EXTENDED POLICY-BASED MANAGEMENT FRAMEWORK TO PROVIDE ALWAYS BEST CONNECTED SERVICES IN HETEROGENEOUS MOBILE ENVIRONMENTS

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Doctor of Philosophy
in the
University of Canterbury
by
Mayank Keshariya

Examining Committee

Doctor Winston Seah	Examiner
Doctor Le Phu Dung	Examiner
Associate Professor Ray Hunt	Supervisor
Doctor Richard Pascoe	Associate Supervisor

UNIVERSITY OF CANTERBURY
2009

To my parents...

Acknowledgments

This dissertation would not have been possible without the help of many people. I would like to take this opportunity to express my deep appreciation to all those who helped me in this journey.

First and foremost, I would like to thank my supervisor, Associate Professor Ray Hunt, for his support and guidance throughout my research. He gave me the freedom to pursue my ideas and constantly encouraged me by refining my research direction through his evaluations and feedback.

I am thankful to my co-supervisor Dr. Richard Pascoe, whose constant guidance has helped me to shape my research work.

I am deeply grateful to my parents, for their invaluable support. They have stood by me every step of the way and have always supported me through their guidance and encouragement.

A special thank you to my wife Nadine, for her understanding and loving support throughout the course of my research.

And to my beautiful son, Ari, little did he know that a “wa wa bliff” went a long way when a smile was needed.

Special thanks to my sisters Astha and Abhilasha, for their unwavering support. I could always rely on them for friendship and advice or for a joke or two when needed.

I am thankful to Mr. Prasan De Silva for his knowledge and support during the course of the Technology Industry Fellowship (TIF). His help in gathering real world requirements, to be incorporated in this work, is invaluable.

I would also like to thank Dr Malcolm Shore for his technical support during my interactions with Telecom.

I am thankful to MediaLab and Mr. Phil Shepherd for his valuable help during the Technology Assessment Project (TAP). This project shaped our ABC client model from the user’s perspective. Regular interactions with the TAP project manager, Mr. Mishul Prasad was of great help and fun.

I am thankful to my employer Mr. Bob Ward and R A Ward Limited for providing a flexible and learning work environment. I am thankful for the study leave and support the company provided during the writing of this thesis.

During this work I have enjoyed being part of a productive infrastructure and have collaborated with many colleagues for whom I have great regard. In enabling me to do this I wish to thank the University of Canterbury for providing the Internet Security Laboratory facility and the opportunity to attend and take part in seminars and conferences.

Finally, research funding plays an extremely important role in a graduate student’s life, and I was no different. This research was funded by the Technology Industry Fellowship (TIF) and the University of Canterbury Doctoral Scholarship for which I am indebted.

Abstract

The growth in the popularity of Internet services, increasing demands of mobile users together with a wide range of access technologies and mobile-networked devices, introduces the notion of integration and inter-working of heterogeneous access networks. Sometimes referred to as 4th generation (4G) networks, the overall objective of this research is to provide a managed Always Best Connected (ABC) service over underlying heterogeneous wireless and mobile platforms while maintaining negotiated security and Quality of Service in a scalable and modular environment.

This research proposes a new model and its architecture for policy-based management (PBM) to provide a framework for the centralised management of networks based on business-level policies. This work extends existing IETF Policy-based Network Management (PBNM) model by introducing a new layered-approach which facilitates the negotiation of management services over interconnected heterogeneous mobile platforms, thus achieving an ABC scenario.

The proposed layered-approach provides flexibility to the organisations so that they can choose favourable semantic and syntactic approaches and facilitates the separation of management policies from their implementation in a distributed and heterogeneous environment. The extended Policy Information Model and a new policy conflict detection technique are also introduced.

Further, we have proposed and implemented a new model of a policy-managed mobile client and its architecture to support seamless handoff across multiple access networks. The proposed mobile client supports multi-domain authentication and security along with downloadable user profiles. We have also proposed and implemented a network selection algorithm and introduced a new *Infrastructure* parameter, which assists in selecting an optimum time and the best available access network to handoff. We present performance analysis to validate our architectural approach.

Research Papers Presented/Published

- Keshariya, M. and Hunt, R. A New Architecture for Performance-based Policy Management in Heterogeneous Wireless Networks, ACM International Workshop on Performance and Analysis of Wireless Networks, Ilan, Taiwan, 2008.

- Keshariya, M. and Hunt, R., Extended Policy-Based Management Framework to Provide Always Best Connected Services in Heterogeneous Mobile Environments, Proceedings of the Fourth IASTED Asian Conference on Communication Systems and Networks, AsiaCSN2007, 2-4 April, 2007, Thailand.

- Hunt, R. and Keshariya, M., Optimised Transparent and Automated Handoff Between Wireless LANs and 3G Networks, Proceeding of the 3G World Congress, Hong Kong, 14-18 November 2005.

- Keshariya, M. and Hunt, R., Implementation of an Integrated Secure Mobile Wireless Architecture, Proceedings of IASTED NCS2005 - Networks and Communication Systems, 18-20 April, 2005, Krabi, Thailand.

- Hunt, R. and Keshariya, M., Proposal For An Automated Algorithm For Optimised Handoff In Integrated Wireless Networks, Proc. of IASTED NCS2005 - Networks and Communication Systems, 18-20 April, 2005, Krabi, Thailand.

Table of Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

The growth in the popularity of Internet services, increasing demands of mobile users together with a wide range of access technologies and mobile-networked devices, introduces the notion of integration and inter-working of heterogeneous access networks. Sometimes referred to as 4th generation (4G) networks, the overall objective of this research is to provide managed Always Best Connected (ABC) service over underlying heterogeneous mobile environments in a modular and scalable environment.

1.1. Background and Motivations

There have been many changes in the face of computing in the past decade, one of which is undoubtedly the introduction of mobile computing. Over the past few years, mobile computing has become widely accepted by the general public as a useful productivity tool, with wireless hotspots becoming more and more common. The ability to travel whilst accessing the Internet provides the opportunity to increase productivity and makes Internet access easier and more flexible.

This wide acceptance of mobile technology originated from its unparalleled ability to provide ubiquitous access and low prices regardless of underlying networking technologies, allowed the rapid development of the Internet and Internet Protocol (IP)-based technologies in private and corporate areas. The IP layer is considered as the unique glue to bridge diverse applications and telecommunication technologies with changing user requirements.

1.1.1. **History**

Since the early 90's, the problem of feature interactions has been recognized as crucial in the design and implementation of telecommunications distributed systems. As distributed systems evolve in terms of features, technology, complexity, and size, the feature interaction problems expand, become distributed and more complex requiring sophisticated solutions. Fundamentally, feature interactions arise in the creation, maintenance, and evolution of new services (e.g. telephony, electronic commerce, Web services, multimedia, mobile services, etc.) and in the implementation of these services across distributed, sometimes heterogeneous, platforms.

The current industrial trend is towards merging different existing communications technologies (such as videoconferencing, email, Voice over Internet Protocol), with new technologies (such as WiFi, Digital Subscriber Line, 3G), and device control, together with a move to greater mobility (e.g. wireless communications, mobile telephony, and ad hoc networking). As a result, the end user will be an always connected entity. However, users might not always wish to be disturbed, or at least not by everyone or for any type of inquiry. Future services need to provide support for users to control their *availability*, and availability is highly dependent on the context of the user.

Various research initiatives such as Next Generation Internet (NGI), Canada Advanced Internet Developments (CANARIE), Internet2, etc are progressing to provide unlimited bandwidth for Internet users, while in parallel, based on the conventional Internet architecture, the Internet Engineering Task Force (IETF) is performing a “bottom-up” development of Internet protocols and techniques, to fulfil upcoming requirements from applications, users and providers.

However, developing and deploying any network services, i.e. services which operate on the IP layer, through best practice and standardisation is too slow, and cannot match the rate at which requirements of various applications are growing. Examples of such services are transparent sessions and always connected to the best interface available, signalling for Quality of Service (QoS), reliable connectivity and security. Similar to the intelligent network architecture in the Public Switched Telephone

Network (PSTN), the current Internet architecture needs to be enhanced in order to allow for a more rapid introduction of such services.

1.1.2. Evolution

Telecommunication Standardisation Sector [1] defined that the network management consists of a set of functions required to control, plan, assign, deploy, coordinate and monitor telecommunication network resources, including functions such as planning the initial network, assign frequencies, route traffic to support load balancing, cryptographic key distribution and authorisation, configuration management, fault management, security management, performance management and accounting management. The complexity of network management tasks lie in the fact that the managed components have evolved from an isolated, homogenous and controllable set of systems to a large, heterogeneous and distributed communication environment.

Being faced with such challenges several standards have been specified with the goal of supporting cross-system, multi-vendor networks. Management communication frameworks like the Simple Network Management Protocol (SNMP) [2] and Common Management Information Protocol (CMIP) [3] have dominated from the early days of network management. The emergence of advanced technologies, such as Common Object Request Broker Architecture (CORBA) [4] and the Remote Method Invocation (RMI) [5], addressed the distribution of software environments and interoperability of systems. These technologies ease the development of more open, interoperable, flexible, and scalable management architectures. However, there are still steps to be taken when it comes to the management of networks that are required to cater for continuously changing requirements of the users and in particular roaming users and consequently their expectations from a management platform [6].

1.1.3. Motivation

Since the beginning of the 1990’s, the concept of policy-based network management was considered as a panacea for these problems [7] and has attracted significant industry interest [8]. Currently, it is promoted by several network equipment vendors in the form of forums like Distributed Management Task Force (DMTF) [9] or is standardised within the IETF Policy Framework group [10]. Policy-Based Network

Management (PBNM) opened a new window of opportunity to operators as it enables them to homogenously perform their network management tasks, raise the level of interoperability across different vendors' equipment thereby creating a new range of different customisable services.

These advantageous properties of PBNM technologies for managing networks have inclined us to use this technology in our proposed solution. The proposed research aims at building a framework for seamless inter-working of heterogeneous access networks upon which users can enjoy always best connected services. In the following sub-sections we will provide an overview of heterogeneous networks, always best connected services and policy-based network management technology.

1.2. **Overview of Heterogeneous Networks**

With the widespread deployment of second-generation (2G) cellular systems, such as Global System for Mobile Communications (GSM) during the 1980s and roaming agreements between GSM operators, a mobile user can stay connected for major geographical areas, while reachable through the same phone number. This concept introduced the notion of being *always connected*. The traditional mobile GSM networking applications focus on ensuring continuity of services under the assumption that devices connect to the infrastructure using a single access technology.

In recent years, General Packet Radio Service (GPRS) has been deployed to give IP connectivity to GSM users, and third-generation (3G) cellular systems, such as Universal Mobile Telecommunications System (UMTS) and Code Division Multiple Access (CDMA2000), are enhanced for mobile Internet solutions. In parallel with the evolution of cellular systems, a number of other access technologies have emerged. Wireless local area networks (WLANs) like IEEE 802.11b deployed in hotspot areas provides Internet access at offices, airports, hotels, and conference centres. WLAN technology is also commonly used in home environments, being connected to fixed access networks such as digital subscriber lines (DSL) or cable modems.

Considering the three lowest layers of the OSI model, it is clear that the network connectivity is based on a significant number of heterogeneous technologies, which are integrated at the IP layer. Technologies such as 802.11x (WLAN) standards, 3G (CDMA 2000, UMTS), Wideband-CDMA and Broadband Wireless Access Standards (802.16) are commonly used for network access (Figure 1-1).

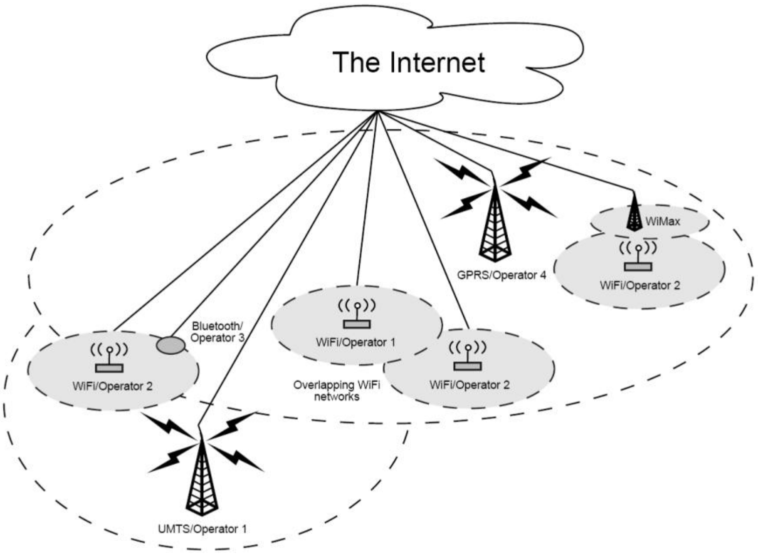


Figure 1-1 Overview of heterogeneous networks

The motivation for heterogeneous networks arises from the fact that no single technology, service or architecture can provide ubiquitous coverage and high throughput across all geographical areas. These access technologies vary in bandwidth, delay, communication range, power consumption, security, reliability, implementation complexity, end-user cost and several other aspects. For example, while 2G cellular systems evolve into 3G systems such as UMTS or CDMA2000 providing worldwide coverage, wireless LAN solutions have been extensively deployed to provide hotspots of high-bandwidth Internet access. At the same time, fixed access networks such as DSL and cable modems tied to wireless LANs appear in home and office environments.

There is a consensus that the next generation telecommunication infrastructure will consist of a set of partially overlapping heterogeneous access networks, where each network provides different coverage, price and performance. With the current trend of mobile communication systems, heterogeneous access networks (such as GSM, GPRS, WLAN, CDMA2000, UMTS, etc.) having different characteristics will coexist in a complementary manner in an all-IP based heterogeneous mobile environment, also referred to as the 4G wireless system [11] [12]. Inexpensive and higher bandwidth WLAN (WiFi) connectivity will for example be available with limited range in *hotspot* areas and will complement more traditional cellular connectivity offering wide area coverage, but with reduced bandwidth and being slightly costlier. If the multiple access networks are present, mobile users will have a choice of how to access the Internet through the *best* available network, i.e. the network offering the best price-performance ratio while being always connected, hence introducing an always best connected concept.

1.3. Overview of Always Best Connected (ABC) Services

An *Always Best Connected* (ABC) service is one of the many provided by an integrated collection of heterogeneous networks that is favoured by a user for a particular set of circumstances. The ABC concept (Figure 1-2) allows a user to connect to applications using the devices and access networks that best suits his or her needs, thereby combining the features of technologies such as DSL, Bluetooth, WLAN and cellular systems to provide an enhanced user experience. The existence of multiple access networks opens up the possibilities of selecting or sharing appropriate access networks to ensure that all applications receive an acceptable QoS. For example, when a network condition changes, an application may be transparently handed over to some other better available option.

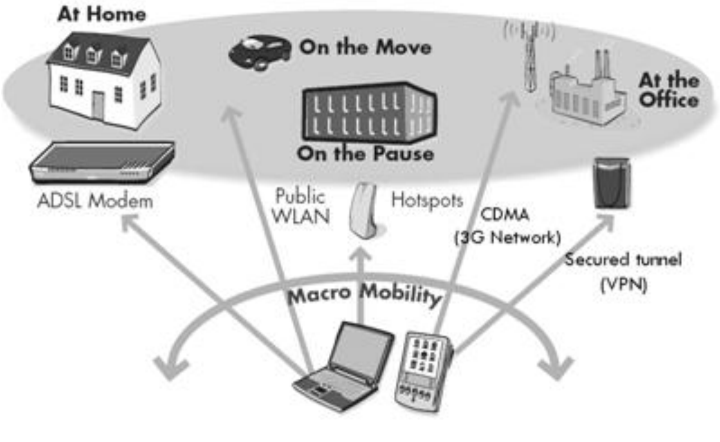


Figure 1-2 Always Best Connected concept

This notion of “Always Best Connected” enables users to be able to connect seamlessly to the network in a way that best suits their application needs. The main requirement for offering such service is to maintain a transparent handover of a mobile device when it is moving from one network to another, while maintaining negotiated QoS and security. The notion of “best” is often based on a number of user and application dependent factors – such as personal preferences, device capabilities, application’s QoS needs, cost, security, available network resources and network coverage [13].

1.4. Overview of Policy-based Management – A centralised solution

The main problem with offering such always best connected service is that different access networks are not particularly integrated and users in most cases are forced to manually interact with the system when switching between networks, including reconfiguring their mobile device with security credentials for the new network, restarting running applications, and if demanded by running applications, manually adjusting bandwidth utilisation. Ideally, Internet connectivity should be easy to use and applications should automatically adapt to current bandwidth and security constraints.

However, to provide access to any roaming mobile user, a service providing network needs to authenticate and authorise the user before they are given access to the requested network resources. The administrators also need to dynamically manage their network and to be able to adapt according to the current network conditions to continually provide agreed services to their home and roaming users. Furthermore, network administrators need to cater for changing requirements of network users (home and guest) and be able to negotiate service levels with other service providers.

Hence, management of a network is a two facet problem. Managing internal networks requires *intra-domain* management and configuration rules to manage all the network elements and users within the network; while managing external networks require *inter-domain* agreements and negotiation rules between service providers. Hence, a centralised network management framework is required to manage internal networks and to provide methods for interaction with external networks.

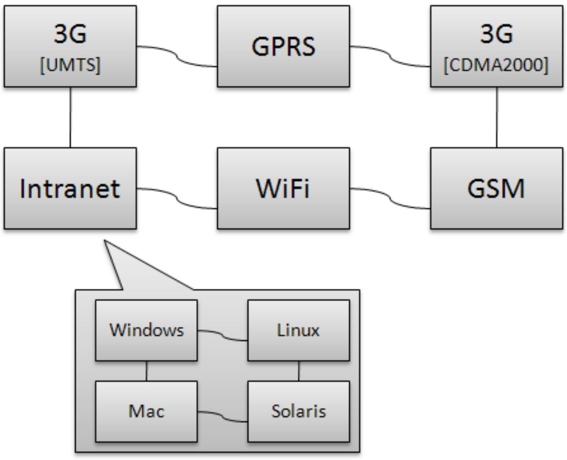


Figure 1-3 Management of network - Two facet problem

Policy-Based Management (PBM) has emerged as a promising solution for a centralised management of networks and distributed systems. In comparison with previous traditional network management approaches such as TMN (Telecommunications Management Network) or TINA-C (Telecommunications Information Networking Architecture Consortium), PBM focuses on users and applications rather than devices and interfaces [14].

