoS components negotiates the QoS on any segment along this path and decides which application-level down-scaling operations to perform at which nodes. UbiQoS components also perform application-level admission control and reserve local resources. The UbiQoS framework takes advantage of the agent technology to perform code migration along with the execution status to manage different critical scenarios such connection failure. A dynamic adjustment of bandwidth requirements of multimedia applications by a linear regulator was proposed in [38]. A run-time decision to increase, hold or decrease the utilized bandwidth based on information obtained by the RTCP protocol.

In general, techniques for adaptation performed at the application-level share some common limitations. The concept of application-level adaptation encompasses an underlying assumption that application designers are responsible for building sophisticated applications that can dynamically change amount of resources required in order to tolerate resource availability on every system it is run on. This significantly complicates the task of application designers and shifts their focus away from their main goals related to the application specific functionalities. Furthermore, adaptive applications attempt to lower their resource usage to a sustainable level by changing operational parameters. Most approaches assume no a priori knowledge of resource requirements or resource reservation mechanisms. As each application attempts to minimize the resource usage in these systems, without having a global view of the network, running a number of concurrent adaptive applications may result in unstable system behavior.

### **2.5.2 QoS Routing Mechanisms**

Adaptive routing schemes [39] are mainly based on the assumption that in order for each traffic connection in the network to be set up successfully, a certain level of resources needs to be al-

located. Failure to reserve the resources results in rejection of the connection request. Routing schemes operate by augmenting an underlying link-state protocol through the calculation of alternative paths or paths with particular resource-related properties and through control of the frequency of link-state updates. For example, the path chosen may be specially computed so as to be the shortest, the least congested, or the one with the least expected latency. In [40], Korkmaz et al. proposed a new heuristic algorithm for the selection of a feasible path that satisfies all the QoS requirements of multimedia applications while maintaining high utilization of network resources. A new nonlinear cost function was formalized whose minimization provides a continuous spectrum of solutions of the QoS routing problem. In [41], the authors developed a new algorithm based on Lagrange relaxation techniques to find the route with the least expected delay for delay sensitive traffic.

In general, due to the required link state information, routing mechanisms inherit the space, message, and computational complexities of link-state routing, especially when the networks are large and highly dynamic. Furthermore, the complexity of such algorithms increases proportionally with the increased number of QoS requirements constraints.

### **2.5.3 Network level Reservation Based Approaches**

Aiming to provide stringent QoS guarantees for real-time applications, networks and operating systems have incorporated mechanisms for resource reservation. Reservation mechanisms have to keep track of the use of the limited set of resources provided by the system, and receive requests from new users interested in using these resources. New requests are subject to admission tests based on the usage of the resources and the guaranteed levels that satisfy the users. Reservations are then accepted, if there are available resources, or rejected if not. As mentioned earlier, the RSVP protocol [30], is widely used to reserve the network bandwidth along the path between the sender and the receiver for a predetermined period of time.

Another example of a resource reservation scheme is the global resource management system GRMS [42]. In GRMS, end-to-end QoS negotiation and adaptation is performed through the utilization of resource agents. A two-phase atomic negotiation and adaptation protocol is used by the agents to allow for a dynamic resource reservation each time QoS specifications are changed. A similar work presented in [43] describes another QoS-based Resource Allocation Model (Q-RAM) based on utility functions. The main objective of the work is to allocate resources to the various applications such that the overall system utility is maximized under the constraint that each application can meet its minimum needs. In [44] a bandwidth broker (BB) architecture implies that resources allocation must be based on admission control decisions made at a central location for each administrative domain.

While these approaches significantly enhance the delivered service quality, they have an obvious scalability problem due to the maintained per flow states. Moreover, resource reservation of predefined bandwidth leads to poor resource utilization. Furthermore, despite the ability of resource reservation schemes to customize resource utilization according to applications requirements, it assumes a static view of the overall end-to-end resource management strategy that may be configured statically based on estimated traffic characteristics. It is usually hard, if not impossible, to give precise estimates of the amount of required resources for the lifetime of a specific, and usually complex, application.

#### **2.5.4 Network Level Adaptive QoS Management**

The aforementioned challenges and difficulties in reservation-based mechanisms motivated the study of network-level adaptation-based approaches. The key idea of adaptive approaches is to define different customizable network configurations to suit QoS requirements of different applications. In this section, some of the proposed schemes that adopt this direction are discussed.

The Lancaster QoS Architecture (QoS-A) developed by Campbell et. al. is closely related to the ISO QoS reference model. It is composed of a number of layers and plans. The lower four layers of the suggested framework are equivalent to the lower four layers of the OSI referent model. Management of QoS is carried out through three vertical plans; the protocol plan, which is divided into two parts, a user sub-plan and a control sub-plan. The QoS maintenance plan is comprised of one QoS manager per layer. Each manager is responsible for monitoring and fine-tuning various components at that layer. QoS negotiation and establishment is carried out by the flow management plane. The OMEGA architecture [45] developed at the University of Pennsylvania supports the definition of thresholds for transit delay and deals with bandwidth demands from clients. The OMEGA architecture consists of a QoS broker, a centralized end-point entity for handling QoS at the edge routers, and end-to-end communication protocols using resources negotiated by the broker. The QoS broker performs QoS translation, negotiation and admission control for network resources. Nevertheless, the main contribution of the OMEGA architecture is mainly concerned with the QoS management and configuration during connection establishment phase. The architecture was developed under the assumption that applications are well behaved. Hence, it did not provide any adaptive means for resource reservation or adaptation at run-time. The HeiProject [46] at IBM's European Networking Center, Heidelberg, Germany, has developed an-end-to-end model that provides guarantees for the end-systems and at the network. It includes a continuous media transport system (HeTS/TP) that provides QoS mapping and media scaling. The Heidelberg Inter-networking layer supports both guaranteed and statistical levels of service. It provides both QoS-based routing and filtering. At the network layer, HeiRAT, a resource administration technique, is used to perform QoS negotiation, calculation and admission control. Once flows are admitted, QoS enforcement and scheduling techniques are used to guarantee the delivered QoS.

The QoS for CORBA Objects (QUO) [47] framework has been developed to support QoS in the CORBA layer. It extends the CORBA functionalities with a description Language (QDL).

QDL is used to specify application requirements in terms of an expected usage pattern of a certain Object rather than the communication itself. Each object specifies QoS regions that represent the status of the QoS for an object connection. The application can adapt to changing conditions by changing behavior based on these QoS regions. Each application uses a contract, similar to a finite state machine, to specify actions to be taken based on the state of the distributed system and the desired requirements. A recent work [48] further extends CORBA functionalities by applying reflective middleware techniques to implement QoS-enabled mechanisms by automating the selection and adaptation of key QoS properties. According to Wang et al., reflective middleware is a term that describes a loosely organized collection of technologies designed to manage and control hardware and software system resources. Using containers for each of the system components, each component's QoS is configured reflectively by its container. Meanwhile, a component configurator is used to decouple the implementation of services from the time they are configured.

Similar to QUO, The BBN Adaptive Quality of Service Architecture (AQUA) [49] is a resource management framework to deliver QoS guarantees to multimedia applications. In AQUA, applications specify their resource requirements in advance. The operating system (OS) then allocates the requested resources using an estimate of available resources. When the application changes its requirements, it renegotiates with the OS for resource reallocation. When adequate resources are not available applications are expected to dynamically adapt. In AQUA, each resource is associated with a controller and a scheduler while resource consumers, i.e., applications, user are connected to a QoS manager that estimates their resource needs and negotiate on their behalf. Two libraries, namely, a resource negotiation and a usage estimation libraries, are used as interfaces to legacy applications.

The Real-Time Adaptive Resource Management platform (RTARM) [50] provides mechanisms for QoS negotiation and adaptation. Service managers are used to represent resource management components and manage both specific resources and computing nodes. They can

be organized in a tree with higher-level managers constructed on top of lower-level managers. At the top level, they coordinate end-to-end resource negotiation and adaptation. At the lowest level, service managers model individual resources, such as CPU and network resources, within a node. The lower-level managers provide an adapter mechanism, which supports the incorporation of current and future components implementing scheduling algorithms or protocols. A two-phase negotiation and adaptation protocol is used to perform an admission control test on the associated tree and propagates either a commit or an abort command along the tree.

The Quartz architecture [51] utilizes agents to perform application-to-network requirements mappings as well as lower level functionalities. At the network-level, Quartz employs the resource reservation protocols available in the target network. Resource adaptation is performed by a *balancing agent*, which tries to compensate for the loss of one of the network or system resources by requesting more resources from the other. If re-balancing at lower levels fails, Quartz requests the application to adapt its requirements in order to decrease the consumption of resources. In the work proposed in this dissertation, the adaptation process is more proactive and broader in the range of adaptation mechanisms it adopts.

The European research project, Adaptive resource control for QoS Using an IP-based Layered Architecture (AQUILA) [52], defines a DiffServ-based architecture for delivering QoS. AQUILA implements an overlaid distributed control layer, the Resource Control Layer (RCL), implementing a mechanism for dynamic control of intra-domain resources. Each defined network service in AQUILA is meant to support a class of applications with substantially similar requirements and characteristics (e.g., Real-time applications, Streaming real-time applications). RLC performs three functionalities; it offers an interface to the QoS infrastructure to legacy applications, performs policy control and admission control on incoming traffic. Finally, it is responsible for monitoring and controlling resources. Similar to most approaches, AQUILA's resource control layer assumes a pre-configured underlying DiffServ network. The DiffServ code points (DSCP) and the PHBs of this network are assumed to be predefined by

management.

Most of the QoS architectures proposed so far target only a specific configuration of operating system software and communication infrastructure. This tight dependency on a specific environment constrains their application in open systems, where heterogeneity is always present. Furthermore, adaptation operations are either limited to small adjustments in the service rates or relies on static configurations stored a-priori in the system.

### **2.5.5 Differentiated Services Extensions**

Generally, research efforts concerned with extending the functionalities of the Differentiated Service model fall under one of two categories. The first category includes efforts to enhance edge routers operations through adaptive techniques for admission control and offered service rates. In the second category, existing techniques enhance core routers functionalities through adaptive traffic provisioning mechanisms. In the schemes proposed in [53], an admission control function is provided over DiffServ networks by means of the Endpoint Admission Control (EAC). EAC builds upon the idea that admission control can be implemented purely in an end-to-end manner, involving only the source and destination hosts. At connection setup, each source-destination pair starts a probing phase to determine whether a connection can be admitted to the network. The source node sends probing packets that reproduce the traffic characteristics of the connection to be established. Upon reception of the first probing packet, the destination host starts monitoring probing packets statistics (e.g., loss ratio, inter-arrival times) for a given period of time. At the end of the measurement period, the destination makes the decision as to whether to admit, or reject, the connection and notifies the decision to the source. In [54], Knightly et al. apply mathematical modelling techniques to provide statistical QoS guarantees through adaptive admission control. A measurement based approach has been introduced in [55] to calculate the admissible service rate. A survey of admission control tech-

niques can be found in [54]. The work presented in [56] addresses the problem of provisioning DiffServ traffic aggregates at core routers. Parameters of class-based weighted fair (WFQ) schedulers and queues are adjusted at run-time based on the obtained network measurements. The joint buffer management and scheduling ( JOBS) algorithm [56] adjusts service rate allocation and buffer parameters at run-time in order to satisfy a set of constraints related to service delay and loss. Nevertheless, these approaches provide significant contributions to enhance the performance of DiffServ-enabled networks, and in turn provide promising solutions to the QoS management problem. These individual works lack a global view of the overall management functionalities and may fail to cover different requirements of the management problem simultaneously.

### 2.5.6 Policy-Based Network Management Approaches

Policies have been introduced as an efficient tool for managing QoS at the network level. It has been widely supported by standard organizations such as the IETF and DMTF to address the needs of QoS traffic management. The Policy Working Group [57] is chartered to define a scalable and secure framework for policy definition and administration. The main goal is to support QoS management. This group has defined a framework for policy-based management that defines a set of components to enable policy rules definition, saving and enforcing. A policy [58] is a *“set of policy rules that instructs network nodes on how to operate”*, i.e., a policy represents an instruction for network nodes on how to manage requests for network resources. It is essentially a mechanism for encoding business objectives concerning the proper use of scarce resources. For example, policies define what resources a given resource consumer (e.g., predefined group of users, a single user, users in a certain domain, etc. ) can use in the context of a given scenario, e.g., (time of the day, location, type of application, etc.).

In the IETF model, the policy management system (or policy server) is referred to as Policy

Decision Point (PDP) [58]. It weighs the policy request sent by a Policy Enforcement Point (PEP), as a result of policy event against a corresponding set of policy rules. As a response to a policy request, the PDP either evaluates the policy rules for that request or retrieves the set of policy rules relevant for the request. The policy decision or the set of policy rules is then transported back to a PEP using a policy transaction protocol such as Common Open Policy Service (COPS) [59]. The PEP is a network device where the policy is enforced. Enforcement of policy decisions is carried out by the specific hardware/software features residing in the device such as packet filtering, marking, shaping, policing, bandwidth reservation, traffic prioritization, multiple forwarding queues, etc.

In the proposed work, we envision policies as a very powerful tool that can be used in automating the management communication system. This belief stems from the following premises. Firstly, policies can be used to describe agreements between customers and service providers as well as among service providers at different domains. Secondly, while allowing high level policies representing business objectives to be determined by administrators, policies translation from enterprise goals and service level agreements between different network domains into network level parameters can be easily automated. Furthermore, policies are persistent; once applied it remains active during its lifetime. Moreover, changing system behavior without modifying underlying Software/Hardware can be easily accomplished by changing the previously applied policies or by enforcing a new set of policies. Finally, by adopting the notion of *policy domains* for grouping of objects, scalable and efficient management of a large pool of resources can be achieved. Following existing approaches, the proposed work utilizes policies to configure the behavior of the system's different components. However, The proposed scheme furthers the use of policies by automating the creation, assembly and selection of the applied policies at a given instance of time; thereby, automating the adaptation in the behavior of the system without human involvement.

### 2.5.7 QoS Management Using Active Networks Technology

Active networks are frameworks where network elements, primarily routers and switches, are programmable. In active networks, programs are injected into the network and executed by the network elements to achieve higher flexibility and to present new capabilities. Active networks can be used in two different manners. Administrators can inject program codes directly into routers, otherwise, they can also program data packets, which then transport code to nodes along the way to their destination.

Recently, active networks technology has been geared to aid network management functionalities. Active networks have been used in [60] to build a prototype system where legacy routers are enhanced with an adjunct active engine, which enables the safe execution and rapid deployment of new distributed management functionalities in the network. Network nodes can execute and exchange a collection of programs that are injected to the network by authorized users. These programs are used for different management tasks such as router configurations, element detection, network mapping, network security management, etc. Smart packets have been introduced in [61] for network management and monitoring. Smart packets are programs that are executed at nodes on the path to one or more target hosts. Firstly, smart packets are used to tailor the amount of information delivered from nodes to the management centers. Consequently, management rules are embodied in programs sent to the managed nodes in order to identify and correct network problems at run-time. Another management architecture, the Active Distributed Management (ADM) architecture, has been proposed in [62] which exploits active network technologies and mobile agent paradigms to perform network management functionalities. A management task is executed as a program on a management station. The execution of this program involves interactions between manager-agents that control the information of the underlying network nodes. In [63] an active QoS Broker Interface provides services with the configuration and the management capability for the routers network inter-

faces in order to support QoS functionality. Examples of these functionalities are: establishing a connection with the broker, configuration of a DiffServ class at edge and core routers, adding makers and policers, adding monitoring components, etc.

Although active networks based management seems to provide some promising solutions, introducing more programmability into routers also implies adding more complexity to their management functionalities. In addition, excessive utilization of active packets results in network performance deterioration due to their high utilization of network resources. One solution to this problem is to restrict the functionality of the programs carried by the active packets, resulting in architectures with decreased capabilities. Furthermore, the dispatched active packets or programmable codes introduce new safety and security concerns.

### **2.5.8 Individual Efforts at Different Protocol Layers**

Besides the intensive research efforts that tackle the management problem at both the network and application layers, several individual research approaches provide different solutions at the operating system (OS) level. The Nemesis [64] framework, developed at the University of Cambridge, is an example of a QoS enabled OS which provides fine-grained guaranteed levels of all system resources including CPU, memory, network bandwidth and disk bandwidth based on different applications requirements. Another framework for Adaptive Resource Allocation (FARA) at the OS level has been introduced by Georgia Institute of Technology [65] for adaptive management of the operating system components, such as the memory, disk and CPU cycles. FARA is a hierarchical adaptation model which applies a discrete set of runtime configurations for distributed applications. Feedback adaptation is performed through resource reallocation. FARA allows users to specify their requirements at the start up time and adjust these requirements at run-time.

## 2.6 QoS Management Approaches for Wireless Networks

Recent progress in computing technology and wireless communications has made portable computers such as laptops, palms, and personal digital assistants easily available. This, in turn, has led to a growing number of mobile users that demand the same real-time services available to wired network users. However, the emergence of wireless environments has been accompanied with new challenges that were not considered in QoS architectures defined for wired counterparts. In wireless communications there is at least one hop over a wireless link with less predictable properties and less bandwidth than the wired links. Hence, traditional networking assumptions such as fixed topology and fixed amount of available resources do not hold in such wireless mobile environment. The Mobile IPv6 (MIPv6) [66] protocol provides the necessary extensions to enable IPv6 with true and transparent mobility support for the applications. However, upon every movement of the mobile terminal to a new station, a registration operation takes place, where routing entries need to be registered with a new Care-of-Address (CoA) and new paths have to be established. This, in turn, adds to the latency encountered by the mobile user.

In the view of the proposed work, handoff operations regarding latency and signaling overhead can be optimized, essentially by being able to predict the users' movement and prepare for handoff ahead of time. Another important issue that contributes to the enhancement of the QoS in wireless issue is the ability to renegotiate and reestablish the QoS configurations in the changed paths in an efficient manner. This can be achieved by allowing applications and users to predefine or dynamically specify their QoS in a simplified manner and allow the QoS management framework to autonomously perform reconfiguration based on these pre-specified objectives.

In the following, an overview of the challenges in wireless communication systems is given along with an analysis of how they contribute to further complicate the problem of QoS in such

environments. Consequently, a discussion of some of the existing wireless architectures which support QoS is given.

### 2.6.1 QoS Management Challenges in Wireless Networks

- **Connectivity:** In contrast to today's wired networks, wireless links are characterized with a high bit error rate (BER) and fading. In turn, these characteristics result in a higher packet loss in the wireless medium, which further contributes to a higher packet delay and jitter. In general, there is a tradeoff between BER and the bandwidth, and a reduction in packet loss can translate into higher packet transmission delay and jitter. Another important aspect in wireless networks is the mobile terminal handoff. As the mobile node roams and hands off from one access point to another, there is a change in wireless resources. This change results in a lot of fluctuation in the resource availability. Thus, resource availability is one of the main issues to be addressed while designing an end-to-end framework for QoS support in mobile and wireless networks.
- **Link errors:** Typical mobile data communications such as wireless links are prone to more errors than the wired links. Typical possible sources of errors in wireless networks are attenuation, signal interference and background noise. Different link layer mechanisms such as Automatic Repeat reQuest (ARQ) and forward error correction (FEC) have been proposed to reduce link error rate.
- **Terminal characteristics:** The heterogeneity of the users' terminals, provide additional challenges for the delivery of the required services. For example a mobile computing device could range from a web-enabled telephone to a notebook size portable computer, with considerable variations in processing power, memory availability and input-output capabilities.

- **Network resources scarcity:** While the capacity of wired networks continues to grow, the capacity of wireless networks is fairly limited. In wireless networks the expansion of available bandwidth is difficult; data compression is one way of reducing the application requirements in terms of Bandwidth in wired networks. However, data compression and decompression requires significant processing capabilities that are not available in wireless terminals.

Similar to the classifications of QoS management schemes for wired network, mechanisms for wireless networks can be further divided into application and network level approaches that will be summarized in the following sections.

### 2.6.2 Application-Level Approaches

CMU researchers have developed the Odyssey system [67] for application-aware adaptation in mobile environments. It is a proxy based approach to provide smooth audio/video streaming. The system includes client components to request lower fidelity of data. For example, when faced with a sharp decrease in bandwidth, a Web browser might ask for more highly-compressed images; a video player may reduce frame rate or frame quality of the stream; and a map viewer may filter out small or irrelevant features. The model is based on the assumption that the adaptive decisions are driven entirely by the clients' applications, and servers are able to support these adaptive decisions by providing data at various qualities, but not take an active role in quality selection or resource monitoring. In general, the Odyssey framework requires some modifications to be done with the hosts kernel, therefore it is not suitable for many hosts that are already connected to the Internet. CARISMA [68] is a mobile computing middleware which exploits the principle of reflection to enhance the construction of adaptive and context-aware mobile applications. CARISMA models a mobile distributed system as an economy, where applications compete to have a common service delivered according to their

preferred quality-of-service level. Bids are collected from different applications and from applications and a policy is selected such that it maximizes the utilization of resources. A generic framework for developing network-aware applications is proposed by Bolliger in [69]. The core of this framework is a feedback closed-loop that controls adjustment of an application to network properties. This feedback loop is designed as an adaptation layer sitting between the application layer and the lower layers in a common network model. The model requires the utilization of different transformation algorithms that are heavily application dependent, and therefore cannot be specified in a general framework.

### **2.6.3 Network-Level Reservation Based Approaches**

One way to provide mobile users with guaranteed levels of QoS is to directly apply existing resource reservation schemes for wireline networks to its counterpart of wireless environments. However, due to users' mobility, resource reservations need to be updated every time the user moves from one cell to another. Moreover, resources might not be available in the new location. Furthermore, it may not be compatible with the nature of the wireless environments which necessitates a frequent change in the path followed by the transmitted packets. In [70], Talukdar et al. described a resource reservation protocol, MRSVP, for mobile networks in IntServ enabled networks. MRSVP is an extension of the reservation protocol RSVP. The proposed solution assumes that for some users their next access point can be predicted, and hence they proposed three mobility related service models, namely; Mobility Independent Guaranteed (MIG), Mobility Independent Predictive (MIP) and Mobility Dependent Predictive (MDP) services. For MIG and MIP services, users are provided with guaranteed or predictive services, respectively, as long as they move in a regular manner that can be predicted by pre-allocating resources for all routes needed by a mobile host. On the other hand, MDP service provides users with a non weakly guaranteed level of service by using sources that have been allocated to other flows, but

are currently underutilized. The main feature of their proposed protocol is the concept of active and passive reservations which is used to provide mobility independent service guarantees. It is clear that their proposed approach still suffers from the scalability problem inherited from the IntServ model. Furthermore, there is no flexibility to adjust QoS levels to deal with changes in the underlying environment. The Localized RSVP protocol (LRSVP) [71] defines a mechanism to enable a mobile terminal to request QoS support from its own access network. Proxies are used to intercept the RSVP signaling packets and answer requests of the mobile terminal for local resources. In turn, the mobile terminal can send Path request messages directly instead of scheduled refresh messages. Hence, it contributes in reducing the latency encountered during handoff. In [72], similar to protocol layer transparency concept which keeps node mobility invisible to transport layer protocols, the authors defined a network layer flow transparency to reduce the incurred changes in the flow identity due to node change of address during handoff. To minimize handoff renegotiation delay, the routers which reside in the common portion of the new and old path are exempted from performing path update and only those routers that are in the newly added portion of the path are involved in the update process.

In general, although these schemes can reduce the latency encountered during handoff, which in turn reduces service degradation during handoff, these schemes cannot provide any guarantees for a new session establishment. In addition, the existence of the old reservation during the establishment of the new request result in decreasing the utilization of the network resources. To avoid the scalability problem, Lo et al. [73] use RSVP to establish a static subnetwork path between any two neighboring subnetworks. These subnetworks paths are configured with resource reservations in an aggregate manner. Buffers are used to store in-flight packets during handoff, while advance resource reservations along neighboring subnetwork paths is carried out to reduce the packet loss. The Hawaii project [74] combines resource reservation with application dynamic adaptation to provide the best possible quality of transmission considering the level of resource reserved and the current network load. Resource reservation is

carried-out repeatedly by the user rather than by the system. Although their model is simpler than other reservation based techniques, it does not provide any specific quality, but rather only the best quality that can be provided under the experienced conditions.

#### **2.6.4 Network-level Based Adaptive Management Approaches**

In order to support different applications requirements in the presence of scarce and variable resources in wireless environments, different network level adaptive architectures have been proposed in literature. In the Havana framework [75] Gomez et al. and Campbell propose an adaptive QoS framework for wireless networks. Their work is based on three different control mechanisms that operate over distinct adaptation time scales. At the packet transmission level, a packet based channel predictor determines whether to transmit a packet or not depending on the state of the wireless channel. At the packet scheduling time scale, a compensator credits and compensates flows that experience bad link quality. In the third mechanism, an adaptor regulates flows taking into account the ability of the wireless applications to adapt to changes in the available bandwidth. Ahn et al. proposed SWAN [76]; a network model that uses feedback based control mechanisms to support real-time services and service differentiation in mobile ad hoc networks. An explicit congestion notification (ECN) is used to dynamically regulate admitted real-time sessions in the face of network dynamics brought on by mobility or traffic overload conditions. Rate control is designed to restrict best effort traffic yielding the necessary bandwidth required to support real-time traffic. While dynamic admission control can be very effective in resolving the congestion problem in the resource-limited wireless networks, generalizing admission control to be the only correct solution to all encountered congestion situations may not always represent an effective policy. The Timely framework [77], developed at the University of Illinois, provides mechanisms for adaptation based on capturing the requirements of the mobile flows. A revenue model for resource usage along with a re-

source adaptation algorithm that seeks to maximize the network revenue while satisfying QoS requirements of the flows is used. The AQuaFWiN architecture [78] utilizes a generic feedback mechanism to support adaptability at all layers of the wireless network. Mobile hosts are grouped into clusters. Each cluster has a cluster head chosen by a distributed election protocol. Clusters are connected to base stations which provide connection to wired infrastructure and mobility support. A group of base stations are connected to a special node called the supervisor node (SN). The network layer uses feedback information to initiate a handoff. During handoff the SN can indicate to the gateway via the feedback packet that future packets need to be sent to both the current SN and the potential new SN. INSIGNIA [79] is a simple signalling mechanism which can be combined with a variety of routing protocols. The INSIGNIA protocol provides per-flow QoS by piggy-backing soft-state reservations onto data packets in an IP option field. If soft-state reservations are not renewed with a certain time interval, they are withdrawn.

While these approaches clearly address the needs of scalability and service quality, they do not address the important issue of mobility, which complicates the delivery of QoS in wireless networks. Without any flexibility to frequently adjust the delivered QoS according to the status of the underlying network, any guarantees for hard QoS are easily violated as the users move from one location to the other.

### **2.6.5 Extensions for Differentiated Services**

In [80], a new model for supporting DiffServ in wireless networks has been proposed. Each radio cell (or the related base stations) in a geographic area is associated with a registration domain. The registration domain is responsible for identifying new users as well as performing admission control. A bandwidth broker manages the resource allocation over the DiffServ registration domain. Once the mobile terminal is admitted to a registration domain, it can hand

off to other cells within the domain without the involvement of further call admission control in the bandwidth broker. In [81], the bandwidth broker configures the resources allocated to different service classes on different paths as well as the resources on the possible new paths for handoff.

The main limitation of the above schemes is that parameters such as the number of cells belonging to each domain, the resources allocated to each service class in each base station, and the resource commitment statically to users are dependent on many varying parameters such as the traffic loads and resource availability. These parameters have enormous effects in blocking and handoff dropping probabilities and hence the delivered QoS. We argue that, in a wireless network characterized with resource availability fluctuates static configurations do not present efficient suitable management solutions.

## **2.7 End-to-End QoS Schemes for Heterogeneous Networks**

The BRAIN QoS architecture [82] is based on the IETF Integrated Services and RSVP and Differentiated Services (DiffServ) architectures. The fundamental concept is to use IntServ parameters and RSVP signalling to communicate application requirements to the connecting network, and to provide the actual service differentiation with the DiffServ scheme. DISCIPLE [83] is another ongoing project at the Center for Advanced Information Processing, Rutgers University, to achieve adaptive collaboration for wired and wireless platforms using XML as the focus of a data-centric architecture to dynamically adapt data, shared between different platforms. An end-to-end negotiation protocol (E2ENP) have been proposed in [84]. E2ENP uses session initiation protocol (SIP) [85] to transfer control data and an extensible description model, based on extensible markup language (XML), to specify system characteristics and QoS parameters. The E2ENP model consists of a tree of QoS specifications. These specifications are defined at different levels of abstractions. The root of the tree is associated with a QoS

specification that imposes general constraints on the amount of resources used by all of the sessions along with their corresponding streams. Each leaf of the tree is associated with a QoS specification that represents the application QoS contract for the given stream. Negotiation is performed by resolving an adaptation-rule predicate associated with each subtree. Resolving this predicate instructs the system to enforce the QoS contract for that child.

## 2.8 Automated Management of QoS-Enabled Networks

As illustrated earlier, current adaptation approaches in the literature present simple adaptation algorithms which offer suboptimal solutions to the QoS management problem. Dynamic self-adaptation in response to changing QoS needs, resources availability, service cost, perceived performance of the network components or even neighboring networks, will become an essential operation in future networks. To our knowledge, automated adaptation have seldom been exploited at the network layer. This can be attributed to the lack of practical solutions for adaptive control in general. In the following, we investigate some of the few trials for automating one or more of the network management functionalities.

In [86], a self-adaptive routing mechanism has been proposed. Users are allowed to specify requirements related to the paths used in routing their packets, such as selecting the path with the highest QoS or the path with the least power consumption. The network then observes its state in a distributed manner. These observations are then fed to a random neural network (RNN)-based reinforcement learning mechanism which runs autonomously at each node to make routing decisions based on an estimate of QoS. These routing decisions are restricted to certain packets, which then inform the source about the paths they have found with the best QoS. These paths are then used by the payload carrying packets until a better path is found. An application level self-adaptive mechanism has been presented in [36]. Applications adapt in real-time their internal settings (i.e., video sizes, audio and video codecs) to the unpredictably

changing capacity of the network. A genetic algorithm is used to decide when to trigger the adaptation process while a rule induction algorithm is used to select the new QoS settings. The CADENUS (Creation and Deployment of End-User Services in Premium IP Networks) project [87] attempts to automate network service delivery. Mediation components are used to represent the main actors involved, namely, users, service providers, and network providers, and define their automated interactions. By defining roles, responsibilities, and interfaces, the service deployment process is decomposed into a set of sub-processes whose mutual interactions are standardized.

Autonomic Computing, launched by IBM in 2001 [88], is an emerging technology that aims at allowing users to transit transparently and dynamically between different providers and service domains. To gracefully perform this transition, automated selection of service configurations, reallocation and monitoring must be carried out with minor intervention of users and system administrators. The system must be able to make service selection and adaptation decisions autonomously and intuitively, without the users' awareness. However, these decisions must be frequently evaluated to ensure their feasibility in a dynamic environment with continuously changing parameters. Although, in theory, autonomic computing essence seems to provide the ultimate solution for the complex management problem, in general, research efforts towards autonomic management are still in their infancy and are still faced with many challenges before being realized as a successful solution.

## 2.9 Discussion

This chapter discussed, in details, various approaches proposed to address the issue of providing QoS support to multimedia applications in wired, wireless and hybrid systems. The development of QoS management schemes such as IntServ and DiffServ made it possible to provide some QoS guarantees to the users in a wireline network. Nevertheless, it is generally

difficult to promise a specified level of QoS to a mobile user since there may not be enough resources in the part of the network that the user is moving into. However, in order to provide real-time services, it is necessary to provide some level of QoS guarantees, at least statistically, to these mobile users. By investigating current research contributions in literature, the following key conclusions are reached.

- Current QoS mechanisms mostly cover only a single facet of the global QoS management problem (e.g., either limited to application or network components reconfigurations). An adequate QoS coordination mechanism should cover all components and layers (e.g., users and application requirements and mobility effects). Furthermore, it should incorporate mapping between different classes and types of QoS parameters, orchestrating, respectively, various system facets.
- Static network components configuration is inefficient to manage traffic at the aggregate levels.
- Current approaches lack the ability to learn from current and past management experiences resulting from the applications of different QoS management strategies.
- Due to heterogeneity in methods used to specify users' requirements, little research has been carried out to include the effects of these requirements and their dynamicity on management functionalities.
- Users have little or no control over their perceived quality of service. While current applications allow limited users interaction, (e.g., the choice between low, medium, high quality of a streaming media). Users must be involved in the selection of adaptation methods based on the available services, cost of service, their locations and their terminal characteristics. For instance, a user will sacrifice the quality of an interactive gaming

application for a reduced cost, while the quality of a distance learning application is appreciated over the cost of this service.

- Higher costs of maintenance of existing management models due to the reliance on human operators,

Earlier architectures for QoS management primarily focused on providing applications with APIs, while manually configuring adaptation strategies in response to changes in the applications requirements. The work in this dissertation is concerned with developing an autonomous management system that can perform QoS management without human operators' intervention. Furthermore, the proposed work is the first to utilize users' movement prediction and detected network anomalies proactively, in order to avoid experienced latency and packet loss due to users' movements and handoff operations. The framework also actively monitors the behavior of the underlying network in order to proactively adapt system configuration while maintaining a smooth delivery of services.

## **Chapter 3**

# **Autonomous Management of Hybrid Systems**

As mentioned earlier, the main thrust of this dissertation is focused on developing novel approaches that can be employed to attain an automated management of networking systems based on a prediction-based adaptation paradigm. To this end, the following chapters will develop schemes to predict/adapt the network status. These schemes are intended to be developed as components in a unified framework to achieve the objective of management automation. However, in order to fully describe the role of these components, it would be necessary to first provide a general overview of their functionalities and how they can, through their interactions, automate the management operations. For that purpose, this chapter lays out an architecture that ties these components together and coordinates their functionalities.

The chapter proceeds as follows. Section 3.1 describes the assumed underlying network model. A detailed description of the requirements of an autonomous QoS management system are presented in Section 3.2. An outline of the proposed approach is given in Section 3.3. Different functionalities of the framework components are discussed in Sections 3.4, 3.5, 3.6 and 3.7. The interactions between these components are discussed in Section 3.8. Finally,

salient features of the proposed framework are presented in Section 3.9.

## 3.1 The Underlying Network Model

### 3.1.1 Network Model

In this section, a sample network which will act as the basic infrastructure for subsequent discussions is presented. Figure 3.1 illustrates a simple reference model for the assumed underlying hybrid communication system. As shown, a typical communication system consists of one or more access networks connected together. An access network is an IP network to which users are directly connected. Examples of access networks are corporate networks, Internet service providers (ISPs) and educational organizations. Each access network represents a single administrative domain. Access networks can implement different technologies, such as Universal Mobile Telecommunications Systems (UMTS), wired Local area networks (LANs) or wireless LANs (WLANs). Each access network consists of clients, edge routers, core routers, and policy servers. All access networks provide differentiated services.

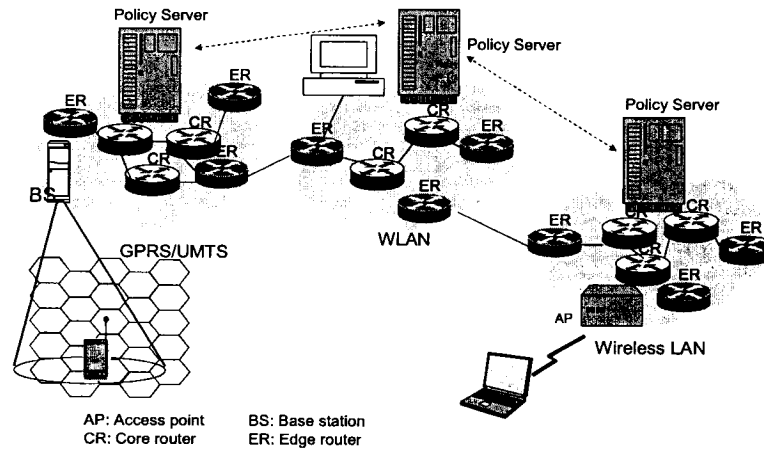


Figure 3.1: The network reference model.

A client's terminal can be a fixed terminal, a wireless IP phone or mobile/wireless device.

While clients have the option of having pre-configured QoS requirements in terms of Service level agreements (SLAs) [23], they can further specify and dynamically change their requirements as will be described later in details.

An edge router (ER), located at the boundaries of any domain, is responsible for performing multiple functions for QoS managements. It acts as a gateway for supporting clients requesting services. One of the main functionalities of edge routers is to enforce policies related to accepting/rejecting incoming requests. It is also responsible for performing traffic shaping and policing on all incoming traffic. In addition, edge routers aggregate all incoming flows into different QoS classes. Edge routers can be further classified into ingress router that are responsible for receiving all incoming traffic from different domains and egress routers which are the traffic end points that dispatch traffic to other domains.

A core router (CR) is any IP router connected only to edge routers and in the path of packets between clients. They are responsible for differentiating the treatments of different flows according to the classes assigned to these flows by edge routers.

A network management component, referred to as the policy adaptation agent (PAA), is responsible for controlling all QoS usage related functionalities. Details of these components will be described later in subsequent chapters. The focus of this work is the management components and their functionalities.

### **3.1.2 Layered Model**

The assumed layered model, shown in Figure 3.2, is comprised of end users, using end-system devices which run applications communicating with its peers. Applications utilize the device's operating system components, e.g., buffers, shared CPU and Input/output interfaces, to access the underlying network. The network cloud consists of hardware devices (e.g., switches and routers) and software components (e.g., buffers, schedulers, classifiers). The components of

the proposed architecture span different levels of the layered model. At both the user and application levels the architecture components are responsible for monitoring and predicting possible changes in the demanded QoS. Meanwhile, the network management components, residing at the network level, proactively performs adaptation functionalities to best meet those changing requirements.

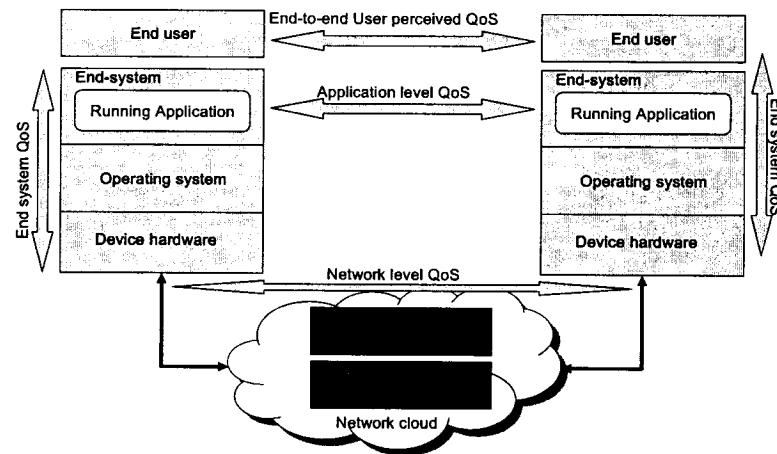


Figure 3.2: The layered reference model.

## 3.2 QoS Management System Requirements

This section summarizes the requirements of an autonomous QoS management architecture. These requirements refer to the desirable features that the architecture should possess.

- *Flexibility*: The management system must support a broad range of applications and provide services for different users with diverse QoS requirements.
- *self-learning*: The system must be able to learn from knowledge gained from past experiences. In other words, the system's QoS management strategies should be evolving gradually with the gained system experience.

- *Autonomous*: The management architecture must be able to dynamically reconfigure its components and adapt its behavior to maximize both system utilization and users satisfaction.
- *self-observing*: The management system must include a layer in its architecture to monitor its components behavior patterns and detect service deterioration, violations and anomalies.
- *self-diagnosis and repair*: The system must possess the ability to restore and correct any dysfunctional management task with minimal external intervention.
- The management system must predict and react to changes to the requested QoS during the lifetime of a user session.
- The management system must satisfy different business requirements as imposed by network administrators.
- The management system functionalities must abide with a set of network-level constraints that specifies availability of underlying resources, mechanisms and physical limitations.
- The system must provide users with a flexible tool to specify a their QoS requirements as functions of different parameters such as their locations, time of the day, type of running application and number of applications running simultaneously.
- The management system must shield applications from the complexity of the underlying QoS specification and QoS management.
- Novice users must be shielded from the burden of having to specify QoS requirements in terms of complicated QoS specifications.

- Experienced users can negotiate with the management system to specify their QoS requirements in either an abstract level or a more detailed network related parameters.

### 3.3 Proposed Management Architecture

The proposed framework, henceforth referred to as: the AUTONomous NETwork Management Architecture, or AUTONEMA for briefly, is based on a prediction-based adaptation mechanism. The rationale behind AUTONEMA is to employ all available knowledge pertaining to users, applications, network components and surrounding environment to predict the future status of the managed entities and requested services and to proactively act to maintain the system in an optimal state with respect to two major objectives. The first is to maximize resource utility relative to a set of pre-dictated higher-level business goals, whereas, the second objective is to maximize the satisfaction of serviced users according to their own services preferences.

AUTONEMA is modelled as a quadri-layer system in which the different components are referred to as agents. However, the presented work does not impose any restrictions on the technologies adopted for realizing the proposed system.

As depicted by Figure 3.3, agents are categorized by their mainstream functionalities, where agents performing similar functionalities are grouped in a single layer. Typically, those functionalities belong to one of four major categories, namely, information gathering, prediction, service adaptation and network monitoring. Agents in the first layer of AUTONEMA perform information gathering, and are further subdivided into agents that act on behalf of users, applications, domain and network administrators. Information gathered by agents in the first layer is then supplied to agents in the second layer which are responsible for performing the prediction functionality. Those agents cooperate to predict future changes in the network through utilize user's information, such as her profile, location and terminal characteristics, to predict possible future changes that would affect the characteristics of the requested and/or the

delivered service. In addition, they utilize network -related information, such as alarms and traffic loads, to predict conditions of network components. Predicted information is subsequently fed to agents in the third layer to adapt the network behavior accordingly. A feedback mechanism is achieved through information obtained by monitoring agents, existing on the fourth layer, to ensure that the applied adaptation strategies perform as intended.

In AUTONEMA, agents are aided with policies as a tool that facilitates their tasks. On one hand, policies are used to assist in the process of describing service preferences with respect to various conditions. For example, a user may specify a policy indicating her preference in a cheap service when she is away from her home network, and her willingness to pay more to get a better service when she is in her home network. However, when her device is battery operated she would prefer to reduce its utilization regardless of her location. Another example is a policy specified by an administrator dictating that on weekends, a higher preference should be given to media streaming traffic, whereas, during weekdays, file transfer traffic receives a higher priority. On the other hand, at the network-level, policies are used to aid adaptation agents in controlling behavior of network components through the dynamic assembly/disassembly of control policies. An example of this type of policies is a policy that shapes traffic belonging to a certain type of traffic when the network is congested.

Agents at the first and fourth layers are merely specialized workers that operate on data with no assumption about any embedded smartness. Conversely, intelligence is built into agents at the second and third layers to support the autonomous reasoning about supplied information and the decision making with regard to specific adaptation strategies, respectively. Figure 3.4 provides a schematic representation of the data flow among the different agents in the framework. The following sections discuss, in further details, the design and functionalities of the agents at different layers.

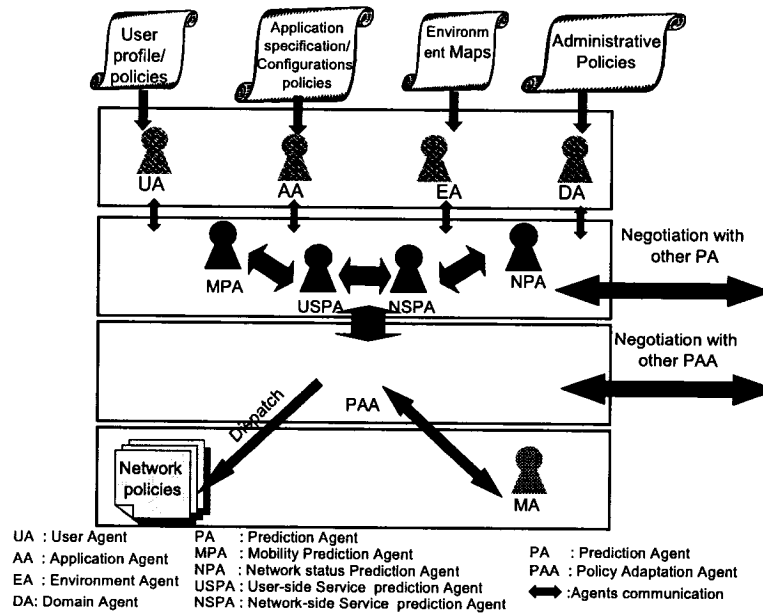


Figure 3.3: AUTONEMA layered model

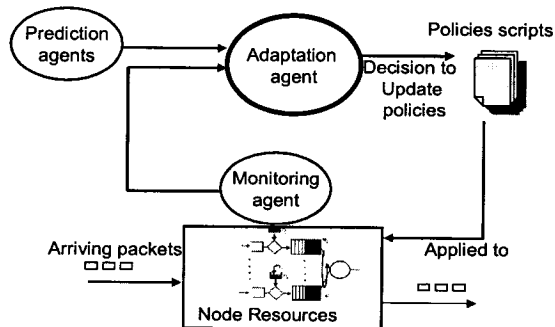


Figure 3.4: A schematic description of the adaptation process.

### 3.4 Information Gathering layer

Agents at this layer are concerned with capturing the necessary information that will be used by other agents to infer future changes in the underlying environment. The idea behind utilizing agents at this level is to build wrappers around available information sources in order to allow them to conform to one uniform convention of the developed architecture. Hence, to this end, we assume the availability of information providers that can deliver the data requested by prediction agents, concerning users, devices, applications and network components. It is also assumed that agents at this layer can integrate different sources of data to obtain the required knowledge and are capable of transforming collected data into a uniform format as described elsewhere [89].

Due to the dynamics of underlying environment, it is expected that users will continuously desire various service requirements depending on their current context. Hence, to minimize the user's involvement in such operations, agents at this layer maintain a list of preferences, as described by users, applications and administrators, in terms of policy structures. Table 3.1 presents a sample of these policies. The following subsections present, in further details, the various types of information gathering agents.

#### 3.4.1 User Agent (UA)

The involvement of the user's actions and preferences plays a major role in the process of future predictions. A user agent (UA) utilizes pre-specified *user-policies* along with a list of user's preferences and terminal characteristics to encompass a view of users' future service selections. In addition, it acts as an information source to lower-layer agents. One way to access user-related information is through the content capabilities/profile preferences (CC/PP) framework [90] or a user-profile provider (e.g., [91], [92]). Throughout the rest of this work it is assumed the availability of a reliable user-profile provider such that it can deliver the required

Table 3.1: Sample of policies as collected by different information gathering agents

Layer	Sample Policy
User policies	$P_1^u$ : <b>if</b> (Location $\neq$ home $\wedge$ Application = "VoD") <b>then</b> Service Quality = high $\wedge$ Availability $\geq$ 80% $\wedge$ Response Time $\leq$ 300 ms $\wedge$ priority(Quality) > priority(Availability) $P_2^u$ : <b>if</b> (Application= "Transaction") <b>then</b> Service Quality = low $\wedge$ Availability $\geq$ 95% $\wedge$ response time $\leq$ 150 ms $P_3^u$ : <b>if</b> (Destination= "work") <b>then</b> Security = high $\wedge$ Information Loss < 5%
Application policies	$P_1^a$ : <b>if</b> (Application= "interactive" and Video Quality= low) <b>then</b> Frame Rate > 15fps $\wedge$ Frame Size > "640x480" $P_2^a$ : <b>if</b> (Application = "VoD" $\wedge$ Available Bandwidth > 2Mb/s) <b>then</b> Frame Rate > 15fps $\wedge$ Color Depth = 24 bits
Administrative policies	$P_1^b$ : <b>if</b> (Network-status= "congested") <b>then</b> decrease assigned BW to Bronze class by 30 % $P_2^b$ : <b>if</b> (Destination= "BuildingA" $\wedge$ 9:00 < Time < 12:00 ) <b>then</b> priority > all other traffic

information through a query/response manner. The UA continuously analyzes maintained information and monitors events that would trigger one or more of the user's policies. Once a policy is triggered, the UA reports the necessary actions to be taken to a specialized prediction agent.

### 3.4.2 Application Agent (AA)

Similar to the UA, an application agent (AA) maintains information specific to running applications, typically represented in the form of a profile. A typical application profile [93] describes various characteristics of the expected requirements and respective performance of an application as well as policies, or adaptation strategies, an application is expected to follow. Examples of application characteristics are: bandwidth utilization rate, acceptable loss range in data rate and accepted data loss patterns (e.g., distributed and bursty). An example of an application-policy is to reduce the frame rate in case of reduced bandwidth if the user is satisfied with a medium image quality, otherwise, it should reduce the frame size. Applications profiles, supplied by the AA will be used by prediction agents to map expected users' demands in terms of specific resources needs.

While the creation of applications profiles is out of the scope of this work, without loss of generality, we also assume the existence of such profiles for most used applications that serviced users might run such as ftp, http, and video streaming.

### 3.4.3 Environment Agent (EA)

Environment agents (EA) gather and describe information regarding the landscape and environment of the user using modified *Spatial Conceptual Maps* (SCMs) [94]. Details concerning EA functionalities and SCMs will be discussed in Chapter 4.

### 3.4.4 Domain Agent (DA)

A domain agent (DA) represents a wrapper agent around network related information. It maintains a list of high-level administrative policies and objectives (e.g.,  $P_1^b$  and  $P_2^b$  in Table 3.1). and provides the means for their translation into network level objectives [19], [20]. When the conditions of one or more of these policies are satisfied it translates their corresponding

actions and communicates the resulting network level objectives to the network side prediction agent (NSPA) at the third layer of the architecture. It also maintains a list of network related information sources such as the domain's management information base (MIB) [95]. Hence, the DA provides an interface between the network information sources and the other agents in the management architecture. It can answer queries related to characteristics of network components such as the physical limitations that can be imposed by routers features (e.g., memory size, CPU speed and link bandwidth).

### 3.5 Prediction Layer

A key component to proactive management is the ability to predict more complex knowledge about involved entities (e.g., users and network components) and possible events (e.g., new/modified service request) that can trigger changes in the delivered and/or requested services. In AUTONEMA, service prediction is carried out through a two-step process. In the first step, concepts of evidential reasoning of Dempster-Shafers theory [96] are utilized to infer more complex knowledge about users and network conditions using information supplied through the top layer agents. The goal of the second step, is to ensure that the predicted information is simplified into a formalism that can be used by agents in the lower layer to perform the necessary automated service adaptation. This step is realized through the dynamic construction of several multi-attribute utility functions [97] that efficiently describe predicted service requests.

The prediction step is performed by two types of specialized prediction agents (PA), namely, mobility Prediction agents (MPA) and network-status prediction agents (NPA). The step of formalizing future service requirements is carried out by the user-side service prediction agents (USPA) and network-side service prediction agents (NSPA) to represent services at users and aggregate levels, respectively. A more detailed picture for the operation of these agents is given

next.

### **3.5.1 Mobility Prediction Agent (MPA)**

An important factor that affects the quality of the delivered services is the user's movement. To tackle this problem, the MPA is responsible for predicting future location of the user, assessing its effects on the QoS, and delivering such information to the USPA, where a change in the user's location, particularly from one domain to the other, may trigger the renegotiations about services qualities. The problem of user's location prediction has been the subject of an intensive research and results have been reported in [98], [99], [100]. Details of the functionalities of the MPA which involve the development of a novel scheme for mobility prediction are discussed in Chapter 4.

### **3.5.2 Network-status Prediction Agent (NPA)**

A network-status prediction agent (NPA) is a specialized agent that infers the behavior of the underlying managed network components through the analysis of different sources of data. More precisely, it generates predictive alarms with regard to the network conditions based on the measurements of network monitoring variables (e.g., buffer overflow and CPU utilization level). Examples of these predictive alarms are a congested network, a failed server, a broadcast storm and users misuse. These alarms are passed to the DA which in turn compares these alarms against the maintained administrative policies in order to decide whether a policy may be triggered in response to those alarms and communicates any resulting new network objectives back to the NSPA. The later may decide to change and renegotiate service offerings based on these objective, or directly prompt the policy-based adaptation agent (PAA) at the fourth layer to reconfigure the underlying network components.

The NPA adopts a novel approach for automated fault diagnosis through the utilization of

statistical analysis and evidential reasoning [96]. Details pertaining to NPA functionalities will be discussed in Chapter 5.

### 3.5.3 User-side Service Prediction Agent (USPA)

The user-side service prediction agent (USPA) is the entity that communicates the user's specific service requirements with network agents. It analyzes different service requirements, based on information collected through UAs, AAs, and MPAs. (Figure 3.5(a) depicts agents interactions with the USPA), and consolidates these requirements into a mathematical formalism. When a higher-level user or application policy condition is satisfied, upper layer agents notify the USPA to perform the necessary service reconfigurations to satisfy the triggered policy's actions. This functionality is realized in two basic steps. The first is concerned with the translation of higher-level goals into a corresponding multi-attribute utility function (MAUF) that formalizes those goals. The second step utilizes the constructed function to evaluate service offers by different service providers and to select the service that best matches user requirements.

#### Dynamic construction of MAUF

Figure 3.5(b) depicts an example of translating a user-policy into a MAUF. The key idea of dynamically constructing MAUF is rooted in the utilization of a decision theory known as the multi-attribute utility theory (MAUT) [97]. A USPA agent formulates the user/application preferences by a dynamically constructed MAUF, denoted by  $\mathcal{U}(X)$  and has the following form,

$$\mathcal{U}(X) = \sum_{i=1}^n \omega_i \sum_{j=1}^m \omega_{ij} u_{ij}(x_{ij}) \quad \sum_{i=1}^n \omega_i = 1, \quad \forall i \quad \sum_{j=1}^m \omega_{ij} = 1 \quad (3.1)$$

In (3.1),  $x_{ij}$  quantifies the preference assigned by the user/application for the  $j$ -th attribute, in feature  $i$  and  $\omega_i$  and  $\omega_{ij}$  are their relative weights. Examples of the utility features and the

corresponding attributes have been given in Section 2.2 and in [101].  $X$  represents a matrix whose  $(i, j)$  entry provides  $x_{ij}$ .

$u_{ij}(\cdot)$  is the normalized utility of  $x_{ij}$ . The utilization of  $\mathcal{U}(X)$  reduces the problem of satisfying user requirements to finding the value of a matrix  $X$  that maximizes  $\mathcal{U}(\cdot)$ . The utilization of the MAUT allows for a unified manner to evaluate different and sometimes conflicting preferences of users.

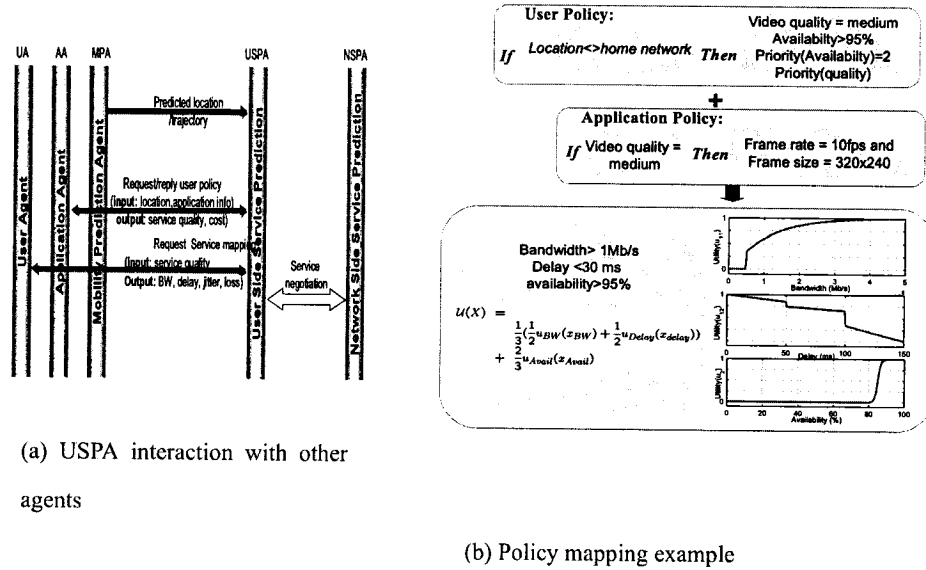
The construction of the user MAUF involves the calculation of the relative attribute weights as well as the selection of the utility function that best describes the relevance of a certain feature. Utility weights are selected based on both users and applications policies according to a normalized linearly increasing function calculated based on the degree of importance of each feature with respect to other ones. The utility function is selected from a set of predetermined normalized utilities where several parameters are dynamically adjusted based on the application nature and user's tolerance for variation in the corresponding service feature. Details of mapping methodologies can be found in [19], [20], [21].

### Service selection

Once constructed, the MAUF is used to aid the negotiation process with different network-side service prediction agents (NSPAs) to construct a dynamic service-level agreement. Through the negotiation process, the USPA obtains different offers described by a matrix  $X_{\text{offer}}^i$ . The utility of each offer is calculated using (3.1) and the service offer that maximizes the utility value is selected.

Once agreed upon,  $X_{\text{offer}}^i$  is used by the adaptation agent to perform network related functionalities to deliver the required service.

Figures 3.6(a) and 3.6(c) depict two different two-dimensional utility functions,  $U_Q^{P^u}(x_{\text{delay}}, x_{BW})$  and  $U_Q^{P^u}(x_{\text{delay}}, x_{BW})$ , with respect to the user's preference for service quality. Similarly, Figures 3.6(b) and 3.6(d) depict two different functions,  $U_R^{P^u}(x_{\text{Availability}}, x_{\text{Response}})$  and  $U_R^{P^u}(x_{\text{Availability}},$



(a) USPA interaction with other agents

(b) Policy mapping example

Figure 3.5: USPA MAUFs construction steps

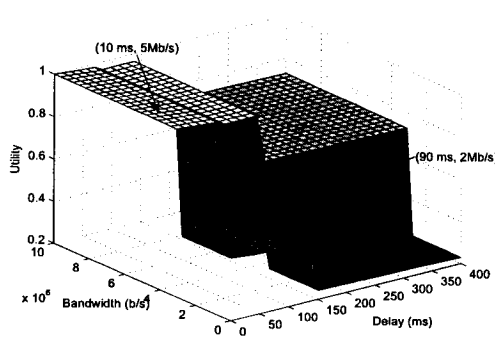
$x_{Response}$ ), describing user’s preference in service reliability. The first feature is expressed in terms of allocated bandwidth and experienced delay. Similarly, service reliability is represented in terms of service response-time and availability.

As shown,  $U_Q^{P_1^u}(\cdot)$  expresses a more strict preference for the service quality over  $U_Q^{P_2^u}(\cdot)$ . On the other hand,  $U_R^{P_2^u}(\cdot)$  expresses the user stringent requirements for service response-time and availability over,  $U_R^{P_1^u}(\cdot)$ .

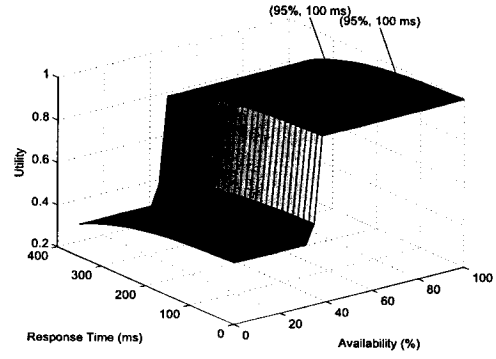
These four functions are used to construct  $\mathcal{U}_1$  and  $\mathcal{U}_2$ , Figure 3.6, describing a single user’s preferences with respect to two applications, video streaming and transaction processing, as expressed through the user’s policies  $P_1^u$  and  $P_2^u$  (table 3.1), respectively.

### 3.5.4 Network-side Service Prediction Agent (NSPA)

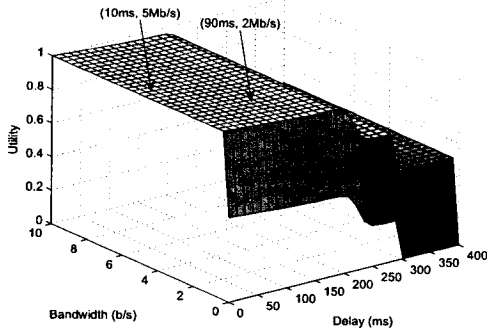
The network-side service prediction agent (NSPA) is responsible for maintaining a set of network-level objectives  $\mathcal{O} \subset \mathbb{O}$  that describe requirements for offered services. This set



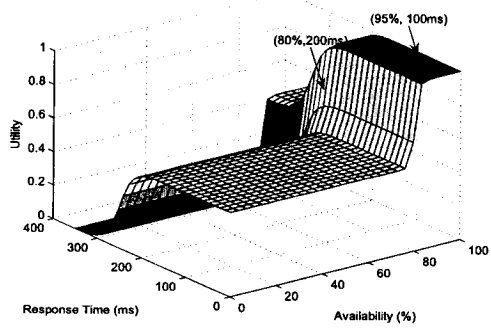
(a)  $U_Q^{P_1^u}(x_{delay}, x_{BW})$



(b)  $U_R^{P_1^u}(x_{Availability}, x_{Response})$



(c)  $U_Q^{P_2^u}(x_{delay}, x_{BW})$



(d)  $U_R^{P_2^u}(x_{Availability}, x_{Response})$

Figure 3.6:  $\mathcal{U}_1(\cdot) = 0.5U_Q^{P_1^u}(\cdot) + 0.5U_R^{P_1^u}(\cdot)$ ,  $\mathcal{U}_2(\cdot) = 0.3U_Q^{P_2^u}(\cdot) + 0.7U_R^{P_2^u}(\cdot)$

reflects goals of administrative policies as obtained from the DA. The objectives space,  $\mathbb{O}$ , is the Cartesian product  $\mathbb{O} : \mathbb{C} \times \mathbb{X}$ , where  $\mathbb{C}$  is the space of classifications that a traffic may belong to. Therefore, an element  $C_i \in \mathbb{C}$  can be a class of service for flow aggregates or an individual flow specification, e.g., IP source/destination addresses or MAC addresses.  $\mathbb{X}$  is the space of features that characterizes a class  $C_i$  such as the service quality and cost. Examples of two objectives are  $\mathcal{O}_1(\text{Gold}, 10c/\text{min}, 10\text{Mb/s}, 99\% \text{ availability}, 10\text{ms delay}, 3\text{ms jitter})$  and  $\mathcal{O}_2(\text{IP\_source} = 122.145.1.20, 10c/\text{min}, 2\text{Mb/s}, 100\text{ms delay})$ .

The NSPA also relies on the interactions between the DA and the NSPA to dynamically change the set of network objectives. For example, a network performance deterioration due to a predicted server failure or a congested network as determined by the NPA (as will be described in details in Chapter 5, may result in triggering an administrative policy stored by the DA (see  $P_1^b$  in Table 3.1) which dictates that the bandwidth assigned to the "Bronze" class of service should be reduced by 30%. In this case, the triggered policy actions will be communicated through the DA to the NSPA in order to change its current set of objectives.

Finally, NSPA utilizes the objectives set to negotiate with other NSPA in neighboring domains. The end result of the negotiation process is a set of contracted inter-domain Service agreements. Similar to the USPA, the NSPA uses service objectives to construct a network-side MAUF that represents the utility of each service features such as service quality, cost and reliability. This function is used to evaluate offers from neighboring domains during negotiation. Finally, as described before, the NSPA interacts with USPAs, acting on behalf of users, to construct dynamic service agreements.

### 3.6 Adaptation Layer

The goal of this layer is the continuous adaptive management of the behavior of the underlying network components to best meet the current users' demands and administrative objectives as

delivered by the NSPA. In this dissertation, policies are used as tools to guide the dynamic reconfiguration of network routers. The proposed scheme at this layer is mainly concerned with adapting network-level policies to meet changing objectives as they are obtained from the NSPAs. This problem is approached from a new perspective through posing this task as a problem of learning from current system behavior while creating new policies at run-time in response to changing requirements. Given sets of network objectives and constraints, policies are assembled at run-time. These tasks are carried out by the policy adaptation agent (PAA), which will be discussed in details in Chapter 6.

### 3.7 Monitoring Layer

Network monitoring refers to collecting raw data about network status such as bandwidth and latency. Based on the traffic generated, network monitoring methods can be classified as either active or passive methods [102].

In active monitoring, network measurements are done by sending additional testing messages. This approach inevitably introduces extra traffic to the network. Active monitoring can be established either through simple probing services, e.g., standard ping [103], running from a single host or through complex monitoring systems with probes distributed in the entire network. For example, to measure packet delays, probe packets are assigned a timestamp, and on receipt the round trip time (RTT) can be calculated. Another example of complex active probes is the agent system proposed in [102]. The IETF One-Way Active Measurement Protocol (OWAMP) is another example of a monitoring protocol supporting one way active delay measurements [104]. With the cost of extra traffic, active monitoring can have more control in the monitoring process. It can easily measure the characteristics of an entire network path between two hosts such as: packet RTT, average packet loss, and available bandwidth, but it is hard to get information about a single point in a network.

In contrast to active methods, passive monitoring techniques [105], rely only on the traffic that applications generate as they communicate with other nodes in the network. The network status information is piggybacked on packets traversing the network, so it will not cause extra traffic in the network. The most well known passive monitoring used is the Simple Network Management Protocol (SNMP).

Although, passive monitoring is more accurate than active techniques, since actual user packets are measured, fast processing is necessary to measuring all actual user packets securely. Otherwise, it can easily lead to outdated knowledge. In our proposed framework we rely on passive monitoring techniques, particularly through SNMP, to capture the status of the network.

### **3.8 Agents Interactions for Service Negotiation**

One key issue in AUTONEMA is the automated interactions and negotiation among agents at various layers. One typical problem occurs as agents with conflicting objectives operate independently to achieve their tasks. An example of such situation occurs during service negotiation between NSPAs and USPAs as well as among NSPAs. To resolve this issue, both agents must also have to embed the desire to cooperate to achieve their respective goals. This is achieved through a simple form of a predicated negotiation protocol. In the context of AUTONEMA, we adopt an argumentation-based approach [106] where agents interact through a negotiation protocol that includes exchanging proposals, critiques, and counter-proposals. The negotiation process is initiated by either USPAs or NSPAs due to either the initiation of a new user/administrator request or a change in preferences. Once the required preferences are captured, they are broadcasted to servicing NSPA agents to serve as an initial criterion that describes the service request for initial service negotiation. Upon receiving a service request, each NSPA in the contacted domains matches these preferences with one of its available service classes. The agent then replies with a proposal for a service class. In turn, the requesting

agent evaluates all received service offerings and utilizes its constructed MAUF to select the best matching offer. Optionally, negotiation can proceed by sending a preference update in the form of a service critique, in which the agent can request one or more incentives, such as a better level of security or better guarantees on service timeliness. The contacted agent, will then have the choice of offering a counter-proposal with its final offerings. Finally, the negotiation is terminated by the requesting agents accepting the best available services.

In the following chapter, details describing functionalities of each of the presented framework components will be presented. Performance analysis and results for each of these components will also be demonstrated.

### **3.9 Features of the Proposed Framework**

The main salient features of the proposed framework may be summarized, based on the previous discussions, as follows,

- Separation between three major correlated functionalities; network status prediction, mapping of users and application requirements into network level objectives and adaptive configurations of network components.
- Incremental developments of functionalities of different components at each layer.
- Consideration of the effects of users mobility on the delivered QoS in wireless networks.
- Consideration of the importance of earlier network anomaly diagnosis to maintain a smooth service delivery.
- Proactive response to possible deteriorations in the delivered QoS aiming to achieve seamless services.

- Utilization of knowledge of past experiences learnt from previous management decisions to adapt network components to best suit current environment status.

## **Chapter 4**

# **Mobility Prediction Scheme via Dempster-Shafer**

As illustrated in the previous chapter, user mobility prediction represents a key component in assisting handoff management, resource reservation and service pre-configuration and hence in achieving seamless QoS delivery. This chapter presents a novel scheme for user mobility prediction that can accurately predict the travelling trajectory and destination using knowledge of user's preferences, goals, and analyzed spatial information without imposing any assumptions about the availability of users' movements history. This chapter is organized as follows. Section 4.1 introduces concepts of mobility prediction schemes and highlights the motivations behind the proposed scheme. In Section 4.2 related work and existing approaches are briefly discussed. The key idea of utilizing contextual knowledge for mobility prediction is introduced in Section 4.3. Section 4.4 introduces the necessary background of Dempster-Shafer's theory of Evidence. The proposed scheme is described in Section 4.5. In Section 4.6 a simulation model and results are presented, along with an analysis of prediction performance. Finally, Section 4.7 concludes the chapter.

## 4.1 Introduction

With the current advances in the field of wireless technology, fast and accurate mobility prediction techniques have become one of the main topics in current research efforts. The importance of mobility prediction techniques can be seen at both the network and service levels.

At the network level, there are several management tasks that are deeply influenced by the user's mobility. These tasks include handoff management, flow control, resource allocation [99], congestion control, call admission control, and QoS provisioning [107], [108].

At the application level, the importance of mobility prediction techniques stems from the Mobile Location Services (MLS) [109], [110], [111] which provide the users with enhanced wireless service based on a combination of their profile and their current or predicted location. Examples of such services are pushed online advertising, map adaptation, user-solicited information such as local traffic information, weather forecasts, instant messaging for communication with people within the same or nearby localities, mapping/route guidance, and directing people to reach their destination.

While there have been several attempts to address the issue of mobility prediction [112], [113], [114], [115], [110], [116], [117], [118], [107], [119], [120], [99], [98], [121], [108] most of the existing techniques are based on the use of historical movement patterns pertinent to the users in order to calculate their possible future locations. These techniques are based on the assumption that the user's movements follow a specific pattern and exhibit some regularity. In this case, a training phase is first required during which regular movement patterns are detected and stored. User's movement behavior may be highly uncertain and assumptions about user's movement patterns should be made with utmost care. Therefore, an obvious disadvantage arises whenever the user is situated in new locations, or when there is a slight change in the user's mobility patterns. Examples of scenarios where past history of users' movements may not be available occur when trying to predict the location of a tourist while touring a city, or a

student navigating for the first time in university campus.

This chapter proposes a novel mobility prediction scheme that can accurately predict user's mobility trajectory, while overcoming the aforementioned drawbacks. The robustness of the proposed work is the result of a two-fold contribution. The first aspect of the contribution incorporates environment and user contextual information such as real-world maps and user profile and preferences. This is motivated by the fact that contextual information is now becoming a common denominator in mobile environment for adapting services according to the user's specific demands and surrounding environment. Therefore, the goal of this contribution is to gear this available information towards the objective of location prediction.

The second contribution is based on the utilization of the rich mathematical theory of evidence as a tool of reasoning to investigate the user's behavior concerning his decisions about his future location. The idea behind such combination of knowledge and reasoning is to improve the overall competence of the prediction algorithm to be able to handle new situations. A spatial conceptual map (SCM) along with user's available information, such as his profile, schedule, and preferences are used to extract important pieces of evidence concerning groups of possible future locations. These groups are then refined to predict the user's future location when evidence accumulates using Dempster rule of combination.

## 4.2 Related Work and Motivation

A number of schemes for user movement prediction have been reported in literature. In [112] Tabbane suggests that the mobile user's location may be determined based on her quasi-deterministic mobility behavior represented as a set of movement patterns stored in a user's profile. In [113] Liu et al. further pursued this method by modelling the users movement behavior as repetitions of some elementary movement patterns. A matching/recognition-based mobile motion prediction algorithm (MMP) is used to estimate the future location of the mo-

mobile user. The main drawback of these two algorithms is their high sensitivity to changes in the user's pattern of movement. As the degree of randomness in the user's movement increases the accuracy of these algorithms decreases linearly. Simulation results for the MMP algorithm, as reported in [114], show that the prediction accuracy of the MMP decreases linearly with the increase in the randomness factor. For example, prediction accuracy can reach 70% for users who have 30% randomness in their movements, however, it drops to almost 45% if the user's randomness factor increases to 50%.

To increase the accuracy of location prediction, recent research trends predict user's movement through modelling the user's moving behavior by storing all the possible movement paths and related mobility patterns that are derived from the long-term history of moving events of the mobile user. The time-varying location probability, estimated from the movement behavior, is used to predict the future location and arrival time of the user. In Doppelganger [122] information concerning the user movement is gathered through active badges, Unix logins, and schedule files. A limitation imposed by machine learning techniques is the large number of required example locations before the modelling system can be used. In [115] a dictionary of individual user's path updates is used as an input to a Markov model. As users move between cells, or stay in a cell for a long period of time, the model is updated and the network has to try fewer cells to successfully deliver a call. A closely related work has been carried out by Ashbrook et al. [110], where a GPS system is used to collect location information over time. The system then automatically clusters GPS data taken into meaningful locations at multiple scales. These locations are then incorporated into a similar Markov model to predict the user's future locations. In ComMotion [111], the location model is constructed from a set of learnt destinations that the user has categorized. Different pattern recognition models including Markov models and Bayes models are used for route learning and route prediction.

In general, for statistical prediction methods (e.g., Markov predictors) to succeed, history patterns must exist a-priori, i.e., sufficient data must be generated to provide reliable estimates

of transition probabilities. Furthermore, these statistical models are built upon a stability assumption that the user will follow the same behavior for a long period of time to present a reasonable approximation of reality; an assumption that might not be always true. For example, a college student might have a model that was built using the locations of his classes for an entire semester. When the next semester starts he may have an entirely different schedule; in this case it might take the entire semester for the model to be updated to correctly reflect the new information.

A fuzzy inference system was proposed in [116] to estimate the current mobility information based on real-time measurements. In [117], a prediction-based location management using multi-layer neural network (MNN) has been proposed. The MNNs are trained with respect to the data obtained from the movement pattern of the mobile user for a period of time, before the prediction scheme is used.

A simulation study was performed in [118] using data collected within ORL (Olivetti and Oracle Research Laboratory) to track the movement of its staff members for a period of 15 days using various prediction schemes. Roughly 60% of the data was used to establish a history database and 40% of the data was used to carry out movement estimations. Although this case study involved regular movement patterns followed by employees in their usual workplace, simulation results showed that most prediction schemes have a prediction accuracy ratio in the range 50 to 70%. Furthermore, higher levels of randomness in the users' movement can further degrade the prediction accuracy of these schemes.