PBM aims to provide system administrators a centralised administration window for managing their networks, with the flexibility to adapt to changing requirements by changing the rules that govern the behaviour of network elements without having to recode functional components of the system. It provides a way to allocate network resources, primarily network bandwidth, QoS, and security, according to defined business policies. Policy definitions are a response to questions such as:

- Who and what can access which resources on the network?
- What are different priorities of traffic?
- What traffic must have guaranteed delivery and relative bandwidth allocation?
- What steps are to be performed when any active network element fails?

It allows the administrator to define rules based on these types of questions. Network administrators can manage and configure network devices with simple commands similar to the business policies. The policy framework then translates these policies into configuration rules and automatically implements configurations across the network (Figure 1-4).

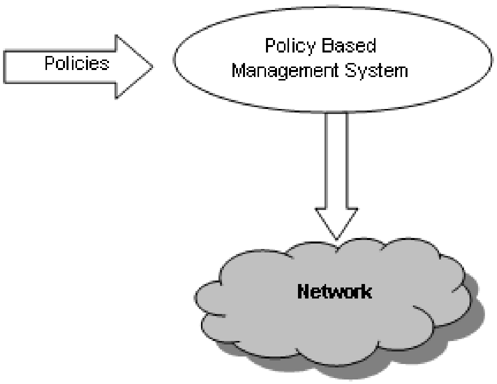


Figure 1-4 Overview of Policy based management system

PBM systems are well suited for large networks where large numbers of devices are easier to manage from a central location. Public networks can also use a form of policy management to allocate resources, but resource allocation is based on service agreements established with network's users. Policies are particularly suited for delegating management responsibility which is essential to enable the customisability of network resources. Policies also permit a more automated and distributed approach to management by making decisions based on current network conditions and selecting the most applicable sets of rules [15]. The device independent property of policies is optimum for the management of heterogeneous network technologies.

These functionalities of policy based networking have influenced our approach for our proposed research idea of providing always best connected services in a heterogeneous mobile environment.

1.5. **Problem Statement**

It is inevitable that future network environments will not consist simply of one access technology but will integrate multiple access technologies, adding complexity to mobility management systems. Users will want to be connected at all times, and preferably with the best access network available. The seamless inter-networking required to provide ABC services will likely become a basic feature in mobile terminals to allow connectivity in heterogeneous environments.

To achieve this ABC scenario, the mobile devices need more intelligent solutions to offer seamless connectivity to mobile users while network service providers need to be able to dynamically manage themselves so as to provide requested services in real-time and simultaneously satisfy agreed service level agreements with their home and guest users.

The following section discusses the requirements imposed for centralised management of networks and management of mobile entities (mobile users and mobile devices).

1.5.1. Requirements for centralised management of networks

A typical enterprise network system consists of a large number of heterogeneous network devices such as routers, and servers running a variety of applications and offering services to a large number of users. Today, state-of-the-art network systems are becoming increasingly complex in terms of:

- Multiple access methods: wired/wireless; PC/PDA/Cell Phone
- New applications with new challenges: Peer-to-Peer (P2P), large scale multimedia applications
- Multiple vendors and different hardware/software-platforms
- Growing demands for QoS for time sensitive applications such as VoIP
- Growing demands of mobile users for mobility and uninterrupted services
- Addressing the increasing number of intrusions and offer network security

Hence, network administrators need support in order to handle these issues more easily. They need:

- Better security tools and, moreover, a better coordination among these tools
- A reliable/guaranteed answer to the question: “Will my network fulfil the desired needs?”
- An automated support for handling alerts within the networks; a proactive system that continuously monitors the networks and provides self-learning mechanisms through handled alerts, attacks and other critical events.

While it is advocated that PBM systems become the pillar technology of self-managing and autonomic computing systems, problems with the paradigm itself are being debated in the policy community. The complexity of the managed systems results in high administrative costs and long deployment cycles for business initiatives, and imposes two requirements on their management systems. Although these requirements have long been recognised, their importance is now becoming increasingly critical: (i) management must be distributed in order to be scalable and to be able to cope with the size of enterprise networks, and (ii) management procedures must be automated to reduce administrative cost. Manual management is expensive and the effort and time needed for management increases exponentially as a system expands.

Furthermore, while defining policies for distributed, large enterprises two issues need consideration: (i) the model (or language) adopted for representing policies and (ii) approach adopted for interpreting, distributing and enforcing policies to the network. There are other important issues in PBM, including policy specification and refinement, policy analysis, conflict detection and resolution, policy enforcement and policy negotiation. All the requirements outlined above demand for a modular and scalable policy framework to be designed.

1.5.2. **Requirements for management of mobile entities**

As users become more mobile and change their point of attachment to the network more frequently, the problem of automation of management is becoming more and more critical. Users want to have access to resources no matter when, and how, and with the constant availability of services. To achieve this goal, systems have to adapt to constantly changing situations. Static and long-term configuration of the network cannot be used in this case.

Furthermore, to realise the mobility requirements of mobile entities, the network service providers need to:

- Authenticate and authorise roaming mobile entities before they are given access to the network resources

- Be able to download the current profile of a roaming user and be able to negotiate the service level with its home network
- Dynamically manage their networks to continually provide agreed services to their home and guest users
- Be able to grant/deny users real-time requests
- Minimise administrative message sent over wireless links. The link by which a mobile device is directly attached to the Internet may often be a wireless link, which has substantially lower bandwidth and higher error rate than traditional wired links. Moreover, mobile devices are likely to be battery powered, and minimising power consumption is important. Therefore, the number and size of administrative messages sent over the link by should be minimised.

We have also identified other requirements which introduce the complexity in providing always best connected services to roaming mobile entities:

- With every handover of a mobile entity from one network to another, both the mobile entity and service providers are susceptible to attacks. Therefore, all the messages which are used to update information must be authenticated in order to protect against remote redirection, man-in-the-middle or rogue service provider attacks.
- Connection setup needs to be fast to minimise disruption to the mobile user's current sessions
- Support for multi-domain security, authentication and authorisation models
- A procedure where a mobile user can request other services (such as higher bandwidth, voice/video services) on a temporary or subscription basis

Considering all the requirements of the mobile entity, a *policy-managed mobile client* is required which can interact with the service providing networks and control the behaviour of the mobile device and the mobile user to be in harmony with the network's offerings for those services.

1.6. **Objective of Research**

Many communities (such as IETF, DMTF, Object Management Group, TMF, etc.), academia (Imperial College University of London, etc.) and industry (HP, Allot, Tivoli, Cisco, etc.) are focusing on general frameworks, languages, or easy-to-use products, respectively, for using policy-driven mechanisms to cope with the requirements of centralised management of network. While these efforts are useful in their own aspects, we will focus on providing a flexible and modular integration of access network technologies to offer managed ABC services to support seamless roaming of mobile users to access offered services from any device in an integrated heterogeneous mobile environment.

Following the hypotheses that inter-working of heterogeneous access networks can provide ABC services more efficiently than tight integration or single network approaches, the main objective of this research is:

> *"To provide managed Always Best Connected (ABC) service over underlying heterogeneous mobile environments in a modular and scalable environment"*

To achieve this objective, we have presented:

> *A proposal of a management framework for the management of heterogeneous networks supported on the proposed extended policy-based management system model and facilitates to provide always best connected services to its users*, and
>
> *A proposal for a policy-managed mobile client framework for the management of the client-side for always best connected services which interacts with the service providing network and controls the behaviour of the mobile entity*

There are three main aspects of this proposed research.

- *Managed Networks*: Each network should be able to provide the best possible services (resource utilisation, adaptable and dynamically managed)
- *Intelligent Mobile Terminals*: To move between systems along with the capability of choosing the best available options

- *Service Level Agreement*: If a mobile user changes service from one system to another, managed network systems must continue to honour service agreements with their user, which implies that the network must support agreed QoS, mobility, authorisation and security.

Hence, the proposed *management framework* for management of heterogeneous networks must exhibit the following characteristics:

i. The proposed framework must be modular to leverage the proof-of-concept of existing technology and research approaches.

ii. The proposed framework should be flexible. It must be able to cope with different underlying devices, technologies and services.

iii. The proposed framework must be adaptable to support the dynamic extension of management functionality.

iv. The proposed framework should be able to view the network and users as a set of entities rather than individual devices.

v. The proposed framework must be scalable in both model and architecture views to be applicable for large enterprises or service providers.

vi. The proposed framework should incorporate some policy conflict methodology.

vii. The proposed framework should offer an interoperable platform to be able to communicate with other service providers to authenticate and authorise users, and to negotiate different service levels.

The framework for *policy-managed mobile client* must provide:

i. Support for seamless handoff between network while maintaining user's current sessions

ii. Maintain transparent and negotiated security

iii. Sophisticated movement detection and ability to choose *best available* access network

iv. Selection of optimum time to handoff

v. Capability to download a user profile and control the behaviour based on the profile and a network's current support for that profile

vi. Support for standardised protocols

The main objectives of our research is summarised in Figure 1-5.

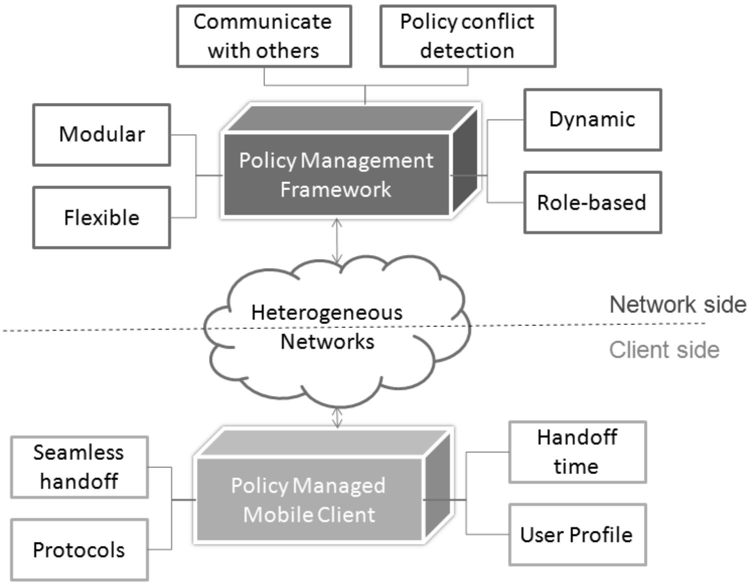


Figure 1-5 Objectives of proposed research

1.7. Document Structure

In this chapter, we have provided an overview of our proposed research work. We have further discussed our motivations for the research and discussed the current technologies involved. A problem statement is then defined so as to provide ABC services in heterogeneous mobile environments. Following this, we have listed the requirements of service providing networks and mobile entities followed by the overall objective of the research work. The following chapters detail different aspects of the work developed.

Chapter 2 provides an overview of the trends and evolution of the most relevant projects in both the academic and commercial worlds necessary to provide solutions to the requirements imposed by both centralised management of networks and of mobile entities.

Chapter 3 discusses the requirements of a new policy-based management framework to provide ABC services to roaming entities in heterogeneous mobile environments. A four-layered management framework model is then proposed. The proposed model introduces separate layers for policy creation, analysis and deployment which are referred to as *Policy Model Layer*, *Semantic Layer*, *Syntax layer* and *Enforcement Layer*. An analysis of the proposed model is then presented. Furthermore, a four-layered management framework for a policy-managed mobile client is proposed and analysed.

Chapter 4 describes the first layer of the proposed policy framework model. The Policy Model Layer uses the concept introduced in the Role Based Access Control (RBAC) model [16] and discusses the concept of Domains and Roles. The proposed Policy Schema and the proposed Information Model are then introduced.

Chapter 5 discusses the Semantic Layer and provides an overview of the methodologies proposed by various researchers. A semantic analysis method to detect policy conflicts is then proposed and discussed.

Chapter 6 describes the third layer of the proposed policy framework model. The Syntax Layer discusses different approaches for distributing policies and different methodologies proposed by different working groups. *Intra-domain* policy distribution and *Inter-domain* policy negotiation concepts are then introduced.

Chapter 7 focuses on the proposed Enforcement Layer used for deploying policies together with the proposed Mapping Translators to translate policies from higher layers to lower layers.

Chapter 8 discusses the proposed architecture of an extended policy-based management system and introduces different components of the architecture.

Chapter 9 proposes a new model of the policy-managed mobile client and discusses the proposed layered-approach for the management of the mobile client. The proposed architecture of a mobile client is then presented with the implementation details and testbed results.

Finally, *Chapter 10* summarises the main outputs from the work realised. It suggests future work that can be developed to enhance the proposed models of managed networks and policy-managed mobile clients.

Chapter 2
Trends and Evolution

From the previous chapter we have gained the basic knowledge about the objectives of this thesis as well as about the main technologies involved. Although our research objective is to provide ABC services across heterogeneous mobile environments, it is the management of these heterogeneous networks that has a significant influence on the research model, design and implementation together with the management of mobile entities. The particularities of these technologies motivate a set of requirements that must be handled by the management framework and the mobile entity.

2.1 Network Management Frameworks

The concept of using policies in management applications has been applied selectively for quite some time. Only recently have attempts been made to apply this concept to virtually all the management functions, as well as to the development of architectural frameworks, protocols and data models. The activities of the IETF (e.g. Policy Framework [17] and Resource Allocation Protocol [18]) WG on policy-based management have created a strong interest in the subject domain, and have inspired numerous projects in both research labs and the commercial world.

There have been numerous efforts to evaluate various aspects of different policy frameworks for different application scenarios. Damianou in [19] focused on comparing the nature of policy specification languages. Duflos in [20] evaluated the suitability of some policy languages for security management in distributed systems. In [21], Aib et al. evaluated both the specification languages and the policy management models of some frameworks for Quality of Service (QoS) management. Tonti in [22] compared policy specification and reasoning capabilities of three semantic web-based policy frameworks for the management of Multi-Agent Systems, while Phan et al. [23] performed a survey of policy-based management approaches for Service Oriented Systems. Vivero [24] presented an analysis of different policy management solutions for active and programmable networks.

While these policy frameworks are useful in their own respects, to the best of our understanding, there have been no attempts to evaluate existing frameworks for providing ABC services in heterogeneous mobile environments. In this chapter, we present our analysis of existing frameworks from the angle of our specific needs i.e. management of networks and mobile clients to provide seamless mobility, security and QoS services. It serves as our first step towards developing a suitable policy framework for ABC services management.

2.2 Approaches for Centralised Management of Networks

Policy-based management is an emerging technology for the management of networks that is based on our requirements, and can be adopted for providing ABC services in heterogeneous mobile environments.

Many research projects have covered the field of policy-based management. Among these, the most relevant projects for our framework are the IETF proposed Policy-Based Network Management (PBNM) model [25] and DAIDALOS [26]. The IETF PBNM model has been one of the first standardisation efforts and manufacturer-independent policy-based management frameworks. Furthermore, it defines a policy model and a policy architecture from where many concepts have been used in the proposed framework. The DAIDALOS project explores a framework for the integration of heterogeneous network technologies to provide services such as voice, data, and multimedia services. These are also properties of relevance in our proposed framework.

Another approach, though not directly related to the policy management is Role-based Access Control (RBAC) [27]. This presents an attractive solution for providing access control for services offered by web-based e-commerce and e-governance applications as well as operating systems. Usually, such systems involve a large number of users interacting with the system under different rights and obligations. The ability of

RBAC to provide an authorisation and access control system to assign roles derived from the organisational structure, has been of particular interest in our proposed policy management framework and provides an authentication-authorisation link to our proposed policy-managed mobile client.

Hereafter, we describe some of the work on policy-based management and comment upon those characteristics that are relevant to our work.

2.2.1 **Standardisation Approaches**

A PBNM is a set of technologies developed to control the use of network resources and provide an abstraction level of management to the network administrators. The IETF working groups have proposed a set of standards that aim at defining a framework for the representation, management, sharing and reusing of policies and policy information in a "vendor-independent, inter-operable, and scalar manner" [28].

IETF have defined the PBNM components [25] which operate in the following manner (Figure 2-1). An administrator inputs high-level business policies through the Policy Management Tool (PMT) and stores the Policy Information Model (PIM) in the form of Policy Rules in the Policy Repository. The Policy Decision Point (PDP) chooses the best-suited policy based on business rules and current state of the network, whereas the Policy Enforcement Point (PEP) enforces those policies to the network devices.

For example, a network access policy of a software company that states
"*If the user is a software developer and accessing network services during week days, then provide TypeA Internet access but block the Yahoo messenger application and Doom game*"

```
iptables -R INPUT 1 -s 10.1.1.0/24 -dport 80 -j ACCEPT
iptables -R INPUT 1 -s 10.1.1.0/24 -dport 5050 -j DROP
iptables -R INPUT 1 -s 10.1.1.0/24 -dport 666 -j DROP
```

will be deployed on a Linux-based router as:

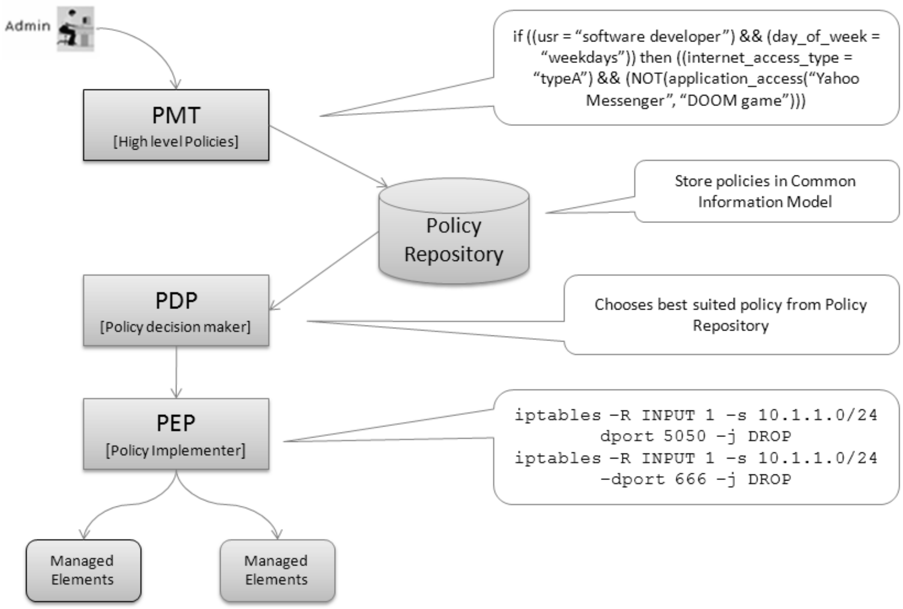

Admin
PMT
[High level Policies]
if ((usr = "software developer") && (day_of_week = "weekdays")) then ((internet_access_type = "typeA") && (NOT(application_access("Yahoo Messenger", "DOOM game")))
Policy Repository
Store policies in Common Information Model
PDP
[Policy decision maker]
Chooses best suited policy from Policy Repository
PEP
[Policy Implementer]
iptables –R INPUT 1 –s 10.1.1.0/24 dport 5050 –j DROP
iptables –R INPUT 1 –s 10.1.1.0/24 –dport 666 –j DROP
Managed Elements
Managed Elements

Figure 2-1 IETF proposed PBNM components

Policies are abstracted to apply across a variety of different devices so there is no need to create separate rules for each policy client. At the device level, policies are implemented by means of an "If/Then" proposition. That is, if certain conditions are present, then specific actions are to be taken. An "If" condition can be a time of day, a type of traffic, an IP address, a person, a group, or combination of these. A specific action might request the configuration of priority tagging or set security at certain levels. Other possibilities are actions related to access control, load balancing, and more sophisticated traffic-shaping.

A. ***Standards and Working Groups***

The use of policies for network management has recently gained interest by the Internet community. However, for the deployment of PBNM systems and to ensure inoperability amongst equipment from different vendors and PBNM systems from different developers, a standardisation process is required.

Both the IETF and Distributed Management Task Force (DMTF) are currently working on the definitions of standards for PBNM. DMTF is mainly focused towards the representation of policies and the specification of a corresponding information model and schema. IETF, in co-operation with DMTF, is also working in that field while also trying to define a generic framework for PBNM systems, as well as protocols that could be used for implementations.

a. ***DMTF work on Policy Based Network Management***

The DMTF has defined the Common Information Model (CIM) [29] management schema, which consists of an object-oriented model for the representation of policies defined for a managed network. The information model generated with this representation is stored in the directory of a Directory Enabled Network (DEN) [30]. The CIM has been the starting point for the specification of the Policy Core Information Model (PCIM) [31] by the IETF. PCIM is an object-oriented information model (discussed in section 4.3.1) for representing policy information developed jointly by the IETF Policy Framework Working Group and as extensions to the CIM.

b. ***IETF work on Policy Based Network Management***

IETF does not define a specific policy language, instead generic Object Oriented policy information models are proposed. While DMTF CIM is engineered for general representation of a managed system, the PCIM and its extensions [32] are used to represent policy-related information. In these models, the logical and physical elements and their relationship in a managed environment (such as system, services, and users) are represented as Object Oriented classes. Both PCIM and CIM are vendor and network independent and are useful for defining and modelling high-level policy systems.

There are several groups within the IETF where PBNM work is ongoing. The IETF working groups that are more related to PBNM work are the Policy Framework (policy) workgroup [17], and the Resource Allocation Protocol (RAP) [33] workgroup.

i. ***IETF Policy Framework Working Group***

The target of the Policy Framework working group is (a) specification of a framework for PBNM, (b) definition of PCIM for representation of generic policy data, and (c) extension of PCIM to support policies related to specific fields, such as QoS traffic management [34] and Internet Security Protocol (IPSec) configuration [35].

The IETF policy framework includes a policy framework definition language, a policy model, a set of policy terminologies, and a policy architecture Meta model. It defines a policy deployment model comprising a *policy manager* for managing the life cycle of policy objects, a *policy repository* for storing policy objects, a *PDP* for deriving policy actions based on the state of the environment, and distributed *PEPs* for enforcing the policies at the managed element site.

The PBNM framework (Figure 2-2) consisted of four elements. The *policy management console* offers a user interface for introducing policies in the PBNM system. These policies are stored in the *policy repository* from where they are retrieved by the *Policy Decision Point* (PDP) to decide when they should be enforced. Finally, the *Policy Enforcement Point* (PEP) is in charge of configuring the managed device(s) accordingly when policy conditions are met.

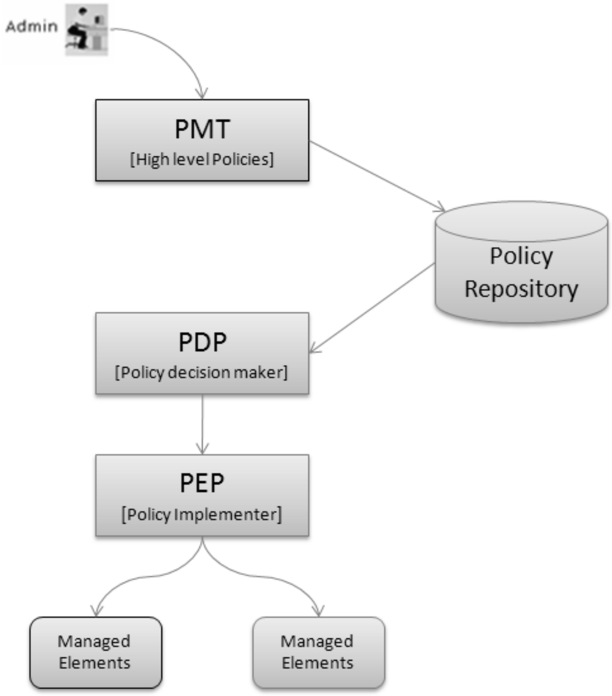


Figure 2-2 IETF proposed PBNM architecture

A PDP generally takes form of a policy server, which interacts directly with PEPs. It is the ‘decision-maker’ and the decisions are based on policies retrieved from the policy repository, as well as other management entities such as authentication servers, if they exist.

A PEP is the ‘policeman’, enforcing policies at the packet level as data passes through this point, also known as *policy agents*. The PEP may generally be implemented on the network components, such as routers, switches, network access servers (NAS), private branch exchanges (PBX), virtual private networks (VPN) and VoIP gateways.

ii. ***Resource Allocation Protocol Working Group***

The Resource Allocation Protocol (RAP) working group has defined the Common Open Policy Service (COPS) protocol [36] (discussed in section 6.2.1) for communication between PDP and PEP, and defined general-purpose objects that facilitate the manipulation of policies and provisioned objects available through COPS.

c. ***Other groups***

The IETF’s initial focus has been towards network policies to control the QoS and IP Security. Specifically, attempts such as the QoS Policy Information Model (QPIM) [34] have been made to extend and translate CIM and PCIM for QoS management. In the QPIM model, QoS control is performed by adjusting network device configuration according to the predefined policies. There have also been efforts to define the mapping of the IETF models into implementation-specific schema such as Web Based Enterprise Management schema [37].

Other working groups in PBNM are the IETF SNMPConf working group [38] defining objects that enable policy-based configuration management of Simple Network Management Protocol (SNMP) infrastructures [39] and the IPSec Policy System [40] defining a Management Information Base (MIB) module for managing IPSec, Internet Key Exchange (IKE) protocols and associated policies.

In summary, the Policy Framework and RAP WG promote mainly the standard for a generic policy-based framework while other working groups focus on expanding PBNM to specific solutions. PBNM shares the MIB and the PIB using protocols such as SNMP, COPS, Lightweight Directory Access Protocol (LDAP), and the Hypertext Transfer Protocol (HTTP). Table 2-1 provides a summary of the IETF Working Group standards activities.

Working Group	**Activity**	**Field of standard**
Policy	- Policy Server Structure - Information Model	Policy-based framework standard
RAP	- COPS - COPS-PR	Policy distributing protocol standard
DiffServ	- QoS service PIB	QoS DiffServ standard
SMNPConf	- Defining Policy MIB	Defining SNMP infrastructures
IPSP	- IPSec policy system - IPSec policy modelling - IPSec specification language	IP security service framework standard

Table 2-1 IETF Working Group related with PBNM Framework

2.2.2 Other Approaches

A. Ponder

The Ponder project [41] has had a good acceptance within the research community and its results have been used in various research projects based on policy-based management. The Ponder project defined a *language* and a *framework* for specifying security policies that map onto various access control implementation mechanisms for firewalls, operating systems and databases in the Java programming language. It supports policies that are event-triggered and based on condition-action rules for management of networks and distributed systems. Ponder can also be used for security management activities such as registration of users or for logging and auditing events to deal with providing access to critical resources or security violations. The key concepts of the language include *roles* to group policies relating to a position in an organisation, and *relationships* to define interactions between those roles and management structures pertaining to an organisational unit.

To define different components of the management framework, Damianou in [42] states that the number of levels of policy hierarchy can be arbitrary but three levels of policy specification have been accepted (Figure 2-3). A Policy hierarchy is referred to as steps of refinements where general high-level policies are translated into a number of more specific policies [43]. Each policy level is defined as follows:

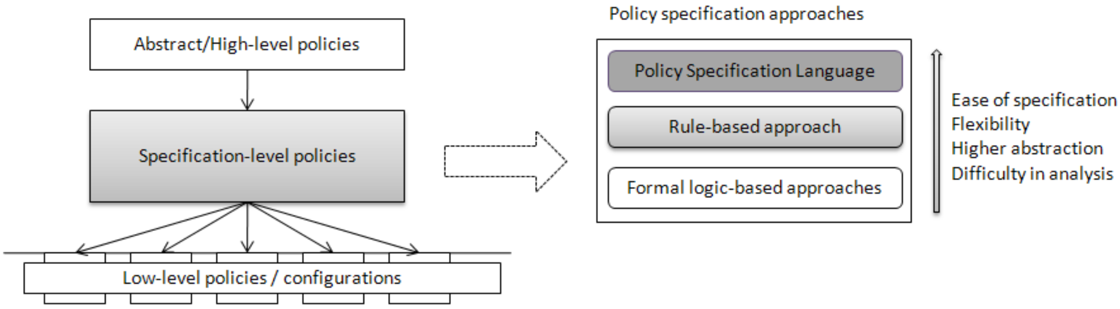

Abstract/High-level policies
Specification-level policies
Low-level policies / configurations
Policy specification approaches
Policy Specification Language
Rule-based approach
Formal logic-based approaches
Ease of specification
Flexibility
Higher abstraction
Difficulty in analysis

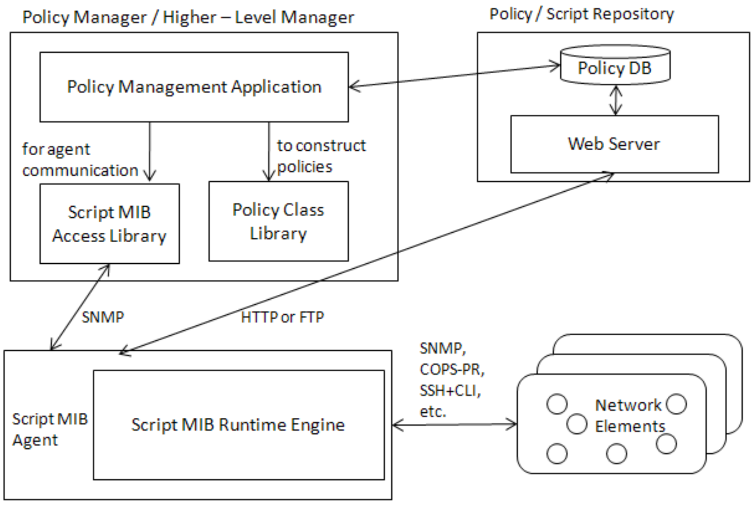
Figure 2-3 Three-level policy hierarchy

- *High-level abstract* policies (also referred to as management goals), which can be business goals, service level agreements (SLA), trust relationships or even natural language statements. High-level abstract policies are not enforceable and their realisation involves refining them into one of the other policy levels.
- *Specification-level* policies sometimes referred to as network-level or business-level policies. These are the policies specified by a human administrator to provide abstractions for low-level policies but in a precise format. These policies relate to specific services, or objects and how their interpretation can be automated. These policies are further categorised as: policy specification languages, rule-based specifications, and formal logic languages.
- *Low-level policies* or configuration commands such as device configurations, security mechanism configuration (e.g. access control entries, firewall rules), directory schema entries and so on.

The Ponder project has defined a specification of the policy language and a framework for policy processing. However, in their preferred model, the line separating low-level policies and device configuration is not clear, while directly specifying policies at this level often proves to be a bottleneck to both scalability and interoperability. Furthermore, the actual management functionality is not considered in the project and is oriented towards the management of passive networks. Therefore, it cannot be applied to mobile environments. Also, policy negotiation between heterogeneous networks is not supported.

B. **Jasmin**

The Jasmin project [44] aims to evaluate, enhance and implement the distribution and invocation of network management scripts with distributed network management applications. The Jasmin project adds a set of classes to support policy-based configuration management of Linux DiffServ nodes (Figure 2-4). The implementation supports multiple languages and run-time systems. In particular, general policy management language extensions, domain specific policy management language extensions and drivers mapping between domain specific policies and the underlying device-level mechanisms, have been realised.



Policy Manager / Higher – Level Manager
Policy Management Application
for agent communication
to construct policies
Script MIB Access Library
Policy Class Library
Policy / Script Repository
Policy DB
Web Server
SNMP
HTTP or FTP
Script MIB Agent
Script MIB Runtime Engine
SNMP, COPS-PR, SSH+CLI, etc.
Network Elements

Figure 2-4 Jasmin Script MIB based management architecture

As with Ponder, Jasmin is focused on the management of passive networks via scripts. It explores the distribution of policy condition monitoring and policy action enforcement although in many cases the decision is still made at the policy manager station. Furthermore, the following are some of our requirements of our proposed model which are not supported by Jasmin: (i) capacity for creating different management infrastructures based on service provider needs, (ii) abstracted management functionality which can be assigned to more specific nodes, (iii) capability of dynamically extending the management framework and (iv) ability to negotiate policies between different service-providers. However, the concepts used in Jasmin for the distribution of policy tasks, especially the MIB Runtime Engine and automation of policy decisions, are also considered in this thesis.

C. **ANDRIOD**

The Active Network DistRibuted Open Infrastructure Development (ANDRIOD) project [45] proposed a policy- and event-driven architecture for the management of Application Layer Active Networking (ALAN) networks [46], [47]. ALAN network states that the action of the policy "*if <condition> then< action>*" is triggered when particular *events* are received. Both events and policies are distributed with the Management Information Distribution (MID) system.

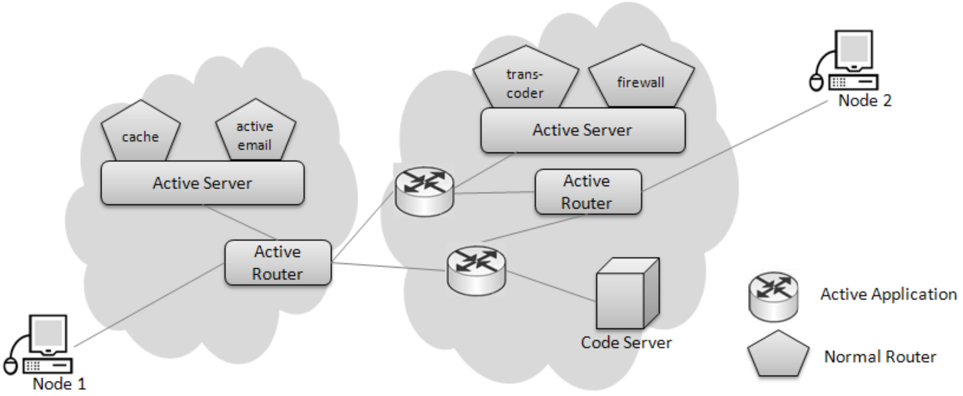


Figure 2-5 ANDRIOD Architecture

Each ANDRIOD framework instance runs at least one MID server. Inside the MID, policies and events are introduced into a new XML document called Notification. The event destinations as well as the protocol that should be used to communicate with those destinations are specified in the MID by means of policies.

The XML policy defined in ANDRIOD carries at least six fields:

i) *Creator*: Specifies the source of the policy to establish its access rights

ii) *Info*: Contains policy related information other than the policy itself, such as the expiration time, policies replaced by this one, etc.

iii) *Sender*: Lists the forwarding path followed by the policy

iv) *Subject*: Identifies the entities that pertain to a role that must process this policy

v) *Trigger*: Enumerates the events that will activate the policy. When the trigger field of the policy is empty the system assumes that the policy should be enforced immediately

vi) *Actions*: This field can include, in addition to the policy actions, optional conditions that should be assessed before enforcing the actions

Events are also defined in XML in ANDRIOD. Its structure is as follows:

i) *Event-id*: Unique identifier of the event

ii) *Time*: Specifies the time when the event was launched. It can be used by the receiving entity to reject the event if it is too old

iii) *TimetoLive*: Establishes the time during which the information carried by the event will be relevant

iv) *Source*: Identifies the entity that created the event

v) *Sequence*: Integer which is incremented every time an event is sent from a particular source

vi) *Information*: Explanatory text about the event information

vii) *Data*: Structured information carried within the event

When a user wants to install a new service inside a server, it sends an event to the network operator. The operator initiates the resources and security checks, based on available policies, and then loads the service. The services are continuously monitored, so that if unexpected behaviour is detected corrective policies can be enforced.

XML policies are used for managing routers, resources and security in servers and MID systems.

ANDRIOD is focused towards the management of servers within an application-layer network. In particular, it is focused towards the management of security and resource access by services inside the ANDRIOD server and considers reduced management of the routers i.e. configuration of user's routes forwarded to their assigned server. Instead, our focus is targeted towards supporting the management of heterogeneous mobile environments, as well as multiple functionalities ranging from security and resource allocation to traffic engineering. However, the usage of XML to manage user routes in routers, resources and security, provides an interesting aspect of XML as an interoperable standard for policy negotiations between heterogeneous mobile environments.

D. **Policy eXtension by Policy (PxP)**

The Policy eXtension by Policy (PxP) project [48] suggests a mechanism for the dynamic extension of a PBM system. The mechanism uses policies within an active network environment to realise the extension. The PxP proposal is limited to the extension method, so it must be included within some other management architecture.

The method defined two types of policies for realising this extension, i.e. Policy Definition (PD) policies and Policy Extension (PE) policies. On the one hand, PD allows a user to add a new type of policy into the Policy Server specifying the correct syntax and restrictions. Then, through PE policies users can specify the corresponding methods for translating the new policies types into commands on different types of network nodes. Both PD and PE policies are defined by either network operators or an application.