In the authors' opinion there are two limitations for schemes relying on individual user's mobility patterns. The first is concerned with the change of user's behavior. When a user visits new places in which no past history is available, pattern history-based prediction schemes fail to work. The second limitation can be related to logging the mobility history for a long time. This causes recent changes of the user's behavior to have insignificant impact on the prediction scheme which must go through a learning phase again before it can function accurately. In

[107] Soh et al. propose a scheme to overcome these limitations by assuming that a user's next move tends to follow the movement pattern of other people in nearby places if they move in the same direction. Behavioral information can be used to enhance prediction schemes in order to accurately and adequately represent and deal with incompletely specified situations which are characterized by partial or even complete absence of knowledge about the users pervious movement patterns in current location. To emphasize the importance of developing a mobility prediction scheme that does not impose any assumptions about the existence of a user's movement history, the following three scenarios [109] are considered. The first scenario concerns a tourist visiting a foreign town. The second concerns a new computer science student at campus. The final scenario concerns a businessman on a business trip away from home. It is clear that in these three scenarios no movement history can be used to aid in the mobility prediction schemes and thus other alternative information must be used.

### **4.3 Contextual Knowledge and Mobility Prediction**

Recent trends of mobile computing and communication technologies have highlighted the central role of context-awareness in mobile applications and services. The objective of incorporating context is to acquire and utilize information pertinent to the context of the environment, the user, or the mobile device to provide services that are tailored to a particular user, place, time and events. As a consequence, contextual information has become an integral part of various mobile applications. With the promising advances in context sensing and modelling systems [123], [124], and the abundance of applications that are geared towards context awareness (e.g., Sensay [125], Aura [126], CybreMinder [127]) it is evident that context information will be an integral key component in mobile environments. However, in spite of the recent advances in the research work of context aware computing, the dominant part of this work assumes that location information is an already existing piece of contextual knowledge and aims

at utilizing it mainly in service adaptation for location aware applications.

The primary focus of the proposed scheme is to effectively employ the available information about the context of the user and the environment in the domain of mobility prediction. To our knowledge this work is the first to utilize available contextual knowledge to predict users' mobility patterns.

As will be illustrated through simulation results of the proposed work, combining the user and spatial context dramatically enhances the performance and accuracy of user's mobility prediction, particularly in situations that lack the availability of a reliable history for movement patterns.

The proposed scheme has been motivated by recent advances in the research area of modeling user context (e.g., [122], [91], [92], [128]) and previous work developed by the authors' research group [129] that proposed a unified framework for a context negotiator/provider to automate the acquisition of users context. Based on these developments, the availability of a reliable user context provider is assumed. It is also assumed that the proposed mobility scheme can interact with such providers through a query/response form. Further details concerning these models will be discussed in section 4.5.

## 4.4 Review of Dempster-Shafer Theory

Dempster-Shafer theory [130], [96] has attracted considerable attention as a successful approach in dealing with problems of combining different bodies of evidence to reach to decisions in situations characterized by a high degree of uncertainty (e.g., [131]). The main advantage of the underlying theory of evidence over other approaches is its ability to model the narrowing of a hypothesis with the accumulation of evidence, and to explicitly represent uncertainty in the form of ignorance or reservation of judgment. The theory is based on two ideas: the idea of obtaining degrees of belief for a related hypothesis, and the idea of applying Dempster's rule

for combining such degrees when they are based on different bodies of evidence. The net effect of Dempster's rule is that concordant bodies of evidence reinforce each other, while conflicting evidence erode each other.

#### 4.4.1 Mass belief assignment

Dempster-Shafer theory of evidential reasoning starts by assuming a Universe of Discourse  $\Theta$ , also called *Frame of Discernment*, which is a set of mutually exclusive and exhaustive propositions about a domain. Let  $2^\Theta$  denote the set of all subsets of  $\Theta$ . Elements of  $2^\Theta$  are the general propositions in the domain with which the theory is concerned.

A function  $m:2^\Theta \rightarrow [0, 1]$  is a basic probability assignment *bpa* if it satisfies that for the empty set,  $m$  is 0; and that the sum of  $m$  over all subsets of  $\Theta$  is 1. That is:

$$m(\phi) = 0, \quad \sum_{A_i \subseteq \Theta} m(A_i) = 1 \quad (4.1)$$

The basic probability assignment  $m$  is referred to as *mass distribution* to distinguish it from the probability distribution. Note that it applies directly to the evidence (subsets of the frame of discernment  $\Theta$ ), not to the elements of  $\Theta$  as in traditional probability theory.

#### 4.4.2 Evidence combination

Dempster-Shafer's theory provides a means for combining beliefs from distinct sources, known as Dempster-Shafer rule of combination. Suppose  $m_{E_i}$  and  $m_{E_j}$  are two *bpas* of the same  $\Theta$  from independent bodies of evidence,  $E_i$  and  $E_j$ . The combined *bpa* can be computed as:

$$m_{E_i} \oplus m_{E_j}(C) = \frac{\sum_{X \cap Y = C} m_{E_i}(X) m_{E_j}(Y)}{1 - K}, \quad (4.2)$$

for all non-empty  $C$

where  $K = \sum_{X \cap Y = \phi} m_{E_i}(X) m_{E_j}(Y)$ . The denominator  $1 - K$  is a normalization factor, making the sum of  $m_{E_i} \oplus m_{E_j}(C)$  range between 0 and 1.

It is worth mentioning here that the orthogonal sum does not exist when  $K$  equals 1. In this case, the sources are totally contradictory, and it is no longer possible to combine them using (4.2). The application of this rule is valid only when the sources are sufficiently in agreement. Recently, a number of methods and combination operations have been developed to circumvent this problem [132]. Nevertheless, this case does not occur under closed-world assumptions provided that an appropriate modelling approach for the sources of information has been adopted. Since evidence regarding users' intentions are generally in partial agreement, (4.2) can be used to combine the underlying evidence.

The mass function can be discounted by means of the so called discount operation. Let  $m_E$  be a mass function on  $2^\Theta$ . Given a real number  $\beta$  and a proper subset  $X$  of  $\Theta$ , the mass function  $m_E^\beta$ , defined by

$$\begin{aligned} m_E^\beta(X) &= (1 - \beta)m_E, \\ m_E^\beta(\Theta) &= (1 - \beta)m_E(\Theta) + \beta \end{aligned} \tag{4.3}$$

( $\forall X \subset \Theta, \beta \in [0, 1]$ ), is said to take into account the reliability of the information source providing the belief distributions. Higher  $\beta$  values indicate a lower degree of reliability.

The Dempster-Shafer's combination rule computes a measure of agreement between two bodies of evidence concerning different propositions discerned from a common frame of discernment. An important property of the rule is that it is commutative and associative. This is desirable because evidence aggregation to reach one conclusive decision given all collected evidence should be independent of the order of evidence gathering. The rule focuses only on those propositions that both bodies of evidence support.

Using *bpa* values, a lower bound of the interval that contains the precise probability can be derived. This lower bound is referred to as the *belief* value. A *belief function*  $Bel : 2^\Theta \rightarrow [0, 1]$

describes quantitatively all reasons to believe in a hypothesis  $A$  such that

$$Bel(A) = \sum_{B|B \subseteq A} m(B) \quad (4.4)$$

Once evidence is accumulated using the Dempster combination rule, deciding which hypothesis is the most reliable can be realized based on the calculated belief values. This scheme adopts a decision rule that selects the hypothesis that maximizes the belief value, i.e.,

$$D = \arg_{A \in \Theta} (\max(Bel(A))) \quad (4.5)$$

## 4.5 Proposed Mobility Prediction Scheme

Figure 4.1 presents a schematic description of the proposed Mobility Prediction Agent (MPA) located on the user's Mobile Terminal (MT). The MPA gathers the necessary information for the prediction process and analyzes this information using Dempster-Shafer's theory in order to predict the future location of the mobile user. The process of the location prediction is carried out in four main phases: information gathering, evidence extraction, decision making, and finally, path finding. As shown in Figure 4.1 the MPA is composed of two main modules: a *reasoning engine* which is responsible for the first three stages of the prediction process and a *path finder* module that performs the final stage of finding the user's expected trajectory. The following subsections present further details of the prediction process described above.

### 4.5.1 Information Gathering

The information gathering stage, performed by the agents at the first layer of the architecture, is concerned with capturing the necessary contextual information that will be used in the following stage to infer the user's future behavior. Contextual knowledge at this stage is comprised of two categories: environment context and user context. This knowledge will be collected and

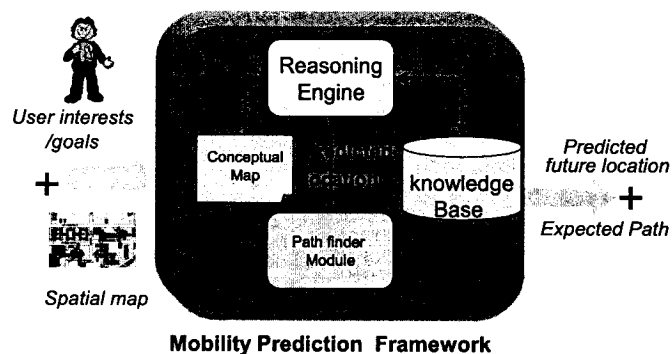


Figure 4.1: Mobility Prediction Agent

presented through agents at the information gathering layer, namely, the user agent(UA) and the environment agent as will be described in the following subsections.

### Environment Context

The environment context describes information regarding the landscape and environment of the user using a *Spatial Conceptual Map* (SCM) [94]. As defined in [94] an SCM is “an abstraction of a real map representing a portion of the urban environment”. An SCM contains representations of a set of *landmark objects*  $O_{SCM} = \{O_1, O_2, \dots, O_n\}$  and a set of *way areas*  $W_{SCM} = \{W_1, W_2, \dots, W_m\}$ . Way areas define areas on which users can move (e.g., streets, roads, highways, or simply trajectories and virtual connections between objects). Landmark objects are places of interests for users such as buildings and monuments. A way area  $W_x$  is further partitioned into a set of  $k$  consecutive segments called *Way Elementary Areas* (WEA), such that  $W_x = \{\alpha_1^x, \alpha_2^x, \dots, \alpha_k^x\}$ . According to [94], WEAs can be categorized as follows:

- $(\alpha_1)$ : corresponds to a WEA that is marked by at least one landmark object.
- $(\alpha_2)$ : corresponds a WEA that is a crossing of two or more Ways.
- $(\alpha_3)$ : corresponds to a WEA that is an intersection between a crossing of a Way and one

or several landmark objects.

- ( $\alpha_4$ ): corresponds to a WEA that is a straight unremarkable segment (there are no landmarks nor intersection of way in the neighborhood).

Figure 4.2 shows a portion of the university of Ottawa campus map (Figure 4.2(a)) and its SCM representation (Figure 4.2(b)).

A characterization function  $C_O$  is associated with each landmark object  $O_i$ , where  $C_O(O_i)$  represents basic information about  $O_i$ . For example, if  $O_i$  is a restaurant, then  $C_O(O_i) = \{c_1 = \text{Food}, c_2 = \text{Chinese}, c_3 = \text{moderate prices}\}$ . For a museum  $O_j$ ,  $C_O(O_j) = \{c_1 = \text{Historic}, c_2 = \text{Greek}, c_3 = \text{no cost}\}$ . In general,  $C_O(O_i)$  can be represented by  $n$  characteristics of a landmark object  $O_i$ , such that  $C_O(O_i) = \{c_1, c_2, \dots, c_n\}$ .

An SCM is represented by a sparse matrix known as: *the matrix of orientation, adjacency, and characterization* (MOAC). The columns and rows of an MOAC represent the WEAs of the SCM. Each cell  $(i, j)$  of the MOAC corresponds to the relation between the two WEAs  $\alpha_i$  and  $\alpha_j$ . When  $i \neq j$  MOAC( $i, j$ ) contains information about the orientation of possible displacement from WEA  $\alpha_i$  to WEA  $\alpha_j$ . This orientation can be expressed in degrees ( $0^\circ - 360^\circ$ ) or, for simplicity, relative to the four directions north, south, east and west. When  $i = j$ , MOAC( $i, i$ ) contains a reference to nearby landmarks and their characteristics. For example, if a landmark  $O_x$  is close to (CLT)  $\alpha_i$  then MOAC( $i, i$ ) = ( $O_x, C_O(O_x)$ ). Figure 4.2(c) represents the MOAC of Figure 4.2(b).

It is assumed here that the SCM information is maintained at each base station (BS) and can be obtained through MT requests of MOACs to the BS.

### User Context

The user context maintains different information concerning selected aspects of the user. In the proposed work, these aspects are used to collect evidence pertinent to the user's selec-



tion for his future destination. User modelling for context has been the focus of intensive research work [122], [91], [92], [128]. Existing approaches for user modelling rely on different methods for collecting user related data. Examples of these methods are: body-worn sensors [125], observed user actions and interactions with the surrounding environment [91], inference rules [123], or explicitly through a user feedback [122]. For example, knowledge about a student's schedule can be obtained through the campus server once she registers to certain courses. In the same manner, related tasks, such as handing in an assignment, can be realized. Furthermore, knowledge about daily tasks can be extracted from an electronic diary. User's interests can also be collected through a questionnaire list [122] or deduced from the user's activities. Throughout the rest of this work it is assumed the availability of a reliable user context provider (e.g., [122], [91], [92], [128]) such that it can deliver the required data about the user to the MPA through a query/response manner. However, to maintain lucidity in illustrating the basic notion of the proposed mobility prediction scheme, the existence of a simple student model comprised of three classes: user's interests, user's tasks and goals and user's schedule is considered.

It is also to be noted that the proposed scheme is not restricted to the characteristics of the user model listed above. However, it is rather the intention to limit the discussion to those three classes in order to keep the presentation of the proposed scheme as concise as possible. The selection of more extensive characteristics can be tailored to the nature of the users and their activities, as will be discussed later.

#### **4.5.2 Evidence Extraction**

The goal of this step is to apply concepts of Dempster-Shafer's theory to the information gathered in the previous step along with the information available concerning the mobile user to generate hypotheses and bodies of evidence that will form the input to the next phase. The

functionalities of this phase is carried out by the reasoning engine of the MPA.

The first step is to generate a frame of discernment  $\Theta$  from all potential future destinations. i.e.,

$$\Theta = \{O_i : O_i \in O_{SCM}\} \quad (4.6)$$

where  $O_{SCM}$  represents the target locations obtained from the SCM map.

Next, the reasoning engine uses the domain  $2^\Theta$ , the set of all subsets of  $\Theta$ , to generate different hypotheses and applies different rules to assign a belief mass value to each hypothesis. This process is performed based on different premises inferred from the user's context to represent different bodies of evidence that can foretell the user's future behavior. To be precise, the following types of premises are used by the reasoning engine to structure the frame of discernment into groups of candidate future locations supported by different bodies of evidences and represented by different corresponding hypotheses.

- **User's interests**

The MPA uses its knowledge base to store the user's interests in certain activities in the form of certain attributes. Pieces of evidence are generated from this set of attributes by enforcing a set of rules that try to match each attribute with the user's current schedule and availability of nearby suitable locations.

This process conduces to a set of pieces of evidence that undergo a threshold process to determine the plausibility of the evidence. For example, the evidence suggesting that a student, with a high interest in exercising, would be going to the gym is eliminated if his schedule does not allow enough spare time.

To formalize this process, denote by  $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$  the set of all user's interests, such as sports, shopping and going to movies. Each domain of interest  $I_i$  is paired with the following attributes,

- $t_{\text{earliest}}^{(i)}, t_{\text{latest}}^{(i)}$ , which indicate, respectively, the earliest and latest preferable time of the day to perform this activity.
- $t_{\text{duration}}^{(i)}$  specifies the average time required by the user to perform activity  $I_i$ .
- $V^{(i)}$  reflecting the degree of importance of activity  $I_i$  for the user.
- $C^{(i)} = \{c_1^{(i)}, c_2^{(i)}, \dots, c_n^{(i)}\}$  the set of characteristics related to activity  $I_i$ .

Now a set of evidences based on the user's interests  $\mathcal{I}$  may be defined as  $\xi^{\mathcal{I}} = \{\xi_1, \xi_2, \dots, \xi_k\}$  such that,

$$\xi^{\mathcal{I}} = \{\xi_i : \forall I_i \in \mathcal{I} \mid t_{\text{earliest}}^{(i)} < t_{\text{prediction}} < t_{\text{latest}}^{(i)} \vee t_{\text{duration}}^{(i)} < t_{\text{Available.time}}\} \quad (4.7)$$

where  $t_{\text{prediction}}$  is the current time of the prediction process, and  $t_{\text{Available.time}}$  is the user's amount of spare time before his next scheduled meeting.

An impact  $(1 - \beta_i)$ , which ranges from 0 (absolutely insignificant) to 1 (highest possible impact), is assigned to each generated evidence. The impact  $(1 - \beta_i)$  of evidence  $\xi_i$ , related to interest  $I_i$ , is calculated as

$$(1 - \beta_i) = V^{(i)} \quad (4.8)$$

For each evidence  $\xi_i$ , based on an interest  $I_i$  with a set of  $n$  characteristics, a group of hypotheses  $\mathbf{H}^{(i)} = \{H_1^{(i)}, H_2^{(i)}, \dots, H_n^{(i)}\}$  is constructed such that:

$$H_j^{(i)} = \{O_k : O_k \in \Theta \mid c_j^{(i)} \in C_O(O_k)\} \quad (4.9)$$

A belief mass value  $m$  is associated with  $H_j^{(i)}$  such that:

$$m(H_j^{(i)}) = \frac{1}{n} \quad (4.10)$$

i.e., evidence  $\xi_i$  supports all locations having a characteristic  $c_j$  with a degree of belief equal to  $1/n$ . Thus, locations with more characteristics matching a user's certain interest are more supported by  $\xi_i$  over locations with fewer matching characteristics. For example, consider a user's interest in sports with two characteristics:  $c_1$ = outdoors and  $c_2$ = running. Now, consider two candidate locations a gym,  $O_1$ , and a tennis court,  $O_2$ . Then,  $m(H_1) = m(\{O_2\}) = 0.5$  and  $m(H_2) = m(\{O_1, O_2\}) = 0.5$  resulting in  $Bel(O_1) = 0$  and  $Bel(O_2) = 0.5$ .

The resulting belief mass function  $m(H_j^{(i)})$  is further reevaluated with respect to the impact of evidence  $\xi_i$  as detailed in (4.3).

- **User's schedule constraints**

User's schedule constraints represent another strong body of evidence that can help in predicting the user's future location. If  $O_s$  represents the location of the earliest scheduled appointment and  $t_s$  the time left before the scheduled appointment takes place, then an evidence  $\xi^S$  can be inferred which consists of a set of  $n$  hypotheses  $\mathbf{H}^{(s)} = \{H_1^{(s)}, H_2^{(s)}, \dots, H_n^{(s)}\}$ , representing sets of locations grouped according to the traveling time needed to reach these locations. Each hypothesis,  $H_j^{(s)}$ , thus, is constructed such that

$$H_j^{(s)} = \{O_k : O_k \in \Theta | (t(O_c, O_k) + t(O_k, O_s)) \leq j \times \frac{t_s}{n}\} \quad (4.11)$$

where  $O_c$  is the user's current location and  $t(O_i, O_j)$  is the time taken to visit  $O_j$  starting from  $O_i$ . It is straightforward to deduce here that  $H_{j-1}^{(s)} \subseteq H_j^{(s)}$ .

A belief mass value  $m$  is associated with  $H_j^{(s)}$  such that:

$$m(H_j^{(s)}) = \frac{1}{n} \quad (4.12)$$

To ensure that this evidence becomes more vital as the scheduled meeting approaches, an impact  $(1 - \beta_s)$ , is assigned to  $\xi^S$ , such that the impact of evidence  $\xi^S$  increases as

the appointment time approaches.

$$(1 - \beta_s) = \frac{t(O_c, O_s)}{t_s} \quad (4.13)$$

The selection of  $n$  in (4.12) depends on the user's moving speed. For example, for a pedestrian,  $n$  can be selected such that  $t_s$  is divided into intervals of 10 minutes. In this case  $H_j^{(s)}$  will contain places that the user can visit within  $(j \times 10)$  minutes or less.

• **User's goals and tasks**

Having an idea of the user's future goals and tasks provides additional information aids in constructing other pieces of evidence. The knowledge base stores a set of user's goals and tasks  $\mathcal{G} = \{G_1, G_2, \dots, G_n\}$ . Each goal  $G_i$  is associated with the following attributes:

- $t_{\text{earliest}}^{(i)}$  The earliest date on which task  $G_i$  can be accomplished.
- $t_{\text{deadline}}^{(i)}$  deadline before which task  $G_i$  has to be performed.
- $t_{\text{duration}}^{(i)}$  The necessary amount of time to perform  $G_i$ .
- $C^{(i)} = \{c_1^{(i)}, c_2^{(i)}, \dots, c_n^{(i)}\}$  the set of characteristics related to task  $G_i$ .
- $V^{(i)}$  the degree of importance of task  $G_i$

Now a set of evidences based on the user's goals  $\mathcal{G}$  may be defined as  $\xi^{\mathcal{G}} = \{\xi_1, \xi_2, \dots, \xi_k\}$  such that:

$$\xi^{\mathcal{G}} = \{\xi_i : \forall G_i \in \mathcal{G} \mid t_{\text{earliest}}^{(i)} < t_{\text{prediction}} < t_{\text{deadline}}^{(i)} \vee t_{\text{duration}}^{(i)} < t_{\text{Available\_time}}\} \quad (4.14)$$

The impact  $(1 - \beta_i)$  assigned to an evidence  $\xi_i$  is dependent on the degree of importance of goal  $G_i$ .

$$(1 - \beta_i) = V^{(i)} \quad (4.15)$$

For each evidence  $\xi_i$ , based on an interest  $I_i$  with a set of  $n$  characteristics, a group of hypotheses  $\mathbf{H}^{(i)} = \{H_1^{(i)}, H_2^{(i)}, \dots, H_n^{(i)}\}$  is constructed such that:

$$H_j^{(i)} = \{O_k : O_k \in \Theta | c_j^{(i)} \in C_O(O_k)\} \quad (4.16)$$

A belief mass value is associated with  $H_j^{(i)}$  such that:

$$m(H_j^{(i)}) = \frac{1}{n} \quad (4.17)$$

The resulting belief mass function  $m(H_j^{(i)})$  is then reevaluated with respect to the impact of evidence  $\xi_i$  according to (4.3).

Since the Dempster rule of combination mainly work under uncertainty with regard to the collected data, there is an underlying assumption that the values of the collected data represents only estimates and not necessary precise values.

Finally, although the proposed work focused here on building bodies of evidence based only on three different classes, namely: user's interests, goals and schedule, further knowledge of the user's context (e.g., used devices, nearby people and mood) and the surrounding environment (e.g., weather and light) can also be used to build other bodies of evidence pertinent to the user's predicted location. Since the Dempster-rule of combination is commutative and associative, i.e., supports reasoning with incremental evidence, other bodies of evidence can be easily incorporated with no changes to the basic scheme.

### 4.5.3 Decision Making

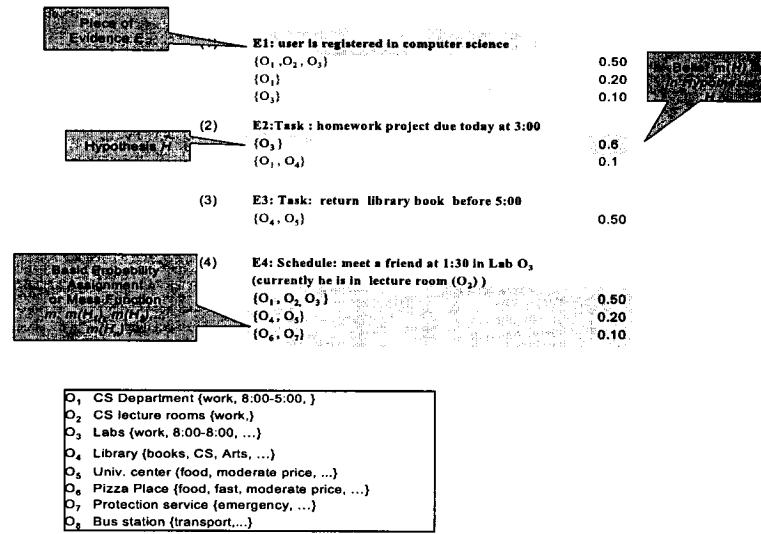
To determine the user's future predicted location, the reasoning engine combines each pair of hypothesis-belief mass using the Dempster rule of combination of (4.2) to conclude the user's

most likely future location.

Since the Dempster rule of combination is commutative and associative, these available bodies of evidence can be combined in any order. The result is a list of candidate locations together with their corresponding belief values which describe the degree of support for each candidate location. The location with the highest belief value is the predicted future location of the user. Figure 4.3 shows an illustrative example of the prediction process for a student in campus. Applying (4.2) on  $m_{E_1}$ ,  $m_{E_2}$ ,  $m_{E_3}$ , and  $m_{E_4}$  yields the combined belief distribution in Figure 4.3(b). The results show the belief in each of the candidate hypotheses. The predicted future location is the computer science laboratory represented by  $O_3$ .

#### 4.5.4 Path Finding

Once the future location of the user is predicted, the path finder module uses the orientations of the MOAC to determine the path going from the current user's location represented by a WEA  $\alpha_{\text{current}}$  to the predicted future location presented by the WEA  $\alpha_{\text{predicted}}$ . The outcome is a list of tuples  $(\alpha_x, \textit{orientation})$ . Consider the SCM of Figure 4.7, where  $\alpha_{\text{current}} = \alpha_5$  and  $\alpha_{\text{predicted}} = \alpha_6$ , then the user's expected trajectory can be represented by  $(\alpha_5, S), (\alpha_4, S), (\alpha_3, E), (\alpha_6, *)$ , where  $*$  is the target location. If more than one path from  $\alpha_{\text{current}}$  to  $\alpha_{\text{predicted}}$  exists, then different strategies can be used based on the user's common behavior. It has been shown in [133] that pedestrians choose the route that minimizes their distance as a primary strategy. In addition, route pleasantness is the second attribute that can be used to select between paths with equal lengths [133]. Similarly, drivers tend to avoid routes with high traffic. In general, a weight value can be assigned to each path according to the user's activity. The predicted path selection is then carried out using a shortest path algorithm such as Dijkstra [134].



(a) Sample Prediction

#	Hypothesis	Belief
1	$\{O_1\}$	0.1523
2	$\{O_3\}$	0.538
3	$\{O_1, O_2, O_3\}$	0.204
4	$\{O_4\}$	0.018
5	$\{O_1, O_4\}$	0.006
6	$\{O_4, O_5\}$	0.052
7	$\{O_6, O_7\}$	0.09
8	$\{O_1, O_2, \dots, O_8\}$	0.018

(b) Combination of Evidence

Figure 4.3: An example of evidence-based mobility prediction

### 4.5.5 Illustrative Examples

This section further illustrates the advantage of the proposed scheme over other existing history based approaches (e.g., [115], [119]), especially when users are located in new areas. Two different scenarios for mobility prediction are briefly discussed and the behavior of both the proposed work and existing history based approaches are compared in these cases. Consider a scenario where a frequent traveller tourist is visiting a city for the first time. Based on the user's interest in certain types of hotels series, cost and features, the MPA predicts the user selection of his preferred hotel. Hence, based on the predicted location, the MPA selects the trajectory that avoids streets with high traffic. Now the MPA utilizes knowledge about the tourist's interests (e.g., historical places, recreation, etc.), time of the day, cost preferences to build different bodies of evidence that when combined will result in the candidate site for the tourist's future location. In the case of history based approaches, (e.g., UMP scheme [119] and LeZi scheme [115]), no history for previously visited locations will be available. Thus, each time the user enters a new cell (in the case of UMP scheme) or a new location area (in the case of LeZi scheme), the new candidate location can not be determined and the history list must be updated with the new location. Given the fact that tourists usually will not repeat their visits to previously seen sites, this update in the history list is useless and even in some cases misleading. Another situation where a similar scenario arises is that of a business man in a trip. Although no data are available for previously visited locations, the MPA can accurately perform mobility prediction based on the tight known schedule usually available though access to his diary while inferring his selections for places of leisure through knowledge of his general interests.

## 4.6 Simulation Details and Results

This section summarizes simulation results of the proposed scheme. A simulation architecture was constructed to model and predict students' travelled trajectories in the campus of university of Ottawa. Figure 4.4 illustrates a schematic diagram for the simulation model which is comprised of the following four modules.

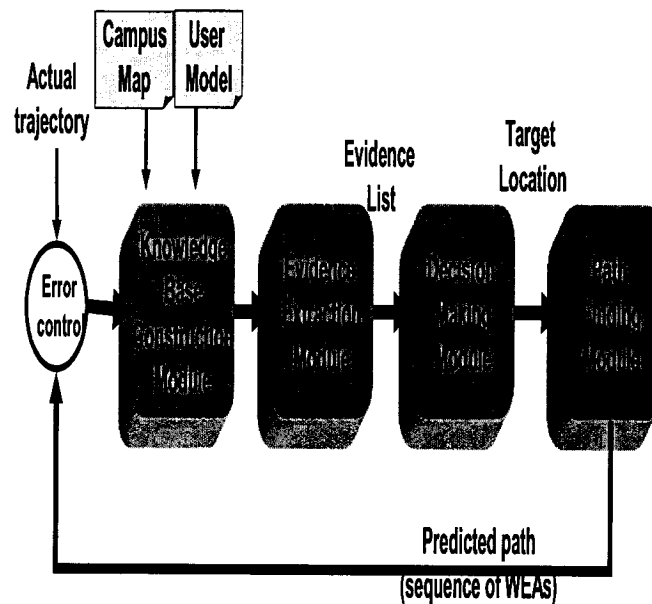


Figure 4.4: Schematic representation of the simulation model.

### 1. Knowledge Base Construction Module.

This module is responsible for constructing the knowledge base of the MPA. It is fed with two sources of information. The first source represents an SCM map of the university campus, while the second is related to the user context.

To facilitate the simulation of the proposed scheme, a simplified context model has been developed. As shown in Figure 4.6, the model is divided into the following six main categories. Personal context which is related to the users personal information such

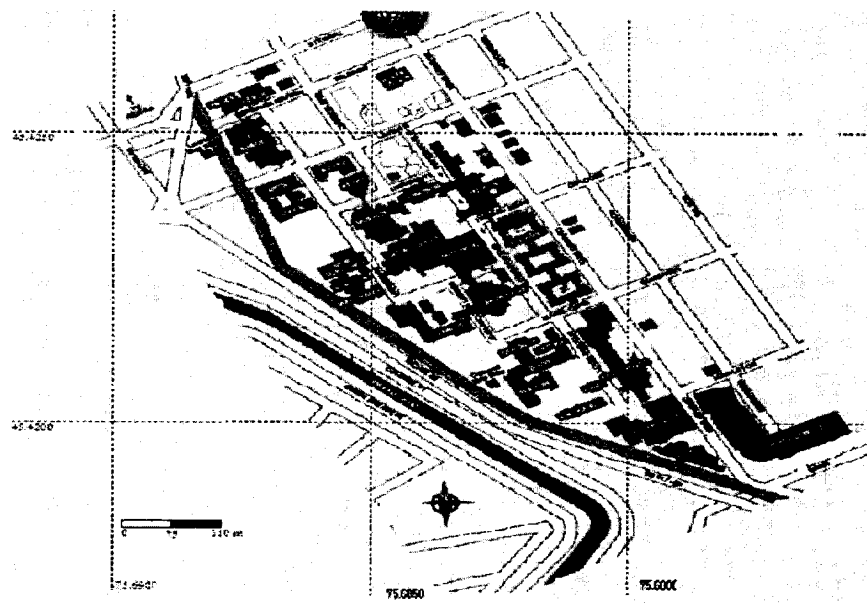


Figure 4.5: Campus map

as his age and name. Task context which describes the user tasks, such as shopping, handing assignments, etc. The interest context category describes the different fields of user's interests, such as doing sports, going to movies, watching TV, etc. The calendar context contains the user scheduled appointments. The activity context describes what kind of activity the user is doing (e.g., walking, riding a bike, or driving). Finally, the environment context captures the user's spatial surroundings. Figures 4.5 and 4.7 show a geographical map for the university campus and its corresponding SCM representation, respectively. The user model is converted into an XML description of the student's general preferences, tasks and calendar. Figure 4.8 describes a representative part of a typical XML file for a university student.

## 2. Evidence Extraction Module

This module utilizes the knowledge base to extract different bodies of evidence and their corresponding hypothesis as discussed in section 4.5. It is assumed here that students

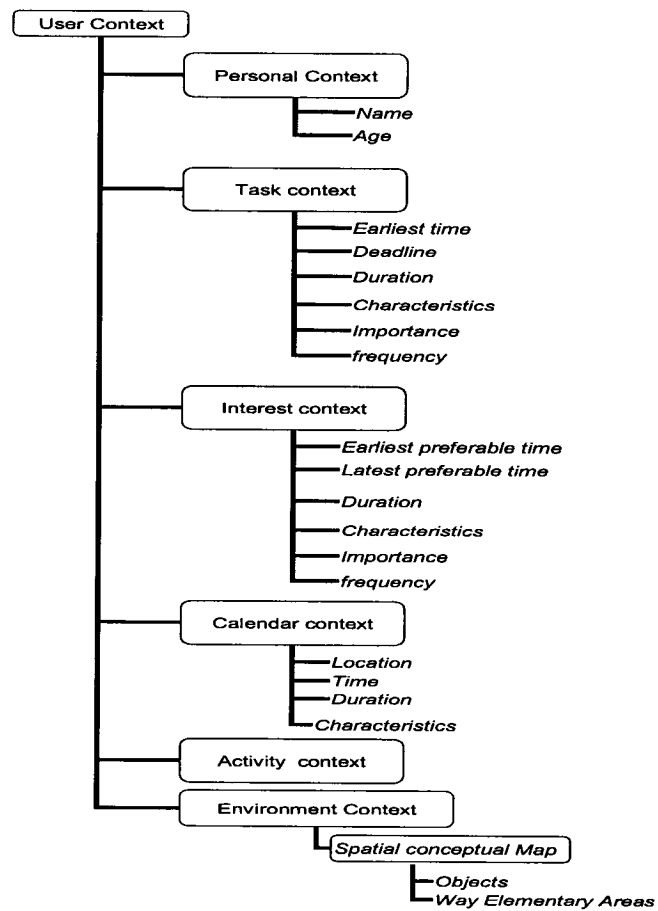


Figure 4.6: User Context Model

walk from one destination to another with an approximate speed of 5–5.2 km/hour [110].

### 3. Decision Making Module

This module applies Dempster-Shafer rule of combination to reach a decision about the user's future target location.

### 4. Path Finding Module

This module calculates the expected trajectory of the student from her current location to the predicted location. Based on the assumption that students try to minimize their

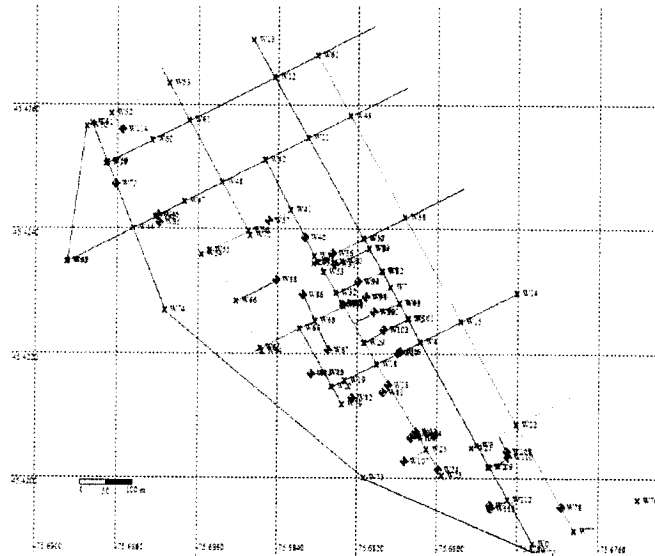


Figure 4.7: SCM representation of campus map

walking distances by selecting shortest path, finding the shortest path is done using the Dijkstra shortest path algorithm [134].

A periodic feedback mechanism using current location obtained from real trajectory triggers the prediction process whenever the actual trajectory of the student deviates from the set of previously calculated predicted points. This resembles a threshold-based location update.

#### 4.6.1 Mobility Data and User Context Collection

Many researches have either modelled user mobility by transition probabilities from cell to another (e.g., [116]) or used some variations of a random-walk mobility mode (e.g., [98]). It can be argued that these are simplified mobility models that fail to take into account neither certain aspects of the user's movement behavior such as his intentions and goals nor the impact of the surrounding environment, such as streets layouts. In order to validate the performance of the proposed scheme, trajectory data for different students travelling in and around Ottawa

```

<user>
  <name>
    <forename>Louis</forename>
    <surname>Lemon</surname>
  </name>
  <age> 19</age>
  <activities>
    <type>walking </type>
    <speed> 5.1km</speed>
  </activities>
  <Schedule>
    <description>
      Lecture Network management
    </description>
    <location> CUBE </location>
    <date>02/27/2003</date>
    <time>3:00:00 PM</time>
    <duration>180 min.</duration>
  </Schedule>
  <Interests>
    <description>
      do sports whenever time is available
    </description>
    <type> Entertainment</type>
    <Preferable_time>
      <from>16:00:00 </from>
      <to>22:00:00</to>
    </Preferable_time>
    <importance>80%</importance>
    <characteristic>Sport</characteristic>
    <characteristic>jogging</characteristic>
    <characteristic>Indoor</characteristic>
  </Interests>
  <Tasks>
    <description>
      Food Shopping outside campus
    </description>
    <earliest_deadline>
      <date>02/23/2003</date>
      <time>9:00:00 AM</time>
    </earliest_deadline>
    <latest_deadline>
      <date>02/29/2003</date>
      <time>08:00:00 PM</time>
    </latest_deadline>
    <expected_duration>
      120 min.
    </expected_duration>
    <characteristic>buy</characteristic>
    <characteristic>Food</characteristic>
    <characteristic>cheap</characteristic>
    <characteristic>
      transportation
    </characteristic>
  </Tasks>
</user>

```

Figure 4.8: Part of an XML file for a student

university campus for two weeks at the beginning of a semester was synthesized. It is worth noting, that in this scenario these students had no previous history of their visited locations. Students' collected trajectories were represented as a series of way points. Each way point is defined by the location in terms of an altitude and a longitude, time of day, and speed. To enhance the accuracy of the collected tracks' points, a software filter was used to match each collected track against streets layouts to eliminate any jitter caused by position errors. It is also assumed that users moved with an average speed of 5 km/hour [112], [119], [115]. Thus, collected way points' times were normalized to imitate a constant speed movement. However, this assumption can be later relaxed by considering that pedestrians tend to accelerate at the initiation of their trips and as they approach towards their destinations [135].

A simplified user model is used as shown in Figure 4.6. The user context model for each student was built using two steps. In the first step, students were asked to answer a questionnaire concerning their interests with regards to different places within the university campus. Secondly, students were asked to continuously register both their tasks and scheduled appointments. A general approach to collect these data automatically can be attained through a user context provider built on top of the MPA [122], [91], [92], [128].

## 4.6.2 Mobile User Predictability

Before deriving a measure for the accuracy of the MPA, the nature of user's trajectories is analyzed and a method for categorizing these users into different groups, for which different levels of accuracy will be obtained, is derived. Based on the data collected from different students, they were categorized in three different categories according to the level of *predictability* in their travelled trajectories. Predictability in this context refers to the level of certainty with which the student's future location can be determined. More precisely, the degree of predictability  $\mathcal{P}(O_i)$  of a future location  $O_i$  is equal to the total belief in  $O_i$  being the future lo-

cation after accumulating all evidence using (4.2), i.e.,  $\mathcal{P}(O_i) = Bel(O_i)$ . The total degree of predictability of a student's trajectory during an experiment of a total time  $T$ ,  $\mathcal{P}_T$ , is calculated as,

$$\mathcal{P}_T = \frac{\sum_{i=1}^{i=n} Bel(O_i)}{n} \quad (4.18)$$

where  $n$  is the total number of visited locations during  $T$ . In these experiments  $T$  is equal to 9 hours a day for 14 days. As  $T \rightarrow \infty$ , call  $\tilde{\mathcal{P}} = \lim_{T \rightarrow \infty} \mathcal{P}_T$ , the general degree of predictability of the user.

To further illustrate this, assume that only bodies of evidence based on the user's schedule are collected. In this case as shown in (10-12)  $Bel(O_s) \propto t(O_c, O_s)/t_s$ . Thus, as the time available before the scheduled appointment  $O_s$  ( $t_s$ ) decreases, the certainty that  $O_s$  is the future location increases.

In other words, students with a tight known schedule leave little chances for uncertainty and therefore enjoy a high degree of predictability. On the other hand, students with frequent intermitting breaks and plenty of spare time come low on the scale of predictability since their location is hardly known in advance even with the availability of their movement history. Figure 4.9 shows the distribution of predictability across students subject to the experiments.

It is to be noted here that predictability has been chosen as the main parameter for demonstrating the performance of the proposed scheme under various conditions of uncertainty in the users' behavior. This is in contrast to using *regularity* (e.g., as in [119], [118]) since the performance of the proposed scheme is not influenced by the existence (or the lack of) a regular history for the pattern of movements of the users.

The remainder of the section presents the performance evaluation of the proposed scheme for tracking students belonging to the above mentioned categories.

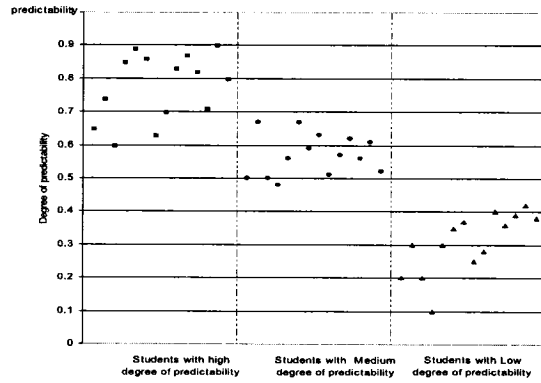


Figure 4.9: Predictability degrees of different students

### 4.6.3 Prediction Accuracy

In this section, an estimate for the degree of accuracy in predicting the trajectory of a user with a predictability  $\tilde{\mathcal{P}}$  is derived. The derived accuracy estimate is generally dependent on the probability of predicting the right destination as well as the probability of predicting the correct path. With the underlying assumption that users tend to minimize their walking distances [133], then the latter probability can be ignored, without loss of generality.

In the last section, it has been shown that on the average the belief in predicting the user's future location,  $O_p$ , is  $\tilde{\mathcal{P}}$ . Hence,  $1 - \tilde{\mathcal{P}}$  represents the degree of uncertainty in addition to ignorance in allocating the future location of the user. Without loss of generality, assuming that the system's degree of ignorance concerning the user is negligible, then the remaining uncertainty, which is equal to  $(1 - \tilde{\mathcal{P}})$ , will be divided over the entire range of candidate future locations predicted by the MPA,  $\{O_1, O_2, \dots, O_p, \dots, O_N\}$ , where  $N$  is the total number of those candidate locations. Furthermore, one can assign a general degree of confidence to those locations including  $O_p$  in the following manner,

$$m_{i \neq p}(\{O_p, O_i\}) = 2m_{i \neq j \neq p}(\{O_p, O_i, O_j\}) = \dots = (N - 1)m(\{O_1, O_2, \dots, O_p, \dots, O_N\}) \quad (4.19)$$

In other words, (4.19) states that the MPA's belief in that the candidate future location is either

$O_i$  or  $O_p, i \neq p$ , is twice stronger than the belief that the future location is in  $\{O_i, O_p, O_j\}$ ,  $i \neq j \neq p$ , and is three times stronger than the belief that it is in  $\{O_i, O_p, O_j, O_k\}$ ,  $i \neq j \neq p \neq k$ , and so forth.

Hence, the uncertainty  $1 - \tilde{\mathcal{P}}$  can be expressed as the following summations,

$$1 - \tilde{\mathcal{P}} = \sum_{i=1, i \neq p}^{i=N} m_{i \neq p}(\{O_p, O_i\}) + \sum_{i,j=1, i \neq j \neq p}^{i,j=N} m_{i \neq j \neq p}(\{O_p, O_i, O_j\}) + \dots + m(\{O_1, O_2, \dots, O_p, \dots, O_N\}) \quad (4.20)$$

In general, using (4.19) and (4.20), it can be deduced that  $m(A), A \in 2^\Theta$  is given by,

$$m(A) = \frac{(1 - \tilde{\mathcal{P}})(N - k - 1)}{\sum_{i=1}^{i=N-1} \binom{N-1}{i} (N-i)}, \quad A \in 2^\Theta, \quad O_p \in A \quad \text{and} \quad k = |A| \quad (4.21)$$

To derive an estimate for the accuracy of the MPA based on the predictability  $\tilde{\mathcal{P}}, E(\tilde{\mathcal{P}})$ , the definition of the relative atomicity of  $\{O_p\}$  with respect to  $A, a(\{O_p\}/A)$ , is used, which is defined by [136],

$$a\left(\frac{X}{Y}\right) = \frac{|X \cap Y|}{|X|}, \quad X, Y \in 2^\Theta, \quad Y \neq \phi, \quad (4.22)$$

where  $|X|, X \in 2^\Theta$ , is the number of states it contains.

Thus  $E(\tilde{\mathcal{P}})$  can be obtained as a weighted sum of beliefs  $m(A)$  with  $a(\{O_p\}/A)$  as the weighting coefficients, i.e.,

$$E(\tilde{\mathcal{P}}) = \sum_{A: O_p \in A, A \in 2^\Theta} m(A) a\left(\frac{\{O_p\}}{A}\right) \quad (4.23)$$

substituting from (4.21) and (4.22) into (4.23) we get

$$\begin{aligned} E(\tilde{\mathcal{P}}) &= \tilde{\mathcal{P}} + (1 - \tilde{\mathcal{P}}) \sum_{i=1}^{i=N-1} \frac{\binom{N-1}{i} (N-i)}{(i+1) \sum_{k=1}^{k=N-1} \binom{N-1}{k} (N-k)}, \quad N > 1 \\ &= \tilde{\mathcal{P}} + \frac{(1 - \tilde{\mathcal{P}}) 2^{N+1} (N+2) - 4(N^2 + N + 1)}{N (2^N (N+1) - 4N)}, \quad N > 1 \end{aligned} \quad (4.24)$$

where  $N$  is the number of candidate locations.

Figure 4.10 shows a plot of the expected prediction accuracy versus the number of candidate locations for  $\tilde{\mathcal{P}} = 0.8, 0.6$  and  $0.4$ . As can be seen, at  $N = 2$ ,  $E(\tilde{\mathcal{P}}) = \tilde{\mathcal{P}} + (1 - \tilde{\mathcal{P}})/2$ , i.e., the uncertainty  $1 - \tilde{\mathcal{P}}$  is divided equally between the two possible locations. On the other hand, as  $N$  increases the expected accuracy decreases. In fact, as  $N \rightarrow \infty$ ,  $E(\tilde{\mathcal{P}}) \rightarrow \tilde{\mathcal{P}}$ , i.e., as the number of candidate locations increases it is expected that the degree of certainty in the candidate location asymptotically approaches  $\tilde{\mathcal{P}}$ .

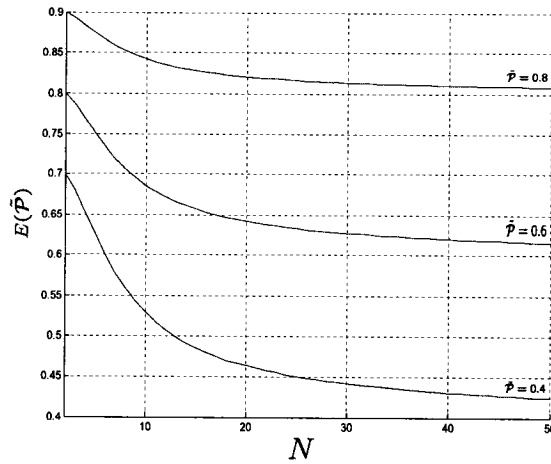


Figure 4.10: Expected prediction accuracy vs. number of locations

#### 4.6.4 Accuracy Measurements

In this section, a formula that will be used to measure the accuracy of the MPA's predicted trajectory is derived. Trajectory accuracy is the measure between the error in the path predicted by the MPA and the actual trajectory chosen by the user as it is collected. In general, differences between two trajectories can be partitioned into two basic errors: spatial and temporal. Since users are assumed to move with a constant speed in both the actual and predicted trajectories, temporal errors are eliminated. In the following, an accuracy measure for errors resulting from spatial errors is derived.

Assume that the actual and predicted trajectories,  $C_A$  and  $C_P$ , are represented by a set of points  $\{a_1, a_2, \dots, a_n\}$  and  $\{p_1, p_2, \dots, p_m\}$ , respectively, where each point  $x \in C_A \cup C_P$  is described by a latitude  $x^{\text{lat}}$ , longitude  $x^{\text{long}}$ , and a time  $t(x)$  at which this point has been reached, such that  $x \equiv (x^{\text{lat}}, x^{\text{long}}, t(x))$ . Then, an approximate Hausdorff-like distance measure formula [137] can be adopted to estimate the error in predicting a certain point  $p_i$  with respect to the actual trajectory,  $C_A$ ,  $\tilde{\delta}(p_j, C_A)$ . The error measure is based on determining the distance between  $p_i$  and the actual trajectory as follows,

$$\tilde{\delta}(p_j, C_A) = \|a_i - p_j\|_{\min_{a_i} |t(a_i) - t(p_j)|}, \quad (4.25)$$

where  $\|\cdot\|$  is the Euclidian norm and is calculated as  $\|a_i - p_j\| = \sqrt{(a_i^{\text{lat}} - p_j^{\text{lat}})^2 + (a_i^{\text{long}} - p_j^{\text{long}})^2}$  for  $a_i \in C_A$  and  $p_j \in C_P$ .

Thus, the overall error of predicting the trajectory  $C_P$  with respect to the actual trajectory  $C_A$ ,  $\delta_m(C_P, C_A)$ , can be calculated as,

$$\delta_m(C_P, C_A) = \frac{1}{m} \sum_{p_j \in C_P} \tilde{\delta}(p_j, C_A) \quad (4.26)$$

Figure 4.11 depicts a typical measured error using the above formula for the first 200 meters of the predicted trajectory in Figure 4.12 (details of this experiment will be presented later in this section).

The accuracy of a predicted trajectory  $C_P$ ,  $\sigma_P$ , is the ratio between the above measured error and the length of the actual trajectory and can be calculated as,

$$\sigma_P = \frac{\delta_m(C_P, C_A)}{\sum_{i=1}^{i=n} \|a_{i+1} - a_i\|} \quad (4.27)$$

Throughout the rest of the chapter  $\sigma_P$  will be used as the measure of accuracy of the trajectories in the performed experiments.

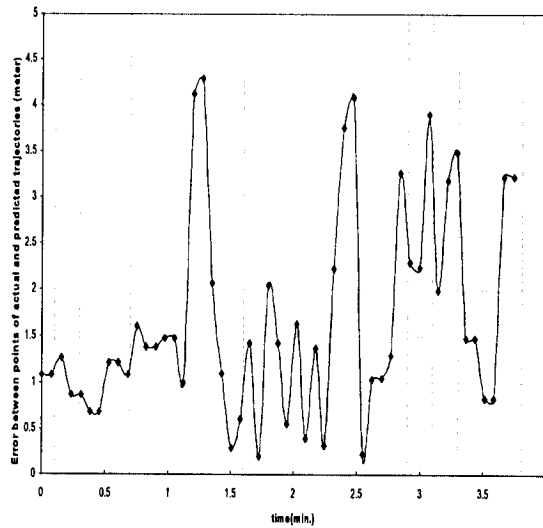


Figure 4.11: Error between points of actual and predicted trajectories vs. time

#### 4.6.5 Results of Trajectory Prediction

Figure 4.12 shows a comparison between the actual trajectory of an undergraduate student and the trajectory predicted using the MPA. This experiment shows the performance of the proposed algorithm in locating users from the category with the highest level of predictability; namely undergraduate students with a tight schedule. The movement trajectory of those students can be inferred mainly based on their schedule. In other words, the dominating bodies of evidence in this case are the ones extracted from the students' schedule. The solid line depicts the actual trajectory and the dashed line shows the predicted trajectory. For this trajectory  $\sigma_P$  was found to be 7% and therefore the trajectory was predicted with 93% accuracy.

In another set of experiments, the proposed scheme is used to track the movement of graduate students characterized by a lower degree of predictability. Figure 4.13 shows the actual and predicted trajectory of a student with 0.6 degree of predictability. In this case,  $\sigma_P \simeq 18\%$ , i.e., the trajectory is predicted with 82% accuracy. Figure 4.14 shows the predicted trajectory of a graduate student performing field work research. The student's movement pattern is characterized by a degree of predictability  $\tilde{P} = 0.4$  where the measured accuracy decreased to 65%.

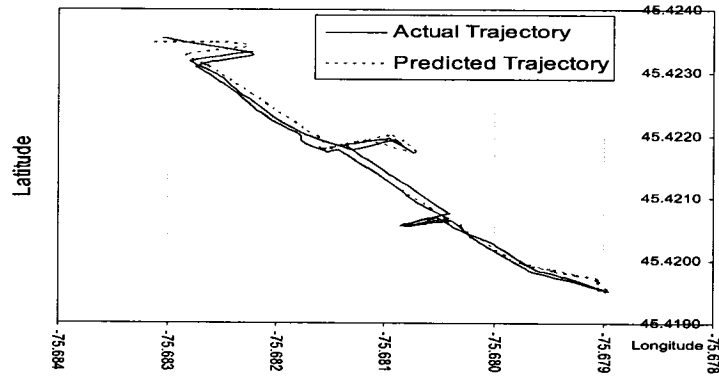


Figure 4.12: Actual trajectory vs. predicted trajectory of a student with 0.8 predictability.

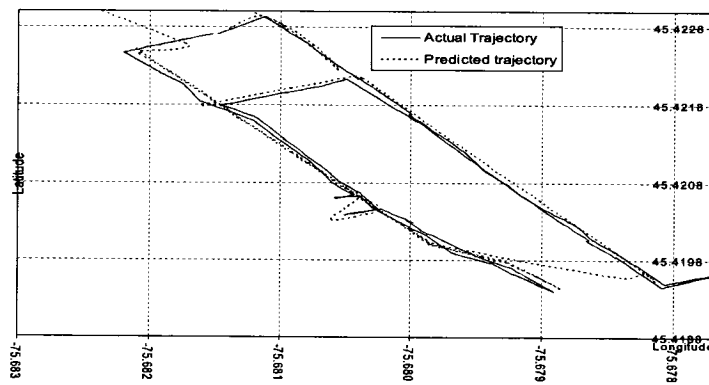


Figure 4.13: Actual trajectory vs. Predicted trajectory of a student with 0.6 predictability.

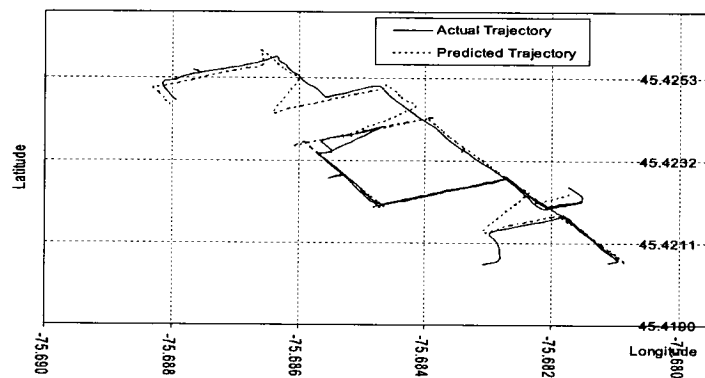


Figure 4.14: Actual trajectory vs. Predicted trajectory of a student with 0.4 predictability.

#### 4.6.6 Accuracy of the MPA vs. Number of Visited Locations

Figure 4.15 displays the hit ratio against the number of visited locations  $n$  over an experiment length  $T$  for different degrees of users' predictability. A *hit* in Figure 4.15 refers to a correct

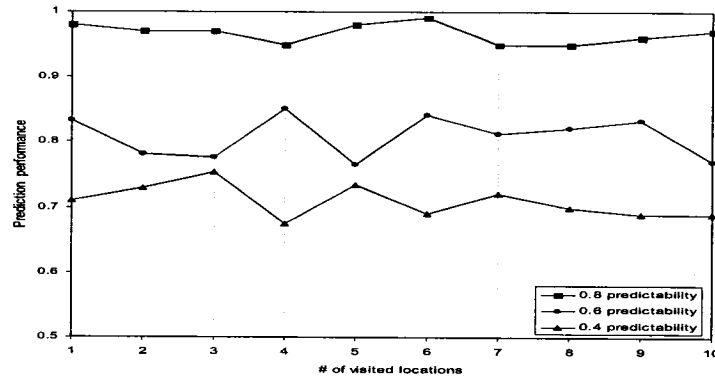


Figure 4.15: Hit ratio vs. number of visited locations

prediction by the MPA that matched the actual target location [119]. On the other hand, an incorrect future location prediction is referred to as a *miss*. In general, it is clear that the hit ratio is not affected by the number of visited locations (regardless of whether these locations have been visited before). The MPA reaches an average of a 97% hit ratio for students with the highest predictability. For students with a 0.6 predictability degree, the hit ratio decreases due to the relative uncertainty about the students destinations. However, an average hit ratio of 80% is maintained as the number of the visited locations increases.

#### 4.6.7 Prediction Accuracy and the MPA Warm-Up Period

It can be noticed that in contrast to other approaches (e.g., [119], [110]) the MPA does not require a warm-up period to collect history of previously visited locations. Therefore, the MPA maintains the same degree of accuracy in predicting students' movements independently of the time span of the experiments, as was shown in the previous section.

Although that in order for the MPA to accurately predict the user's mobility, an initial

phase is required to build the user's context model which will contain the user's interests, it can be noticed that this initial phase is required once to build a generic view about the user's interest. Furthermore, both schedules and tasks are collected continuously ahead of the related prediction step and it is performed as part of updating the user context. Hence, interest collection phase can be considered as an initial setup phase which will only be performed once and will be successfully used even if the user is situated in new places. On the other hand, in history based approaches, warm-up period is proportional to the regularity of the users [119] and is required each time the user moves to newer places.

#### 4.6.8 Complexity Analysis

To calculate the time complexity of the proposed scheme, the time requirements for each stage of the MPA prediction process individually is examined. It is assumed that the time required for information gathering is negligible since it mainly depends on the query/response time between the MPA and the context provider. In the second phase, evidence extraction complexity, using(4.7)-(4.13), is linear with respect to size of the frame of discernment  $\Theta$ , i.e.,  $\mathcal{O}(|\Theta|)$ . The complexity of applying the Dempster rule of combination, in the third stage, is  $\mathcal{O}(|\Theta|\log|\Theta|)$  [138]. Finally, the complexity of path calculation using Dijkstra shortest path algorithm is  $\mathcal{O}(|E| + |V|\log|V|)$  using a binary heap implementation, where  $|V|$  is the number of WEAs and  $|E|$  is the number of edges between all WEAs. Hence, the total complexity of the MPA is equal to  $\mathcal{O}(|\Theta|) + \mathcal{O}(|\Theta|\log|\Theta|) + \mathcal{O}(|E| + |V|\log|V|)$ , i.e.,  $\mathcal{O}(|\Theta|\log|\Theta|) + \mathcal{O}(|E| + |V|\log|V|)$ . Since the number of WEAs is always bigger than the number of candidate locations ( $|V| > |\Theta|$ ), then the total complexity of the MPA scheme is  $\mathcal{O}(|E| + |V|\log|V|)$ . Furthermore, if by assuming that shortest paths between different locations are already stored, then the MPA prediction complexity is reduced to  $\mathcal{O}(|\Theta|\log|\Theta|)$ .

The MPA prediction process is triggered once for each new target location. In the above

experiments, students visited between 2 to 10 locations per day. This can be compared to other history based approaches which requires the MT to perform an update whenever it enters a new cell [119]. In general, a history list update requires an  $\mathcal{O}(n)$  comparisons to search if the new cell has been visited before, where  $n$  is the number of stored locations in the history list.

#### 4.6.9 Memory Requirements Analysis

The SCMs are represented as MOACs which are symmetric sparse matrices. The size of an MOAC depends on the number of WEAs that represent streets and landmarks layouts. On the average, each row in a MOAC has  $2 \sim 3$  entries. Thus, the average storage complexity of an MOAC is  $\mathcal{O}(|V|)$ , where  $V$  is set of WEAs. For example, in the SCM of the university campus in Figure 4.7,  $|V| = 114$  resulting in a density of 2.6%.

### 4.7 Summary

In this chapter, a novel scheme for a mobility prediction based on the use of contextual information has been presented. Uncertainty of the user's navigation behavior was captured by gathering pieces of evidence concerning groups of candidate locations. These groups were then refined to predict the user's future location when evidence accumulate using Dempster rule of combination. In contrast to existing approaches the proposed approach did not impose any assumptions concerning the availability of a history of user's movements. The Dempster-Shafer evidence theory has been chosen because it offers an appropriate scheme for the modelling of the differences between information acquired at the two considered spatial scales.

Furthermore, since the Dempster-Shafer theory supports reasoning with incremental evidence, evidence other than the ones used in the proposed work can be included at a later stage. Simulation results were presented to demonstrate the performance of the proposed scheme.

## **Chapter 5**

# **Anomaly Diagnosis via Statistical analysis and Evidential Reasoning**

This chapter presents a robust scheme to reliably diagnose network anomalies in real-time. The contribution of this chapter is two fold. In the first contribution, a bi-cycle application of auto regression is used to model increments in the values of network monitoring variables to accurately detect network anomalies. The second contribution presents a new scheme to classify the root-cause of the anomalies, detected in the first step, using concepts of evidential reasoning of Dempster-Shafer theory.

This chapter proceeds as follows. Section 5.1 introduces concepts of anomaly diagnosis schemes and highlights the motivations behind the proposed technique. Discussion of related work is presented in Section 5.2. Section 5.3 outlines the main ideas of the proposed anomaly diagnosis scheme. Sections 5.4 and 5.5 provide details of the proposed anomaly detection and classification steps, respectively. Two alternative classification schemes particularly suited for biased training sets are presented in Sections 5.6 and 5.7. Performance analysis and results are presented in Section 5.8. Finally, Section 5.9 concludes the chapter.

## 5.1 Introduction

Computer networks are increasingly becoming an integral part of humans' everyday life. Networking systems can be viewed as the sole enablers of a wide range of services in mission critical domains including defense, transportation, manufacturing and health care. An essential feature of such systems is their sustainability, which can be measured in terms of their ability to tolerate faults and maintain an acceptable performance in the presence of failures or attacks. In general, network anomalies can be attributed to various causes; in addition to hard failures due to hardware problems (e.g., equipment failure and power outage), soft failures can result from inappropriate use of network resources, protocol failures, temporary congestion, mischievous users and denial of service attacks. While anomalies in communication networks are usually unavoidable, early detection and classification of such faults is crucial to providing networking services with a high level of availability and reliability.

In the past, as there were fewer network equipments, fault management was handled manually by experienced administrators who relied on the analysis of a limited set of device variables. As networks continue to grow in size and complexity, the task of diagnosing network faults is increasingly involving the collection and analysis of an immensely growing amount of high dimensional data. Fortunately, advances in network monitoring and capturing tools, such as the simple network management protocol (SNMP) [139], remote network monitoring (RMON) [140] and NetFlow [141] facilitates the capturing of the necessary measurements to describe the fine grained behavior of each networked equipment. Nonetheless, analysis of such data in order to diagnose network anomalies is still considered an unmet challenge and remains to be the focus of numerous ongoing research efforts (e.g., [142], [143], [144], [145], [146]).

This chapter presents two contributions to address two particular challenges in the domain of network anomaly diagnosis. The first contribution presents a novel technique to accurately detect abrupt changes in measurements network parameters in order to flag potential anomalies.

The second contribution, uses the detected anomalies to classify the root-cause of a network node failure using the Dempster-Shafer theory of evidential reasoning.

In general, anomaly detection, the target of the first contribution, involves the identifications of time-points where performance variables show a sudden deviation from their normal behavior. A particular difficulty in performing such operation arises from the non-stationary and stochastic nature of data representing those variables in addition to the erroneous nature of the collected data due to possible glitches in the monitoring systems. The proposed anomaly detection approach presented builds on recent developments in the area of statistical analysis [147] where a bi-cycle of statistical modelling is used to represent the analyzed data while effects of isolated sudden changes are diluted using windowed averaging mechanisms.

While the outcome of performing the former operation identifies a possible occurrence of a failure in a network component as viewed by an individual variable, anomaly classification deals with the collective analysis of groups of variables in order to provide a higher-level view of the most probable root-cause of the detected anomalous behavior. Classification of anomalies' root-cause is challenging as it requires a-priori knowledge of models of all types of anomalies which are usually cumbersome to obtain. One way to automate such operation is to utilize a set of a continuously learnt anomalous patterns to aid the classification of newly detected anomalies. Nonetheless, this task is usually hurdled by the noisy nature of the collected data, the high dimensionality of the problem domain, and the large size of required training data in order to perform the classification step with an acceptable degree of confidence.

We present a robust anomaly classification scheme based on the framework of the evidential reasoning theory [96]. In the proposed technique, different pieces of evidence concerning possible candidate root-causes of a newly detected anomaly are constructed from a training set of previously detected and classified anomalies. To handle cases of imbalanced training sets, the presented work describes two new approaches to assign mass belief values to the classes under investigation using the constructed evidence. The latter contribution can be of

independent interest to research related to the Dempster-Shafer theory and can be applied in contexts other than the present one.

## 5.2 Related work

The problem of anomaly detection in computer networks has been the focus of intensive research efforts. In the following, we analyze a sample of these efforts.

Statistical analysis of network performance variables has been studied in [148], [144]. In [144], Thottan et. al. detect abrupt changes in the measured traffic management information base (MIB) counts obtained via the SNMP protocol using time series analysis. An auto regression model of each variable is applied where a generalized likelihood ratio test (GLR) is performed sequentially to detect abrupt changes in the modelled time series. Second order statistics were used in [148] to investigate load anomalies in segments of IP networks, particularly for voice over IP traffic.

Principle component analysis (PCA) is introduced in [142], [149] to separate variations in traffic at any point into normal and anomalous components. Once an anomalous behavior is detected, the diagnosis of the anomaly is compared to a vector associated with each anomaly that describes the manner in which a particular anomaly adds traffic to each link in the network. Based on traffic analysis, the authors developed a set of heuristics to for anomaly diagnosis. Barford, et al. [145] use pseudo-spline wavelets as the basis to analyze the time localized normalized variance of the high frequency component to identify signal anomalies. A similar approach for the utilization of wavelets has been introduced in [146].

In comparing various approaches, statistical analysis techniques represent the simplest approaches with a relatively lower computational complexity. Thence, the approach adopted in this chapter is based on the statistical analysis of measurement variables. However, in contrast to existing approaches, it aims at reducing the number of detected false alarms by diluting

effects of isolated small deviations.

In contrast to anomaly detection, anomaly classification has received little attention and usually appeared as a minimal complementary part for the anomaly detection step. Earlier attempts to address anomaly diagnosis included rule-based expert systems [150] and model-based systems [151]. These approaches were limited in their scope and hardly captured the impeded complexity in the underlying networking systems. A method for generating hypotheses about the symptoms associated with each network fault is presented in [152]. The method assumes the existence of prior fault probabilities obtained through long-term statistical observations.

In [144], correlation patterns between a collection of SNMP MIB variables were investigated. In [153], a set of rules were developed for the identification and, hence, the classification of particular traffic patterns that are accompanied by different network anomalies originating from security attacks. Another classification scheme based on the utilization of a neural network classifier has been introduced in [154] to diagnose anomalies caused by network attacks.

Artificial anomalies were injected in [155] into an inductive rule learning system to detect known and unknown anomalies resulting from intrusions and network misuse. Lakhina et.al. [156] treat network faults as events that disturb the distribution of traffic features such as the addresses and ports and applies sample entropy to classify different anomalies.

Various clustering techniques have been also used for anomaly classification. The performance of three different clustering algorithms has been studied in [157], namely a fixed-width clustering-based algorithm, an optimized K-nearest-neighbor algorithm, and an unsupervised variant of the are support-vector machine (SVM) algorithm.

The proposed anomaly classification scheme differs from other existing approaches in that the unique nature of the adopted Dempster-Shafer theory allows decision making under uncertainty and existence of imperfect knowledge. Hence, the theory lends itself easily to the problem of anomaly classification. As will be illustrated, the proposed classification scheme

has the innate feature of accurately identifying the significance of each training sample in making decisions for labelling new anomalies.

### 5.3 Outline of the Proposed diagnosis Scheme

In this section, we outline the adaptive learning approach employed in the proposed scheme for network anomaly diagnosis at real-time. The proposed scheme relies on collected network performance measurements to continuously learn the normal behavior of the network and detect deviations from the norm. Once performance deviations are detected and measured, the scheme analyzes a maintained set of previously detected anomalies in order to decide on the most probable root-cause of the detected new network anomaly.

Figure 5.1 provides a schematic description of the functionalities of the NPA for anomaly diagnosis. An NPA is typically located at a network node. It is assumed here that the node is supported with the necessary performance monitoring and data measurements and collection tools that can be delivered through DAs. As shown, the scheme is divided into two main modules: a change detection module and a decision engine. The first module of the NPA

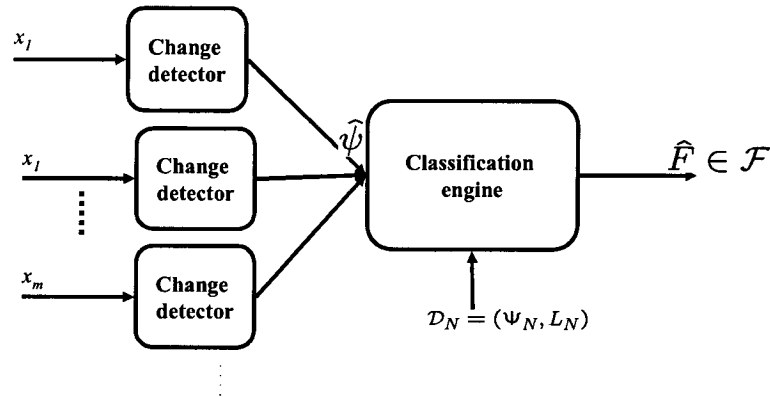


Figure 5.1: A schematic description of the NPA components

statistically analyzes increments of the obtained measurements from each monitored variable

through a bi-cycle of an auto regressive modelling process in order to obtain a normalized measure of possible abrupt changes at a given time point.

The collected deviation measures for each monitored variable are then combined by the decision engine to reach conclusions pertinent to failure cause. The proposed decision engine adopts a novel approach for anomaly classification through the utilization of Dempster-Shafer theory of evidence [96].

The following sections present further details of the diagnosis steps described above.

## 5.4 Step 1: Change detection

A typical input to any network diagnosis system is a collection of continuous stream measurements of performance monitoring variables that measure the behavior of traffic flows (e.g., number of dropped packets and link utilization) as well as the nodes' performance (e.g., CPU utilization and memory load). These measurements can, for example, be presented as set of MIB counts collected via SNMP that are supplied through the DA.

The starting point of the proposed scheme is the declaration of the hypothesis that independent of the particularities of a given network anomaly, they all share one common feature. This common feature entails that they all induce some form of change in the underlying traffic characteristics which in turn results in abrupt changes in one or more of the measured variables. Based on the above premise, network anomalies can be identified through the analysis of the spectral characteristics of the obtained data streams representing performance variables.

In order to perform the first step of abrupt change detection, we follow a statistical analysis approach of measurements obtained from network monitoring variables. Auto regressive (AR) modelling techniques have been widely adopted for the analysis of various network performance measurement (e.g., [158], [147], [159]). This can be attributed to the fact that AR models can easily capture the linear dependency of the future values on the past ones and

hence can be used to identify anomalous behaviors in the sequence of the modelled time series.

In such case, one can reformulate the problem of network anomaly detection to that of detecting an abrupt change in the modelled AR and the estimation of the change time. The proposed scheme adopts ideas from recent developments in the area of statistical analysis [147] based on a bi-cycle application of AR modelling.

We note that there are three main goals desirable to any abrupt change detection mechanism. The first goal is to minimize the number of false alarms. The second is to maintain a very short delay before which actual change is detected. Finally, an efficient algorithm must provide an indication of the magnitude of the occurring abrupt change.

In order to describe the functionalities of the NPA, let the time series  $\{x_j(t_1), x_j(t_2), \dots\}$  denote a sequence of increments in observations obtained from the  $j$ -th network monitoring variable, where  $t_i$  is the time variable. In the rest of this section,  $x_j(t_i)$  will be referred to simply as  $x_i$  where the subscript  $j$  is considered implicit in the manipulations. The series can be expressed using a second time series  $\{y_1, y_2, \dots\}$ , such that,

$$x_i = y_i + \mu, \quad \text{and} \quad (5.1)$$

$$y_i = \sum_{j=1}^k \alpha_j y_{i-j} + \varepsilon \quad (5.2)$$

$\{y_1, y_2, \dots\}$  is a piecewise constant AR process of the  $k$ -th order.  $\varepsilon$  is a white noise following a Gaussian distribution  $\mathcal{N}(0, \sigma_\varepsilon^2)$ . The conditional probability distribution for  $x_i$  can be obtained as [158], [147].

$$\begin{aligned} p_k(x_i | \cdot) &\equiv p(x_i | x_{i-k}, \dots, x_{i-1}; \alpha_1, \dots, \alpha_k, \sigma_\varepsilon) \\ &= \frac{1}{\sigma_\varepsilon \sqrt{2\pi}} e^{-\frac{(x_i - \omega_k)^2}{2\sigma_\varepsilon^2}}, \quad (5.3) \\ \omega_k &= \mu + \sum_{j=1}^k \alpha_j (x_{i-j} - \mu) \end{aligned}$$

It is worth noting here that the AR parameters can be estimated using different methods described in literature, e.g., via minimizing the squared errors [160], [161]. Once those param-

eters have been estimated, the AR model is then considered a representative of the normal behavior of the network node.

Using the conditional probability distribution, a normalized measure of the deviation of a point  $x_i$  in the time series can be calculated as

$$dev(x_i) = \frac{\log p_k(x_i|\cdot)}{\log p_k(x_i|\cdot) - 1} \quad (5.4)$$

where  $\log$  denotes the natural logarithm.