The architecture (Figure 2-6) where this extension method has been conceived is the general IETF PBNM architecture containing a Graphical User Interface (GUI), a policy manager, a database and policy agents.

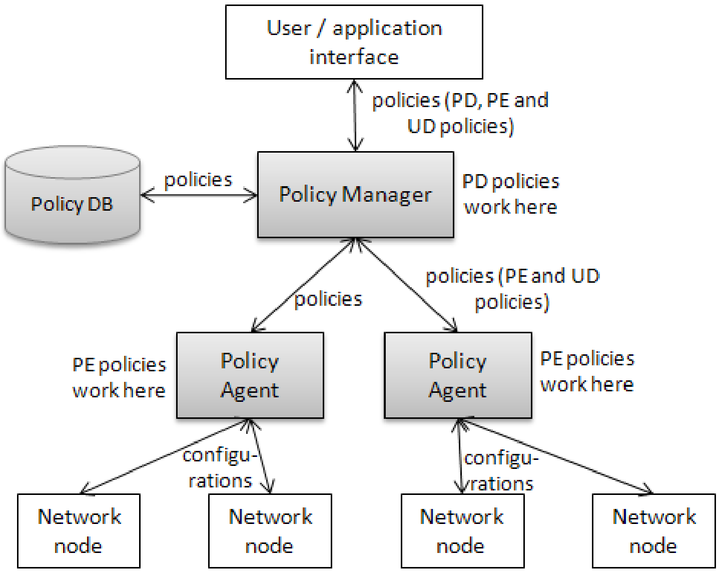


Figure 2-6 Policy Extension by Policy architecture

When an administrator introduces a new policy, the policy manager verifies the correctness of the policy with the information contained in the corresponding PD policy. The policy agent then translates the new policy into managed device commands following the instructions contained in the corresponding PE policy. In consequence, the PE policy depends on the managed node where the policy should be enforced.

The way policies should be translated is described within PE policies by means of templates. These templates are completed with the policy information using *fillers*. The fillers specify what information should be retrieved from the policy to complete the template. A program interpreter can be included inside each policy agent to evaluate fillers, i.e. to allow fillers specifying certain processing of the policy data before been included in the template.

PxP is a method for the “extension” of management functionality in a PBM system. Nevertheless, it only defines the extension mechanism (i.e. it does not cover the decision mechanism or the conflict checking mechanism), which should be included into a complete policy-based architecture like the one described in this thesis, or others. Therefore, the research developed in PxP should be considered as complementary research in relation to the model-architecture presented in this thesis.

E. **DAIDALOS**

Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services (DAIDALOS) [26] is an EU Framework Programme 6 Integrated Project aims at building a framework for the integration of heterogeneous network technologies to provide services such as voice, data, and multimedia services. The DAIDALOS project, started in January 2006 with 46 partners from industry and academia, focuses on the network architectures necessary to create a user-centred manageable communication infrastructure to provide an integrated environment.

The objective of DAIDALOS is to develop and demonstrate an open architecture based on the common network protocol (IPv6). The overall objectives are to:

- Design, prototype and validate the necessary infrastructure and components for efficient distribution of services over diverse network technologies beyond 3G (e.g. 4G),
- Integrate complementary network technologies to provide pervasive and user-centred access to these services,
- Develop an optimised signalling system for communication and management support in these networks,
- Demonstrate the results of the work through strong focus on user-centred and scenario-based development of technology

The project is guided by five key concepts:

- MARQS (Mobility Management, AAA [Authentication, Authorisation and Accounting], Resource Management, QoS and Security) supporting functional integration for end-to-end services across heterogeneous technologies,
- Virtual Identity, which separates the user from a device, thereby enabling flexibility as well as privacy and personalisation,
- Ubiquitous and Seamless Pervasiveness enabling pervasiveness across personal and embedded devices, and allowing adaptation to changing contexts, movement and user requests,

- Seamless Integration of Broadcast which integrates broadcast at both the technology level, such as DVB-S/T/H (Digital Video Broadcasting), and at the services level, such as TV, carousels and data-cast,
- Federation which allow network operators and service providers to offer and receive services, allowing players to enter and leave the field in a dynamic business environment.

The DAIDALOS architecture integrates both wired and wireless technologies, with QoS capability under a common authentication, authorisation, accounting, auditing and charging (A4C) framework in a secure communication environment. The following components have been proposed:

1. **Service Provisioning Platform**

   The architecture defines a Service Provisioning Platform comprised of services for QoS, Network Management, Network Monitoring, Security, A4C and Multimedia. Service Provisioning Platform is the main part of the DAIDALOS architecture and functions as the home environment for the user with respect to different services.
2. **Access Network**

   The Access Network is a network comprised of Access Router(s) that connect mobile devices to the network via different access technologies and Control entities, such as QoS Broker and AAA proxy to administer the authentication and authorisation to mobile devices connected to the network.
3. **Key Interconnection**

   The Key Interconnection enables communication between various management types in different administrative domains and provides secure inter-domain key transport and supports symmetric and asymmetric keys.
4. **(Third Party) Service Providers**

   The (Third Party) Service Providers provide applications (e.g. multimedia) and content to the end-users. They can use the Service

Provisioning Platform to enable access to mobile users, arrange QoS for multimedia content, and to manipulate user's sessions

5. **Interoperability among different service providers**

   The architecture divides the overall next-generation network in different administrative domains, each hosting a Service Provisioning Platform. Components in different administrative domains cooperate when there is a service level agreement between them. This cooperation is necessary to support different services of the Service Provisioning Platform across administrative domains, including the support for seamless mobility of users, end-user terminals and multimedia sessions between administrative domains.

Although related to our research objective for providing ABC services to mobile users, our approach is fundamentally different from the DAIDALOS project. DAIDALOS have identified different (and somewhat precise) modules for providing each service and is very specific in its implementation. For example, it does not support IPv4 and requires service operators to introduce components and protocols *as* proposed in DAIDALOS. Hence, to provide end-to-end QoS architecture in wireless networks, DAIDALOS has defined two different scenarios: the single-hop which is composed of a set of wireless technologies, namely 802.11e, 802.15.1 and TD-CDMA. The two-hop scenario is a concatenation of two wireless technologies, namely 802.16 in the first hop and 802.11 or 802.15.1 in the last hop. Different Interfaces and Drivers [49] with QoS Broker and Access Routers [50] are proposed to achieve this end-to-end QoS. Similarly, for mobility purposes, extensions to the IPv6 fast mobility [51] is used. For secured user authentication and authorisation, EAP over PANA (Extensible Authentication Protocol over Protocols for carrying Authentication for Network Access) protocol is used while for server authentication, EAP over DIAMETER [52] protocol is used.

Furthermore, to enable a standardised exchange of authentication information between different administrative domains (i.e. inter-domain security) and to allow single sign-on for users, the Security Assertions Markup Language (SAML) [53] is proposed and to support the security between the network elements like routers (Intra-domain security), IPSec is deployed as the basic security protocol.

Our research is focused on providing a generic framework to the organisations or network operators to deploy PBM solutions and to be able to use their existing infrastructure and protocols as long as is feasible. However, DAIDALOS architectural solutions are based on different Service Provisioning Platforms for different services and a domain-based Provisioning Platform for interoperability among service providers has been of particular interest.

2.2.3 Commercial Approaches

In policy-based networking, most of tools developed within the industry are based on the IETF framework. The majority of commercial tools are specific to QoS management, while some also include access control. As these tools have a significant influence on the adoption of PBM solutions, this section provides a brief overview of some of the major commercial products.

a. **HP Openview PolicyXpert**

PolicyXpert [54] is a three-tier PBM solution allowing network administrators to define QoS policies and support traffic management actions to offer QoS-based services. The policies are defined using the *if <condition> then <action>* paradigm where conditions can be based on different parameters such as packet information, time of day or higher-level protocol information like HTTP or Uniform Resource Locator (URL). The tool supports other standards including COPS, Differentiated Services (DiffServ) and Resource Reservation (RSVP) protocol.

b. **Allot Communications NetPolicy**

Allot Communications NetPolicy [55] aims to provide QoS-based PBM capabilities to service providers by monitoring traffic to enable guaranteed bandwidth available to the end user. The policies are specified using the condition/action notation, and the conditions can be defined in terms of the packet information parameters. The policy repository is implemented using

LDAP and policy information is passed to target devices using either COPS or Command Line Interface (CLI). NetPolicy also supports management operations on simple access control lists.

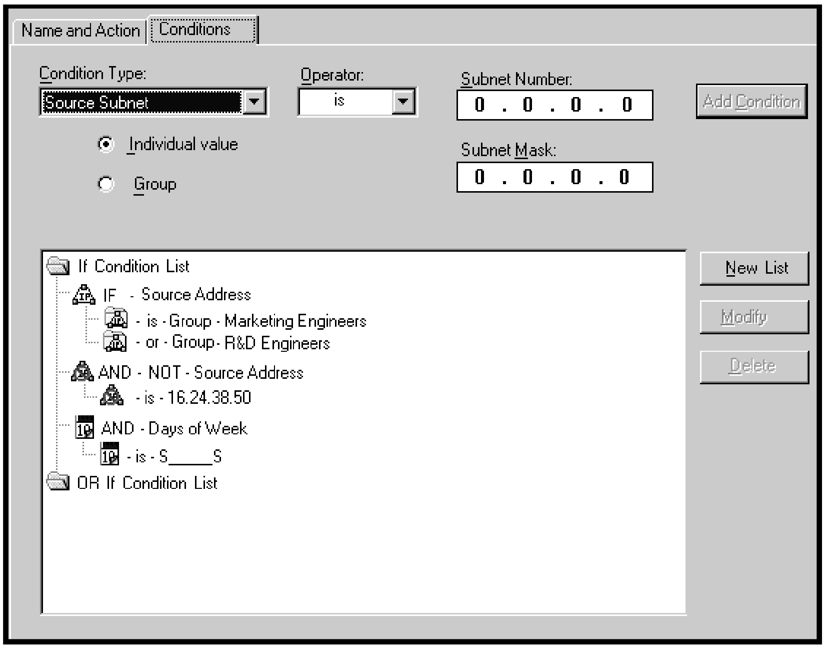
c. **CiscoAssure**

CiscoAssure Policy Manager [56] is also aimed at QoS service management. Although policies are specified using the condition/action approach defined by the IETF CIM standard, the policies themselves are stored in a flat-file [57]. The user interface allows the administrator to specify multiple conditions for triggering policies. The conditions can be specified using a combination of IP addresses (source and destination), application ports, and the protocol being used (IP, TCP or UDP). Policy actions are applied to routers by using CLI language already supported by Cisco hardware. Multi-vendor interoperability is provided with an implementation of COPS. In addition to supporting QoS related management operations, this tool allows administrators to define access control policies for the devices being managed.

d. **Tivoli Access Manager**

The system management framework developed by Tivoli is an extensive suite of applications that provide support for configuration management and user access management. The Access Manager tool [58] allows administrators to configure authorisation policy templates which maps to access control lists in the managed system. If the managed objects are organised into a hierarchy, the policies defined at a given level are propagated to lower level objects. Any inherited policies can be overridden by specifying explicit policies on any given element.

A common component of these commercial tools is a GUI based interface (Figure 2-7) which allows administrators to visually select a network device or managed element and specify the policies in the form of *if <condition>then<action>* rules for selected targets. Different products allow varying degree of conditions in policy rules including time attributes, source or destination IP addresses, service type, port numbers, and some higher-level user-defined data, and allow the user to permit or deny traffic based on conditions.



Condition Type: Source Subnet

Operator: is

Subnet Number: 0 . 0 . 0 . 0

Subnet Mask: 0 . 0 . 0 . 0

● Individual value

○ Group

```
If Condition List
  IF  - Source Address
    - is - Group - Marketing Engineers
    - or - Group - R&D Engineers
  AND - NOT - Source Address
    - is - 16.24.38.50
  AND - Days of Week
    - is - S_____S
OR If Condition List
```

Figure 2-7 GUI support for HP Openview PolicyXpert

An important concern common to the commercial solutions is the support of multi-vendor platforms, which is not adequately supported by most of the currently available products. The implementations are mainly a graphical environment used to configure *their* proprietary network devices. Little effort has been made towards allowing policies to be represented in a generic abstract high-level form and facilitate policy conflict detection and resolution. Typically these policies are static in nature and do not respond to the changes either in network topology or the current status of network nodes.

Furthermore, different standard protocols are implemented to varying degrees by different vendors. For instance, many of the products support COPS as the main communication protocol for policy information between the components of their architecture, while others support HTTP or CLI.

2.3 **Role Based Access Control**

The Role Based Access Control (RBAC) model was proposed by the National Institute of Standards and Technology (NIST) in 2001 [27] and formally adopted as an American National Standard Institute (ANSI) standard in 2004 [16]. The RBAC approach, though not directly related to PBM models provide an attractive solution for defining an overall structure of the network or organisation, where users and network elements are represented as a set of entities rather than individual persons and devices. The policies defined using the RBAC approach treat individual users and network elements based on roles, where roles are assigned permissions to perform an action. The users and network elements are assigned roles hence the RBAC approach provides an authorisation and access control system. The ability of RBAC to define a formal structure of the organisation is of particular interest in this research.

The NIST RBAC model provides a methodical way to understand and consequently implement security policies for access control in a manner that reduces the administrative overhead from the management of those access control policies. Existing work within the RBAC community is focused on specifying access control configuration in terms of roles.

To identify RBAC features that exhibits true enterprise value and are practical to implement, NIST has conducted and sponsored market analysis [59], developed prototype implementations [60], and sponsored external researches [61]. NIST have also proposed a standard reference model to facilitate interoperability among information systems that implement RBAC [62] and views RBAC as a tool to enable the administration of security at a business-enterprise level rather at the user-identity level. Other significant research has been performed at the academic level in developing new RBAC models and applications. One of the prominent efforts for defining a consensus standard for RBAC was proposed by Sandhu et al. [63].

The NIST RBAC model is organized into RBAC Reference Model and RBAC System and Administrative Functional Specification. The reference model defines the scope of features that comprise the standard and provides a consistent vocabulary in support of the specification. The reference model is further divided into several sub-models as shown in Figure 2-8 and comprises different sets of functionalities. The functionality of each sub-model is discussed as follows:

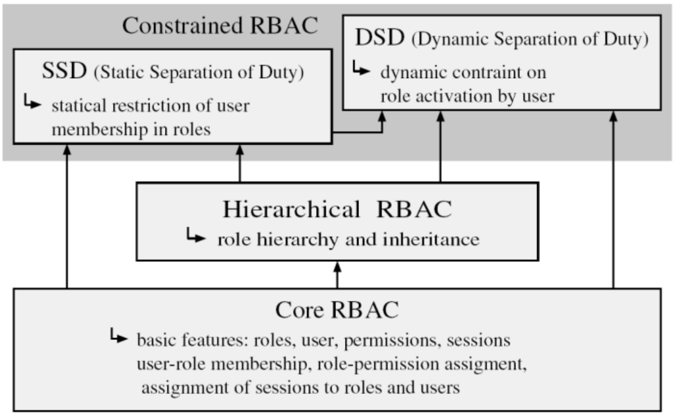

Constrained RBAC
SSD (Static Separation of Duty)
statical restriction of user membership in roles
DSD (Dynamic Separation of Duty)
dynamic contraint on role activation by user
Hierarchical RBAC
role hierarchy and inheritance
Core RBAC
basic features: roles, user, permissions, sessions user-role membership, role-permission assigment, assignment of sessions to roles and users

Figure 2-8 RBAC Reference Model

a. **Core RBAC**

The Core RBAC embodies the essential aspects of RBAC. It is the basis of the entire model, and is the most primitive of the four levels of abstraction. The relationship between users, roles, and permissions constitutes the base of the model. The definitions of the entities and relationships in the model are as follows:

- *User* in this model is a human being or other autonomous agent such as a process or a computer.
- *Role* is a job function or job title within the enterprise with some associated semantics regarding the authority and responsibility conferred on a member of a particular role.
- *Permission* is defined as an approval of a particular node to access one or more objects in the system. Permissions confer the ability of the holder to perform some action or actions in the system. In addition, each permission can be represented as either a permission or prohibition.
- *User-Role (UA) relationship* represents which user is assigned to perform the associated role in the enterprise
- *Permission-Role (PA) relationship* assigns permission or a set of permissions to a specific role or a set of roles.

Users are assigned to roles, permissions are assigned to roles, and users acquire permissions through their assigned roles. The relationships user-role and role-permissions can be many-to-many (Figure 2-9).

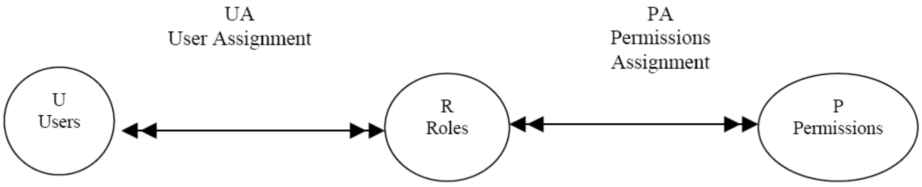


Figure 2-9 Core RBAC

The Core RBAC abstraction supports user-role view, thus provides a basis for defining relationship of users that are granted a specific role and the roles that are permitted for a specific user. Moreover, users can simultaneously exercise permissions of multiple roles, akin to multiple inheritances in an object-oriented model. The requirement for user-role review differentiates the Core RBAC from the group-based access control modelling paradigm. However, the Core RBAC abstraction leaves many issues related to the scalability of the model, the nature of permissions, expression of permission revocation, and representation of role administration.

b. **Hierarchical RBAC**

The Hierarchical model extends the Core RBAC abstraction by introducing the notion of role hierarchies (Figure 2-10). A hierarchy is a partial ordering of roles, whereby senior roles subsume the permissions of their juniors. Role hierarchies are a natural means for structuring roles to reflect an enterprise's line of authority and responsibility. They can be inheritance hierarchies, meaning that the activation of an instance of a senior role by a user (such as at login) implies the inheritance of the permissions of all junior roles (Figure 2-11), or activation hierarchies, in which there is no implication of overall inheritance of permissions. In the activation hierarchy, the inheritance is limited to the roles which are subordinate to the specified role in the tree structure of the model. Therefore, the NIST model has identified two sub-levels, which are general (inheritance) hierarchical RBAC that uses partial ordering of roles and the restricted (activation) hierarchical RBAC that uses simple structures such as trees or inverted trees.

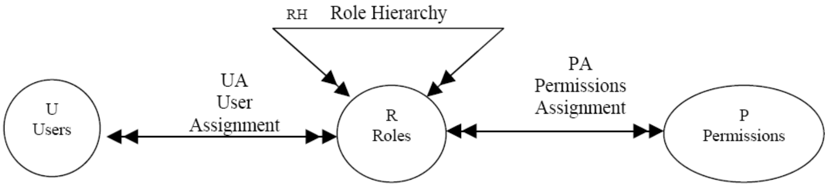


Figure 2-10 Hierarchical RBAC

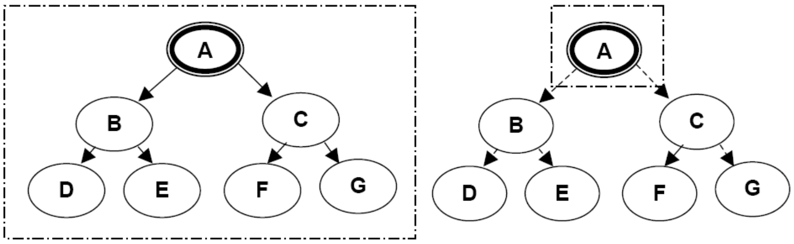


Figure 2-11 Inheritance Hierarchy and Activation Hierarchy

c. **Constrained RBAC**

The Constrained RBAC model introduces the semantics needed to enforce a separation of duty (SOD) [64], which is a time-honoured technique for mitigating the potential for the occurrence of fraud and accidental damage attributed to sharing of duties. It is often used to enforce conflict-of-interest policy that enterprises may employ to prevent users from gaining authorisation for permissions associated with conflicting roles.

The two categories of separation of duty are as follows:

i. **Static Separation of Duty (SSD)**

SSD is defined as a constraint associated with the UA assignment (Figure 2-12). Membership in one role may prevent a user from being a member of one or more mutually exclusive roles, depending on the SSD rule enforced. In other words, the SSD requirement is that no user should be assigned to two roles which are in conflict with each other, or conflicting roles cannot have common users. The SSD policy can be centrally specified and then uniformly imposed on specified roles.

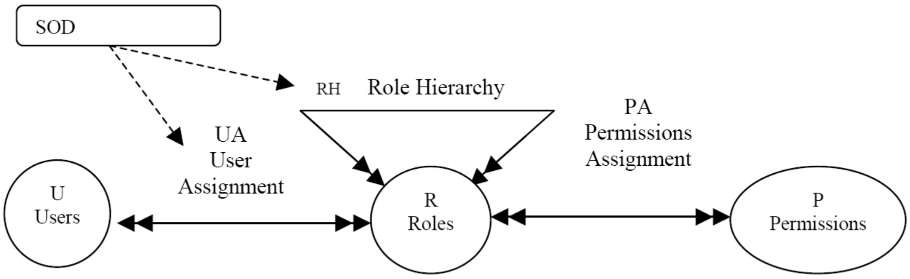


Figure 2-12 Static Separation of Duty

ii. **Dynamic Separation of Duty (DSD)**

DSD is defined as a constraint associated with the activation of roles within user sessions. It is a dynamic SSD property with respect to the roles activated by the users in a single session (Figure 2-13). In other words, conflicting roles may have common users but users cannot simultaneously activate roles that are in conflict with each other. A concept of *sessions* is introduced where a user is assigned to a role based on some constraints (e.g. login, time of day, location of access, etc.).

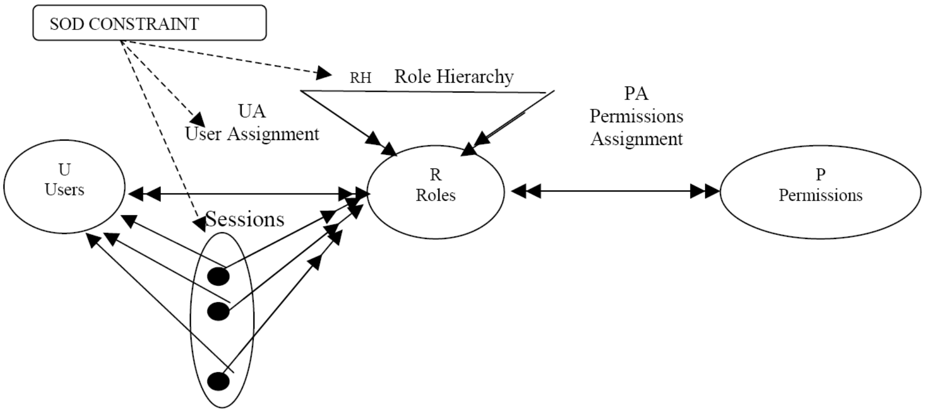


Figure 2-13 Dynamic Separation of Duty

d. **Symmetric RBAC**

Symmetric RBAC extends the semantics of the model to accommodate permission-role review (PA assignment), which is similar to UA assignment: it is possible to determine the role to which a particular permission is assigned as well as the permission assigned to a specific role. The permission-role review interface returns one of the two types of results. The query-symmetric

RBAC will include the semantics necessary for defining direct and indirect assignment of permissions. Direct-permission assignment pertains to the set of permissions that are assigned to the user directly. Indirect permission assignment includes the direct permissions assignment and the set of permissions that are inherited by the roles assigned to the user.

2.4 Approaches for providing ABC service to Mobile Entities

As discussed in section 1.3, ABC service allows users to be able to select and seamlessly connect to the network in a way that best suits their application needs. One of the main requirements for offering such service is to *maintain a transparent handover* of a mobile device when it is moving from one network/access technology to another.

This transparent mobility across heterogeneous networks demand a universal, physical-layer independent, mobility solution preferably at the network layer or higher. The Mobile-IP protocol [65] has been proposed to solve the mobility problem. It provides IP-layer mobility management between access technologies, permitting wireless interface and data-link layers of various access technologies to evolve independently. Most other common approaches are aimed at reducing the handover latency, signalling load, and improving scalability and robustness [66].

Hence, supporting a mobility protocol, similar to Mobile-IP, enables loosely-coupled interworking architectures to support session continuity when mobile devices roam among heterogeneous networks managed by different operators and different access technologies, hence achieving an ABC state. Other requirement of QoS and security is largely dependent on the current context of the mobile user (e.g. number of active sessions, type of active application, etc.) and the state of the network offering such service (e.g. number of working routers, current load on network, available bandwidth to end users, etc.).

2.4.1 **Standardisation Approach**

The IETF IP Routing for Wireless/Mobile Hosts Working Group (IETF mobileip Working Group) [67] has defined Mobile IP as, basically a method for extending the Internet Protocol (IP) to allow a portable computer to be moved from one network to another without changing its IP address and without losing existing connections. Specifically, the Mobile IP protocol offers routing support to permit mobile nodes using either IPv4 or IPv6 to seamlessly roam among IP sub-networks by binding mobile nodes with a permanent IP address (i.e. home address) and a dynamic address offered by the current network (i.e. care-of address). The Mobile-IP protocol supports transparency *above* the IP layer, including the maintenance of active TCP connections and UDP port bindings. It is believed that Mobile-IP is the oldest and probably the most widely known mobility management proposal [68]. Mobile-IP protocol and our implementation for Windows operating system is further discussed in section 9.1.3.

2.4.2 **Other Approaches**

There are other approaches and several testbeds proposed (and some of them implemented) that emulate to provide mobility to roaming mobile nodes [69]. Mobility is usually offered in two flavours which are loosely defined as (i) inter-domain handoff which provides mobility support to mobile nodes while it is moving from one access technology/network to another, and (ii) intra-domain handoff when mobile node is moving within the same network/access technology. Most of testbeds and research approaches are based on existing protocols, such as Mobile-IPv4, Mobile-IPv6, Cellular IP and HAWAII. The following summarises some of the work and testbeds.

a. **Dynamics**

   The Dynamics HUT Mobile-IP system developed at Helsinki University in 2001, is a scalable, dynamical and hierarchical Mobile-IP implementation to support both IPv4 and IPv6 protocols implemented for Linux operating system [70]. Dynamics mobile node is also partially ported for the Microsoft (98SE, NT4 and 2000) operating systems. It is arguably one of the most used implementations of the Mobile IP protocol and since then various other testbeds have been based on it.

b. **BARWAN**

The concept of handoff between different heterogeneous wireless networks (i.e. vertical handoff) was introduced in 1996, as part of the Bay Area Research Wireless Access Network (BARWAN) project at University of California (Berkeley) [71]. It is a pioneering work in the area of mobile networking where overlay internetwork management architecture was proposed allowing mobile applications to operate across a wide range of networks to support media-intensive applications. The architecture and protocols were designed to scale for users clustered in higher densities and cater for their increasing demands for computational resources.

c. **MosquitoNet**

MosquitoNet deployed at Stanford University [72] was one of many testbeds developed to evaluate Mobile IPv4 for intra-technology handovers. MoquitoNet was focused on minimising delays during horizontal handovers (i.e. across the same access technology such as WLAN to WLAN) and later evaluated Mobile-IPv6 performance in heterogeneous environments. The testbed is used to study the integration of different radio access technologies into a single IP-based core infrastructure.

d. **Nomad**

Nomad [73] is a research and development project in the 1st programme funded by the EU commission. The project started at 2002 and was focused on developing a middleware capable of seamlessly integrating different access technologies (i.e. WLAN and UMTS) and IP compatible ad-hoc networks. A performance evaluation based on implementation of Mobile-IPv4 was carried out in this project, where the mobile device roams among different heterogeneous networks. Furthermore, several filter extensions to Mobile IPv4 [74] and Mobile IPv6 [75] were proposed to allow distributed flow of packets among the point of attachments where the mobile node is currently connected to the network(s).

e. **MIND**

The Mobile IP based Network Developments (MIND) [76] project was formed by European’s Telecom Operators in 2001. The project implemented an experimental setup, which integrated IEEE 802.11b, UMTS, and GPRS including ad-hoc, wireless and fixed infrastructure. The project further evaluated Mobile-IPv6 performance during inter- and intra-technology handovers.

f. **Moby Dick**

Moby Dick [77] project, proposed and implemented a global end-to-end Mobile-IPv6 based architecture to offer QoS in heterogeneous environments. The proposed network architecture supported the concept of using standard IP-based protocols and technologies by reusing commonality in different access technologies and support to provide voice services to traditional cellular networks. The testbed included UMTS-like wireless access technology, IEEE 802.11b WLANs, and wired connectivity. Further work is being done as part of DAIDALOS project [26].

g. **IDMP**

The Intra-Domain Mobility Management Protocol (IDMP) [78] was proposed for managing mobility within a specific domain. Like other proposals based on mobility, IDMP envisions that multiple IP-subnets are aggregated into a single domain as long as a mobile node moves within a single localised domain. The IDMP approach focused on enhancing Mobile-IP protocol in micro-mobility environments with high handover frequency to decrease the signalling load. It also offered paging services to locate mobile nodes within a particular domain and power saving techniques to resource-limited handheld devices. IDMP was originally deployed using the Linux Mobile-IP code of the Stanford University MosquitoNet project.

h. **HAWAII**

HAWAII was another domain-based approach for supporting micro-mobility. Developed by Lucent Technologies in 1999, HAWAII proposed a separate routing protocol for mobility rather than the Mobile IP protocol [79] by using specialised path setup schemes which install host-based forwarding entries in specific routers to support intra-domain micro-mobility and defaults to using Mobile IP for inter-domain macro-mobility to provide QoS support.

i. **Cellular IP**

The Cellular IP micro-mobility protocol [80] was proposed by Columbia University and Ericsson Research in 1998 to provide local mobility and handover support. It can interwork with Mobile IP to provide intra-domain mobility support. The mobile nodes, in addition to running Mobile IP implementation engine, have to run a special Cellular IP protocol engine that controls the mobility support.

2.5 Analysis of Different Approaches

In this chapter, we have analysed a number of PBM approaches and presented a brief overview of mobility management research and testbeds. Although, to our knowledge, none of the reviewed projects is focused towards ABC services in heterogeneous mobile environments, their relevance to generic policy-based networking and mobility (and somewhat QoS) support is important.

2.5.1 Analysis of Policy-based Management Approaches

Two IETF Working Groups have considered policy networking. The RAP working group defined COPS protocol for use between PEP and PDP, while the Policy Framework working group defined a framework for representing and managing policies in a vendor independent and scalable manner. An extensible information model, PCIM, for representing policies is proposed.

Although standardisation efforts are underway, key aspects in the area of policy-based management still needs to be addressed. For instance, the impact of inconsistent policies on the network state is barely understood. Also, realising effective mappings of high-level policies into lower-level policies remains unsolved in many cases. The IETF framework does not provide answers to the question of how to best distribute the components of a PBM system in medium and large size networks, let alone in different heterogeneous networks and negotiate service policies among different service providers.

The Ponder project focuses on the specification of a policy language and a framework for policy processing. The actual management functionality is not considered in the project and is oriented towards the management of passive networks. Therefore, it cannot be applied to mobile environments, and policy negotiation between heterogeneous networks is not supported.

As with Ponder, the Jasmin project proposes the management of passive networks via scripts. However, the concepts used in Jasmin for the distribution of policy tasks especially, the MIB Runtime Engine, and automation of policy decisions are also considered in this thesis.

The ANDRIOD project, proposed a policy- and event-driven architecture for the management of active servers. The project is focused on the management of security and resources accessed by services inside the ANDRIOD server.

The PxP project proposed a mechanism for the dynamic extension of a policy-based management system. However, it only defines the extension mechanism (i.e. it does not cover the decision mechanism or the conflict checking mechanism), which should be included in a PBM system.

The DAIDALOS project aims at building a framework for the integration of heterogeneous network technologies however it is very specific in its choice of protocols to provide services (e.g. IPv4 is not supported). Our research approach is to provide a generic framework for network management; however, the concept of domain-based Provisioning Platform is used in this thesis to provide interoperability among service providers.

2.5.2 Analysis of ABC Service Approaches to Mobile Nodes

The presented testbeds and research approaches have produced very interesting results for the improvement of mobility management and wireless networking. With the analysis of individual projects, it is clear that the focus is on offering mobility to the mobile devices, either directly using mobility protocols, such as Mobile IPv4 or Mobile IPv6, or extending these mobility protocols. Although most of the projects are significant in achieving mobility through handoffs, however, they present limitations in terms of offering ABC services (in terms of providing seamless mobility, QoS, security and cater for context-based requirements of the mobile user).

The micro-mobility projects such as HAWAII, analysed extensions of Mobile-IP to support mobility and offer QoS support, however, it results in high control overhead due to the frequent notifications sent to the home networks, and high latency causing disruptions during handover.

The main aspects for offering ABC service which have not been addressed by any of the reviewed projects, from the mobile device point-of-view are:

- Flexibility for the adherence of access technologies and modification of core networking components.
- Integration support between networks to support seamless mobility across heterogeneous environments.
- Support for inter-network security, authentication and authorisation models.
- Capability to automatically control the behaviour of the mobile node and user based on current network conditions and user’s current requirements.

Recapitulating, the three aspects discussed in this research work that have not been addressed by any of the reviewed state-of-the-art projects, from network management point-of-view are:

- The support for management of heterogeneous mobile environments,
- The possibility of employing management techniques that best suits the network’s current infrastructure, and
- Inter-networking between heterogeneous networks and being able to negotiate the service levels.

These three properties represent the innovative aspects of our proposed framework. We also introduced the concept of a policy-managed mobile client. In this chapter and the following chapters, we will observe that there are a number of approaches for managing networks and managing mobile entities proposed by various research groups and standardisation agencies. Our modular framework for network and mobile entity management will *fill-in* the gaps of available approaches and allow us to employ some of the proven methodologies to provide ABC services to mobile entities in heterogeneous environments.

Chapter 3
Proposed Models

In Chapter 1, we have identified the objective of the proposed research - "to provide managed ABC service over underlying heterogeneous mobile environments". We then deduced that to offer such service requires: (i) a *network management framework* for managing heterogeneous networks and (ii) a *policy-managed mobile client framework* for managing mobile entities. The proposed frameworks are divided into network-side and client-side components respectively. However, our analysis in chapter 2 showed that different approaches proposed for management of networks and mobile clients respectively are not suited for providing ABC services.

In this chapter, we propose a new *policy-based network management model* to support centralised management of networks and distribution of policies from high-level business-like rules to low-level device-dependent commands. The proposed framework extends the basic PBM ideas to provide a *layered approach* for step-by-step refinement of policies and facilitates the negotiation of management services over heterogeneous mobile environments to achieve an always best connected state.

We also propose a new model for a *policy-managed mobile client* (section 3.5) which can interact with the service providing networks and control the behaviour of a mobile device and mobile user (collectively termed as a mobile entity) to be in harmony with the current service requirements and network offerings for those services.

3.1 Policy Based Management System

Nowadays, networks are no longer a single entity providing basic connectivity, but rather treated as a service-enabling platform which is open, intelligent and adjustable to offer a variety of services under the authority of different communities.

The network management, either of telecommunication or data networks, has long been argued along the manager-agent model [81] and deals with three fundamental aspects: a) *functionality* grouped according to five areas, namely, Fault, Configuration, Accounting, Performance and Security [82], b) *information modelling* by which network and network element resources are identified and abstracted in a way that underpins specific operational semantics, and c) the *communication method* among different entities involved.

Management tools that attempt to automate this process do exist; however the diversity of these tools, strong vendor dependence and the lack of interoperability among them reduces their efficiency and usability for generic distributed networks. Moreover, networks have grown significantly in terms of size, complexity and heterogeneity with new applications emerging, and network services (such as VoIP) are increasingly in demand. To this end, it must also support co-existence of different networks and interoperation with different vendor equipment, while, dynamically adjusting its services based on the current network status and changing requirements of the mobile entities.

3.2 Requirements of PBMS Model

The primary objective of any management system is to maintain network and system availability and aid in extending the network services, enhance performance, provide security and reduce operating overhead by automating the administrative tasks. However, the problem of translating these goals into actions remains.

Various standardisation agencies as well as academic and commercial groups are concentrating on policy-based management as a promising solution for managing their networks and providing best available services to their users.