The averaged loss or deviation in the time series  $\{x_1, x_2, \dots\}$  can then be calculated by averaging the occurring deviations in individual points over successive time periods of length  $T$  to obtain,

$$z_l = \frac{1}{s} \sum_{i=(l-1)s+1}^{ls} dev(x_i) \quad (5.5)$$

where the index  $l$  refers the  $l$ -th interval, and  $s$  is the number of samples within each interval.

One can further eliminate false alarms by reducing effects of isolated outliers through the analysis of the time series  $\{z_1, z_2, \dots\}$ . This can be achieved by remodelling  $\{z_1, z_2, \dots\}$  using a second AR model  $\{h_1, h_2, \dots\}$ , such that  $z_l = h_l + \bar{\mu}$ . Consequently, a second conditional probability density function  $p_k(z_l|\cdot)$  is obtained and used to calculate an overall deviation measure (obtained using (5.4) by substituting for  $x_i$  with  $z_l$ ).

It is worth noting here that the size of the period under investigation,  $T$ , influences the delay in detecting an abrupt change in the series under investigation. When  $T$  is large the detection of an abrupt change is delayed. On the other hand, a very small  $T$  does not allow for diluting the effects of isolated outliers, and hence increases the chances of detecting false alarms.

As shown in Figure 5.1, the process of abrupt change detection is applied to the time series obtained from the increments in each of the measured network monitoring variable. The final result of this step is a set of normalized measures for the deviation of each variable from the consciously learnt normal behavior.

## 5.5 Step 2: Dempster-Shafer for Anomaly diagnosis

The goal of this step is to apply concepts of Dempster-Shafer's theory (DST) to the calculated deviation measures, obtained in the previous step to classify the root cause failure of the detected anomaly.

Using DST concepts, a-priori available knowledge about pervious network failures are used to generate hypotheses and bodies of evidence that will lead the decision concerning the most probable root-cause of the detected network anomaly.

To formalize this step, let the vector  $\psi(t) = [\psi^1(t_i) \cdots \psi^m(t_i)]$  denote a normalized  $(1 \times m)$  deviation indicator (DI) vector that measures the abrupt changes in  $m$  network variables at time  $t_i$ , i.e.,

$$\psi^j(t_i) = dev(z_{j,l}) \quad (5.6)$$

where  $z_{j,l}$  is assumed to be obtained as shown in the previous section. Note that  $\psi^j(t_i) \in [0, 1]$  corresponds to the deviation in the  $j$ -th performance monitoring variable at time  $t_i$  (henceforth,  $t_i$  will be removed for brevity). A value  $\psi^j=1$  indicates maximum abnormality while a value of  $\psi^i = 0$  reflects normal operation.

Using the DST to classify network anomalies requires establishing an initialization procedure involving the following steps:

- Defining the frame of discernment  $\mathcal{F} = \{F_1, \cdots, F_M\}$  with  $F_i$  describing a certain failure type such as a congested router or an inadequate router buffer size. The set  $\mathcal{F}$  represents a set of mutually exclusive and exhaustive propositions about all possible network failure types.
- Constructing a training set  $\mathbf{D}_N = (\Psi_N, \mathbf{L}_N)$  of  $N$  DI vectors  $\Psi_N = \{\psi_1, \cdots, \psi_N\}$  and  $N$  corresponding labels of failure types  $\mathbf{L}_N = \{L_1, \cdots, L_N\}$ . The  $i$ -th element of  $\Psi_n$ ,  $\psi_i$ , is associated with a corresponding label  $L_i$  where  $L_i \in \{1, \cdots, M\}$  points to the

failure class  $F_{L_i} \in \mathcal{F}$ .

Upon receiving a new DI vector, denoted by  $\hat{\psi}$ , the objective of the DST classification engine is to recognize the most probable root-cause failure  $\hat{F} \in \mathcal{F}$  of  $\hat{\psi}$ . More precisely, the process of diagnosing a possible network anomaly involves the following basic steps,

1. Identifying bodies of evidence,  $\xi_i$ , that can be derived from  $\hat{\psi}$  and the training set  $D_N$ .
2. Computing individual *bpas*,  $m_{\xi_i}(A)$ , which quantify the belief that  $A$  contains the root-cause failure  $\hat{F}$ , where  $A \in 2^{\mathcal{F}}$ .
3. Using the DST combination rule (4.2) to combine the *bpas* obtained from all pieces of evidence regarding the hypothesis that  $\hat{F} \in A$ . The resulting combined *bpa* for each  $A \in 2^{\mathcal{F}}$  can be utilized in (4.4) to find  $Bel(\{F_q\})$  that measures the belief that  $\hat{F} = F_q$ , i.e., the belief that  $F_q$  is the root cause for the detected  $\hat{\psi}$ .
4. Decision making that involves reaching the final conclusion that  $\hat{F} = F_q$  is the most likely failure scenario according to the measured deviation  $\hat{\psi}$ . This step is achieved by selecting  $F_q$  that maximizes the belief  $Bel(\{F_q\})$  according to (4.5).

The main challenging part in implementing the above steps lies in deriving a suitable *bpa*,  $m_{\xi_i}(A)$ . A desirable *bpa* is one that reflects our true belief in the correct classification based on the evidence obtained from each vector in the training set. The difficulty in this regard typically arises from the nature of the training set  $D_N$  and its distribution over the set of failure classes  $\mathcal{F}$ . The rest of this chapter introduces three different approaches aimed at addressing this problem for three types of training sets with different characteristics.

The first approach is based on ideas introduced by Denoeux in [162] and handles training sets characterized by a symmetric distribution over the set of failure classes  $\mathcal{F}$ . This approach is presented in Subsection 5.5.1.

The other two approaches are new ideas presented here to handle imbalanced training sets in which one or few classes dominate the rest of the classes with regard to their representations in the training set. Those techniques are presented in Sections 5.6 and 5.7.

To proceed with presenting these approaches, it will be necessary to construct the evidence  $\xi_i$  that supports the belief that  $\hat{\psi}$  corresponds to  $\hat{F}$ , the root-cause failure under investigation. This evidence may be established based on the relation between the two DI vectors  $\hat{\psi}$  and  $\psi_i$ . An intuitive relation suitable for that purpose is the distance between those two vectors. Throughout the remainder of this chapter, the Euclidean distance is selected as the distance measure between any two vectors and is calculated from

$$d(\hat{\psi}, \psi_i) = \sqrt{\sum_{j=1}^m (\hat{\psi}^j - \psi_i^j)^2} \quad (5.7)$$

As the distance  $d(\hat{\psi}, \psi_i)$  decreases, our belief that  $\hat{\psi}$  belongs to the same failure class as that of  $\psi_i$  must increase and vice versa. Hence, an effective *bpa* should be given by a function that is dependent on  $d(\hat{\psi}, \psi_i)$  and reflects the aforementioned relation.

### 5.5.1 bpa for balanced training sets

In situations where the available training set is not biased towards a particular class, the *bpa* corresponding to an evidence  $\xi_i$  can be defined by

$$m_{\xi_i}(A) = \begin{cases} p_i^{(L_i)} & A = \{F_{L_i}\} \\ 1 - p_i^{(L_i)} & A = \mathcal{F} \\ 0 & A \in 2^{\mathcal{F}} \setminus \{\{F_{L_i}\}, \mathcal{F}\} \end{cases} \quad (5.8)$$

where  $p_i^{(L_i)} = \alpha e^{-\gamma d(\hat{\psi}, \psi_i)^2}$ , such that  $0 < \alpha < 1$  and  $0 < \gamma < 1$  [162].

One can interpret (5.8) as follows: as the values of the deviations of monitoring variables in the two deviation vectors are close, there are higher chances that they will cause the same

network faults. On the other hand, as the distance increases, the training vector is reluctant to provide any valuable information about the cause of the network failure corresponding to  $\hat{\psi}$ .

In this representation of the *bpa*, the focal elements of each evidence  $\xi_i$  are the members of the set  $\{\{F_{L_i}\}, \mathcal{F}\}$ . The amount  $(1 - p_i^{(L_i)})$  represents the degree of uncertainty in addition to the ignorance in labelling the new vector  $\hat{\psi}$  using the knowledge about the label of  $\psi_i$ . Hence,  $(1 - p_i^{(L_i)})$  is equally distributed over all possible failure types represented by the frame of discernment itself.

Denoting the *bpa* resulting from combining  $N_q < N$  evidence,  $\xi_i$ , for which  $L_i = F_q$  by  $m^{(N_q)}$ , i.e.,

$$m^{(N_q)}(A) \equiv \oplus_{i:L_i=F_q} m_{\xi_i}(A) \quad A \in 2^{\mathcal{F}} \quad (5.9)$$

Using (4.2), it can be shown that,  $m^{(N_q)}(A)$  is obtained as

$$m^{(N_q)}(A) = \begin{cases} 1 - \prod_{i=1}^{N_q} (1 - p_i^{(q)}) & A = \{F_q\} \\ \prod_{i=1}^{N_q} (1 - p_i^{(q)}) & A = \mathcal{F} \end{cases} \quad (5.10)$$

and the total *bpa* resulting after combining all  $m^{(N_q)}(A)$ ,  $q = 1, \dots, M$ ,  $m^{(N)}(A)$ , i.e.,

$$m^{(N)}(A) \equiv \oplus_{q=1}^M m^{(N_q)}(A) \quad A \in 2^{\mathcal{F}} \quad (5.11)$$

is given by

$$m^{(N)}(\{F_q\}) = \frac{m^{(N_q)}(\{F_q\}) \prod_{r \neq q} m^{(N_r)}(\mathcal{F})}{K} \quad (5.12)$$

$$m^{(N)}(\mathcal{F}) = \frac{\prod_{q=1}^M m^{(N_q)}(\mathcal{F})}{K} \quad (5.13)$$

where,

$$K = \sum_{q=1}^M \prod_{r \neq q} m^{(N_r)}(\mathcal{F}) + (1 - M) \prod_{q=1}^M m^{(N_q)}(\mathcal{F})$$

It is worth noting here that an important property of the utilized rule is that it is commutative and associative. This is desirable because evidence aggregation to reach one conclusive decision given all collected evidence should be independent of the order of evidence gathering. The rule focuses only on those propositions that both bodies of evidence support.

### 5.5.2 Anomaly diagnosis

To determine the root-cause failure corresponding to the vector  $\hat{\psi}$ , the combined *bpa* obtained by (5.12- 5.13) is used to calculate the total belief associated with every failure. The singleton elements of the calculated belief function are the candidate classes of failure for  $\hat{\psi}$ .

Using (4.4), total belief associated with every failure class  $F_q$  is shown to be as follows

$$Bel(\{F_q\}) = m^{(N_q)}(\{F_q\}), \quad \forall q \in \{1, \dots, M\} \quad (5.14)$$

The class  $\hat{F}$  that maximize the belief value is selected as the most probable root-cause corresponding to the measured DI vector  $\hat{\psi}$  such that

$$\hat{F} = \arg_{F_i \in \mathcal{F}}(\max(Bel(\{F_i\}))) \quad (5.15)$$

## 5.6 bpa for imbalanced training sets

In the pervious section, the evidence extracted from knowledge about the label  $L_i$  of each vector  $\psi_i$  in the training set only influenced our belief about the membership of the new vector  $\hat{\psi}$  to the class  $F_{L_i}$ . However, it did not add to our belief or disbelief about its membership to any other class. In some situations, particularly for the case of failures resulting from network intrusions and novel attacks, an imbalanced training set, where the occurrences of one or more of the anomalies are dominant, degrades the accuracy in the classification of such minority anomalies. This *minority problem* is well known in classification and is the subject of a considerable body

of research in literature [163]. However, to the best of the authors' knowledge, it is still an open issue for the case of DST-based classifiers.

In this section, we develop a second *bpa* particularly suited for imbalanced and small training sets. An optimal *bpa* in such cases is one in which every training pair can strongly direct the outcome of the decision engine to the correct classification. To achieve this,  $N - k$  elements of the training set is used to construct a signature set  $\Theta = [\theta_1, \dots, \theta_M]$ , where  $\theta_i$  represents an estimate of the center mean of each of the failure class  $F_i$  under investigation. Consequently, the remaining  $k$  DI vectors are used to construct pieces of evidence to guide the classification decision. Since the successive applications of DST combination gradually accumulates evidence towards the correct class, these estimates of classes means are not necessary required to be accurate. Rather, they are mostly required to represent rough estimates of each class vectors.

Using this signature set, one can create a new evidence  $\xi_i$  for each training pair  $(\psi_i, L_i)$  with respect to a new DI vector  $\hat{\psi}$  as follows

$$m_{\zeta_i}(\{F_q\}) \equiv p_i^{(q)} = \frac{e^{-d(\theta_q - \psi_i) - d(\hat{\psi} - \psi_i)^2}}{\sum_{j=1}^M e^{-d(\theta_j - \psi_i) - d(\hat{\psi} - \psi_i)^2}} \quad (5.16)$$

$\forall q \in \{1, \dots, M\}$ . In this representation of the *bpa*, the focal elements of each evidence  $\xi_i$  are the singletons  $\{F_1\}, \{F_2\}, \dots, \{F_M\}$ . One important feature of this *bpa* distribution scheme is that in contrast to the first distribution (5.8), every DI vector in the training set is likely to enforce the belief in the correct failure type. This idea is illustrated by two simplified classification examples shown in Figure 5.2. Figure 5.2.a depicts the regions of three equally distanced classes,  $F_1, F_2$ , and  $F_3$ , in a one dimensional space, with estimated means  $\theta_1, \theta_2$  and  $\theta_3$ , respectively. Both vectors  $\psi_i$  and  $\psi_j$  assign the highest mass,  $p_i^{(1)} = e^{-(a_i - b_i)^2} / K$ , and  $p_j^{(1)} = e^{-(a_j - b_j)^2} / K$ , respectively, to the correct failure class  $F_1$  for the new DI vector  $\hat{\psi}$  regardless of their class membership. The constant  $K$  is the normalization factor. It can be shown that, in this particular case,  $p_i^{(1)} = p_j^{(1)} = e^{-c^2} / K$ , where  $c$  is the distance between  $\hat{\psi}$  and  $\theta_1$ .

It is important here to highlight the performance of the  $bpa$  defined in (5.16) under the special case depicted in Figure 5.2.(b). In this case, both the center means  $\theta_2$  and  $\theta_3$  of two classes  $F_2$  and  $F_3$ , respectively, are equally distanced from the center mean  $\theta_1$  of a third one  $F_1$  in a two dimensional space. It is clear here that, in this case, members of  $F_2$  and  $F_3$ ,  $\psi_i$  and  $\psi_j$ , respectively, contribute equally to the belief that  $\hat{\psi}$  is a member of the two other classes ( $F_1, F_3$  for  $\psi_i$  and  $F_1, F_2$  for  $\psi_j$ ), i.e.,  $p_i^{(1)} \simeq p_i^{(3)}$  and  $p_j^{(1)} \simeq p_j^{(2)}$ . However, they do not contradict the evidence from a third vector  $\psi_k$ , shown on the schematic diagram, where it is easily seen that  $p_k^{(1)} \gg p_k^{(2)}$  and  $p_k^{(1)} \gg p_k^{(3)}$ , and hence  $\psi_k$  provides a sufficient guide pointing to the correct class  $F_1$  of which  $\hat{\psi}$  is a member. Once different pieces of evidence are constructed from the

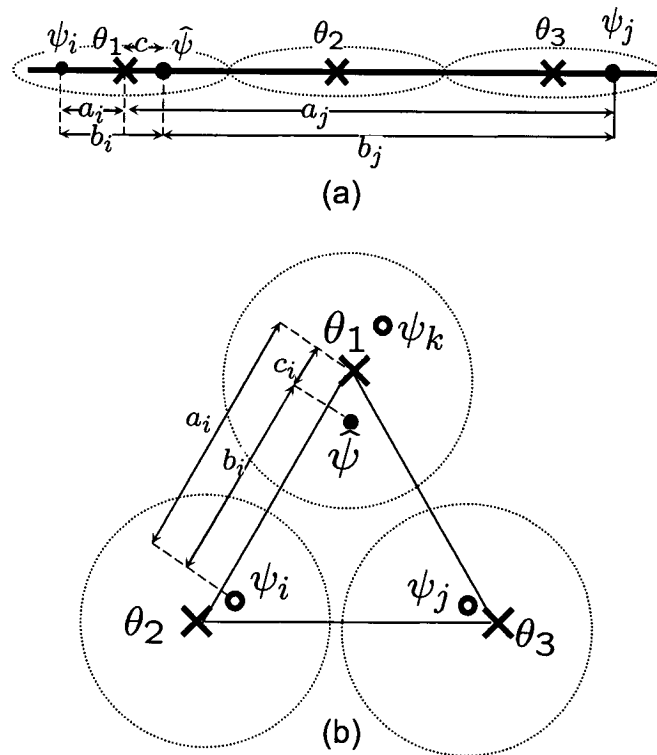


Figure 5.2: Illustration of the performance of the signature-based scheme in two special cases.

training set, they are combined using the Dempster combination rule (4.2). It can be shown that the resulting *bpa* after combining all evidence from  $k$  training vectors  $\{\psi_1, \dots, \psi_k\}$

$$m^{(k)}(\{F_q\}) = \frac{\prod_{i=1}^k (p_i^{(q)})}{K} \quad \forall q \in \{1, \dots, M\}, \quad (5.17)$$

$$K = \sum_{q=1}^M m^k(\{F_q\})$$

Again, these values are used to calculate the belief in each class according to (4.4). The failure type that maximizes the belief function is then selected as the root-cause failure corresponding to  $\hat{\psi}$ .

## 5.7 Adjustable *bpa* for imbalanced training sets

This section presents a third *bpa* distribution that suits cases where it is desired to inject external knowledge into the decision engine. This case is particularly suitable for anomalies resulting from infrequent network faults or in order to distinguish between two or more anomalies that exhibit very similar traffic characteristics. This external knowledge can be obtained, for example, through experts' knowledge or by using a set of a-priori known heuristics.

An example of an expert knowledge is that given a new DI vector  $\hat{\psi}$  with a particularly high value for the  $j$ -th monitoring variable, the expert may be sure that it does not belong to a certain class  $F_q$  and assigns a zero mass value,  $(m_{\xi_i}(\{F_q\}) = 0)$ , to this class in all constructed pieces of evidence  $\xi_i$  during the classification process of that vector. An example of a heuristic rule is a one that assigns a high mass to one of the failure classes representing denial of service (DoS) attacks and flash crowd events according to the value of a particular monitoring variable in the DI  $\psi_i$ .

We demonstrate the advantage of constructing knowledge-based *bpas* via a particular example in which  $p_i^{(L_i)}$  is assumed to be specified by some sort of an expert's knowledge or based on the distance measure as in (5.8). Since  $p_i^{(L_i)}$  represents the belief that  $F_{L_i}$  is the actual failure

that prompted  $\hat{\psi}$  (based on the evidence,  $\xi_i$ , that is deduced from  $\psi_i$ ),  $(1 - p_i^{(L_i)})$  represents the amount of uncertainty in addition to the ignorance in the system.

The key idea is that, in contrast to the *pba* described by (5.8) where the remaining amount of uncertainty and ignorance was assigned to the whole bulk of the frame of discernment  $\mathcal{F}$ ,  $(1 - p_i^{(L_i)})$  here will be distributed over consistent focal subsets in  $2^{\mathcal{F}}$ , where each subset has at least one element that is common to all other subsets with nonzero mass value. As will be confirmed through theoretical derivations and experimental results in Section 5.8, this type of *pba* exhibits a significantly better performance compared to the one given in (5.8) in the case of a biased training set.

To proceed with deriving this distribution, we will need first to identify a set  $\mathbf{D}_k \subset \mathbf{D}_N$  of  $k$ -nearest neighbors of the new DI vector,  $\hat{\psi}$ , based on their Euclidian distances to that vector.

Without loss of generality, we assume that the ignorance of the system is negligible, and hence,  $(1 - p_i^{(L_i)})$  represents only the uncertainty component in the decision making process. The portion  $(1 - p_i^{(L_i)})$  will be divided over the entire range of a subset of failure classes. This subset, denoted by  $\hat{\mathcal{F}} \subset \mathcal{F}$ , is derived from the set  $\mathbf{D}_k$  and is defined by  $\hat{\mathcal{F}} = \{F_1, \dots, F_{\hat{M}}\}$ . Hence,  $\hat{\mathcal{F}}$  groups the  $\hat{M}$  most probable failures that caused  $\hat{\psi}$ , and therefore, it should be advantageous to assign various degrees of confidence to its subsets. We propose the following assignment,

$$\begin{aligned}
 m_{\xi_i: r \neq L_i}(\{F_{L_i}, F_r\}) &= \\
 2m_{\xi_i: r \neq j \neq L_i}(\{F_{L_i}, F_r, F_j\}) &= \\
 \dots & \\
 = (M - 1)m_{\xi_i}(\{F_1, F_2, \dots, F_{L_i}, \dots, F_{\hat{M}}\}) &
 \end{aligned} \tag{5.18}$$

In other words, (5.18) states that the belief in that the candidate failure type is either  $F_r$  or  $F_{L_i}$ ,  $r \neq L_i$ , is twice stronger than the belief that the candidate failure type is in  $\{F_r, F_{L_i}, F_j\}$ ,  $r \neq j \neq L_i$ , and is three times stronger than the belief that it is in  $\{F_r, F_{L_i}, F_j, F_k\}$ ,  $r \neq j \neq k$ .

$L_i \neq k$ , and so forth.

It should be mentioned that the distribution scheme described by (5.18) represents only one approach in which the belief assignment is performed automatically. Nonetheless, it is possible to manually tailor this distribution using some heuristically available information in situations where obvious gains are anticipated from incorporating external knowledge. In fact, the distribution in (5.18) in addition to showing improved classification accuracy, offers a closed-form analytical expression for the *bpa* that can be used to derive a lower bound on the expected accuracy as will be demonstrated in the following section. The closed-form is obtained by noting that

$$\begin{aligned}
 (1 - p_i^{(L_i)}) &= \sum_{j=1, j \neq L_i}^{\hat{M}} m_{\xi_i: j \neq L_i}(\{F_{L_i}, F_j\}) \\
 + \sum_{j,k=1, j \neq k \neq L_i}^{\hat{M}} m_{j \neq k \neq L_i}(\{F_{L_i}, F_j, F_k\}) &+ \cdots + \\
 m_{\xi_i}(\{F_1, F_2, \dots, F_{L_i}, \dots, F_{\hat{M}}\}) & \quad (5.19)
 \end{aligned}$$

In general, using (5.18) and (5.19), it can be deduced that  $m(A)$ ,  $A \in 2^{\hat{F}}$  is given by,

$$\begin{aligned}
 m_{\xi_i}(A) &= \frac{(1 - p_i^{(L_i)})(\hat{M} - h - 1)}{\sum_{j=1}^{\hat{M}-1} \binom{\hat{M}-1}{j} (\hat{M} - j)} \quad (5.20) \\
 A \in 2^{\hat{F}}, F_{L_i} \in A, h &= |A|
 \end{aligned}$$

Different pieces of evidence extracted from the  $k$  neighbors of  $\hat{\psi}$  are then combined and the most probable root-cause failure class is selected as the one that maximizes the belief value.

It is worth noting here that the Dempster rule is difficult to apply when kernels have non-singleton focal elements. However, several approaches have been proposed in literature, e.g., [138] to reduce this complexity.

## 5.8 Performance Analysis of the NPA

### 5.8.1 Performance of Abrupt Change detection scheme

This section summarizes performance evaluation results of the proposed scheme for abrupt change detection. Data for this study consists of SNMP measurements collected at a server at the university of Ottawa for the duration of one month. Five MIB counts were chosen for investigation and are obtained from MIB-II [95], namely, the quantity of input and output of octets (*ifInOctets* and *ifOutOctets*, respectively), the number of IP datagrams received from all interfaces (*ipInReceives*), the number of datagrams correctly delivered to the higher layers as this node was their final destination (*ipInDelivers*), and finally, the number of datagrams passed on from the higher layers of the node to be forwarded by the ip layer (*ipOutRequest*). The diagnosis was focused on detecting four types of network faults  $F_1$ ,  $F_2$ ,  $F_3$  and  $F_4$ , respectively referring to a router software bugs, network congestion, crashed server and user misuse.

Each of the five MIB counts is sampled with a frequency of one sample every 15 sec.. The window size for averaging the deviation measures,  $T$ , was selected to include 20 samples. In both cycles of signal modelling, a first order AR was applied and the model's parameters were estimated using the least square method.

Figures 5.3, 5.4, 5.5, 5.6 and 5.7 show a representative sample of the increments measured for the MIB variables under investigation during a period of approximately 4 hours. The asterisks in the top panel of the figures indicate the deviation measure  $dev(z_t)$ . A value of 1 indicates a maximum deviation from the normal behavior while a 0 value refers to a healthy router.

As shown, of the five variables, *ipInReceives* and *ipInDelivers* tend to show bursts in their deviations. Hence, they are borne to trigger more false alarms as compared to the other variables. For normal traffic, all other MIB variables maintained a *near zero* deviation level during

normal network operations.

An *alarm* is raised by a variable when the deviation measure for the variable at a point in time exceeds a certain threshold (this threshold is set to 0.5 for all variables in these set of experiments). Table 5.1 provides the number of alarms triggered by each variable during one month of observation and the average alarm hourly rate.

A *detected fault* is the result of one or more raised alarms during  $p = 30$  min., where  $p$  is referred to as the estimated average fault period and was set experimentally. A *false negative* fault results from a network fault that was not detected by an monitoring variable through a raised alarm during  $p$ . A *false positive* fault, consequently, results when an actual known/unknown fault occurs at the router during  $p$ . The detected, false negative and false positive faults ratios are the ratios between the number of detected, false negative and false positive alarms, respectively, to the total number of raised alarms. Table 5.2 depicts these three ratios accumulated by the week over one month of testing. The low ratio of false positives is attributed to the two cycle of AR modelling which further dilutes the effects of isolated changes in each variable. It was also noticed that alarms raised by the variable *ipInReceives* where present in almost 85% of the false positive alarms. This can be interpreted as the high sensitivity of the variable to any changes in the underlying router's health.

In the following sections, given the measured deviations for each of the MIB variables, we investigate the correlation among these variables through the DST decision engine.

### 5.8.2 Accuracy estimation of different *bpas*

Before qualitatively analyzing the performance of the second step in the proposed scheme of anomaly classification, we first derive a theoretical estimation of the classification accuracy for each of the three proposed *bpas*.

For briefly, henceforth, we will refer to the first *pba* (5.8) as a *pba* with no distrusting of

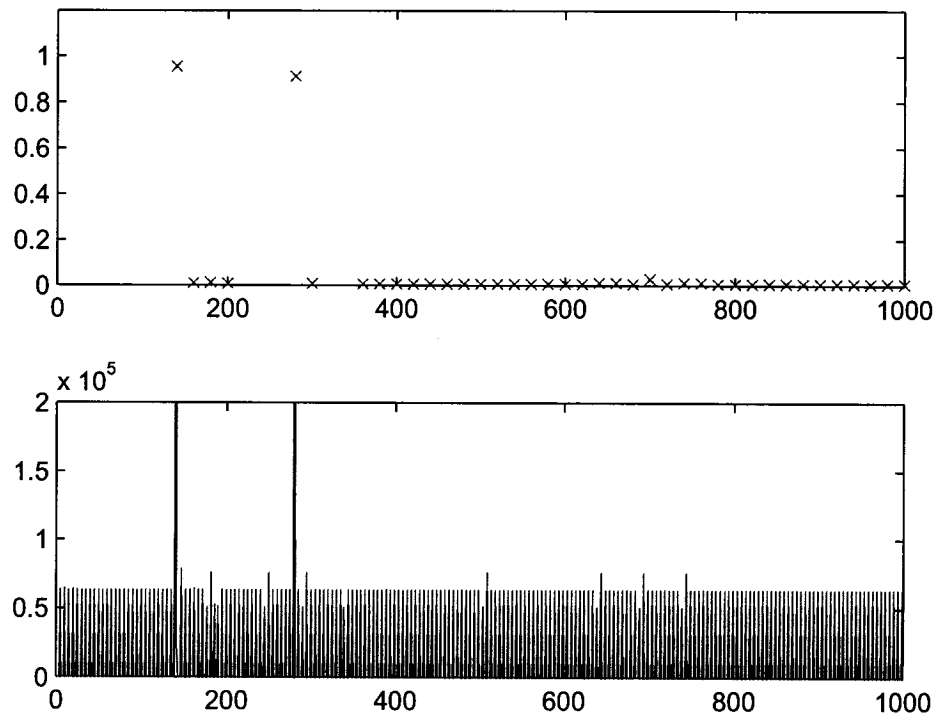


Figure 5.3: Increments of *ifInOctets* (upper panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments).

Table 5.1: Alarm rate over a one month period.

Variable	number of alarms (per month)	alarm hourly rate
<i>ifInOctets</i>	380	0.53
<i>ifOutOctets</i>	442	0.61
<i>ipInReceives</i>	3163	4.39
<i>ipInDelivers</i>	2163	3.1
<i>ipOutRequest</i>	1591	2.21

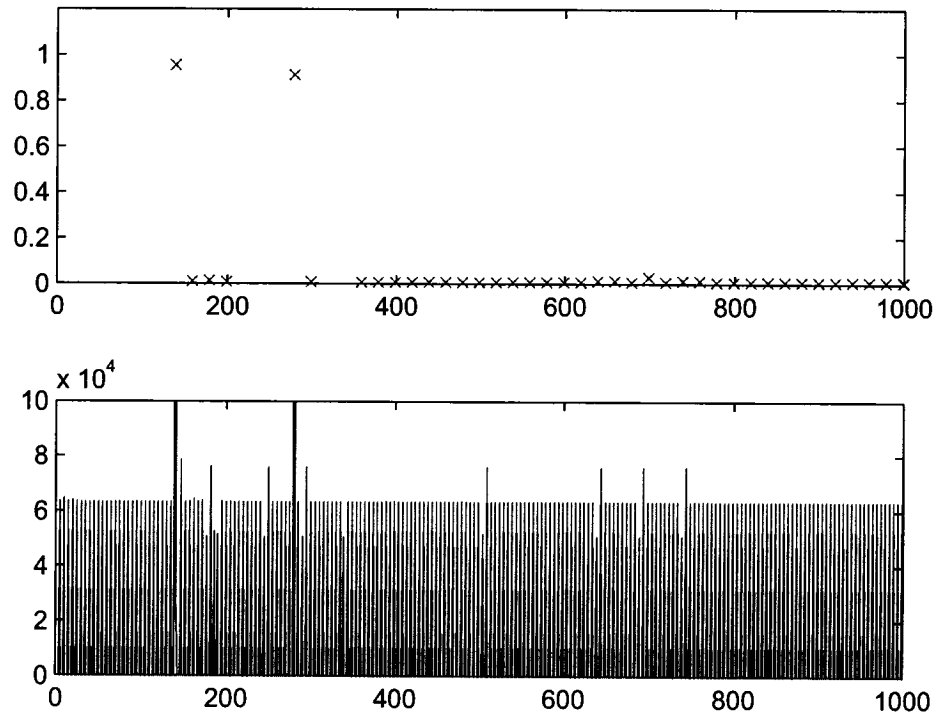


Figure 5.4: Increments of *ifOutOctets* (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments).

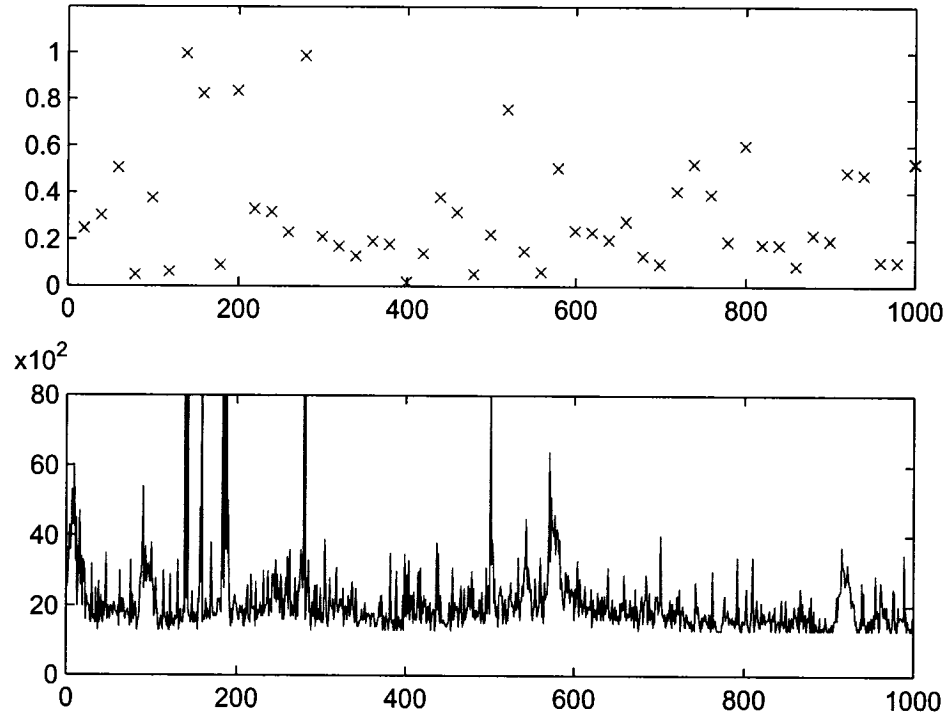


Figure 5.5: Increments of *ipInReceives* (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments).

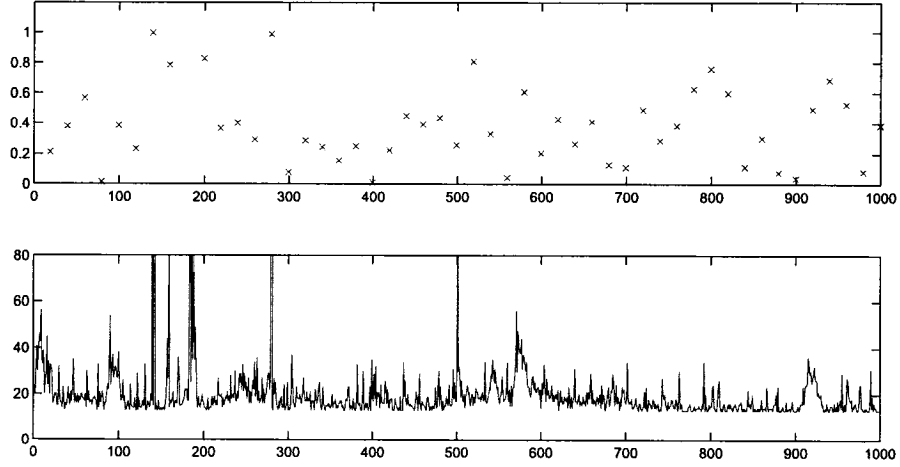


Figure 5.6: Increments of  $iplnDelivers$  (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments).

Table 5.2: Anomaly detection accuracy.

	week#1	week#2	week#3	week#4
detected faults	0.976	0.962	0.96	0.96
False positive	0.016	0.016	0.016	0.018
False negative	0.021	0.03	0.03	0.03

uncertainty (NDU) since the remaining uncertainty,  $(1 - p_i^{(L_i)})$ , is totally assigned to  $\mathcal{F}$ . We will also refer to the  $bpa$ s defined in (5.16) and (5.20) as a *signature-based* (SB)  $bpa$  and a  $bpa$  with *gradual distribution of uncertainty* (GDU), respectively.

It is intuitive that the degree of accuracy in deciding the correct network root-cause failure  $\hat{F}$  corresponding to a new DI vector  $\hat{\psi}$  is dependent on three components. The first component is the accuracy in representing degrees of confidence in each candidate class given a training vector. The second relates to the coverage space of the training set  $D_N$ . Finally, the third

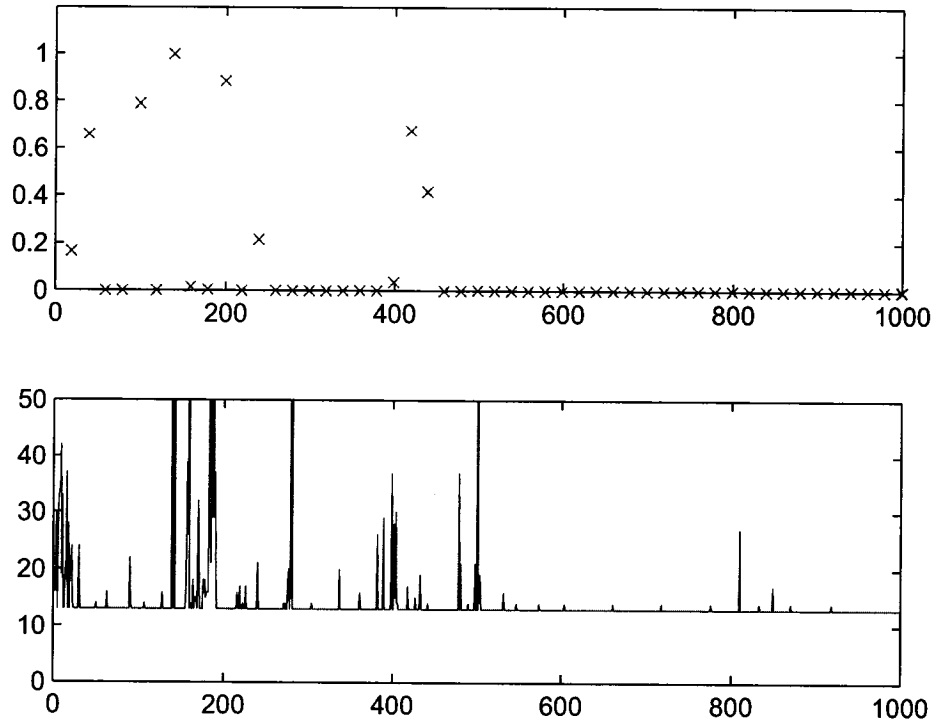


Figure 5.7: Increments of *ipOutRequest* (lower panel) and corresponding deviation level (upper panel) (x-axis represents time in 15 sec. increments).

component has to do with the trend in the behavior of the underlying network in response to the same failure type at different points in time. Without loss of generality, by assuming that monitoring variables are highly responsive to differences between different root-causes, effects of the latter component can be neglected. In the following, we analyze the effects of the two remaining components.