In the following section, we will discuss the requirements of a policy-based management system (PBMS). We will initially discuss requirements for a generic PBMS intended for distributed systems and autonomic computing, and then discuss additional requirements imposed due to our specific requirements of ABC services and heterogeneous mobile environments.

I. Verma in [83] noted that a PBMS model helps in simplifying network management by means of two basic techniques:

1. *Centralised Configuration*

   In a policy-based approach, the network configuration is specified not by configuring each device individually, but specifying the policies for an entire network at a central location. The centralisation point might be the console of a management tool that the network operator is using, or a repository where all the policies are stored. At the central location, where configuration of all the devices is known, various tests and checks can be performed to validate that different policies are mutually consistent. Centralisation of configuration helps to simplify consistency checking.

2. *Simplified Abstraction*

   A simplified abstraction at a higher layer than the physical device configuration, allows network operators to input policies in terms of day-to-day activities. It provides a business-level view of policies rather than specifying the exact marking or rates to be allocated for each network entity.

The author further states that both centralisation and simplified abstractions have associated overheads. A centralised configuration implies that security and integrity of the configuration information is maintained, as well as determines a way to obtain the configuration information from the repository. Although simplified abstractions are great for usability purposes, the simplification comes at the cost of reduced flexibility.

II. Damianou in [42] has identified the following requirements for policy deployment and enforcement in a distributed environment.

- The policy management framework must have the flexibility and necessary abstractions to manage a variety of device types, with different capabilities and limitations, from different vendors. The system architecture should be sufficiently flexible to allow for the addition of new device types with minimal updates and recoding of existing components. To cater for large-scale networks, the management system must be able to apply policy rules to sets of devices rather than individual ones.

- The management framework must be able to adapt to changes in user requirements or changes within the managed network environment. In addition to adapting the behaviour of managed devices, management framework should also adapt its own behaviour, if necessary. Consequently, the management framework must implement mechanisms to modify network behaviour by dynamically changing policies relating to the configuration of managed devices and dynamically selecting which policy should be enforced within the network in order to modify the management strategy.
- The management framework should ensure that the policy is consistent with the functional or resource constraints within the target environment. Static checking should be performed, where possible, prior to deployment, in order to detect inconsistent polices at design time. Furthermore, policy constraints that must be checked at execution times are required for policies related to resource allocation that depend on the current state of the system.

III. Autonomic computing is an approach to self-managed systems with a minimum human interference. It is important to consider the requirements of the autonomic system as they exhibit requirements from the user perspective: flexibility, accessibility and transparency, which are relevant while providing ABC services, where mobile users can demand out-of-profile requirements or request a change in their profile.

IBM research has outlined eight defining characteristics of an autonomic system [84].

- To be autonomic, a system needs to “know-itself”. This implies that the system will need detailed knowledge of its components, current status, and capacity.

- An autonomic system must configure and reconfigure itself under varying and unpredictable conditions. This implies that the policy-managed system should have knowledge about its context - to feedback the managed network, as well as its ongoing behaviour (e.g. connection/disconnection, activity variations and user's preferences).
- An autonomic system never settles for the status quo - it always looks for ways to optimise its operations.
- An autonomic system knows its environment and context surrounding its activity, and acts accordingly.
- An autonomic system cannot exist in a hermetic environment. This implies that the management system should allow for integration of heterogeneous networks, creating an open interworking platform.
- An autonomic system will anticipate the optimised resources needed while keeping its complexity hidden.

The *heterogeneous mobile environments* considered in this research can be viewed as a distributed environment where the components (network elements and processes) are both physically and logically distributed across computers, domains and regions. Also, the *mobile environments* need to be autonomous considering the dynamically changing topology of the networks and ongoing requests from the mobile users. For example, new mobile entities join the network and some mobile entities leave the network, or some user requests to access VoIP or Video-on-Demand functionality– a common scenario in a wireless environment.

3.3 Proposed Characteristics of PBMS Model

A system incorporating all the required elements identified in section 3.2 will be very difficult to build. However, the solution presented in this thesis can be considered as an early attempt to critically examine such concepts. We have considered the above requirements as a benchmark to define specific requirements of our proposed model.

It is clear from the requirements imposed above that (i) a management system must be distributed in order to be scalable (ii) management policies must be expressive and precise enough to be understood by different entities in the hierarchy of policy creation and deployment, and (iii) management procedures need to be automated to proactively behave based on the current network conditions. We have identified that a management framework which needs to offer ABC services, must exhibit the following seven characteristics:

i. ***Modular***

The management framework must be modular. There have been numerous efforts by standardisation groups, academia and the commercial world to offer different functionalities of policy-based management. Some methods and approaches are accepted, standardised or method-of-choice for policy specification, conflict detection, conflict resolution, policy distribution and policy deployment. The management framework should leverage these existing technologies, and provide administrators and network operators the ability to choose their preferred subsystems.

ii. ***Flexible***

The management framework should be flexible. It must be able to cope with different underlying devices, technologies and services. The management framework should be adjustable to changes in protocols, devices or technologies and capable of modification with minimal updates.

iii. ***Adaptable***

The management framework must be adaptable to support the dynamic extension of the management functionality. The framework should tend to implicitly monitor the current status of the network as accurately as possible by sensing network elements. A request/reply methodology should at least be supported. For example, when an active network element fails and could not provide associated services, the management framework should be able to detect (by request and not receiving a reply) and should progressively adjust respective network parameters and service levels.

Furthermore, the management framework should be able to make a decision associated with user's current context i.e. when the user moves in or out of the network, or to support a request for change in a mobile user's profile or a short-term request for a network service. Although, it is not possible for a management framework to intuit user's requirements, however, by maintaining a knowledge base of the current state of network, the framework should be able to accept/reject respective requests of its users.

iv. ***Clustering***

The management framework should be able to cluster the network elements and users as a set of entities rather than individuals. Large-scale networks may contain numerous users and resources. It is not practical to specify policies relating to individual entities – instead, the policy framework should allow policies relating to groups of entities and be applicable within groups of similar attributes.

v. ***Scalable***

The management framework must be scalable in both model and architecture views to be applicable for enterprises or service providers growing in size either conceptually or physically.

vi. ***Conflict Detection***

Policy conflict occurs when the objectives of two or more policies cannot be simultaneously met. For example, a policy P1: "John is allowed to access VoIP services between 9am and 5pm" would be in conflict with another policy P2: "John is not granted access to the VoIP service on weekends". Since, if both these policies are enforced, John is both granted (authorised) and refused (prohibited) permission between 9am and 5pm at weekends.

The management framework should incorporate (or provide a functionality to incorporate) a policy conflict detection mechanism.

vii. ***Interoperable***

The management framework should be interoperable to be able to communicate with other service providers to authenticate and authorise users, and to negotiate different service levels.

The requirements for the proposed management system can be facilitated with policy-based approaches, where support for creating, distributing and deploying of policies can be achieved.

Following the characteristics of the proposed PBMS model, we have also identified other aspects which need consideration while defining the management framework.

Although the following aspects are not considered as core requirements, they present criteria to evaluate the *scalability* and *adaptability* of the system.

3.3.1 Semantic and Syntax analysis

When defining policies for a distributed network, two important issues need consideration: (i) the Semantic model (or language) adopted for understanding and representing abstract policies and (ii) the Syntax approach adopted for interpreting and distributing the policies to network entities.

The Semantic notation provides a common ontology/formal methodology to understand the content of the policies and to detect and resolve conflicts. The Syntax policy notation maps abstract policies to distributable format and support interoperability with different vendor implementations.

3.3.2 Static and Dynamic (run-time) requirements

Although the network community has shown considerable interest in policy-based techniques, proposed solutions are often restricted to condition-action rules where conditions determine which actions should be performed on managed entities. This results in static policy configurations where manual intervention is required to cater for configuration changes and to enable policy deployment.

Whilst most management frameworks may be designed for low-level device configuration, it is important to consider the dynamic adaptation of policies in response to the changes within the managed environment. This ensures, prior to the deployment, that the policies will lead to a feasible implementation for the environment where they apply.

In the following section, we will analyse the IETF PBNM model to evaluate the feasibility of employing this model as our management framework for managing heterogeneous networks.

3.4 IETF PBNM and Missing Links

We have chosen the IETF PBNM model as a management framework to provide always best connected services. The primary reason of adopting IETF PBNM model is that it is one of the first standardisation and manufacturer-independent efforts for defining a formal management framework. Furthermore, it defines a policy meta model and a policy architecture from which many research approaches have been proposed.

However, various adjustments are required in order to use PBNM in a particular application approach for providing ABC services in a heterogeneous mobile environment.

a. The IETF policy framework includes a policy framework definition language, a policy model, a set of policy terminologies, and a policy architecture meta model. However, the IETF does not define a specific policy language; instead different Object Oriented (OO) policy information models are proposed. The *actions* and *conditions* of the policies can be stored in a repository and combined to form different rules. Different IETF working groups are focusing on different functional areas (Figure 3-1): (i) the Resource Allocation Protocol Working group has defined the COPS protocol to standardise the communication and exchange of policy information between PDP and PEP; while (ii) the Policy Framework Working group suggested Policy Core Information Model (PCIM) and Policy Core Information Model Extensions (PCIMe) for mapping abstract policies into object oriented models to be stored in the Policy Repository. However, mapping from one representation model to another is not formally defined [85].
b. In the IETF PBNM model, PDPs do not simply distribute policies to the PEPs. The role of the PDP is (i) to combine the high-level policies with the network state in order to determine the desired behaviour of every device; and (ii) to generate appropriate low-level configuration data for each device (in a supported format and according to its capabilities and limitations) that enforces this behaviour. This implies that if the network state or policies change, the PDP may need to readjust the behaviour of the devices by sending

updated configuration, which makes the PDP a complex component. However, the architectural overview of PDP is not discussed.

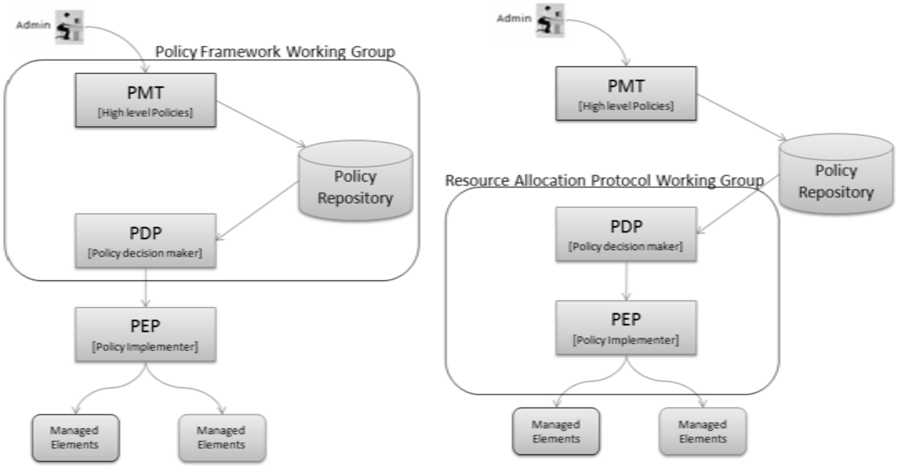


Figure 3-1 Focus of different working groups in IETF PBNM model

c. For policy conflict resolutions, the IETF PBNM Working Group has proposed policy prioritisation where policy rules can be individually prioritised by attaching a priority attribute to the rule. However, as the priority value is manually assigned by the administrator, this approach does not scale well.

d. The Resource Allocation Protocol (RAP) Working group defined COPS for RSVP [33] and proposed COPS-PR protocol for DiffServ policy provisioning [86]. However, in COPS-PR all the intelligence is concentrated at the PDP level. The decisions that a PEP can make are very limited, due to the rigid structure used to store and process policies [87]. This poses some shortcomings in the scalability and distribution of the protocol.

e. The PBNM approach mainly focuses on static policy management and cannot adjust to the dynamic changing environments, as required by mobile entities.

f. While being useful as a general abstract model, the lack of specific policy language makes the IETF’s framework not directly usable. The lack of a policy language also means policy verification and refinement i.e. policy conflict detection and resolution, are not built-in features of the IETF PBNM model. Rather, the support for these depends on the vendor-specific implementation of the framework.

g. As the IETF PBNM model was not designed for providing ABC services, the specific requirements of roaming mobile entities, such as mobility, security, QoS and authorisation have not been supported.

h. The PBNM model mainly focuses on the management of a single organisation or network. This solution is hence somewhat incomplete because of the interoperability issues that arise when heterogeneous environments are operating together. Furthermore, no formal inter-network policy communication methodology has been identified.

Figure 3-2 represents the missing links in the IETF PBNM model while providing ABC services in heterogeneous environments.

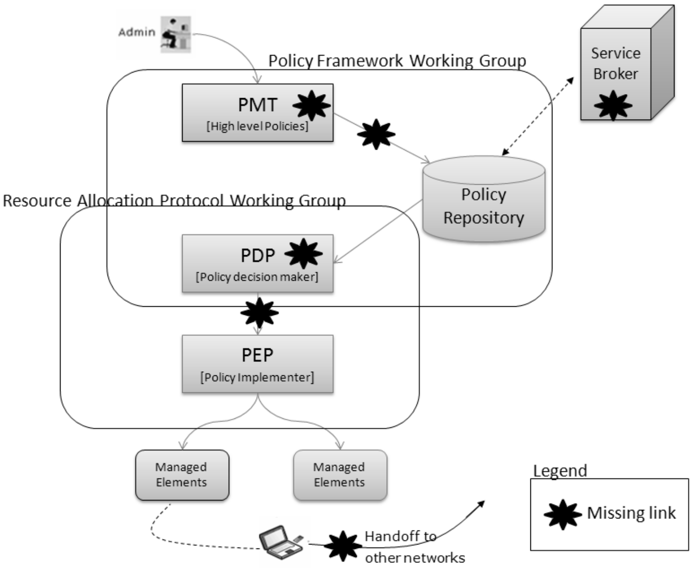


Figure 3-2 IETF PBNM and Missing Links

3.5 Proposed Layered Approach of PBMS

It is generally accepted that there are several layers of policies from the high level business logic to the configuration of network devices. Throughout the literature, we can find different opinions as to the number of levels in policy specification. This, sometimes called policy hierarchy [43] [88], represents different views on policies, relationships between policies at different levels of this hierarchy, or abstractions of policies for the purpose of refining high-level management goals into low-level policy rules whose enforcement can be fully automated.

We propose a four-layer model (Figure 3-3) for the policy-based management framework to allow separate policy representations for both syntactic and semantic analysis, while *mapping translators* between each layer maps policy from one representation to another.

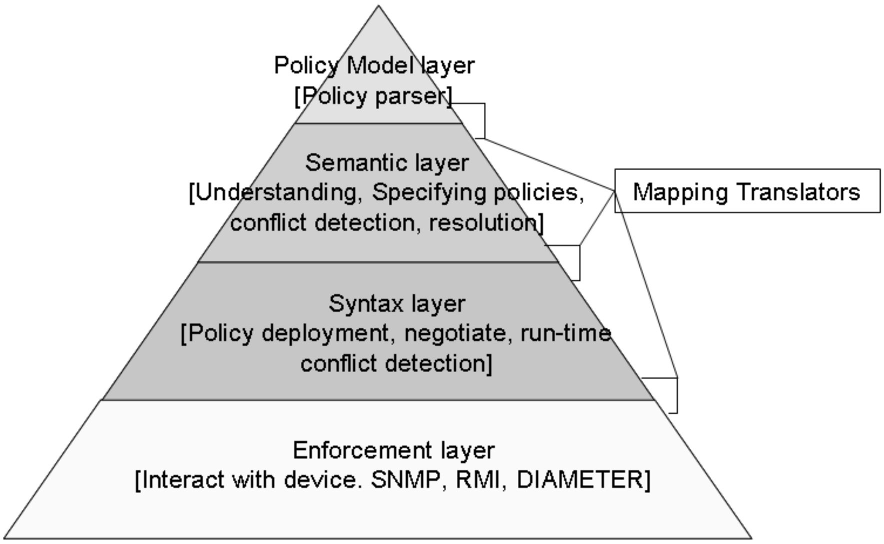

Policy Model layer
[Policy parser]
Semantic layer
[Understanding, Specifying policies, conflict detection, resolution]
Mapping Translators
Syntax layer
[Policy deployment, negotiate, run-time conflict detection]
Enforcement layer
[Interact with device. SNMP, RMI, DIAMETER]

Figure 3-3 Proposed model for managed networks

The functionality of each layer is defined as:

- The ***Policy Model layer*** specifies the high level goal (objective) in general terms, either in a plain natural language text form, or in a semi-structured specification form. In policy parlance, these are business policies, or high level policies.

- The ***Semantic layer*** introduces a formal discipline, where policies are understood and analysed. Policies may be represented as special groupings of rules, with a particular dependency on a conceptual managed entity. The semantically analysed policies need not be at implementable state though they are required to be consistent and unambiguous.

- The ***Syntax layer*** provides a distribution bridge from conceptual semantically correct policies to implementable policies which are understood by the network entities. The syntax layer also communicates and negotiates policies between different service providers.

- Finally, the ***Enforcement layer*** is the bottom layer, where the decisions are implemented in terms of device commands. The rules derived from the policy specifications may use other status data provided to resources, entity access rights, conditions, security rules, or specify certain types of execution engines supporting the prescribed actions.

Hence, while the use of different abstraction layers allows a stepwise refinement from an informal high-level business policy to a set of network operational commands, the proposed model also provides a clear distinction between understanding and implementing policies, and thus combines the advantages of both types of representational schemes. Also, a *layer* can be fully customised or completely ignored, if not required by the organisation.

We have termed the proposed model as a *layered model* rather than the generally used term of *policy hierarchy*, as *hierarchy* implies a system of ranking and organising things or people, where each element of the system (except for the top elements) is a subordinate to a single other element [89]. In our approach, however, the defined layers imply a *conceptual process* of refining a policy from business words to network commands rather than a rigid algorithmic procedure. Hence, by following the modular approach, a *layer* can be fully customised or completely ignored, if not required by the organisation.

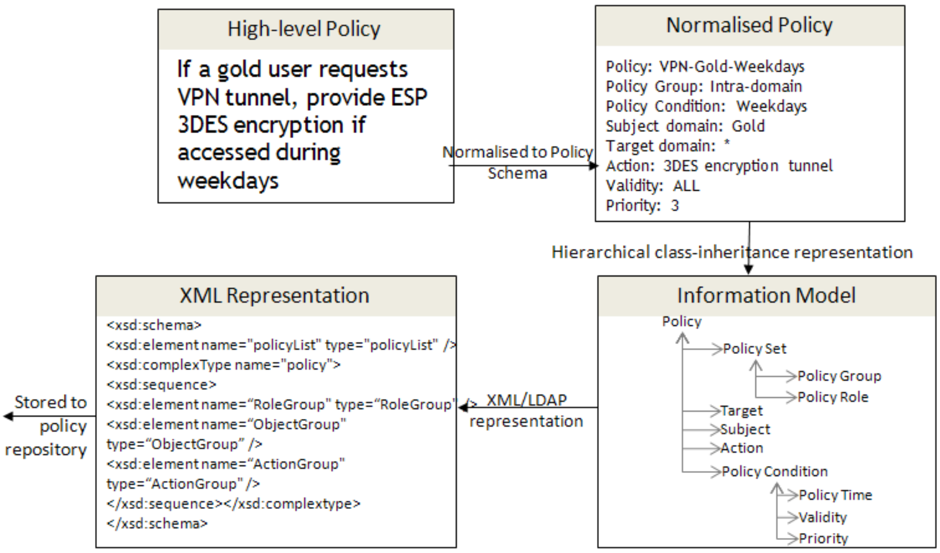
The following subsections will elaborate on each defined layer:

3.5.1 **Policy Model Layer**

The Policy Model layer normalises the policies in accordance with the schema of the organisation. It presents a conceptual framework for policy normalisation and provides an information model for representing high-level policies which can then be stored in a policy repository. The stored policies can be later analysed by the Semantic layer for inherent conflicts.

For sample purposes, we have identified a service profile schema for a company "ABC" where its employees have either Gold, Silver or Bronze profile. The Gold profile users are offered high-level VPN security for remote connections to the company and VoIP service support during weekdays. A typical *GoldSecurity* policy can be defined as: *If a gold user requests a VPN tunnel during weekdays, provide high-level security (i.e. ESP 3DES encryption).*

The Policy Model layer will process the *GoldSecurity* policy as shown in Figure 3-4.



If a gold user requests VPN tunnel, provide ESP 3DES encryption if accessed during weekdays

Policy: VPN-Gold-Weekdays
Policy Group: Intra-domain
Policy Condition: Weekdays
Subject domain: Gold
Target domain: *
Action: 3DES encryption tunnel
Validity: ALL
Priority: 3

```
<xsd:schema>
<xsd:element name="policyList" type="policyList" />
<xsd:complexType name="policy">
<xsd:sequence>
<xsd:element name="RoleGroup" type="RoleGroup" />
<xsd:element name="ObjectGroup"
type="ObjectGroup" />
<xsd:element name="ActionGroup"
type="ActionGroup" />
</xsd:sequence></xsd:complextype>
</xsd:schema>
```

Figure 3-4 Functionality of Policy Model Layer

A high-level management policy is transformed to a normalised form according to a schema defined by the organisation. The normalised policy is then mapped to an object oriented Information Model and finally to XML/LDAP representation to be stored in the policy repository. The derived information model needs to be in a standard format (XML/LDAP) to achieve inter-operability between different vendor implementations.

Note that the presented approach for policy normalisation, mapping to information model and its XML/LDAP representation is our proposed sequence and approach. Organisations may prefer their specific approaches, provided that the basic functionality of the Policy Model layer i.e. policies are in accordance with the policy schema of the organisation, is achieved.

Consider a scenario, where policies are simple management policies and can be directly applied to network elements, for example configuring firewall rules based on network addresses; then the Policy Model layer may itself be sufficient to represent the policies for the organisation (or a testbed) together with the Enforcement layer where rules can be directly applied as network commands.

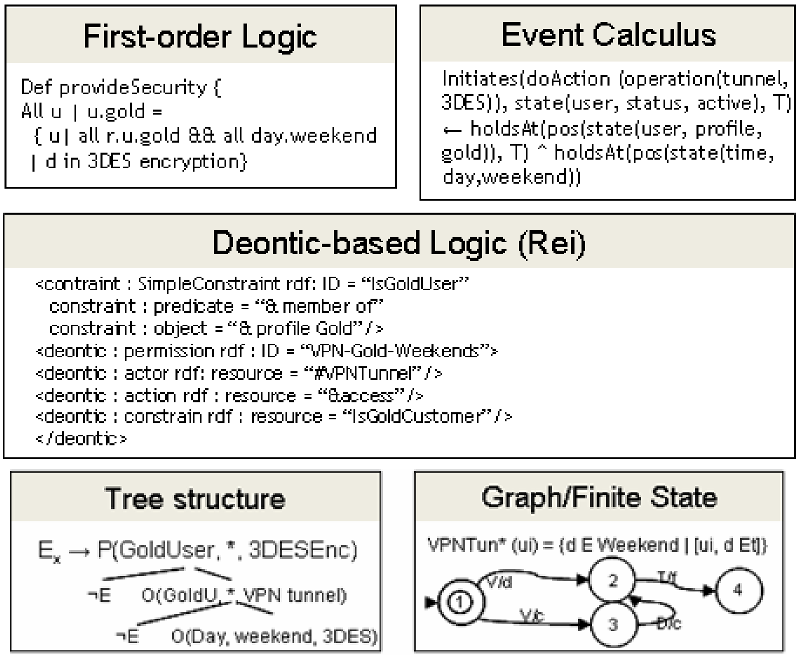
The functionalities of the Policy Model layer is described in detail in Chapter 4 together with the proposed Policy Schema and Information Model.

3.5.2 **Semantic Layer**

The Semantic layer defines a formal methodology to understand and analyse policies for detecting inherent conflicts and for providing conflict resolution.

Previous researchers have proposed various approaches to define semantic notations for describing a policy. Proposals from rule-based notations to RDF/OWL (e.g. KAoS, Rei, SWRL) based ontology [90], to Deontic and Predicate logic have been employed. These conflict detection approaches are generally focussed on modality conflicts, policy propagation, action composition, separation-of-duty and Chinese Wall security policy [91]. Semantic analysis based on graph, tree and finite state automata have also been proposed.

GoldSecurity policy (presented in section 3.5.1) can be analysed by different approaches as shown in Figure 3-5. The Semantic Layer is described in detail in Chapter 5 with an overview of current approaches for semantic analysis and our proposed approach.



```
Def provideSecurity {
All u | u.gold =
  { u| all r.u.gold && all day.weekend
  | d in 3DES encryption}
```

```
Initiates(doAction (operation(tunnel,
3DES)), state(user, status, active), T)
← holdsAt(pos(state(user, profile,
gold)), T) ^ holdsAt(pos(state(time,
day,weekend))
```

```
<contraint : SimpleConstraint rdf: ID = “IsGoldUser”
  constraint : predicate = “& member of”
  constraint : object = “& profile Gold” />
<deontic : permission rdf : ID = “VPN-Gold-Weekends”>
<deontic : actor rdf: resource = “#VPNTunnel” />
<deontic : action rdf : resource = “&access” />
<deontic : constrain rdf : resource = “IsGoldCustomer” />
</deontic>
```

Figure 3-5 Functionality of Semantic Layer

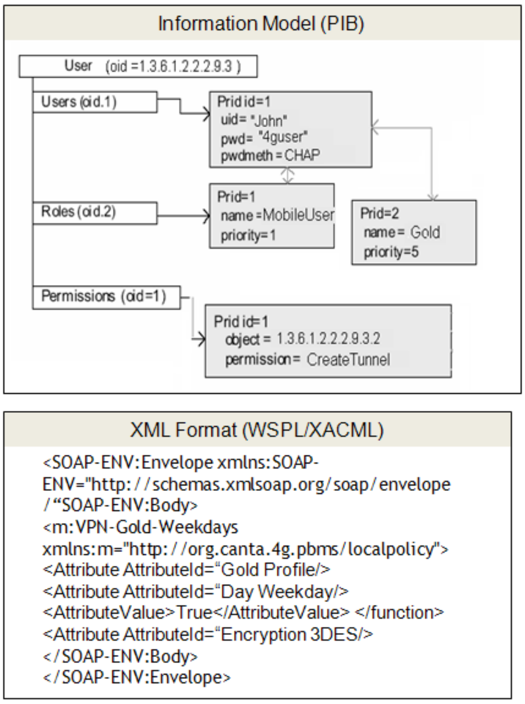
3.5.3 Syntax Layer

The Syntax layer supports a consistent format for the distribution of policies. It focuses on formal policy structure, policy negotiation and dynamic (run-time) requirements of the network elements and users for both intra- and inter- domains.

Several extensions to traditional, instance-specific approaches have been proposed by various IETF working groups (e.g. SNMPConf, COPS and COPS-PR), which could provide the necessary capabilities required for configuration, distribution and monitoring with an integrated approach to management. Also, Web Services [92] is becoming a de facto for managing access control in a request/response format. Although limited in semantic expressability, the major benefits of web services are that a service provider can publish its policies and verify that the received messages harmonise with its own policies and hence could be used for identifying network attacks. Other benefits include determining mutual agreeable policy between service providers or users, and interoperability.

Figure 3-6 shows the distribution of *GoldSecurity* policy represented in structural formats such as Policy Information Base (PIB) [93] (a conceptual tree later to be encapsulated by the COPS protocol) or in a XML format.

In Chapter 6, we will discuss the Syntax layer in detail and analyse our adaptation of COPS protocol for intra-domain policy deployment and eXtensible Access Control Markup Language (XACML) [94] for inter-domain policy negotiations.



```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope
/"SOAP-ENV:Body>
<m:VPN-Gold-Weekdays
xmlns:m="http://org.canta.4g.pbms/localpolicy">
<Attribute AttributeId="Gold Profile/>
<Attribute AttributeId="Day Weekday/>
<AttributeValue>True</AttributeValue> </function>
<Attribute AttributeId="Encryption 3DES/>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

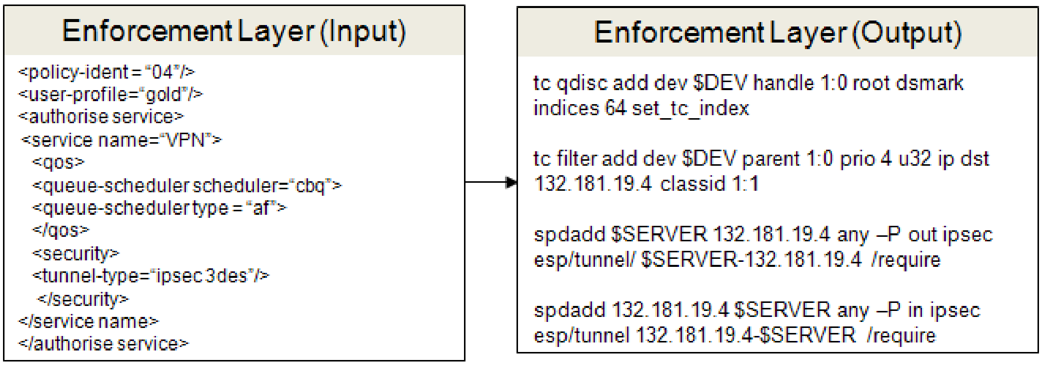
Figure 3-6 Functionality of Syntax Layer

3.5.4 Enforcement Layer

The Enforcement layer translates the policies from Syntax notation to configuration instructions, which are then deployed to the network devices.

The Resource Allocation Protocol (RAP) working group has proposed the COPS protocol for lower layers as a common message bus; however, conventional methods for network management such as SNMP, RMI, Command Line Interface (CLI), or a set of user-defined scripts may also be deployed. Newer approaches such as extending the Diameter protocol [95] or using distributed communication as in the CORBA, are also possible.

Hence, the *GoldSecurity* policy expressed at this level (Figure 3-7) would specify Class Based Queuing (CBQ) to realise Assured Forwarding services in a DiffServ network, and mark the packets arriving from IP address 132.181.19.4 to be treated in accordance with IPSec ESP tunnel requirements.



```
<policy-ident = "04"/>
<user-profile="gold"/>
<authorise service>
 <service name="VPN">
   <qos>
   <queue-scheduler scheduler="cbq">
   <queue-scheduler type = "af">
   </qos>
   <security>
   <tunnel-type="ipsec 3des"/>
    </security>
</service name>
</authorise service>
```

```
tc qdisc add dev $DEV handle 1:0 root dsmark
indices 64 set_tc_index

tc filter add dev $DEV parent 1:0 prio 4 u32 ip dst
132.181.19.4 classid 1:1

spdadd $SERVER 132.181.19.4 any –P out ipsec
esp/tunnel/ $SERVER-132.181.19.4 /require