### Accuracy estimation based on a single training vector

In this section, we derive a theoretical bound to the accuracy achieved in representing degrees of confidence for the correct classification of  $\hat{\psi}$  for each of the proposed *bpas* given a single training pair  $(\psi_i, L_i)$ .

We will use the relative atomicity  $a(A, B)$  of a focal element  $A \in 2^{\mathcal{F}}$  with respect to an arbitrary set  $B \in 2^{\mathcal{F}}$  as defined in 4.22 and restated here for convenience [136]

$$a(A, B) \equiv \frac{|A \cap B|}{|B|}, \quad A, B \in 2^{\mathcal{F}}, \quad B \neq \phi, \quad (5.21)$$

Now, one can use the *probability expectation* [136], of the correct class  $\hat{F}$ ,  $E(\{\hat{F}\})$ , to measure the confidence in deciding  $\hat{F}$  measured by utilizing  $d(\psi_i, \hat{\psi})$ .

The function  $E(A)$  can be obtained as a weighted sum of beliefs  $m(B)$  with  $a(A, B)$  as the weighting coefficients [136], i.e.,

$$E(A) = \sum_{B: A \in B, B \in 2^{\mathcal{F}}} m(B) a(A, B) \quad (5.22)$$

As indicated by (5.22), the probability expectation is a function of the assigned mass values and hence is highly dependent on the representation of *bpas*. In the following,  $E(\{\hat{F}\})$  is calculated for each of the proposed *bpa* distributions given a training pair  $(\psi_i, L_i)$ .

In the case of the NDU scheme, substituting (5.8) in (5.22), we get,

$$E_{\text{NDU}}(\{\hat{F}\}) = \begin{cases} p_i^{(L_i)} + \frac{(1-p_i^{(L_i)})}{M} & \hat{F} = F_{L_i} \\ \frac{(1-p_i^{(L_i)})}{M} & \hat{F} \neq F_{L_i} \end{cases} \quad (5.23)$$

For the SB scheme, since the focal sets of the  $pba$  defined in (5.16) are all singletons, i.e.,  $A \cap B = \phi, \forall m_{\xi_i}(A), m_{\xi_i}(B) > 0$ , then we have

$$E_{\text{SB}}(\{\hat{F}\}) = m_{\xi_i}(\{\hat{F}\}) \quad (5.24)$$

where  $m_{\xi_i}$  is defined in (5.16).

For illustrative purposes, we further focus on the special case where the means of consecutive classes in  $\mathcal{F}$  are equally distanced, such that  $d$  represents the distance between the means of  $F_i$  and  $F_{i+1}$ . In this special case, it can be shown that

$$E_{\text{SB}}(\{\hat{F}\}) = \frac{e^{-c^2}}{K} \quad (5.25)$$

where  $c = d(\hat{\theta}, \psi_i) - d(\hat{\psi}, \psi_i)$ , the normalization constant  $K = \sum_{i=1}^M e^{-(-a+(i-1)d)^2}$  and  $\hat{\theta}$  is the mean of  $\hat{F}$ .

Finally, in case of (5.20),  $E_{\text{GDU}}(\{F_i\})$  can be obtained as follows,

$$E_{\text{GDU}}(\{\hat{F}\}) = p_i^{(L_i)} + \left(1 - p_i^{(L_i)}\right) \sum_{i=1}^{\hat{M}-1} \left( \frac{\binom{\hat{M}-1}{i} (\hat{M}-i)}{(i+1) \sum_{j=1}^{\hat{M}-1} \binom{\hat{M}-1}{j} (\hat{M}-j)} \right) \quad (5.26)$$

for  $\hat{F} = F_{L_i}$  and

$$E_{\text{GDU}}(\{\hat{F}\}) = \left(1 - p_i^{(L_i)}\right) \sum_{i=0}^{\hat{M}-2} \left( \frac{\binom{\hat{M}-2}{i} (M-i-1)}{(i+2) \sum_{j=1}^{\hat{M}-1} \binom{\hat{M}-1}{j} (\hat{M}-j)} \right) \quad (5.27)$$

for  $\hat{F} \neq F_{L_i}$ . Figure 5.8, shows a plot for the probability expectation of  $\hat{F}$ ,  $E(\{\hat{F}\})$ , versus the size of  $\mathcal{F}$ ,  $M$ , for the three *bpa* methodologies using two different training vectors  $\psi_i$  and  $\psi_j$ . The first vector,  $\psi_i$  belongs to the same class of  $\hat{\psi}$  with  $d(\psi_i, \hat{\psi}) = 0.7$  and the second vector belongs to a different class with  $d(\psi_j, \hat{\psi}) = 1.5$  in a three dimensional ( $m = 3$ ) space.

Figure 5.8 shows that, while the GDU method assigns the highest belief to  $\hat{F}$  in the case of  $\psi_i$ , the SB method, shows a lower but a relatively constant probability expectation value for both training vectors, regardless of their class membership.

#### Accuracy estimation for a training set of size $N$

Having derived an accuracy estimate for classification using a single training vector, we proceed to analyze the effects of the distribution of the training set  $\mathbf{D}_N$  on the accuracy of classifying  $\hat{F}$ .

In the case of the SB scheme, the total probability expectation of  $\hat{F}$  after combining evidence from  $N$  vectors,  $E_{\text{SB}}^{(N)}(\{\hat{F}\})$ , can be calculated by substituting for  $m_{\xi_i}$  in (5.22) with (5.17) and noting that  $a(\{\hat{F}\}, \{\hat{F}\}) = 1$ ,  $a(\{\hat{F}\}, A) = 0$ ,  $\forall A \neq \{\hat{F}\}$ , to obtain,

$$E_{\text{SB}}^{(N)}(\{\hat{F}\}) = m^{(N)}(\{\hat{F}\}) \quad (5.28)$$

where  $m^{(N)}$  is obtained from (5.17) by substituting for  $k$  with  $N$ . From (5.28), it is clear that each vector  $\psi_i$  contributes to the overall accuracy of the scheme. Hence, this particular

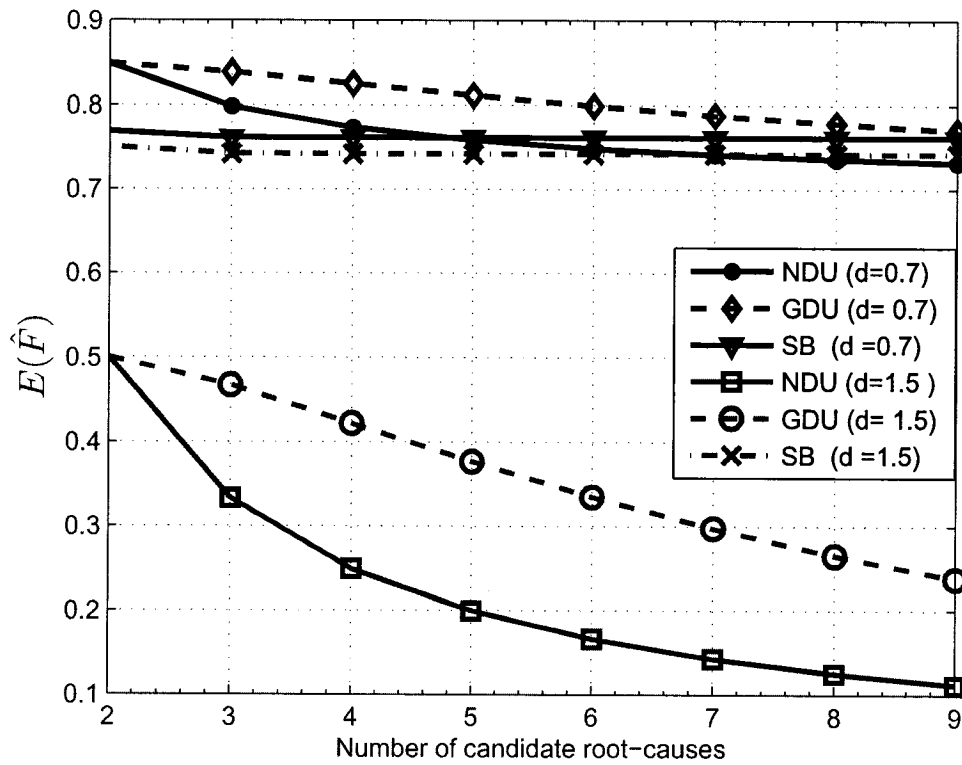


Figure 5.8: Probability Expectation of  $\hat{F}$  for different *bpas*.

distribution is expected to achieve the same high degree of accuracy in the case of a biased set.

In order to derive a similar measure for the NDU scheme, vectors in  $\mathbf{D}_N$  are divided into two sets, the set of  $\hat{N} < N$  vectors with labels pointing to  $\hat{F}$  and the set of  $N - \hat{N}$  vectors representing  $\mathcal{F} \setminus \{\hat{F}\}$ . It can be shown that

$$E_{\text{NDU}}^{(N)}(\{\hat{F}\}) = m^{(N)}(\{\hat{F}\}) + \frac{m^{(N)}(\{\mathcal{F}\})}{M} \quad (5.29)$$

where  $m^{(N)}(\{\hat{F}\})$  is obtained by substituting in (5.12) for  $F_q$  and  $N_q$  with  $\hat{F}$  and  $\hat{N}$ , respectively, and  $\frac{m^{(N)}(\{\mathcal{F}\})}{M}$  is defined in (5.13). A quick analysis of  $m^{(N)}(\{\hat{F}\})$  shows that the first term in (5.29) is highly dependent on  $\hat{N}$ . A smaller value for this term (resulting from a small  $\hat{N}$ ) is reflected by a large increase in the nominator of the second term which in turn is equally distributed over the entire range of classes in  $\mathcal{F}$ . This interprets the decrease in the accuracy of the scheme for biased sets.

A closed-form for  $E_{\text{GDU}}^{(N)}(\{\hat{F}\})$  for the GDU distribution can not be derived due to the existence of non-zero mass values of non-singletons. However, it is intuitive that this value lies somewhere between  $E_{\text{SB}}^{(N)}(\{\hat{F}\})$  and  $E_{\text{GDU}}^{(N)}(\{\hat{F}\})$ . This fact was illustrated in Figure 5.8 as  $E_{\text{GDU}}(\{\hat{F}\}) > E_{\text{NDU}}(\{\hat{F}\})$  for the two vectors.

### 5.8.3 Experimental results

In order to experimentally measure the performance of the classification scheme, the root-causes of the previously measured deviations resulting from monitoring the University of Ottawa server for a period of one month are manually identified and recorded. The frame of discernment  $\mathcal{F}$  was set to the previously defined failure types  $F_1, F_2, F_3$  and  $F_4$ . A subset of this data was used to train the DST-engine while the remaining data was used for testing its performance. In the following, experimental performance is measured in terms of classification accuracies and is compared to that of the majority-voting (MV) scheme [164].

The different *bpas* are defined by setting  $p_i^{(L_i)} = \alpha e^{-\gamma d(\hat{\psi}, \psi_i)^2}$ , where  $\alpha$  and  $\gamma$  are set experimentally to 0.95 and 0.9, respectively.

The experiments described in this section were carried out by randomly selecting a training set, with prescribed criteria, from a total of 45 DI vectors. Each experiment was then repeated 10 times where results obtained were averaged and reported with each experiment.

In the first experiment, 3 representative vectors of each class were randomly selected to construct a balanced training set of size  $N = 12$  (referred to as Training Set #1). These 12 vectors were also used to construct the signature set used in the SB scheme. Figure 5.9 depicts the classification of 45 vectors into the four failure classes with respect to the three variables *ipInReceives*, *ipInDelivers* and *ipOutRequest*, represented by the x, y, and z axes, respectively. As shown in this figure,  $F_1$  represents a failure class accompanied by higher deviations of *ipInReceives*, while small changes in the same variable were indications of  $F_2$  failures.  $F_3$  represented the most frequent type of failures and was characterized by changes in the three MIB variables. The four panels in Figure 5.9 present the classification results obtained from using the three *bpas* approaches in addition to the MV scheme with  $k = 12$  neighbors from the 45 vectors. Incorrectly classified vectors are represented by dark markers. The first column in Table 5.3 shows the ratios of the correctly to incorrectly classified DI vectors for each of the four methods based on the outcome of the 10 runs.

In the second experiment, effects of imbalanced training sets on the performance of the different *bpas* were investigated by biasing the training set towards  $F_3$ . This was achieved by randomly selecting 3, 2, 4 and 2 vectors for  $F_1$ ,  $F_2$ ,  $F_3$  and  $F_4$ , respectively, to construct the second training set (referred to as Training Set #2), which has also been used to construct the signature set for the SB scheme. Figure 5.10 and the second column in Table 5.3 illustrate the performance (averaged over 10 runs) of each scheme. It is important here to stress that GDU and SB approaches demonstrated a considerably better performance than the MV and NDU schemes. However, the *SB* scheme exhibits a slightly less accuracy than GDU. This fact can

be attributed to the natural high variance in the  $F_3$  and  $F_4$  classes, whose center means were not captured accurately by relying only on 4 and 2 vectors, respectively, in the training set.

To further illustrate this issue, we used the available 45 vectors to calculate the center means of each class and construct the signature set. The above experiment was then repeated using a third data set (denoted by Training Set #3) formed by randomly eliminating 1 vector from Training Set #2, where the eliminated vector was selected from those representing classes  $F_1$ ,  $F_2$  and  $F_4$ . The averaged accuracy levels for this experiment is shown in the third column of Table 5.3 and a sample of the obtained results is illustrated in Figure 5.11. As those demonstrate, showed higher accuracy levels when using the SB method in this case.

#### 5.8.4 Computational Complexity

One critical issue for any diagnosis technique is its computational requirements since it will compete over the available router resources with serviced traffic packets. In the following, we provide a rough calculation of the computational complexity required by the proposed scheme.

The first step of abrupt change detection involves a bi-cycle application of AR modelling. This step requires the estimation of the AR model parameters via adaptive least squares algorithms, with computational and storage complexity of  $\mathcal{O}(k)$ , where  $k$  is the number of unknown parameters to be estimated [161]. Throughout the afore-described experimentations, an AR model of the first order ( $k=1$ ) was employed for each MIB variable and therefore this step required  $\mathcal{O}(s)$  complexity, where  $s$  is the number of variables under investigation.

The second step requires calculating the *bpas* whose complexity scales linearly with respect to the size of the training set  $N$ , i.e.,  $\mathcal{O}(N)$ .

In the final step, the Dempster rule of combination (4.2) is applied. It is known that this process is #P-Complete but is drastically simplified in cases where the focal elements are singletons. In the case of the SB and NDU schemes, the complexity of applying Dempster com-

ination rule is  $\mathcal{O}(|\mathcal{F}|)$ , where  $|\mathcal{F}|$  is the number of all possible failure types. For the GDU scheme, this complexity is  $\mathcal{O}(|\mathcal{F}| \log |\mathcal{F}|)$ .

The above analysis shows that the upper bound for the total complexity in the detection and diagnosis of the anomaly is  $\mathcal{O}(s) + \mathcal{O}(N) + \mathcal{O}(|\mathcal{F}| \log |\mathcal{F}|)$ .

## 5.9 Summary

In this chapter, we developed a novel scheme for the diagnosis of network anomalies at run-time. A bi-cycle of auto regressive models was applied to monitoring variables to detect abrupt changes. The detected anomalies were then fed to an anomaly classification engine in order to identify the root-cause of the detected anomaly. Two new approaches to assign mass belief values to the classes under investigation were presented to handle imbalanced training sets. We derived theoretical bounds for the accuracies of the proposed classification schemes. Experimental results demonstrated the effectiveness of the proposed diagnosis scheme.

Table 5.3: Ratio of correctly to incorrectly classified DI vectors.

Method	Training set#1	Training set#2	Training set#3
Majority Voting (MV)	0.85	0.67	0.71
No distribution of uncertainty (NDU)	0.98	0.88	0.8
Signature based distribution (SB)	0.95	0.91	0.987
Gradual distribution of uncertainty (GDU)	0.97	0.96	0.90

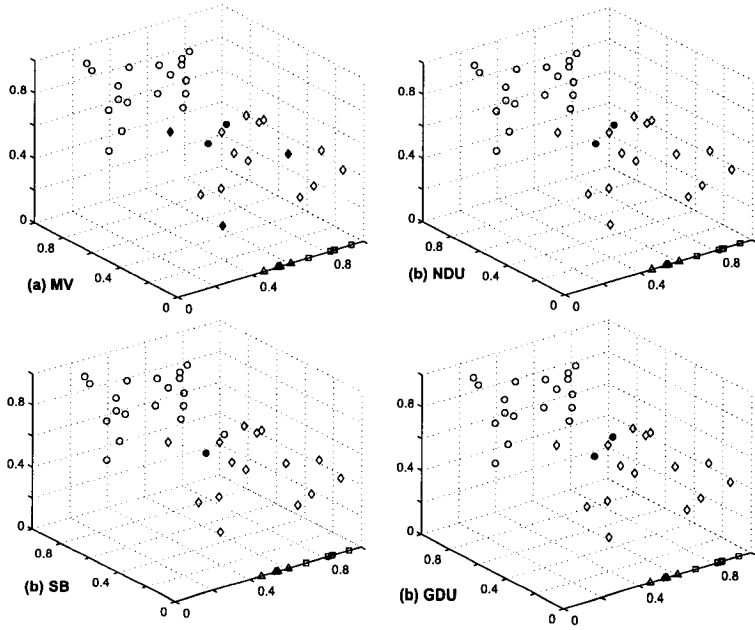


Figure 5.9: Classification of root-cause failures using training set #1 ( $F_1$  as squares,  $F_2$  as triangles,  $F_3$  as circles,  $F_4$  as diamonds, filled shapes indicate a misclassification).

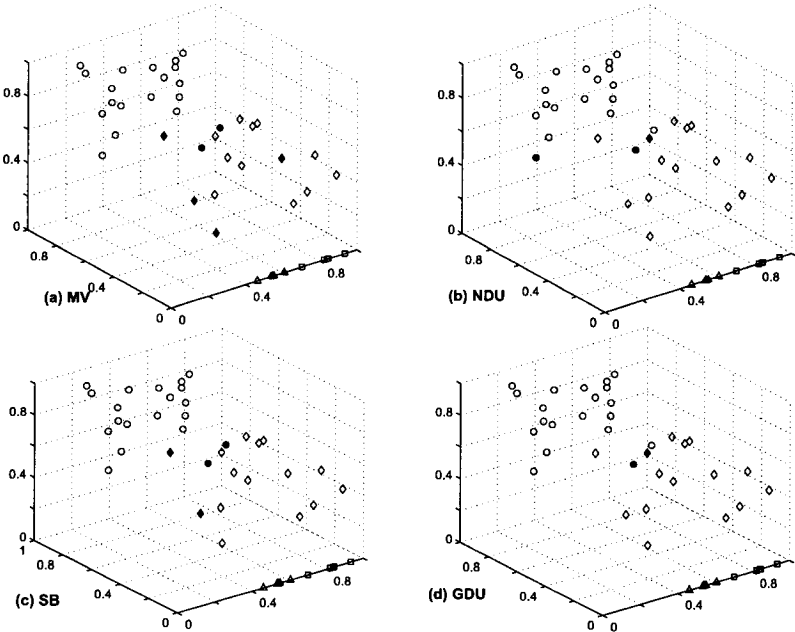


Figure 5.10: Classification of root-cause failures using training set #2 ( $F_1$  as squares,  $F_2$  as triangles,  $F_3$  as circles,  $F_4$  as diamonds, filled shapes indicate a misclassification).

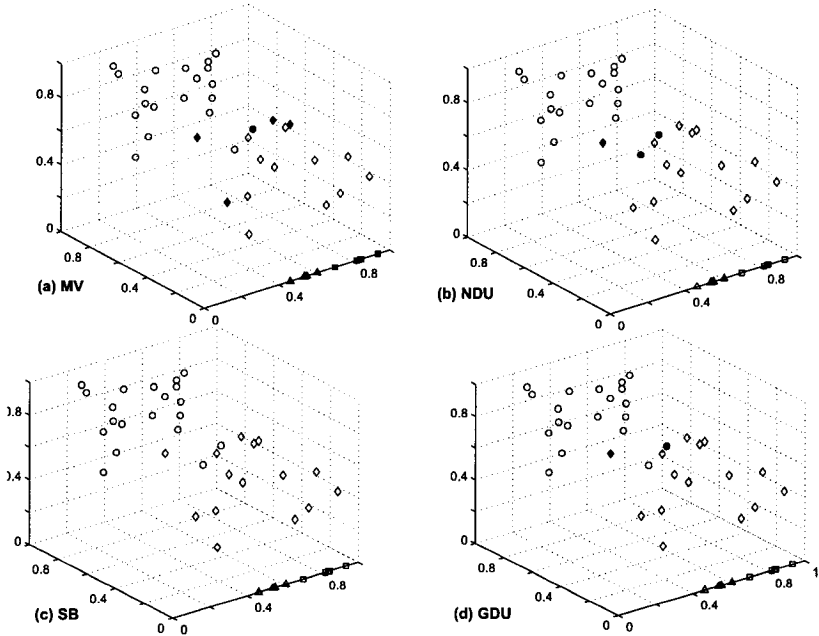


Figure 5.11: Classification of root-cause failures using training set #3 ( $F_1$  as squares,  $F_2$  as triangles,  $F_3$  as circles,  $F_4$  as diamonds, filled shapes indicate a misclassification).

## **Chapter 6**

# **Policy-Based Adaptive Management**

This chapter presents a novel paradigm to approach the issue of autonomous policy-based management of wired/wireless differentiated communication systems. In contrast to existing management approaches which require static a-priori policy configurations, policies are created dynamically. The proposed scheme addresses the management issue from a new perspective through posing it as a problem of learning from current system behavior while creating new policies at run-time in response to changing requirements. Given sets of network objectives and constraints, policies are assembled at run-time. The new approach gives flexibility to users and applications to dynamically change their QoS requirements while maintaining a smooth delivery of QoS through network monitors feedback. The chapter is organized as follows. Sections 6.1 and 6.2 introduce concepts of policy-based network adaptation and basic policy notations, respectively. Related work for adaptive policy-based management is discussed in Section 6.3. A description of the proposed network adaptation scheme is presented in Section 6.4. Simulation results are presented in Section 6.5 to demonstrate the performance of the proposed work. Finally, Section 6.6 concludes the chapter.

## 6.1 Introduction

This chapter proposes a new paradigm to approach the issue of autonomous policy-based management of wired/wireless communication systems. The proposed work is inspired from the process of public policymaking process in the real-world [165]. The novelty of the presented scheme lies in that sets of policies, that are specifically adapted to suit current resources availability and users demands, are dynamically assembled and dispatched at run-time. By decoupling the functionality of adapting network-level policies from the task of mapping business objectives and abstract users' requirements, the proposed work offers users and administrators the freedom to specify and dynamically change their requirements. The first step of mapping higher-level requirements is carried out by the adaptation agents as described in Chapter 3 which results in a set of network level objectives. Given sets of network-level constraints, objectives and sets of possible actions to be taken, decisions for policies customizations are taken at run-time based on values obtained from forecast functions to best utilize the available network resources.

## 6.2 Policies- Basic concepts

The notion of predefined policies has been introduced as a promising solution to address the needs of QoS traffic management. These policies prescribe a set of rules that guide the behavior of network components. Once defined by administrators or network operators, these policies are translated into network level policies and stored in a policy repository where it is subsequently retrieved and enforced as needed.

According to the IETF [166], [167], The term *policy* can be defined as: "A definite goal, course or method of action to guide and determine present and future decisions. Policies are described in the form of one or more rules that specify actions that have to be performed in

response to certain conditions.

IF <condition 1> AND <condition 2> ... AND <condition n>  
THEN <action 1 > AND <action 2> ... <action m>

A policy rule will often comprise of other rules, in other words policies can contain policies. This notion of hierarchy enables complex policies to be built from a set of simpler policies, and simplifies their management.

Policies evolve through a policy life cycle [168] which consists of the following stages: policy specification, policy refinement, policy dissemination, and policy enforcement.

In the first stage in the life cycle, *policy specification*, the system user writes a policy in some language and it becomes represented in some form within the policy support system. The attributes of the policy described in the previous section are defined at this stage.

The next stage in the policy life cycle is that of *policy refinement*. This stage includes expansion of domains, defining procedures, static conflict analysis (use of priorities, meta policies, etc.), and the creation of device specific executable policies. The outcome of this stage is a detailed policy hierarchy.

The following stage is that of *policy dissemination*, i.e. the means by which the policies are distributed to managers or by which the managers look up centrally held policies. Once policy information is distributed to the management system some mechanism is needed in order to undertake *policy enforcement*. This requires that the policy information is translated to an executable form, is checked at the correct point in the management system function, and, depending on the result, then selects certain actions over others. During the course of enforcement, (dynamic) policy conflict may be detected through a manager (or managers) discovering that they have two contradictory policies to enforce. Such conflict may simply be reported to the policy support system or may be resolved by applying previously defined prioritization or meta-policies that choose which policy to apply. Later stages of the life cycle may also include

stages for disabling, withdrawing and deleting policies.

*Policy-based networking* is the ability to control a networking environment by specifying and enforcing policies. Policy-based networking helps manage user and applications priority, quality of service and security rights, based on management policies.

### 6.3 Existing Policy Adaptation Schemes

Policy-based network management has been introduced as a promising solution to the problem of managing QoS-enabled networks. However, static policy configurations built a-priori into network devices lack the flexibility and may not be sufficient to handle different changes in these underlying environments. Various research trends, e.g., [169], have highlighted the notion of policy adaptation and the central role that it can play in QoS management in policy-enabled networks. This notion of policy adaptation is becoming even more crucial as the managed systems become more complicated. In [170], Granville et al. proposed an architecture to support standard policy replacement strategies in policy-based networks. They introduced the notion of policy of policies (PoP). PoPs, acting as meta-policies, are defined to coordinate the deployment of network policies. The definition of PoP requires references to every possible policy that may be deployed besides the identification of events that can trigger a policy replacement. Although their work follows the concepts of policies automation, it puts a burden on the network administrator to define both the standard policies and the PoPs. Planning policies in the existence of PoPs is a complex task. Moreover, reaching an adequate policy replacement strategy requires a complex analysis process. The administrator still has to check which policies deployment strategies were successful and which strategies failed to achieve their goals and manually update these strategies. Active network technology has been proposed as a solution for policies adaptation in [171]. An adaptive policy-based management scheme specifies a policy life-cycle as a sequence of policy enforcement, traffic monitoring, and policy adaptation to

update the deployed policy. Management scripts were used to describe policies along with their life cycles. However, policy adaptation is performed only through adapting policy parameters dynamically according to the current network behavior.

In [172] a genetic algorithm based architecture for QoS control in an active service network has been introduced. Users are allowed to specify their requirements in terms of loss rate and latency and then policies are used to adapt the queue length of the network routers to accommodate these requirements. Policies are treated as though they were genes. An autonomous controller is used to export policies that improve the system performance while it de-activates policies that degrade its performance. The proposed work has the advantage that it is benefits from learning for adaptation. Our work follows the same concept of adaptation based on learning while allowing users/applications to dynamically specify their requirements in terms of high level policies.

Agents are used in [173] to represent *active policies*. The proposed architecture has a hyper-knowledge space, which is a loosely connected set of different agent groups which function as a pluggable or dynamically expandable part of the hyper-knowledge space. Active policies, which are agents themselves, can communicate with agents in the hyper-knowledge space to implement policies and retrieve information from agents. The architecture takes advantage of intelligent agents features such as the run-time negotiation of QoS requirements. However, an active policy by itself has to be created by the administrator, and once deployed to the network it remains static through its life-cycle.

Network monitoring feedback has been introduced in [174] to dynamically decide between different marking and policing policies for multimedia streams at the edge router depending on the network state. A closely related work has been presented in [169], Lymberopoulos et al. proposed a framework for adaptive management of Differentiated Services using the Ponder language [175]. The framework provides the administrator with the flexibility to define rules at different levels. Policy adaptation is enforced by other policies, specified in the same

Ponder policy notation. A goal-based approach to policy refinement has been introduced in [176] where low level actions are selected to satisfy a high-level goal using inference and event-calculus. In contrast to existing approaches, the proposed framework takes advantage of availability of previous experience gained from previously applied policies and their behavior to make decisions concerning the creation of future policies.

Although significant research work has been carried out in the area of policy-based management, existing techniques mainly focus on defining a-priori policies configurations to manage network devices.

If strictly approached in that sense, policies would lend themselves to be of a static nature and thereby introduce an immediate burden on network administrators. Administrators must be able to develop different policies in order to identify and prioritize users/applications and their requirements, provision network resources as well as monitor performance within the network. With the increasing magnitude and complexity of current communication system components, this task places proportional demands on administrators. Another direct consequence of that static nature is that network administrators recourse to using estimates of network traffic and users' requirements in configuring network policies. These estimates can be a major source of inflexibility. For example, initial distribution of network resources to different classes of network services may not come close to optimal utilization of these resources without factoring in past and future forecasted traffic loads. On the other hand, since network level-policies are derived from business objectives and users/applications requirements described in service level agreements (SLA) [23], policy-based management tools have to evolve and adapt with changes in these objectives and requirements in a timely manner. For example, currently developed adaptive applications, such as multimedia applications, are designed to continuously adapt their QoS demands at run-time; thus putting a stringent dynamically changing requirement on communication systems management tools. Also, users should be involved in a QoS selection as they roam across different domains where cost is of primary concern.

In summary, the need for an autonomous self-adaptable policy-based management scheme with inherent dynamic capabilities is becoming inevitable. This scheme must manage, customize and extend the underlying complex infrastructure of communication systems resources in response to the continuously changing business objectives and users' requirements.

## 6.4 Proposed Policy Adaptation Approach

In this section, we discuss the adaptation of network policies either to satisfy new users, applications and business goals delivered through the network side service prediction agents (NSPA) or in response to feedback information reported back by network monitoring agents (MA). Earlier work that addressed the issue of policy adaptation can be classified under one of two categories [169]. Schemes within the first category perform adaptation by dynamically changing different parameters of a QoS policy to specify new attributes values. In the second category, adaptation is carried out by enabling/disabling a policy from a set of predefined QoS policies at run-time. The first category is very specialized and may not cover all situations specially in highly dynamic environments. Meanwhile, policy adaptation through enabling/disabling policies does not scale well with the dynamic users' requirements and the ever changing conditions of underlying wired/wireless environments. The PAA scheme tackles this issue via posing it as a problem of learning from current system behavior and using the results of this learning process to assemble new policies at run-time. Nevertheless, the main challenge in assembling policies at run-time lies in deciding on the appropriate policy actions that can be applied to the different network components. More precisely, the following two issues arise in the decision making process.

The first issue is related to the basic steps taken to reach the optimum decision. This issue is addressed and inspired by novel approaches to the process of public policymaking in the real-world [165]. In these approaches, the policymaking process passes through three

main phases; stage setting, consideration of alternative decisions, and finally reassessment of the applied decision. As shown in Figure 6.1, the proposed scheme follows the same steps to decide network policies, while a feedback mechanism is used to ensure the correctness of the delivered policies. The following sections present details of these three phases and their implementation in the context of network policies.

The second key issue is in fact related to the second step of *consideration of alternative decisions* which we elaborate further in subsection 6.4.2. Of particular importance in this step is the notion of estimating the implications of each of the candidate decisions, on the network performance. For that purpose, a formal prediction process that can measure the impact of each potential decision on different network components is needed. Solving this issue stems from two main observations. The first observation is that the behavior of the network in response to a certain action is typically a function of the network traffic. The second observation is that network traffic has self-similar properties and shows long range dependencies. This observation, reported in [177], suggests that forecasting methods of economical and technical phenomena can be employed to predict the behavior of each candidate action.

Forecasting methods can roughly be classified into one of two categories: time series and causal regression [178]. Causal regression forecasts deal with the relationship between causes and consequences for long term forecasts. In time series methods, the forecast is based on the earlier behavior. Since time series methods proved to be successful in estimating network traffic behavior, we adopt the same methods for forecasting the behavior of different actions [179]. Specific details on the adopted forecasting methodology will be presented in the following sections.

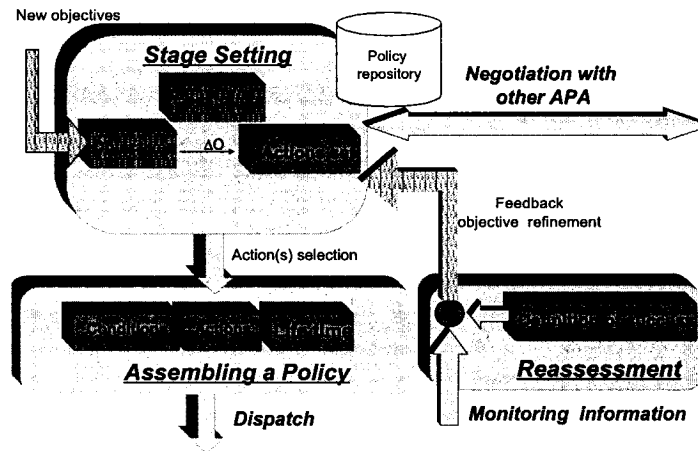


Figure 6.1: Stages of adaptive policies assembly and modification

### 6.4.1 Stage Setting: Information Gathering

The first step in the adaptive policy making process is *stage setting*. At this stage, all information necessary for the process of policy creation is obtained. The information collected at this stage is defined over three spaces: *constraints space* ( $\mathbb{CN}$ ), *objectives space* ( $\mathbb{O}$ ) and the space of all possible *actions* ( $\mathbb{A}$ ).

The constraints space,  $\mathbb{CN}$ , represents the space of all physical limitations that can be imposed by the routers features. Examples of these constraints are: memory size, CPU speed, link bandwidth and available mechanisms for policies implementations (e.g., available classifiers types, metering, shaping and policing mechanisms and scheduling and queuing disciplines).

The objectives space,  $\mathbb{O}$ , of the PAA describes the overall network QoS guarantees for individual flows as well as for different classes of services. Formally,  $\mathbb{O}$  is defined as the Cartesian product  $\mathbb{O} : \mathbb{C} \times \mathbb{BW} \times \mathbb{L} \times \mathbb{D} \times \mathbb{J}$ , where

- $\mathbb{C}$  is the space of classifications that a traffic may belong to. Therefore, an element  $C_i \in \mathbb{C}$  can be a class of service for flow aggregates, e.g., EF and AF, or an individual flow specification, e.g., IP source/destination addresses, port numbers, MAC addresses, or combination thereof.

- $\mathbb{B}\mathbb{W}$ ,  $\mathbb{L}$ ,  $\mathbb{D}$  and  $\mathbb{J}$  are the spaces of allowed values of bandwidth, delay, jitter and loss, respectively.

An objective  $O_i(C_i, BW_i, L_i, D_i, J_i) \in \mathbb{O}$ , or  $O_i$  for short, describes the bandwidth, loss, delay, and jitter requirements for a traffic within the classification  $C_i$ . Examples of two PAA objectives are:  $O_1(\text{EF}, 30\%, 1\%, 20\text{ms}, 3\text{ms})$  and  $O_2(\text{IP\_source} = 122.145.1.20, 5\text{Mbps}, 3\%, 40\text{ms}, 5\text{ms})$ .

The set of network domain objectives, at a certain time  $t$ ,  $O_t \subset \mathbb{C}$ , can then be expressed as  $O_t = \{O_0^t, O_1^t, O_2^t, \dots, O_n^t\}$ ,  $O_i^t \in \mathbb{O}$ .  $O_0$  is a statically configured objective which represents the goal of maximizing the utilization of the available bandwidth capacity. It is worth mentioning here that objectives may take absolute values, e.g., delay guarantees of EF class is less than 100 ms,  $D_{\text{EF}} < 100$  ms, or relative to other objectives values, e.g., delay guarantees of EF class is less one third of the delay guarantees of AF class,  $D_{\text{EF}} < 1/3 D_{\text{AF}}$ . An objective  $O_i^t \in O_t$  will change in response to changes in higher level policies.

Now, as pointed out in the previous section, PAA policy adaptation is triggered due to either a translated change in user/application or business policies or due to a feedback from network monitors. Hence, once the adaptation process is triggered, the input to the stage setting step can be represented as a partial subset  $\hat{O} = \{\hat{O}_1, \hat{O}_2, \dots, \hat{O}_n\}$  reflecting the required new objectives or the actual measured objective, respectively. As the new sub-objectives are fed to this stage, an *objectives change*  $\Delta O = \{\Delta O_1, \dots, \Delta O_m\}$  is calculated as follows,

$$\begin{aligned}
\Delta O &= \hat{O} - O_t \\
&= \{\Delta O_i \mid \forall \hat{O}_i \in \hat{O} : \Delta O_i(C_i, \Delta BW_i, \Delta L_i, \Delta D_i, \Delta J_i) \\
&= \hat{O}_i(\hat{C}_i, \hat{B}\hat{W}_i, \hat{L}_i, \hat{D}_i, \hat{J}_i) - O_j(C_j^t, BW_j^t, L_j^t, D_j^t, J_j^t), \\
&C_j^t = \hat{C}_i\}
\end{aligned} \tag{6.1}$$

The objective change  $\Delta O$  is then used in the next stage of candidate action selection.

Finally, the actions space,  $\mathbb{A}$ , includes possible actions that can be used in the action part of the policies. Associated with each action is an indicator of whether this action can be applied to an edge or a core router. In other words, this indicator specifies whether this action is applied to a single flow or flow aggregates.

Each action is associated with a *Forecast Function (FF)* [178]. *FF* is a mapping of a certain action  $A_k$  to a set of forecasted values. This set of forecasted values is represented by a sparse matrix  $\mathcal{FF}_k$ , written using column-wise notation as follows,

$$\mathcal{FF}_k(n) = [ff_1^k(n) | ff_2^k(n) | \dots | ff_m^k(n)]$$

In general,  $\mathcal{FF}_k(n)$  describes an estimated forecast of the effects of the  $n^{\text{th}}$  application of action  $A_k$  on network performance. More precisely, the  $j^{\text{th}}$  entry of the  $i^{\text{th}}$  column,  $ff_{j,i}^k(n)$ , represents the forecasted change in the  $j^{\text{th}}$  QoS parameter for a certain flow with characteristic classification  $C_i$  in response to applying network action  $A_k$ . The index  $j$ , therefore, can take values of 1, 2, 3 and 4 corresponding to bandwidth, loss, delay and jitter, respectively.

Future values of each entry of  $\mathcal{FF}_k$  at the next step  $n + 1$  can be obtained by using one of the established forecasting techniques [178]. We adopt a general forecasting function with desirable smoothing features; namely, the adaptive-response-rate single exponential smoothing (ARRSES) function. Using the ARRSES function, the value of the  $(j, i)$  entry in  $\mathcal{FF}_k(n + 1)$  is calculated as follows,

$$\begin{aligned} ff_{j,i}^k(n+1) &= \alpha_n \partial Y_{n+1}^j + (1 - \alpha_n) ff_{j,i}^k(n) \\ \alpha_n &= \left| \frac{E_{n-1}}{M_{n-1}} \right|, \\ E_{n-1} &= \beta e_{n-1} + (1 - \beta) E_{n-2}, \\ M_{n-1} &= \beta |e_{n-1}| + (1 - \beta) M_{n-2}, \\ e_{n-1} &= Y_n - ff_{i,j}^k(n), \end{aligned}$$

(6.2)

where  $0 < \beta < 1$ , and  $|\cdot|$  denotes absolute values.  $f f_{j,i}^k(n)$  is the past effect of action  $A_k$  on the  $j^{\text{th}}$  QoS parameter of a traffic with characteristics  $C_i$ , calculated after  $n$  times of enforcing  $A_k$ .

$f f_{j,i}^k(n+1)$  is the forecasted value of the change in the  $j^{\text{th}}$  parameter of a traffic  $C_i$  if  $A_k$  is to be applied.

$\partial Y_{n+1}^j$  is the observed effects of  $A_k$  on the  $j^{\text{th}}$  QoS parameter of a traffic with characteristics  $C_i$  as reported by network monitors.

It can be noticed that the value of the controlling parameter,  $\alpha_n$ , is defined as the absolute value of the ratio of a smoothed error  $E_t$  and a smoothed absolute error  $M_t$ .

ARRSES has been chosen for its simplicity, low memory requirements and accuracy. Furthermore, it adjusts itself by changing the value of  $\alpha_t$  to follow basic changes in the action effects on different QoS parameters. This provides an automated way of following the effects of different traffic loads over the network.

## 6.4.2 Candidate Actions Selection

In the second step, the PAA selects one or more actions from the actions space that best attains the specified change of objectives  $\Delta O$  obtained from (6.1). At this step, the PAA may negotiate with other neighboring PAAs for the selection of the most suitable actions to be taken. The negotiation may also involve the exchange of different actions and their FF. The selection process is carried out by calculating an *Expected Loss value* for each action  $A_k \in \mathbb{A}$ ,  $\mathcal{E}_{\mathcal{L}}(A_k)$ , as follows,

$$\mathcal{E}_{\mathcal{L}}(A_k) = \sum_i \sum_{j=1}^4 w_j (\Delta O_{j,i} - f f_{j,i}^k(n)), \quad A_k \in \mathbb{A} \quad (6.3)$$

where  $\Delta O_{j,i}$  represents the required objective change in the  $j^{\text{th}}$  QoS parameter of class with characteristics  $C_i$ , and is obtained as, described earlier, from (6.1).  $w_j$  is the weight of significance for the  $j^{\text{th}}$  QoS parameter. In other words,  $\mathcal{E}_{\mathcal{L}}(A_k)$  represents the discrepancy between

the forecasted change in the QoS parameters and the actual requested change.

Now the process of choosing the optimum policy action reduces to the task of selecting an action  $\tilde{A}$  that minimizes the expected loss values. This idea can be formally states as follows,

$$\tilde{A} = \min_{A_k \in \mathbb{A}} \mathcal{E}_{\mathcal{L}}(A_k) \quad (6.4)$$

### 6.4.3 Policy Assembly

The third step involves the assembly of a new policy given the actions selected in the previous step. The assembled policy consists of a triggering event as translated by the PAA from higher level policies (e.g., user location), conditions specified by the characteristics  $C_i$  of the satisfied objective, and the selected action  $\tilde{A}$ . The new policy can also be associated with a life time after which it should expire and be deleted. Once a policy is assembled, it is dispatched to be applied at the network level either to an ingress, egress, or a core router according to the selected action type.

### 6.4.4 Reassessment

The final step in the policy adaptation process is performed by the *reassessment module*, as shown in Figure 6.1, in order to evaluate the degree of success of the previously dispatched policy.

Network monitors measure the average values for the actual QoS parameters of different classes. For example, the actual throughput of traffic of class  $C_i$  at time  $T$  is calculated as follows,

$$BW_i^{avg}(T) = \frac{1}{T} \int_{t_0}^{t_0+T} BW_i(t) dt \quad (6.5)$$

In a similar manner, other performance parameters,  $L_i^{avg}(T)$ ,  $D_i^{avg}(T)$  and  $J_i^{avg}(T)$ , are calculated for each class of service  $C_i$ .

The PAA uses a *definition of success (SUC)* to reassess the decision of the applied policies. *SUC* is defined as the difference between the current measured values of QoS parameters and the required objectives, as follows,

$$SUC = \sum_i \sum_j w_j (\hat{O}_{j,i} - X_{j,i}^{avg} - \mathcal{E}_{j,i}), \quad (6.6)$$

$$\mathcal{E}_{j,i} = \log \left( \frac{X_{j,i}^{avg}}{\hat{O}_{j,i}} \right), \quad (6.7)$$

$\hat{O}_{j,i}$  is the new network objective for the  $j^{th}$  QoS parameter of traffic  $C_i$ .  $X_{j,i}^{avg}$  is the measured value for the  $j^{th}$  QoS parameter as obtained in (6.5).

$\mathcal{E}_{j,i}$  is an error function that reflects the degree of tolerance in a certain objective value. We chose the error function to be the natural logarithm of the ratio of the current and desired parameter values since it allows for a wide dynamic range of difference to be measured.

Based on the value of *SUC*, the reassessment module takes one of two actions. In case of successfully achieving the required objectives, it modifies an entry in  $\mathcal{FF}$  of the successfully applied action,  $\tilde{A}$ , using (6.2) and replaces the current network objective  $O_i$  with the newly applied objectives  $\hat{O}$ . Otherwise, the difference between the measured values by the monitors and the required objectives is fed back to the first stage as a new objective change and the adaptation process is repeated.

### 6.4.5 Illustrative Examples

In the following, we give illustrative examples of the elements of the constraints, objectives and action spaces.

- **Constraints set (CN)**

A typical set of constraints is concerned with each router resources. As an example,

it states available memory, disk size, buffers sizes, CPU speed, and supported different queuing disciplines, e.g., FIFO, and round robin.

- **Objectives set (O)**

Table 6.1 gives an example of a possible subset of objectives of the PAA. As shown, for each class of service, e.g., Expedited Forwarding (EF), the objective is to maintain a guaranteed level of QoS. This level is defined in terms of corresponding values for various QoS parameters such as loss rate, latency, jitter, and guaranteed bandwidth.

Table 6.1: PAA Objectives

$O$	$C$	BW	loss rate(%)	avg. delay(ms)
$O_1$	EF	30%	0.1	30ms
$O_2$	AF11	25%	0.15	100ms
..				...
$O_i$	BE	20%	-	-
$O_{i+1}$	IP_source = 122.137.120.1	1Mbps	0.4	100ms

- **Actions space (A)**

Tables 6.2 and 6.3 give a sample of the action spaces in edge and core routers, respectively.

## 6.5 Simulation Details and Results

This section summarizes simulation results of the proposed scheme. A simulation architecture was constructed to model a multi-domain environment used to evaluate the proposed scheme.

Table 6.2: Sample of PAA actions for an edge router

$A$	Action
$A_1$	Increase token bucket generation rate by 1% for flow $C_i$
$A_2$	Decrease token bucket generation rate by 1% for flow $C_i$
$A_3$	Increase token bucket burst (size) for flow $f$
$A_4$	remark with AF
$A_5$	Increase peak burst size of Three color meter by 1% for flow $f$
$A_6$	Add dropper
	⋮
$A_n$	Report back to application to perform adaptation

Table 6.3: Sample of PAA actions for a core router

$A$	Action
$A_1$	Increase weighted round robin priority for class $C_i$ by 1%
$A_2$	Decrease weighted round robin priority for class $C_i$ by 1%
	...
$A_m$	Increase drop tail queue capacity of class $C_i$ by 1%

As shown in Figure 6.2, each simulated domain includes an ingress router, a core router and an egress router. The simulated network has been constructed using the J-Sim network simulator [180], a simulator with a Java based engine. The PAA has been implemented on top of the simulator with a GUI to accept higher level policies that are translated into network level objectives using heuristic rules as those described in [19].

All links for domains A and C and links between all domains are configured with 15Mbps capacity and 5ms propagation delay. Links in domain B are configured with 10Mbps with a

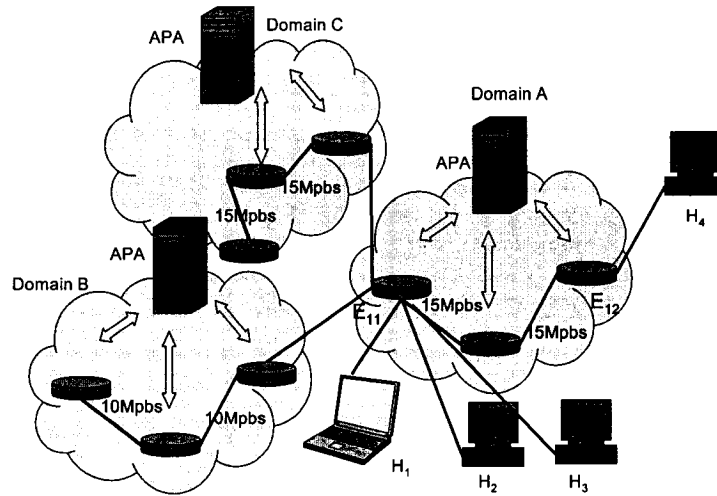


Figure 6.2: Experimental testbed

propagation delay of 5ms. Weighted Round Robin (WRR) scheduling is implemented at each router. Initially, 40%, 30% and 30% of the available bandwidth are set for EF, AF and BE, respectively, for domains A, B and C. Drop tail queues are used as buffering mechanisms for core routers. The edge router  $E_{11}$  is configured with one profile for each traffic source with initial rate as the sending rate of the source. A token bucket-based meter was used for each profile at the edge routers. Unless otherwise stated, traffic is generated by hosts  $H_1$ ,  $H_2$  and  $H_3$  to the same destination  $H_4$ . All network traffic has been modelled with a Poisson distribution with different peak rates and a packet size of 1024 bytes. To simplify the implementation of the proposed scheme, we limited our experimentation to a subset of possible actions, namely,  $A_1, A_2, A_3, A_4$  of table 6.2 and  $A_1$  and  $A_2$  of table 6.3.

### 6.5.1 Experiment 1: User-Based Policy Adaptation

In this experiment we test the performance of the PAA in response to a new user policy. Traffic was generated from  $H_4$  to  $H_1$  with sending rate of 6Mbps while  $H_1$  is moving from domain A to B at  $t=50$  and then from B to C at  $t=80$ . For experimentation purposes, for EF services, we

associated a cost  $3c$ ,  $5c$  and  $3.3c$  per 1Mbps [181] for domains A, B and C, respectively. The user on host  $H_1$  has a registered policy  $P_1$ : *if(location  $\neq$  home\_domain) then cost  $< 1.5c$  per unit time*. As the user moves from one domain to the other,  $P_1$  is translated to different network level objectives by the PAA at domain A depending on the location of the user. Figure 6.3 shows a comparison between the traffic at the source  $H_4$  (source) and the traffic received by  $H_1$  (destination). Initially, at time  $t = 0$ ,  $P_1$  has been mapped to a new objective  $O(C = H_1, BW \leq 6\text{Mbps})$  and action  $A = \text{"Increase Token rate by 1\%Mbps"}$  is successively applied to allow the user a sending rate of 6Mbps. As the PAA is notified with the event that the user is moving to domain B.  $P_1$  is re-mapped to a new objective  $O(C = H_1, BW \leq 3\text{Mbps})$  and action:  $A = \text{"decrease token rate by 1\%"}$  is applied until the token rate of the user profile meter at  $E_{12}$  is set to 3Mbps. Finally, as the user moves to domain C, a new objective  $O(C = H_1, BW \leq 4.5\text{Mbps})$  is formulated and a new action  $A = \text{"Increase token rate by 1\%"}$  is selected repeatedly to reach the target rate.

### 6.5.2 Experiment 2: Application-Based Policy Adaptation

In this experiment we illustrate how adaptation is performed at the network level in response to an application level policy. In this network configuration, EF traffic is generated from two hosts  $H_1$  and  $H_2$  to the same destination  $H_4$ .  $H_1$  transmits at a rate of 3.5Mbps for the duration of the experiment.  $H_2$  sends packets with a target EF service with a throughput requirements of 3.5Mbps from  $t = 30$  to  $t = 70$  and 4.5Mbps from  $t = 70$  up to  $t = 110$ .

An application running on  $H_4$  registers a policy  $P_2$  which indicates that traffic reaching  $H_4$  from source  $H_2$  has double the priority of traffic from  $H_1$ , i.e.,  $P_2$ : *if (Exists(traffic\_source =  $H_1$  and service = EF) and Exists(traffic\_source =  $H_2$  and service = EF)) then priority  $H_2$  traffic = 2 priority  $H_1$  traffic*.

At time  $t = 0$ ,  $H_1$  started transmission with objective:  $O_1(\text{source} = H_1 \text{ and destination} = H_4,$

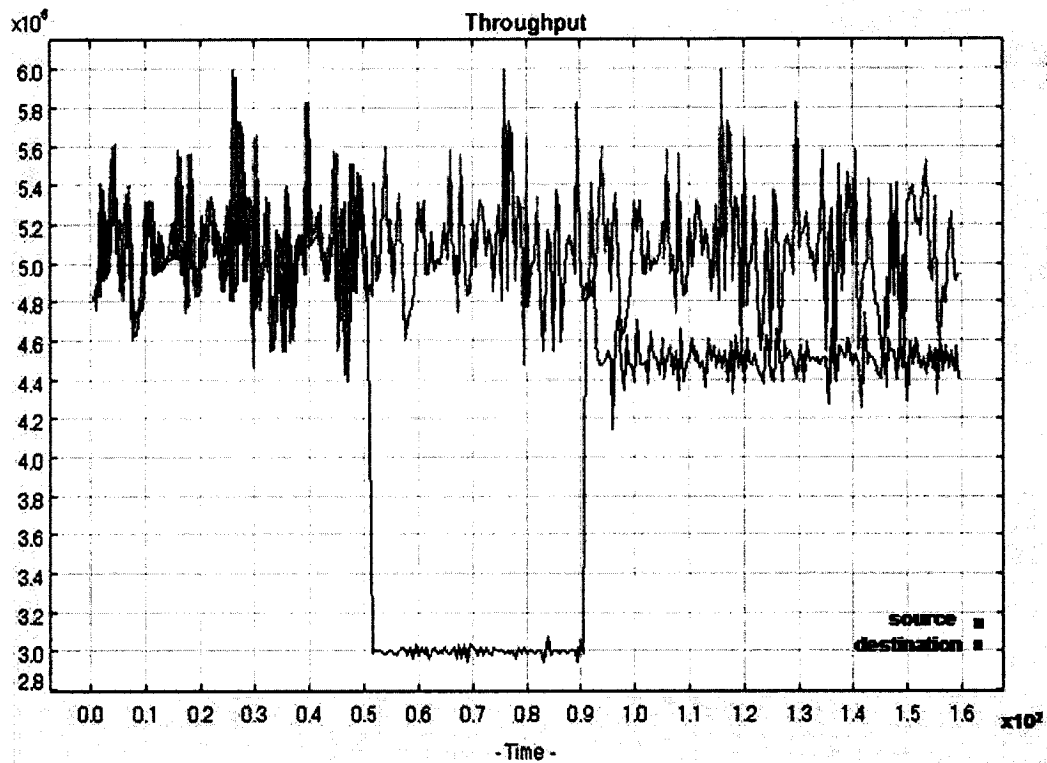


Figure 6.3: Adaptation based on user policy

BW= 3.5 Mbps). In this case the links bandwidth is only utilized by  $H_1$ . The Token bucket meter is initiated with a target rate = 4Mbps and hence all packets are marked with EF DSCP. At time  $t = 70$ , a new objective  $O_2$ (source =  $H_2$  and destination =  $H_4$ , BW = 3.5Mbps) is fed to the PAA and the old objective  $O_1$  is replaced with a objective  $O_1$ (source =  $H_1$  and destination =  $D$ , BW <  $O_2$ (BW)). The target rate of the meter of the first traffic is gradually decreased until  $O_1$  and  $O_2$  are satisfied.

Now at time,  $t = 7$  as  $H_2$  increases its sending rate requirements,  $O_2$  is changed to  $O_2$ (source =  $H_1$  and destination =  $D$ , BW = 4.5 Mbps). A new action that increases the target rate for  $H_2$  is gradually applied. Meanwhile another action for degrading  $H_1$  target rate is applied. Out-of profile packets are remarked as AF. At time  $t = 110$  the objective  $O_1$  is replaced

back with the initial objective and the target rate is increased to 3.5Mbps.

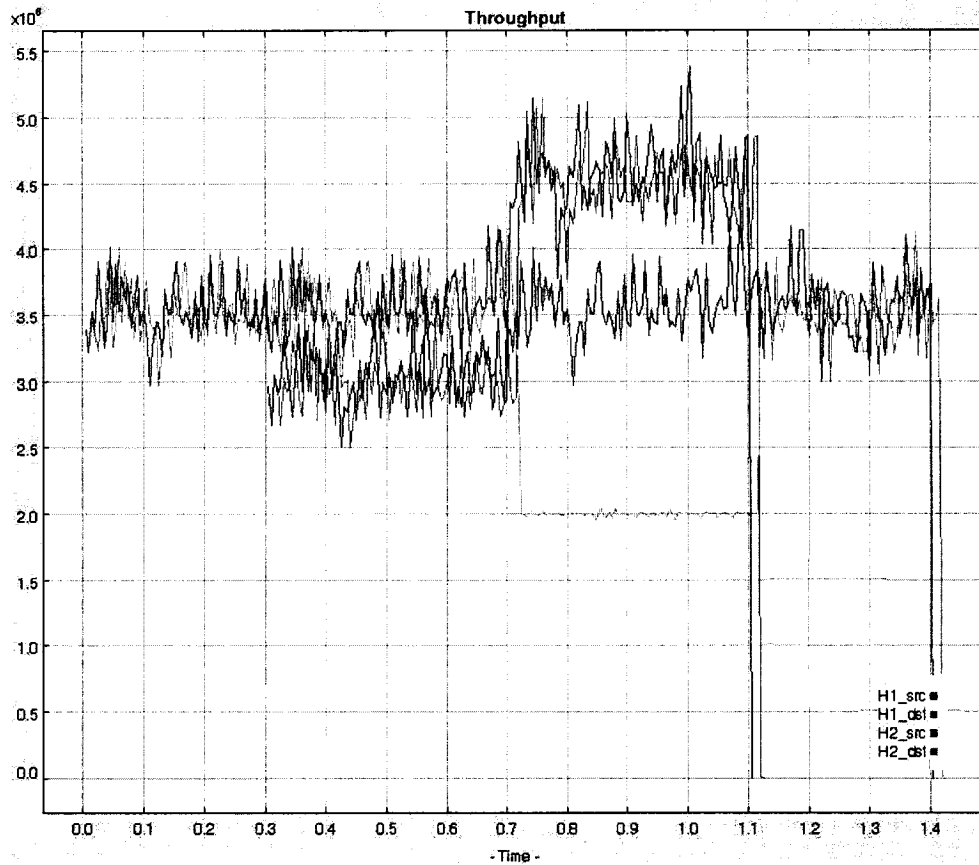


Figure 6.4: Adaptation based on Application policy

### 6.5.3 Experiment 3: Network Adaptation

This experiment illustrates the self-configuration feature of the proposed architecture. We illustrate how PAA performs an automated selection of the appropriate actions under different network level objectives. For simplicity, in this experiment we limit the adaptation process to three actions,  $A_1$ ,  $A_2$  and  $A_3$ , defined as follows,

- $A_1$  : Increase of WRR priority for EF by 1.

- $A_2$  : Increase buffer size for EF aggregates by one packet.
- $A_3$  : Increase the bucket rate of single traffic of  $H_1$  at ingress router  $E_{11}$  by 1%.

All network links are set with the same capacity of 10Mbps. Initially, the network is configured to 60% and 40% of traffic for EF and BE, respectively. Background traffic is generated by  $H_2$  as EF traffic with a constant bit rate of 2Mbps, and by  $H_3$  as BE traffic at a constant rate of 4Mbps. A token bucket with rate equal to the initial sending rate of each flow is selected for each flow.  $H_1$  transmits EF packets following a Poisson distribution with an average rate of 4Mbps.

We start first by analyzing the ability of the selected ARRSES function in following the behavior of the applied actions. Figure 6.5 depicts the forecasted values of the throughput of EF traffic aggregate in response to repetitive applications of  $A_1$ . Figures 6.6(a) and 6.6(b) illustrate the forecasted values of the throughput and loss rate, respectively, of EF aggregate in response to repetitive applications of  $A_2$ .

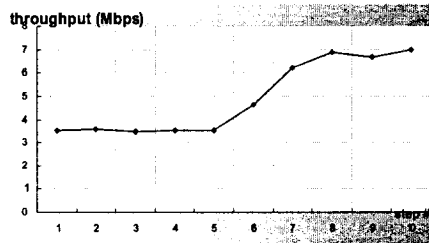


Figure 6.5: Forecasted values for  $A_1$  w.r.t. throughput of EF aggregate.

Figure 4.3 shows a comparison between the achieved bandwidth of the proposed adaptive scheme (PAA) against the bandwidth achieved with static configurations of the meter, drop tail queue and WRR scheduler.

At  $t = 0$ , the following objectives are configured:  $O_1$ (DSCP= EF,  $BW = 6$ Mbps,  $L = 0.2\%$ ),  $O_2$ (source =  $H_1$ ,  $BW = 4$ Mbps,  $L = 0.1\%$ ),  $O_3$ (source=  $H_2$ ,  $BW = 2$ Mbps,  $L =$

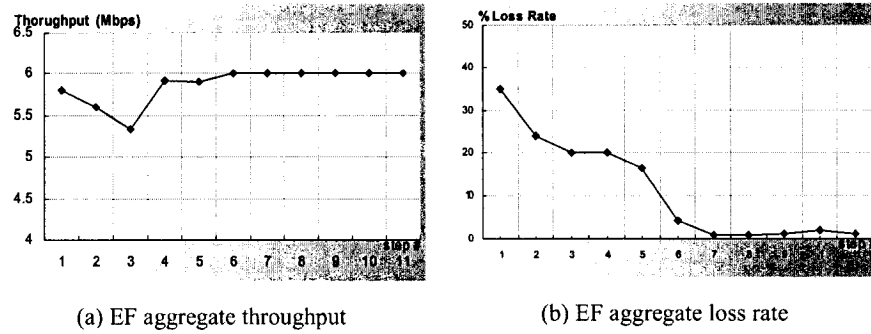


Figure 6.6: Forecast function for  $A_2$  w.r.t. EF aggregates throughput and loss rate.

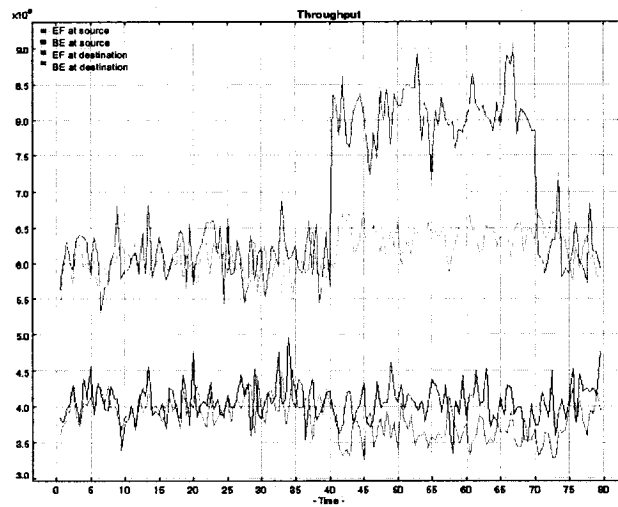
0.2%),  $O_4$ (DSCP = BE,  $BW = 4\text{Mbps}$ ). Initially, the token rate of each flow is set with the values of its sending rate.

Due to feedback from the token bucket meter of the flow from  $H_1$ , an indicated loss rate  $> 0.1\%$  was reported to the PAA. At this stage, only the objective of a single flow was not met, i.e., an objective refinement  $\Delta O_2(\Delta L < 0.1)$  was triggered. At this stage action  $A_3$  is selected for adaptation. The token bucket rate is increased gradually. This caused a slight increase in the rate of flow  $f_1$  from traffic of  $H_1$ . However, since, at the core router, traffic from both  $H_1$  and  $H_2$  is being dropped by the drop tail queue, after a certain period the loss rate of EF class is increased, where,  $O_1, O_2, O_3$  are violated.

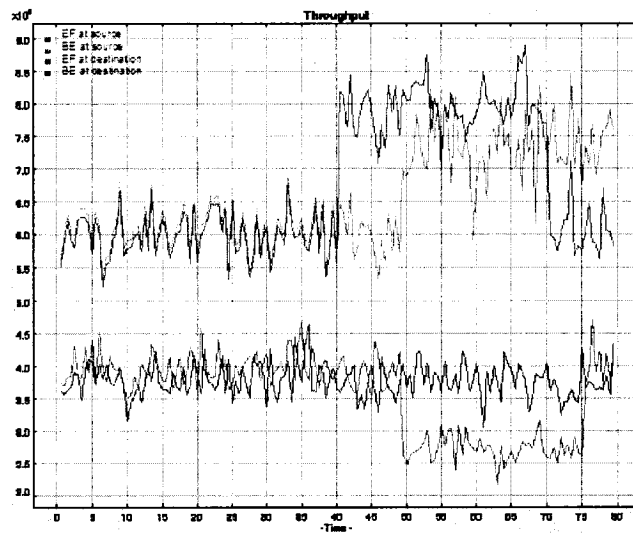
Since action  $A_1$  will result in violating objective  $O_4$ ,  $A_2$  is selected and the size of the drop tail queue is gradually increased. As shown in Figure 6.7, this indicates a temporary increase in the throughput of both flows. However, after a certain queue size, the increase does not affect the throughput of both flows. In response to that, action  $A_1$  is selected and repetitively applied giving priority to  $O_1, O_2, O_3$  over  $O_4$ .

## **6.6 Summary**

In this chapter a novel scheme for adaptive policy-based QoS management in wired/wireless networks has been presented. The novelty of the proposed work lies in that given sets of objectives, constraints and goals, network policies are assembled at run-time. It has been shown that adapting policies at run-time provides flexible means for controlling network behavior as the surrounding environment changes. In addition, it gives more freedom to users and applications providers to describe their requirements, in a continuously changing manner, in terms of policies that are functions of different parameters such as time, location, and cost.



(a) Achieved throughput in case of static configurations



(b) Achieved throughput in case of PAA

Figure 6.7: Throughput comparison between the statically configured network and the proposed scheme

# **Chapter 7**

## **Conclusions and Future Research**

### **Directions**

This chapter identifies contributed research work and discusses planned and future directions. The chapter is organized as follows. Section 7.1 gives a summary of research contributions in the area of automated management. A summary of future research directions is then presented in Section 7.2.

#### **7.1 Dissertation Contributions**

The focus of the conducted research, thus far, has been the development of an autonomous self-adaptive management system for hybrid communication systems. The first step towards achieving that goal included a literature study of the management problem and management difficulties. More precisely, we aimed to address two questions; what are the requirements of any management system? and why is it difficult to satisfy these requirements with current approaches? The results of this phase is a state of the-art survey of major research directions and efforts in the area of management for wired, wireless and hybrid communication systems.

Based on the identified limitations of current research work, a novel framework for an automated prediction-based adaptive management system has been designed. The framework has been presented as a multi-layered model that utilizes knowledge about users, applications and the underlying network to perform adaptive management functionalities. In the following, a summary of the main contributions of current research work is given.

- A state of the art survey of management approaches.
- A complete design and functional specification of the necessary management tasks and architectural components.
- A novel scheme for a mobility prediction based on the use of contextual information has been presented. Uncertainty of the user's navigation behavior was captured by gathering pieces of evidence concerning different groups of candidate locations. These groups were then refined to predict the user's future location when evidence accumulate using Dempster rule of combination. In contrast to existing approaches the proposed approach did not impose any assumptions concerning the availability of a history of user's movements. Simulation results had shown the efficiency of the proposed scheme.
- A robust scheme for network anomaly diagnosis based on statistical analysis and evidential reasoning is developed. The scheme effectively reduces the number of false primitive alarms by diluting effects of isolated outliers. The classification of the detected anomalies is performed via evidential reasoning using a training set of previously detected and classified anomalies. To handle cases of imbalanced training sets, the scheme includes two new approaches to assign mass belief values to the classes under investigation using the constructed evidence. The latter contribution can be of independent interest to research related to the Dempster-Shafer theory and can be applied in contexts other than the present one. Theoretical and experimental analysis demonstrate the accuracy of the

proposed scheme.

- Design of a novel network-level policy-based management scheme has been presented. The key feature in the design is the decoupling between the task of mapping abstract higher level goals into network level objectives from the functionality of adapting the behavior of network components. Furthermore, to automate the mapping process a hierarchical model of policies has been applied and a methodology of translating abstract higher level users' preferences and business goals for QoS into network level objectives has been discussed.
- A novel scheme for adaptive management of QoS based on a new concept of forecasting functions have been presented. The scheme utilizes forecasting functions to predict effects of different management decisions. Dynamic network-level policies assembly at run-time is performed based on values of these functions. The performance of the proposed scheme has been evaluated through simulation.

## **7.2 Future Research Work**

The main focus of the future research work can be divided into two key directions as follows.

### **7.2.1 Case-Based Adaptive QoS Management**

By incorporating the experiences gained from applying different management strategies, one can further enhance the performance any autonomous management system. One way to achieve that is through the utilization of Case-Based Reasoning (CBR) concepts. CBR is a problem solving and learning paradigm that has received considerable attention over the last few years [182], [183]. It has been successfully applied in different domains such as e-commerce [184] and automated help desks [185]. We plan to investigate the feasibility of representing network

policies as cases and in turn the creation of a knowledge base of policies (cases). We will also investigate the applicability of the different phases of the CBR life cycle and well advanced adaptation methodologies to current policy-based management systems. Preliminary results of this research direction have been reported in [186], [187].

### **7.2.2 Network Contextualization**

Another focus of future research work is to investigate the possibility of contextualizing the underlying network information to better perform different management tasks by featuring awareness of the surrounding environment context. As networks are becoming more complex, plain data such as a network monitored bandwidth or error rate will no longer be sufficient to make adaptation decisions in context rich environments. Hence, more sophisticated models that employ context knowledge to better perform various automated management tasks are needed.

Although, different information models have already been used to represent and configure different managed entities, thus far, these models suffer from a lack of a smart representation and encoding of available knowledge in a way which can be used to make the necessary inferences to automate the management process.

On the other hand, despite the recent advancements in the area of context-aware applications, very few research work exists that approaches the problem of network management via a context-awareness perspective.

The main goal of this future direction is to construct a set of inferred network related context properties (e.g., available and preferred services, preferred connections and services that are of interest to users), given knowledge of a set of primitive context (e.g., BW, delay and location). The theory of event calculus will be utilized to formalize context as a dynamically changing model of the world. Initial results of this research direction have been reported in [188], [89].

# List of Publications

## Refereed Journal Publications

1. N. Samaan and A. Karmouch, "A Mobility Prediction Scheme Based on Dempster-Shafer Theory of Evidence", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 5, pp:537 - 551, September/October 2005.
2. N. Samaan and A. Karmouch, "An Automated Policy Based Management Framework for Differentiated Communication Systems", *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 12, pp:2236- 2247, December 2005.

## Papers submitted to refereed Journals

1. N. Samaan and A. Karmouch, "Network Anomaly Diagnosis via Statistical Analysis and Evidential Reasoning", *IEEE/ACM Transactions on Networking*, submitted.
2. N. Samaan and A. Karmouch, "Autonomous Network Management: A Survey and a New Approach", *IEEE Communications Surveys and Tutorials*, submitted.

## Refereed Conference Publications

1. N. Samaan, H. Harroud and A. Karmouch, "PACMAN: a Policy-based Architecture for Context Management in Ambient Networks", *IEEE Consumer Communications and Networking Conference (CCNC07)*, Las Vegas, Nevada, 11-13 Jan. 2007.
2. N. Samaan and A. Karmouch, "Circumscriptive Context Reasoning for Automated Network Management Operations", *IEEE GLOBECOM 2006 - General Conference on Emerging Technologies - Autonomic Communication/Networking Principles*, Nov.27-Dec.1 ,2006, San Francisco, California, USA.
3. N. Samaan and A. Karmouch, "A Novel Approach Towards Autonomic Management in Context-Aware Communication Systems ", in *Mobility Aware Technologies and Applications*, 2nd International Workshop, MATA2005, Montreal, Canada, October 17-19, 2005, Proceedings. Lecture Notes in Computer Science 3744 Springer 2005, pp 374-383.
4. N. Samaan and A. Karmouch, "A user centric mobility prediction approach based on spatial conceptual maps", in Proceedings of the *IEEE International Conference on Communications*, Seoul, Korea, ICC, 16-20 May 2005 Volume: 2, pp 1413- 1417.
5. N. Samaan, B. Benmammar, F. Krief, and A. Karmouch, "Prediction-Based Advanced Resource Reservation in Mobile Environments", in *the 18th Annual Canadian Conference on Electrical and Computer Engineering, CCECE05*, May 1-4, 2005, Saskatoon, Saskatchewan Canada.
6. N. Samaan and A. Karmouch, "A Case-Based Reasoning Approach for Automated Management in Policy-Based Network", in *15th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2004*, Davis, CA, USA, November 15-17, 2004.Proceedings. Lecture Notes in Computer Science 3278 Springer 2004,

pp 76-87.

7. N. Samaan and A. Karmouch, "An Automated Policy-Based Management Framework Using Forecasting Functions", in *the 12th HP-OVUA Workshop*, Porto, Portugal, July 10-13, 2005.
8. N. Samaan and A. Karmouch, "Adaptive policies for management of autonomous communication systems", in *7th International Symposium on Communications Interworking*, Ottawa, Canada, Nov. 2004.
9. N. Samaan and A. Karmouch, "Management in Differentiated networks via an Adaptable Case-Based Policy Framework", in *the 11th HP-OVUA Workshop*, Paris, June 20 - 23, 2004.
10. N. Samaan, A. Karmouch, and H. Kheddouci, "Mobility prediction based service location and delivery" in *the 17th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2004)*, Niagara Falls Ontario, Canada, May. 2-5 2004.
11. N. Samaan and A. Karmouch, "An evidence-based mobility prediction agent architecture", in *the 5th International Workshop on Mobile Agents for Telecommunication Applications*, Marrakech, Morocco (MATA03), October 8-10 2003.
12. N. Samaan and A. Karmouch, "Prediction based policy adaptation for QoS management in wireless networks", in *proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003)*, Italy, Jun. 4-6 2003.
13. N. Samaan and A. Karmouch, "A multi-agent framework for qos adaptation using policies in wireless environments", in *"Gestion de Réseaux Et de Services, Fortaleza, Brazil"*, Feb. 24-27 2003.

# Bibliography

- [1] N. Samaan and A. Karmouch, "A Mobility Prediction Scheme based on Dempster-Shafer Theory of Evidence," *IEEE Transactions on Mobile Computing*, vol. 4, no. 5, pp. 537–551, Sept.-Oct. 2005.
- [2] —, "An Evidence-Based Mobility Prediction Agent Architecture," in *Proceedings of the 5th International Workshop on Mobile Agents for Telecommunication Applications, MATA 2003, Marakech, Morocco*, ser. Lecture Notes in Computer Science, vol. 2881. Springer, Oct. 8-10 2003, pp. 230–239.
- [3] —, "A User Centric Mobility Prediction Approach Based on Spatial Conceptual Maps," in *Proceedings of the IEEE International Conference on Communications (ICC-2005)*, vol. 2, Seoul, Korea, May 2005, pp. 1413–1417.
- [4] N. Samaan, B. Benmammour, F. Krief, and A. Karmouch, "Prediction-Based Advanced Resource Reservation in Mobile Environments," in *the 18th Annual Canadian Conference on Electrical and Computer Engineering, CCECE05, Saskatoon, Saskatchewan Canada*, May 1-4 2005.
- [5] N. Samaan and A. Karmouch, "Network Anomaly Diagnosis Via Statistical Analysis and Evidential Reasoning," *IEEE/ACM Transactions on Networking*, submitted.

- [6] —, “An Automated Policy Based Management Framework for Differentiated Communication Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 12, pp. 2236–2247, December 2005.
- [7] —, “An Automated Policy-Based Management Framework Using Forecasting Functions,” in *the 12th HP-OVUA Workshop, Porto/Portugal*, Jul. 10-13 2005.
- [8] —, “Adaptive Policies for Management of Autonomous communication systems,” in *7th International Symposium on Communications Interworking, Ottawa, Canada*, 29 November - 1 December 2004.
- [9] —, “Autonomic Network Management: A Survey and a New Approach,” *IEEE Communications Surveys and Tutorials*, submitted.
- [10] —, “Prediction Based Policy Adaptation for QoS Management in Wireless Networks,” in *Proceedings of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy*. IEEE Computer Society, Jun. 4-6 2003, pp. 43–51.
- [11] —, “A Multi-Agent Framework for QoS adaptation Using Policies in wireless Environments,” in *“Gestion de Réseaux Et de Services, Fortaleza, Brazil”*, Feb. 24-27 2003.
- [12] N. Samaan, A. Karmouch, and H. Kheddouci, “Mobility Prediction Based Service Location and Delivery,” in *the 17th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2004), Niagara Falls Ontario, Canada*, May. 2-5 2004.
- [13] “ISO 8402: ISO 8402 Quality Management and Quality Assurance: Vocabulary. ISO, 1994. published standard.”

- [14] "ITU-T E.800 (08/94), Terms and definitions related to quality of service and network performance including dependability , CCITT/ITU, Report: Recommendation E.800," Aug. 1994.
- [15] "ISO-QoS, Quality of Service Basic Framework - Outline, ISO/IEC JTC1/SC21/WG1 N1145, International Standards Organisation, UK, 1994."
- [16] E. Crawley, R. Nair, B. Rajagopalan, and H. Sandick, "A Framework for QoS-Based Routing in the Internet, IETF Network Working Group RFC 2386," Aug. 1998.
- [17] F. Fluckiger, *Understanding networked multimedia: applications and technology*. Hertfordshire, UK, UK: Prentice Hall International (UK) Ltd., 1995.
- [18] A. Vogel, B. Kerherve, G. Bochmann, and J. Gecsei, "Distributed Multimedia and QoS - A Survey," *IEEE Multimedia*, vol. 2, no. 2, pp. 10–19, 1995.
- [19] J. Shin, J. W. Kim, and C. Jay Kuo, "Quality-of-Service Mapping Mechanism for Packet Video in Differentiated Services Network," *IEEE Transactions on Multimedia*, vol. 3, no. 2, pp. 217–230, Jun. 2001.
- [20] G. Ghinea, J. P. Thomas, and R. Fish, "Mapping Quality of Perception to Quality of Service: the Case for a Dynamically Reconfigurable Communication System," *Journal of Intelligent Systems*, vol. 10, no. 5/6, pp. 607–632, 2000.
- [21] I. Fukuda, N. Wakamiya, M. Murata, and H. Miyahara, "QoS Mapping between User's Preference and Bandwidth Control for Video Transport," in *5th International Workshop on QoS (IWQoS)*, New York, USA, May 21-23 1997, pp. 291 – 301.
- [22] G. Ghinea and T. J. P., "QoS impact on user perception and understanding of multimedia video clips," in *Proceedings of the sixth ACM international conference on Multimedia*, Bristol, United Kingdom, 1998, pp. 49 – 54.

- [23] E. Bouillet, D. Mitra, and K. G. Ramakrishnan, "The structure and management of service level agreements in networks," *IEEE Journal on, Selected Areas in Communications*, vol. 20, no. 4, pp. 691–699, May 2002.
- [24] R. Steinmetz and K. Nahrstedt, *Multimedia: computing, communications and applications*. Prentice-Hall Series In Innovative Technology, 1995.
- [25] C. Aurrecochea, T. Campbell, A., and L. Hauw, "A Survey of QoS Architectures," *ACM/Springer Verlag Multimedia Systems Journal , Special Issue on QoS Architecture*, vol. 6, no. 3, pp. 138–151, May. 1998.
- [26] H. Lu and I. Faynberg, "An Architectural Framework for Support of Quality of Service in Packet Networks," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 98– 105, Jun. 2003.
- [27] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview, IETF Network Working Group RFC 1633," Jun. 1994.
- [28] S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service, IETF Network Working Group RFC 2212," Sept. 1997.
- [29] J. Wroclawshi, "Specification of the Controlled-Load Network Element Service, IETF Network Working Group RFC 2211," Sept. 1997.
- [30] E. Braden, "Resource Reservation Protocol (RSVP), Functional Specification, IETF Network Working Group RFC 2205," Sept. 1997.
- [31] R. Guerin, S. Herzog, and S. Blake, "Aggregating RSVP-based QoS Requests, Internet Draft, Network Working Group, work in progress," Nov. 1997.
- [32] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services, IETF Network Working Group RFC 2475," Dec. 1998.

- [33] V. Jacobson, K. Nichols, and K. Paduri, "An Expedited Forwarding PHB, IETF Network Working Group RFC 2598," Jun. 1999.
- [34] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group, IETF Network Working Group RFC 2597," Jun. 1999.
- [35] T. Nadeau, C. Srinivasan, and A. Farrel, "Multiprotocol Label Switching (MPLS) Management Overview, IETF Network Working Group RFC 3814," Sept. 2003.
- [36] P. Ruiz, J. Botia, and A. Gomez-Skarmeta, "Providing QoS through machine learning driven adaptive multimedia applications," *Systems, Man and Cybernetics, Part B, IEEE Transactions on*, vol. 34, no. 3, pp. 1398–1411, Jun. 2004.
- [37] P. Bellavista, A. Corradi, and C. Stefanelli, "Application-level QoS control for video-on-demand," *Internet Computing, IEEE*, vol. 7, no. 6, pp. 16–24, Nov.-Dec. 2003.
- [38] I. Busse, B. Deffner, and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP," *Computer Communications*, vol. 19, no. 1, pp. 49–58, Jan. 1996.
- [39] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, "Quality of Service Based Routing: A Performance Perspective," in *SIGCOMM '98: Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM Press, 1998, pp. 17–28.
- [40] T. Korkmaz and M. Krunz, "Routing multimedia traffic with QoS guarantees," *Multimedia, IEEE Transactions on*, vol. 5, no. 3, pp. 429–443, Sept. 2003.
- [41] A. Juttner, B. Szviatovszki, I. Mecs, and Z. Rajko, "Lagrange relaxation based method for the QoS routing problem," in *Proc. IEEE INFOCOM 2001 Conf.*, vol. 2, Apr. 2001, pp. 859–868.

- [42] J. Huang, Y. Wang, and F. Cao, "On Developing Distributed Middleware Services for QoS-and Criticality-Based Resource Negotiation and Adaptation," *Real-Time Syst.*, vol. 16, no. 2-3, pp. 187–221, 1999.
- [43] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek, "A Resource Allocation Model for QoS Management," in *RTSS '97: Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS '97)*. Washington, DC, USA: IEEE Computer Society, 1997, pp. 298–307.
- [44] Z. Zhang, Z. Duan, L. Gao, and Y. Hou, "Decoupling QoS control from core routers: a novel bandwidth broker architecture for scalable support of guaranteed services," in *SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. New York, NY, USA: ACM Press, 2000, pp. 71–83.
- [45] K. Nahrstedt and J. Smith, "Design, Implementation and Experiences of the OMEGA End-Point Architecture," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1263–1279, Sept. 1996.
- [46] D. Hehmann, R. G. Herrtwich, W. Schulz, T. E. Schtt, and R. Steinmetz, "Implementing HeiTS: Architecture and Implementation Strategy of the Heidelberg High-Speed Transport System," in *2nd Intl. Wksp. Network and OS Support for Digital Audio and Video, IBM ENC, Heidelberg, Germany*, 1991.
- [47] J. Zinky, D. Bakken, and R. Schantz, "Architectural Support for Quality of Service for CORBA Objects," *Theory and Practice of Object Systems, John Wiley & Sons, Inc.*, vol. 3, no. 1, pp. 55–73, 1997.
- [48] N. Wang, K. Parameswaran, M. Kircher, and D. C. Schmidt, "Applying Reflective Middleware Techniques to optimize a QoS enabled CORBA Component Model Implemen-

- tation,” in *24th International Computer Software and Applications Conference (COMP-SAC 2000)*, Taipei, Taiwan, 25-28 Oct. 2000.
- [49] Y. Ren, D. Bakken, T. Courtney, M. Cukier, D. Karr, P. Rubel, C. Sabnis, W. Sanders, R. Schantz, and M. Seri, “AQuA: An Adaptive Architecture that Provides Dependable Distributed Objects,” *IEEE Transactions on Computers*, vol. 52, no. 1, pp. 31–50, Jan. 2003.
- [50] I. Cardei, R. Jha, M. Cardei, and A. Pavan, “Hierarchical architecture for real-time adaptive resource management,” in *Middleware '00: IFIP/ACM International Conference on Distributed systems platforms*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2000, pp. 415–434.
- [51] F. Siqueira and V. Cahill, “Quartz: A QoS Architecture for Open Systems,” in *Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS'2000)*, Taipei, Taiwan, Apr. 2000, pp. 197–204.
- [52] T. Engel, H. Granzer, B. F. Koch, M. Winter, P. Sampatakos, I. Venieris, F. Hussmann, H. Ricciato, and S. Salsano, “Adaptive resource control for QoS using an IP-based layered architecture,” *Communications Magazine, IEEE*, vol. 41, no. 1, pp. 46–53, Jan. 2003.
- [53] L. Breslau, E. Knightly, S. Schenker, I. Stoica, and H. Zhang, “Endpoint Admission Control: Architectural Issues and Performance,” *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, pp. 57–69, 2000.
- [54] E. W. Knightly and N. B. Shroff, “Admission control for statistical QoS: theory and practice,” *Network, IEEE*, vol. 13, no. 2, pp. 20–29, Mar.-Apr. 1999.

- [55] C. Cetinkaya, V. Kanodia, and E. Knightly, "Scalable services via egress admission control," *Multimedia, IEEE Transactions on*, vol. 3, no. 1, pp. 69–81, Mar. 2001.
- [56] R. R. Liao and A. T. Campbell, "Dynamic core provisioning for quantitative differentiated services," *Networking, IEEE/ACM Transactions on*, vol. 12, no. 3, pp. 429–442, Jun. 2004.
- [57] R. Yavatkar, R. Guerin, and D. Pendarakis, "A Framework for Policy Based Admission Control, IETF Network Working Group RFC 2753, Informational," Jan. 2000.
- [58] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, M. Huynh, A. Carlson, J. Perry, and S. Waldbusser, "Terminology for Policy Based Management, IETF Network Working Group RFC 3198," Nov. 2001.
- [59] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The COPS( Common Open Policy Service) protocol, IETF Network Working Group RFC 2748," Jan. 2000.
- [60] D. Raz and S. Y., "Active Networks for Efficient Distributed Network Management," *IEEE Communications Magazine*, vol. 38, no. 3, pp. 138–143, Mar. 2000.
- [61] B. Schwartz, A. W. Jackson, W. T. Strayer, W. Zhou, R. D. Rockwell, and C. Partridge, "Smart Packets: Applying Active Networks to Network Management," *ACM Transactions on Computer Systems*, vol. 18, no. 1, pp. 67–88, Feb. 2000.
- [62] R. Kawamura and R. Stadler, "A middleware architecture for active distributed management of IP networks," in *IEEE/IFIP Network Operations and Management Symposium (NOMS 2000), Honolulu, Hawaii, 10-14 Apr. 2000*, pp. 291–304.

- [63] S. Vrontis, I. Sygkouna, M. Chantzara, and E. Sykas, "Enabling Distributed QoS Management Utilizing Active Network Technology," in *IFIP/IEEE International Conference on Network Control and Engineering (Net-Con 2003)*, Muscat, Oman, 11-15 Oct. 2003.
- [64] E. Hyden, "Operating System Support for Quality of Service," *Ph.D. thesis, University of Cambridge Computer Laboratory*, Feb. 1994.
- [65] D. Rosu, K. Schwan, and S. Yalamanchili, "FARA: A Framework for Adaptive Resource Allocation in Complex Real-Time Systems," in *Proceedings of the 4th IEEE Real-Time Technology and Applications Symposium, 1998, Denver, Colorado, USA, IEEE Computer Society*, Jun. 1998, pp. 79–84.
- [66] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6, IETF Network Working Group RFC 3775," Jun. 2004.
- [67] B. Noble, "System Support for Mobile, Adaptive Applications," *IEEE Personal Communications*, vol. 7, no. 1, pp. 44–49, Feb. 2000.
- [68] L. Capra, W. Emmerich, and C. Mascolo, "CARISMA: context-aware reflective middleware system for mobile applications," *Software Engineering, IEEE Transactions on*, vol. 29, no. 10, pp. 929 – 945, Oct 2003.
- [69] J. Bolliger and T. Gross, "A Framework-Based Approach to the Development of Network-Aware Applications," *IEEE Transactions on Software Engineering*, vol. 24, no. 5, pp. 376–390, May 1998.
- [70] A. Talukdar, B. R. Badrinath, and A. Acharya, "MRSVP : A reservation protocol for integrated services packet networks with mobile hosts," *ACM/Baltzer Journal of Wireless Networks*, vol. 7, no. 1, pp. 5–19, Jan 2001.

- [71] J. Manner and K. Raatikainen, "Extended Quality-of-Service for Mobile Networks," in *IEEE/IFIP Ninth International Workshop on Quality of Service (IWQoS 2001)*, Karlsruhe, Germany, Jun. 2001, pp. 275–280.
- [72] C. Q. Shen, W. Seah, A. Lo, H. Zheng, and M. Greis, "Mobility Extensions to RSVP in an RSVP-Mobile IPv6 Framework," in *Internet draft, work in progress*, Jul. 2002.
- [73] S.-C. Lo, G. Lee, W.-T. Chen, and J.-C. Liu, "Architecture for mobility and QoS support in all-IP wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 4, pp. 691 – 705, May 2004.
- [74] R. RAMJEE, T. LA PORTA, S. THUEL, K. VARADHAN, and S. WANG, "HAWAII: a domain-based approach for supporting mobility in wide-area wireless networks," in *Proc. of the 7th International Conference on Network Protocols (ICNP)*, 1999, p. 283–292.
- [75] J. Gomez and A. T. Campbell, "Havana: supporting application and channel dependent QoS in wireless packet networks," *Wireless Networks*, vol. 9, no. 1, pp. 21–35, Jan. 2003.
- [76] P. Agrawal, E. Hyden, P. Krzyzanowski, P. Mishra, M. Srivastava, and J. Trotter, "SWAN: a mobile multimedia wireless network," *Personal Communications, IEEE*, vol. 3, no. 2, pp. 18–33, Apr. 1996.
- [77] V. Bharghavan, K.-W. Lee, S. Lu, S. Ha, J. R. Li, and D. Dwyer, "The TIMELY Adaptive Resource Management Architecture," *IEEE Personal Communications Magazine*, vol. 5, no. 4, Aug. 1998.

- [78] B. Vandalore, R. Jain, S. Fahmy, and S. Dixit, "AQuaFWiN: adaptive QoS framework for multimedia in wireless networks and its comparison with other QoS frameworks," in *Local Computer Networks LCN'99, Lowell, MA, USA*, Oct. 1999, pp. 88–97.
- [79] S. Le, G. Ahn, X. Zhang, and A. Campbell, "INSIGNIA: An IP-based Quality of Service Framework for Mobile Ad Hoc Networks," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 374–406, Apr. 2000.
- [80] Y. Cheng and W. Zhang, "DiffServ resource allocation for fast handoff in wireless mobile internet," *IEEE Communications Magazine*, vol. 40, no. 5, pp. 130–136, May 2002.
- [81] A. Misra, S. Das, A. Mcauley, A. Dutta, and S. K. Das, "Integrating QoS support in TeleMIPs mobility architecture," in *IEEE Int. Conf. Personal Wireless Communications*, 2000, pp. 57–64.
- [82] L. Burness, A. Kassler, P. Khengar, E. Kovacs, D. Mandato, J. Manner, G. Neureiter, T. Robles, and H. Velayos, "The BRAIN Quality of Service Architecture for Adaptable Services," in *the PIMRC 2000, London, United Kingdom*, vol. 1, Sept. 2000, pp. 445–450.
- [83] S. Faizullah and I. Marsic, "An Architecture for Guaranteeing QoS and Charging in Future Internetworks," in *In the IASTED International Symposium on Information Systems and Engineering (ISE2001), Las Vegas, NV*, Jun. 2001, pp. 202–208.
- [84] T. Guenkova-Luy and D. Kassler, A. J. and Mandato, "End-to-end quality-of-service coordination for mobile multimedia applications," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 5, pp. 889–903, Jun. 2004.

- [85] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol, IETF Network Working Group RFC 3261," Jun. 2002.
- [86] E. Gelenbe, R. Lent, and A. Nunez, "Self-aware networks and QoS," *Special Issue IEEE Proceedings*, vol. 92, no. 9, pp. 1478 – 1489, Sept. 2004.
- [87] G. Cortese, R. Fiutem, P. Cremonese, S. D'antonio, M. Esposito, S. P. Romano, and A. Diaconescu, "Cadenus: creation and deployment of end-user services in premium IP networks," *Communications Magazine, IEEE*, vol. 41, no. 1, pp. 54 – 60, Jan. 2003.
- [88] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer Magazine, IEEE*, vol. 36, no. 1, pp. 41–50, Jan. 2003.
- [89] N. Samaan and A. Karmouch, "A novel approach towards autonomic management in context-aware communication systems," in *MATA 2005*, 2005, pp. 374–383.
- [90] M. Nilsson, "Composite Capabilities/Preference Profiles: Terminology and Abbreviations," in *W3C Working Draft*, Jul. 2000. [Online]. Available: <http://www.w3.org/TR/2000/WD-CCPP-ta-20000721>
- [91] L. Razmerita, A. Angehrn, and A. Maedche, "Ontology based user modeling for Knowledge Management Systems," in *Proceedings of the User Modeling Conference, Pittsburgh, USA, Springer-Verlag*, 2003, pp. 213–217.
- [92] M. R. Tazari, M. Grimm, and M. Finke, "Modelling User Context," in *The 10th International Conference on Human-Computer Interaction (HCI2003)*, vol. 2. Lawrence Erlbaum, Jun. 2003, pp. 293–297.

- [93] D. Liu and F. Huebner, "Application Profiling of IP Traffic," in *27th Annual IEEE International Conference on Local Computer Networks (LCN02)*, Nov.06–08 2002, pp. 220–229.
- [94] D. Kettani and B. Moulin, "A Spatial Model Based on the Notions of Spatial Conceptual Map and of Object's Influence Areas," in *Conference On Spatial Information Theory, COSIT 1999, Stade, Germany*, Aug. 1999, pp. 401–416.
- [95] J. Galvin and K. McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2), RFC 1445, Trusted Information Systems, Hughes, LAN Systems," Apr. 1993.
- [96] G. Shafer, *A Mathematical Theory of Evidence*. Princeton Universal Press, 1975, ch. 3.
- [97] R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: Cambridge University Press, 1993.
- [98] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 6, pp. 922–936, Aug. 1998.
- [99] J. Chan and A. Seneviratne, "A Practical User Mobility Prediction Algorithm for Supporting Adaptive QoS in Wireless Networks," in *IEEE International Conference on Networks (ICON'99)*, Sept. 28 - Oct. 1 1999, pp. 104 – 111.
- [100] V. Bharghavan and M. Jayanth, "Profile-based next-cell prediction in indoor wireless LAN," in *Proceedings of IEEE SICON'97, Singapore*, Apr. 1997.
- [101] L. O'Brien, P. Merson, and L. Bass, "Quality attributes and service-oriented architectures," Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMU/SEI-2005-TN-014, 2005.

- [102] W. Caripe, G. Cybenko, K. Moizumi, and R. Gray, "Network awareness and mobile agent systems," *IEEE Communications Magazine*, vol. 36, no. 7, pp. 44–49, Jul. 1998.
- [103] J. Postel, "Internet Control Message Protocol, IETF Network Working Group, RFC 792," Sept. 1981.
- [104] S. Shalunov, B. Teitelbaum, and M. Zekauskas, "A One-way Active Measurement Protocol (OWAMP), IETF Network Working Group RFC 4656," Sept. 2006.
- [105] I. Graham, S. Donnelly, S. Martin, J. Martens, and J. Cleary, "Nonintrusive and Accurate Measurements of Unidirectional Delay and Delay Variation on the Internet," in *INET98*, Jul. 1998.
- [106] I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg, "Argumentation-based negotiation," *The Knowledge Engineering Review*, vol. 18, no. 4, pp. 343–375, Dec. 2003.
- [107] W. S. Soh and H. S. Kim, "QoS Provisioning in Cellular Networks Based on Mobility Prediction Techniques," *IEEE Communications Magazine*, vol. 41, no. 1, pp. 86–92, Jan. 2003.
- [108] A. Aljadhai and T. F. Znati, "Predictive mobility support for QoS provisioning in mobile wireless environments," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 1915–1930, Oct. 2001.
- [109] B. Schmidt-Belz, M. Makelainen, A. Nick, and S. Poslad, "Intelligent brokering of tourism services for mobile users," in *ENTER2002 Annual Conference of the International Federation on Information Technology in Tourism*, Innsbruck, Jan. 2002.

- [110] D. Ashbrook and T. Starner, "Learning Significant Locations and Predicting User Movement with GPS," in *Sixth International Symposium on Wearable Computers, 2002. (ISWC 2002)*, Oct. 2002, pp. 101–108.
- [111] N. Marmasse and C. Schmandt, "A User-Centered Location Model," *Personal and Ubiquitous Computing*, vol. 6, no. 5-6, pp. 318 – 321, Dec. 2002.
- [112] S. Tabbane, "An alternative strategy for location tracking," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 5, pp. 880 –892, Jun. 1995.
- [113] G. Liu and G. Maguire, "A Class of Mobile Motion Prediction Algorithms for Wireless Mobile Computing and Communication," *Special issue: routing in mobile communications networks, ACM International Journal on Wireless Networks*, vol. 1, no. 2, pp. 113–121, Oct. 1996.
- [114] —, "A predictive mobility management algorithm for wireless mobile computing and communications," in *Fourth IEEE International Conference on Universal Personal Communications, 6-10 Nov., 1995*, pp. 268–272.
- [115] A. Bhattacharya and S. K. Das., "Lezi-update: An information-theoretic approach to track mobile users in PCS networks," *Mobile Computing and Networking*, pp. 1–12, 1999.
- [116] X. Shen, J. W. Mark, and J. Ye, "User Mobility Profile Prediction: An Adaptive Fuzzy Inference Approach," *Wireless Networks*, vol. 6, pp. 363–374, 2000.
- [117] V. Kumar and P. Venkataram, "A Prediction based Location Management using Multi-Layer Neural Networks," *Journal of the Indian Institute of Science*, vol. 82, no. 1, pp. 7–21, 2002.

- [118] J. Chan, S. Zhou, and A. Seneviratne, "A Qos Adaptive Mobility Prediction Scheme for Wireless Networks," in *IEEE Global Telecommunications Conference, GLOBECOM 98*, 8-12 Nov. 1998, vol. 3, 1998, pp. 1414–1419.
- [119] E. Cayirci and I. F. Akyildiz, "User mobility pattern scheme for location update and paging in wireless systems," *IEEE Transaction on Mobile Computing*, vol. 1, no. 3, pp. 236–247, Jul.-Sept. 2002.
- [120] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional pcs networks," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 714–732, Oct. 2003.
- [121] K. Wang, J.-M. Liao, and J.-M. Chen, "Intelligent location tracking strategy in PCS," *IEE Proceedings of Communications*, vol. 147, no. 1, pp. 63–68, Feb. 2000.
- [122] J. Orwant, "Doppelganger goes to school: Machine learning for user modeling," Master's thesis, Department of Media Art and Sciences, Massachusetts Institute of Technology, Sept. 1993.
- [123] R. DeVaul, M. Sung, J. Gips, and A. Pentland, "MITHril 2003: Applications and Architecture," in *7th IEEE International Symposium on Wearable Computers, NY, 2003*, Oct. 2003, pp. 4–11.
- [124] A. K. Dey and G. D. Abowd, "The Context Toolkit: Aiding the Development of Context-Aware Applications," in *the Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland*, Jun. 2000.
- [125] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. Wong, "SenSay: A Context-Aware Mobile Phone," in *7th IEEE International Symposium on Wearable Computers (ISWC2003)*, Oct. 2003, pp. 248–249.

- [126] D. Garlan, D. Siewiorek, S. A., and P. Steenkiste, "Project Aura: Towards Distraction-Free Pervasive Computing," *IEEE Pervasive Computing*, vol. 1, no. 2, pp. 22–31, 2002.
- [127] A. K. Dey and G. D. Abowd, "CybreMinder: A Context-Aware System for Supporting Reminders," in *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K)*, Bristol, UK, Sept. 2000, pp. 172–186.
- [128] A. Goker and H. Myrhaug, "User Context and Personalisation," in *European Conference on Case Based Reasoning, Workshop on Case Based Reasoning and Personalization, invited paper*, Sept. 2002.
- [129] M. Khedr and A. Karmouch, "Negotiating Context Information in Context Aware Systems," *IEEE Intelligent Systems Magazine*, vol. 19, no. 6, pp. 21–29, Nov.-Dec. 2004.
- [130] A. P. Dempster, "A Generalization of Bayseian Inference," *Journal of Royal Statistics Society Series*, vol. 30, pp. 205–247, 1968.
- [131] T. Chen and V. Venkataramanan, "Dempster-shafer theory for intrusion detection in ad hoc networks," *IEEE Internet Computing*, vol. 09, no. 6, pp. 35–41, 2005.
- [132] K. Sentz, "Combination of Evidence in Dempster-Shafer Theory," Ph.D., Systems Science and Industrial Engineering Department, Binghamton University, 2002.
- [133] P. Senevarante and J. Morall, "Analysis of Factors Affecting The Choice of Route of Pedestrians," *Transportation Planning and Technology*, vol. 10, no. 1, pp. 147–159, 1986.
- [134] E. W. Dijkstra, " A note on two problems in connexion with graphs," *Numer.Math.*, vol. 1, pp. 269–271, 1959.

- [135] N. Verlander, , and B. Heydecker, "Pedestrian Route Choice: An Empirical Study," in *Proceedings of 25rd PTRC European Transport Forum, Brunel University, Sept. 1997*, pp. 39–49.
- [136] A. Josang, "A logic for uncertain probabilities," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 9, no. 3, pp. 279–311, 2001.
- [137] E. Belogay, C. Cabrelli, U. Molter, and R. Shonkwiler, "Calculating the Hausdorff distance between curves," *Information Processing Letters*, vol. 64, no. 1, pp. 17–22, 1997.
- [138] M. Bauer, "Approximation algorithms and decision making in the Dempster-Shafer theory of evidence : An empirical study," *International Journal of Approximate Reasoning*, vol. 17, no. 2-3, pp. 217–237, 1997.
- [139] J. D. Case, M. Fedor, M. L. Schoffstall, and C. Davin, "Simple network mangement protocol (SNMP), IETF Network Working Group RFC 1157," May 1996.
- [140] S. Waldbusser, "Remote Network Monitoring Management Information Base for High Capacity Networks, IETF Network Working Group RFC 3273, Jul. 2002."
- [141] C. Corporation, "Netflow services and applications. <http://www.cisco.com/>, 2000."
- [142] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," New York, NY, USA, pp. 219–230, 2004.
- [143] J. D. Brutag, "Aberrant behavior detection and control in time series for network monitoring," in *14th Systems Administration Conference (LISA 2000), (New Orleans, LA, USA), USENIX*, Dec. 2000.
- [144] M. Thottan and C. Ji, "Anomaly detection in ip networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2191– 2204, Aug. 2003.

- [145] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *in ACM SIGCOMM Internet Measurement Workshop, (Marseilles, France)*, Nov. 2002.
- [146] V. Alarcon-Aquino and J. A. Barria, "Anomaly detection in communication networks using wavelets," *IEE-Proceedings-Communications.*, vol. 148, no. 6, pp. 355–62, Dec. 2001.
- [147] K. Yamanishi and J. ichi Takeuchi, "A unifying framework for detecting outliers and change points from non-stationary time series data," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2002, pp. 676–681.
- [148] M. Mandjes, I. Sanicee, and A. Stolyar, "Load characterization and anomaly detection for voice over ip traffic," *Neural Networks, IEEE Transactions on*, vol. 16, no. 5, pp. 1019 – 1026, Sept. 2005.
- [149] A. Lakhina, M. Crovella, and M. Diot, "Characterization of network-wide anomalies in traffic flows," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. New York, NY, USA: ACM Press, 2004, pp. 201–206.
- [150] e. Liebowitz, *Expert System Applications to Telecommunications*. John Wiley and Sons, New York, NY, USA, 1988.
- [151] Y. A. Nygate, "Event correlation using rule and object based techniques," in *IFIP/IEEE Int. Symp. on Integrated Network Management IV*, 1995, pp. 278–289.
- [152] E. Tang, Y. Al-Shaer and R. Boutaba, "Active integrated fault localization in communication networks," in *IFIP/IEEE International Symposium on Integrated Network Management (IM'2005)*, 15-19 May 2005, pp. 543– 556.

- [153] M. S. Kim, H. J. Kang, S. H. Hung, S. C. Chung, and J. W. Hong, "A flow-based method for abnormal network traffic detection," in *IEEE/IFIP Network Operations and Management Symposium, Seoul*, Apr. 2004.
- [154] C. Manikopoulos and S. Papavassiliou, "Network intrusion and fault detection: a statistical anomaly approach," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 76 – 82, Oct, 2002.
- [155] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowledge and Information Systems, Springer*, vol. 6, no. 5, pp. 507 – 527, Sept. 2004.
- [156] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, 2005, pp. 217–228.
- [157] A. Eskin, E. nd Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data," in *Data Mining for Security Applications, Kluwer, Boston, MA.*, 2002.
- [158] M. Basseville and A. Benveniste, "Sequential detection of abrupt changes in spectral characteristics of digital signals," *Information Theory, IEEE Transactions on*, vol. 29, no. 5, pp. 709– 724, Sept. 1983.
- [159] C. Hood and C. Ji, "Intelligent agents for proactive fault detection," *IEEE Internet Computing*, vol. 2, no. 2, pp. 65–72, 1998.
- [160] M. B. Priestley, *Spectral Analysis and Time Series*. London, U.K.: Academic, 1981, vol. I and II.

- [161] Z.-S. Liu, "Qr methods of  $o(n)$  complexity in adaptive parameter estimation," *Signal Processing, IEEE Transactions on*, vol. 43, no. 3, pp. 720–729, Mar 1995.
- [162] T. Denoeux, "A k-nearest neighbor classification rule based on dempster-shafer theory," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 804 – 813, May 1995.
- [163] F. Probst, "Machine learning from imbalanced data sets 101," in *Invited paper for the AAAI'2000 Workshop on Imbalanced Data Sets*, 2000.
- [164] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behaviour and performance," *IEEE Trans. on Syst. Man and Cyebrrn., Part A: Systems and Humans*, vol. 27, pp. 553 – 568, 1997.
- [165] E. Walker, W., "Adaptive policies: An approach for dealing with structural uncertainty in public policy making," in *The 5th International Conference on Technology, Policy and Innovation, Netherlands*, Jun. 2001.
- [166] B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, "Core Information Model, <http://www3.ietf.org/proceedings/99nov/I-D/draft-ietf-policy-core-info-model-02.txt>, Internet Draft, Oct. 1999."
- [167] Y. Snir, Y. Ramberg, J. Strassner, and R. Cohen, "Policy Quality of Service Information Model, IETF Network Working Group RFC 3644," Nov. 2003.
- [168] M. D. J. Cox and R. G. Davison, "Concepts, activities and issues of policy-based communications management," *BT Technol journal*, vol. 17, no. 3, pp. 155–169, Jul. 1999.
- [169] L. Lymberopoulos, E. Lupu, and M. Sloman, "An Adaptive Policy Based Management Framework for Differentiated Services Networks," in *IEEE 3rd International Workshop*

*on Policies for Distributed Systems and Networks (POLICY'02)*, Monterey, California, Jun. 2002, pp. 147–158.

- [170] Z. Granville, L., A. Faraco de S Coelho, G., M. Almeida, and L. Tarouco, “An Architecture for Automated Replacement of QoS Policies,” in *The 7th IEEE Symposium on Computers and Communications (ISCC02)*, Italy, Jul. 2002.
- [171] K. Yoshihara, M. Isomura, and H. Horiuchi, “Distributed Policy-based Management Enabling Policy Adaptation on Monitoring using Active Network Technology,” in *12th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'01)*, Nancy, France, Oct. 2001.
- [172] I. Marshall and C. Roadknight, “Provision of Quality of Service for Active Services,” *Computer Networks*, vol. 36, no. 1, pp. 75–85, Jun. 2001.
- [173] T. Hamada, P. Czezowski, and T. Chujo, “Policy-based Management for Enterprise and Carrier IP Networking,” *FUJITSU Scientific and Technical Journal*, vol. 36, no. 2, pp. 128–139, Dec. 2000.
- [174] T. Ahmed, R. Boutaba, and A. Mehaoua, “A Measurement-Based Approach for Dynamic QoS Adaptation in DiffServ Network,” *Journal of Computer Communications, Special issue on End-to-End Quality of Service Differentiation*, Elsevier Science, vol. 28, no. 18, pp. 2020–2033, 2005.
- [175] N. Damianou, E. Dulay, and M. SLoman, “The Ponder Policy Specification Language,” in *IEEE 2nd International Workshop on Policies for Distributed Systems and Networks (POLICY'01)*, Bristol, UK, Jan. 2001, pp. 18–39.

- [176] A. Bandara, E. Lupu, J. Moffet, and A. Russo, "A Goal-based Approach to Policy Refinement," in *5th IEEE Wrkshp on Policies for Distributed Systems and Networks (Policy 2004)*, New York, USA, Jun. 2004.
- [177] W. E. Leland, M. S. Taqq, W. Willinger, and D. V. Wilson, "On the Self Similar Nature of Ethernet Traffic," in *ACM SIGCOMM93*, Sept. 1993, pp. 183–193.
- [178] S. G. Makridakis, S. Wheelwright, and R. J. Hyndman, *Forecasting: Methods and Applications*, 3rd ed. John Wiley and Sons, Inc, 1998.
- [179] N. Papagiannaki, K. amd Taft, Z. Zhang, and C. Diot, "Long Term Forecasting of Internet Backbone Traffic: Observations and Initial Models," in *ACM SIGCOMM93*, Apr. 2003.
- [180] H.-Y. Tyan and C.-J. Hou, "JavaSim on-line manuals and tutorials."
- [181] J. H. and S. Jordan, "A Pricing Model for High-Speed Networks with Guaranteed QoS," in *IEEE INFOCOM*, Apr. 1996, pp. 88–95.
- [182] J. Kolodner, *Case-based reasoning*. Morgan Kaufmann Publishers Inc., 1993.
- [183] A. Aamodt and E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," *AI Communications*, vol. 7, no. 1, pp. 39–59, 1994.
- [184] R. Bergmann and P. Cunningham, "Acquiring Customers' Requirements in Electronic Commerce," *Artif. Intell. Rev.*, vol. 18, no. 3-4, pp. 163–193, 2002.
- [185] M. Goker and T. Roth-Berghofer, "Development and Utilization of a Case-Based Help-Desk Support System in a Corporate Environment," in *3rd Intl Conf on Case-Based Reasoning, ICCBR-99, Seeon Monastery, Germany, Jul., 1999*.

- [186] N. Samaan and A. Karmouch, "Management in Differentiated networks via an Adaptable Case-Based Policy Framework," in *poster in the 11th HP-OVUA Workshop, Paris*, Jun. 20-23 2004.
- [187] —, "A Case-Based Reasoning Approach for Automated Management in Policy-Based Network," in *in 15th IFIP/IEEE Distributed Systems: Operations and Management (DSOM2004), Davis, California, USA*, Nov. 15-17 2004.
- [188] —, "Circumscriptive context reasoning for automated network management operations," in *IEEE Globecom 2006, General Conference on Emerging Technologies - Autonomous Communication/Networking Principles*, Nov.27-Dec.1 2006.