spdadd 132.181.19.4 $SERVER any –P in ipsec
esp/tunnel 132.181.19.4-$SERVER /require
```

Figure 3-7 Functionality of Enforcement Layer

3.5.5 **Mapping Translators**

The Mapping Translators (Figure 3-3) provide a formal process to translate the policies from one representation to another. The key requirements are:

1. The mapping of the policies between layers needs to be unambiguous to preserve the policy semantics (meaning).
2. To support interoperability between different vendor implementations and flexibility to select different approaches at Semantic and Syntax layers, the input and output of the mapping translator needs to be coded in XML format with predefined tags (Document Type Definition).

The use of XML as a standard mapping translator has advantages. XML is ideal for transferring information between heterogeneous platforms as XML parsers are available for many platforms and hence supported by many Semantic and Syntactic analysis approaches. Furthermore, XML policy documents can be validated using a XML policy schema and the syntax for any mapping is done intrinsically by XML parsers through the validation of the XML policy using its schema.

Organisations may employ DMTF CIM-XML based [96] translators, or define their own XML schema for mapping. Customised translators can provide specific functionalities and offer well-suited support (e.g. Policy Model Layer → Semantic Layer), while standards based translators provides inter-operability.

Note that the choice of XML for policy-based translation is not meant for detecting or resolving any policy conflict but for the translation of policies either from higher-layer to lower-layer (i.e. Semantic layer to Syntax layer) or distribution of policies to

Semantic analysis module or Syntax deployment modules (if semantic analysis is not performed) when they are distributed across the network. Considering the diversity of specific options chosen for the Policy Model layer, Semantic layer and Syntax layer approaches, and the assumption of ignoring any specific layer, the choice of Mapping translator is very specific to the particular management framework where it is applied.

In the previous sections, we have identified the requirements of the management framework for managing networks and presented our proposed model. As discussed in the objective of the research work, a mobile client is required to interact with the service providing networks and control the behaviour of the mobile entity to provide an ABC scenario to the user. In the following section, we will discuss the requirements of such *policy-managed mobile client* and propose a layered approach for designing its framework.

3.6 Policy Managed Mobile Client

With the proliferation of handheld devices and growth of different wireless technologies offering multitude of services, the authentication, authorisation and security of mobile entities need to be addressed together with transparent handoffs from one service provider to another service provider (offering better services at that time) and seamlessly maintaining active connectivity sessions.

Hence, the general requirements, such as mobility, security, authentication and QoS, for managing mobile entities need to be considered.

3.6.1 Requirements of Policy Managed Mobile Client

The ABC service requires that a mobile entity needs to be able to connect to different access technologies (such as WLAN, GPRS, CDMA, etc.), at different locations, and switch between them in a transparent way. The handoff must be fast and transparent with minimal effect on a user's current sessions and occur *only* when necessary. The optimal handoff occurs where users are not constrained by the drawbacks of connection to a specific network and the ability to switch to a better available network. The requirements of a framework for policy managed mobile client are identified as:

i. ***Transparent handoff***

The seamless roaming and transparent handoff of all active sessions from one access network to another is considered as one of the fundamental issues in deployment of ABC services (and fourth generation wireless networks).

ii. ***Optimum Time to handoff***

Any handoff procedure involves a set of protocols to notify all the related entities of a particular connection that a handoff has been performed, and that the connection has to be redefined. With every handoff, users may lose data packets and even lose active sessions. Hence, the mobile client framework should be able detect whether a handoff is required and if so, select an optimum time to handoff.

iii. ***Support for protocols***

The framework should be able to support the protocols of the services to be offered to the mobile user. Standardised protocols, such as Mobile-IP for mobility, Radius/Diameter for authentication, WEP/802.1x for security in data-link layer while IPSec/VPN support for security in IP-layer [97], need to be supported.

This requirement does not impose a list of protocols which have to be implemented/ supported by the management framework, though it merely states that a support for protocols is required for the services being offered. For example, if the *home agent* supports integrated protocols such as MIP-AAA [98], then mobility and authentication concerns can be solved together. Similarly, if the mobile user wants to enjoy multimedia communication sessions such as voice and video over the Internet, then both network and mobile device need to support protocols like the Session Initiation Protocol [99] (SIP).

iv. ***Support for Profile***

The framework should support a profile-based management system. This profile can be handled manually or may be downloaded from the service provider directly into the mobile device.

A profile contains two types of information. An *internal profile* maintains information which is managed internally by the mobile device. It includes information regarding the personal preferences of the user (e.g. priority of access networks based on bandwidth, services and cost), security-profile for different locations (e.g. connecting from home network to office requires VPN tunnel) and authentication information (e.g. connecting to office VPN requires digital certificate). An *external profile* is supported by the network and may contain the role/status of the mobile user (such as backup operator of company; or gold user of cellular network). Depending on this profile information, the management framework of a mobile device can control the behaviour of a device and communicate with the service providers.

3.6.2 Proposed Model of Policy Managed Client

We have defined the client model (Figure 3-8) as a four-layer stack: Policy layer, Mobility layer, Enforcement layer and Network layer.

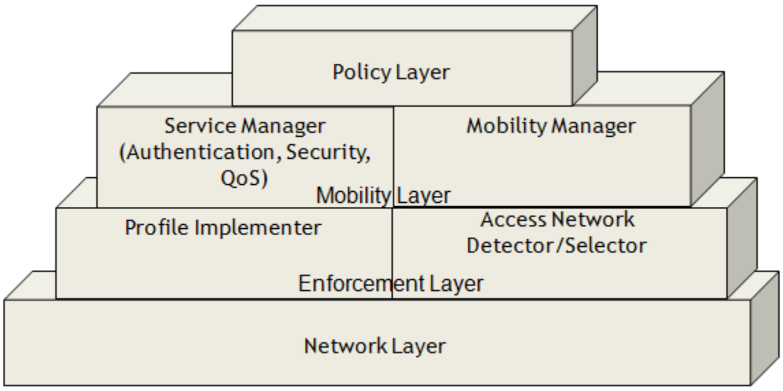


Figure 3-8 Proposed model of Policy-managed Mobile Client

a. The ***Policy layer*** downloads the user profile from its home network and negotiates with the offerings from the roaming network. It can also act as a PEP in the IETF PBNM model.

   We have introduced a new *Infrastructure* (I) parameter I+/I- to ensure whether the offered infrastructure from the service provider is acceptable to the mobile entity. This then assists the mobile device to behave according to the user's current profile and network's present conditions.

b. The ***Mobility layer*** is mainly concerned with providing seamless mobility to the mobile entity. It contains a set of *managers* to manage a set of protocols which are used to offer a set of services.

   The Mobility layer contains two major components (i) the Mobility manager, which manages the mobility protocol (Mobile-IPv4, in our case) and maintains the current sessions when the mobile device is in the state of handoff between different access networks; and (ii) the Service manager.

   The Service manager is a set of *managers* to support all the services for a mobile entity. It may contain *managers* to support Authentication, Security and QoS (as in our case), or specialised protocols like SIP for specialised services. The role of the Service manager is generic as there are different services (authentication, QoS, security) to be managed, for example, to provide credential for authentication and maintain acceptable security state. There may be different authentication and security requirements for user and device imposed from its home and guest access network, which must be fulfilled for a successful connection and a handoff.

c. The function of the ***Enforcement layer*** is to implement the *internal* and *external* profile either created by the mobile user or downloaded from the network.

   The Enforcement layer has two main components (i) Profile implementer, which implements the downloaded high-level profile to device relevant information. It maintains both the statistical information and the abstract information of the profile. For example, an *external profile* states that a *gold* user is allotted *bandwidth = 1.2Mbps* and *delay = 120ms,* while *internal profile* network selection priority list is defined as *Ethernet > Wireless > CDMA* interfaces; and (ii) Access Network Detector/Selector module, which incorporates sophisticated network detection and optimum network selection algorithms. The Access Network Detector/Selector continually monitors available access networks. It maintains a list of available options for mobile devices and selects the best time and access network to perform the handoff, based on *internal* and *external* profiles.

d. The ***Network layer*** functions in accordance with the decisions of the Policy Implementer by implementing profile-based policies into low-level device specific commands. The command interpretation is specific to the mobile device with support for various wireless and wired access technologies such as 802.11x (wireless), 3G, GRPS, GSM and Bluetooth.

   The Network layer directly interacts with the network interfaces (wireless radio) and is dependent on the device drivers from diverse laptops/notebooks (such as from HP, Sony, Acer - running Windows; or Mac) or mobile devices such as from Nokia, Sony Ericsson (Symbian), Microsoft (Windows Mobile), Blackberry (RIM), Google (Andriod) and many others.

In this section, we have discussed the requirements of a policy managed mobile client to offer ABC services and proposed our model. The following section presents an analysis of the proposed *layered approach* and proposed models.

3.7 Analysis of Proposed Layered Approach

Historically, network management has focused on setting parameters of individual interfaces of a device one at a time. Recent innovations of policy management, ranging from new protocols to the use of information models to represent policy rules, have helped in simplifying the management task. Most research groups consider policies as to be a set of rules that express a set of conditions, and if those conditions are met, one or more actions will be executed. However, there are two keys issues which need to be considered: users and processes [100].

Different types of people use policy. Business people do not want to express their policies in network terminology, and networking people do not want policies written using business concepts. However, both business and network policies should be consistent to ensure that network services are managed according to the business goals of the organisation. A formal approach is needed that can translate business needs into device configuration.

The second important feature is process. Every configuration change has an underlying set of business rules that govern its deployment. Business procedures define who can access what, and any policy definition needs to be consistent with the overall objective of the organisation. Policies define how resources of the organisation (or network) are accessed and allocated. Different users and services have different requirements, and policy enables appropriate processes to be applied.

The realisation that business rules and processes, device configuration, and access control of users are tightly bound together. Hence, a modular, scalable and interoperable management framework is needed which can represent the managed environment as a set of entities. The proposed research, therefore, focuses towards proposing a management framework for managing mobile *environments* rather than mobile networks. In our case, the term *environment* includes the users, devices, rules and processes.

The proposed management framework transforms high-level business-like policies into low-level device dependent commands which are used for configuration, monitoring and management of network entities. By placing separate layers of translators between the administrator and the system, our proposed model greatly simplifies the task of enforcing consistent policies throughout large and distributed systems. The layered approach is a way of splitting the vast number of policies into smaller groups of different levels of abstraction, which can be further processed into distinct steps and transformed into applicable low-level policies.

The proposed *layers* also represent different *views* on policies hence defining a policy hierarchy within the management environment in which policies are applied. In other words, the lower the level of abstraction, the more precise and detailed will the definition becomes, i.e. the granularity of the criteria increases, hence identifying policy’s subjects, targets and actions more precisely. This provides two main benefits: (i) as policies are applied within their respective levels (i.e. at Policy Model layer, Semantic layer, Syntax layer or Enforcement layer), a suite of policy conflict detection and resolution methods can be employed. Further, with this *layered-abstraction* offered at higher layers than the physical device configuration, the network administrator is required to input policies in terms of day-to-day activities, hence simplifying the management, conflict detection and resolution tasks; and (ii) feedback and monitoring of the offered services is simplified. As an example, the administrator specifies the priority for different applications or the response time desired for an application, rather than specifying the exact marking or rates to be allocated for application flows. Because of the monitoring and feedback methodology, the managed network entities can also provide a means of specifying the adaptive behaviour in networks to support dynamic and real-time changes.

Furthermore, the proposed Syntax layer provides a bridge for *interoperability* between different service providers to communicate policies and negotiate service level agreements and authorisation of any roaming mobile entity. Hence, the proposed management framework can be put into practice by service providing networks to achieve interoperability, and the proposed policy managed mobile client can roam between the managed networks to achieve always best connected services (Figure 3-9). The proposed models focus on providing a means for profile-based service differentiation, access control and roaming management for seamless mobility of mobile users across heterogeneous mobile environments, while taking into consideration QoS and security requirements of both mobile users and networks.

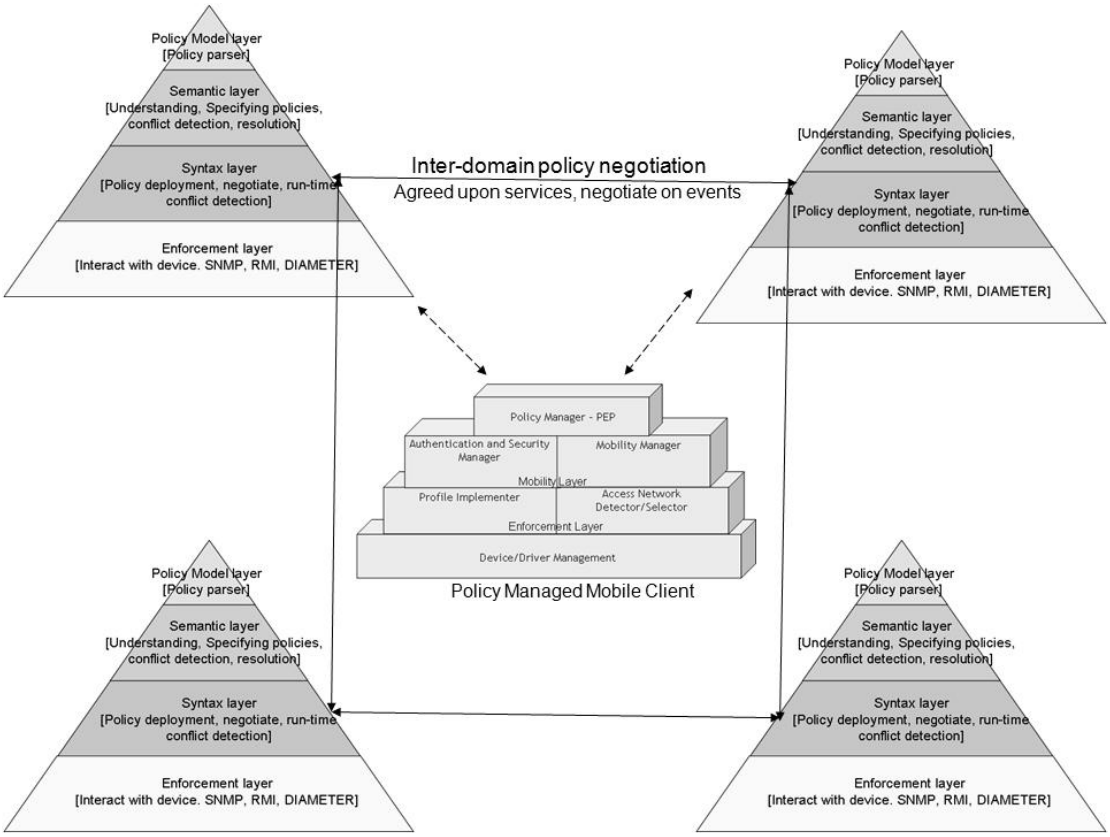

Policy Model layer
[Policy parser]
Semantic layer
[Understanding, Specifying policies, conflict detection, resolution]
Syntax layer
[Policy deployment, negotiate, run-time conflict detection]
Enforcement layer
[Interact with device. SNMP, RMI, DIAMETER]
Inter-domain policy negotiation
Agreed upon services, negotiate on events
Policy Manager - PEP
Authentication and Security Manager
Mobility Manager
Mobility Layer
Profile Implementer
Access Network Detector/Selector
Enforcement Layer
Device/Driver Management
Policy Managed Mobile Client

Figure 3-9 Interoperability between managed networks and mobile client

As the presented layered-approach provides a modular functionality where different approaches, methods or standards can be employed for Policy Model layer, Semantic layer, Syntax layer and Enforcement layer, the proposed model can also be considered as an extension of the IETF PBNM framework (Figure 3-10).

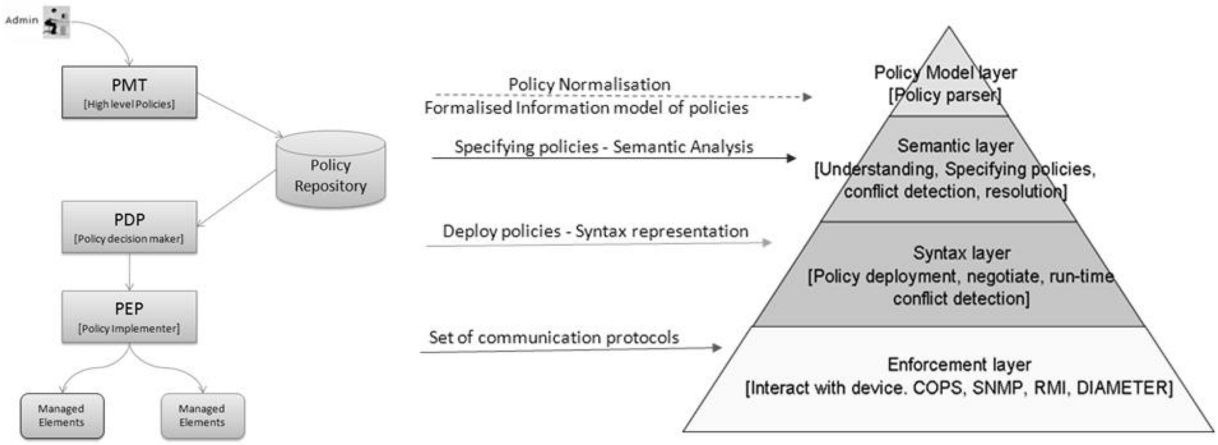

Admin
PMT
[High level Policies]
Policy Repository
PDP
[Policy decision maker]
PEP
[Policy Implementer]
Managed Elements
Managed Elements
Policy Normalisation
Formalised Information model of policies
Specifying policies - Semantic Analysis
Deploy policies - Syntax representation
Set of communication protocols
Policy Model layer
[Policy parser]
Semantic layer
[Understanding, Specifying policies, conflict detection, resolution]
Syntax layer
[Policy deployment, negotiate, run-time conflict detection]
Enforcement layer
[Interact with device. COPS, SNMP, RMI, DIAMETER]

Figure 3-10 Layer mapping from IETF PBNM and Proposed model

The Semantic and Syntax layers can be collectively referred to as the PDP; while the Enforcement layer can be considered as the PEP, responsible for communicating with the PDP and installing the configuration policies onto edge network devices. The communication between PDP and PEP could be managed using the COPS protocol and PIB using mapping translators.

Chapter 4
Policy Model Layer

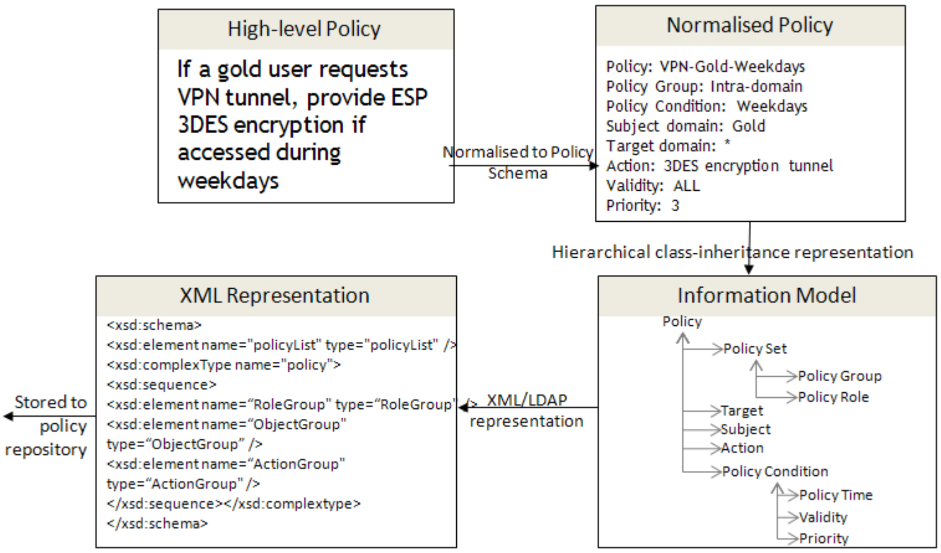
In previous chapters, we have discussed that management systems are derived from business goals. The policies are principles that drive the management systems and define behaviour of applications and networks. We also discussed the process of refining these abstract, business goals into policies relating to specific services, and then into policies implementable by specific devices supporting the service. To apply and deal with this idea, a number of concepts have been pioneered. Numerous policy definitions, policy hierarchies and policy models have evolved. However, these approaches are diverse, as they were developed from diverse points of view and without a common management framework model.

In Chapter 3, we have identified that a complete management framework should include a unified model for representing policies, users and resources and also mechanisms for distributing and enforcing these policies among heterogeneous environments. We further deduced that the IETF PBNM model is not sufficient for providing ABC services and a new management framework is required. The required management framework must be modular, flexible, adaptable and scalable. We then proposed a management framework based on a layered-approach where a stepwise refinement of high-level business-type policies to low-level machine commands is possible.

Following the modular approach of our proposed model, we state that organisations can employ any method, procedure or language best suited to their specific requirements to *fill-in* for Policy Model layer, Semantic layer, Syntax layer and Enforcement layer. In this chapter, we will discuss the functions and requirements of a generic Policy Model layer, followed by our proposed approach.

4.1 Policy Model Layer

As introduced in section 3.5.1, the Policy Model layer provides a conceptual framework for policy normalisation and provides an information model for representing high-level policies which are then stored in a policy repository. For example, a *GoldSecurity* policy defined as "*If a gold user requests a VPN tunnel during weekdays, provide high-level security (i.e. ESP 3DES encryption)*" and can be processed as shown in Figure 4-1.



If a gold user requests VPN tunnel, provide ESP 3DES encryption if accessed during weekdays

Policy: VPN-Gold-Weekdays
Policy Group: Intra-domain
Policy Condition: Weekdays
Subject domain: Gold
Target domain: *
Action: 3DES encryption tunnel
Validity: ALL
Priority: 3

```
<xsd:schema>
<xsd:element name="policyList" type="policyList" />
<xsd:complexType name="policy">
<xsd:sequence>
<xsd:element name="RoleGroup" type="RoleGroup" />
<xsd:element name="ObjectGroup"
type="ObjectGroup" />
<xsd:element name="ActionGroup"
type="ActionGroup" />
</xsd:sequence></xsd:complextype>
</xsd:schema>
```

Figure 4-1 Functionality of Policy Model Layer

The Policy Model layer steps can be identified as:

1. The input policy is *normalised* to different tokens based on the Policy Schema.

   A *normalised policy* is defined as a policy which can be categorised into tokens based on the Policy Schema of the organisation. A Policy Schema (discussed in section 4.2) represents the overall structure of the organisation defining the entities and relationships between those entities.

   For the *GoldSecurity* policy presented above, the business-level policy can be transformed into structural entities (as shown below), based on a conceptual framework of the organisation.

   > GoldSecurity: If a gold user requests VPN tunnel, provide ESP 3DES encryption if accessed during weekdays

2. The normalised policy is then represented by an *information model*.

   An *information model* (discussed in section 4.3) is a formal way to represent the conceptual structure of the organisation. It provides a uniform and consistent representation of the entities in the organisation. An entity can be a person, a computer, a router, or even a protocol message.

3. The structured policy is then stored in a policy repository in a standard format (e.g. XML or Lightweight Directory Access Protocol (LDAP) format [101]). Policy storage is discussed in section 4.4.

Herewith, the objectives of the Policy Model layer can be identified as:

a. To provide a conceptual framework for policy normalisation

The Policy Model layer should provide a conceptual framework of the organisation and define a *schema* for which policies are to be written.

In general terms, a schema defines the structure and contents of any information resource. In database terms, a schema identifies the entities and the types of attributes for those entities. IBM has defined a schema as a set of statements, expressed in a data definition language to completely describe the structure of a database [102]. We have identified schema such that if the users and network elements of a managed network are to be collectively referred to as "entities", then a *schema* represents the structure of these entities, hence providing an overall structural model that identifies how different entities are structured and the relation between those entities in a managed environment.

b. To provide an information model for representing high-level policies

The Policy Model layer should use an information model. An information model is defined as an abstract but formal representation of entities including their properties, relationships and the operations that can be performed on them [103]. It provides a common language in which different types of management entities can be defined.

The relationship between *schema* and *information model* can be defined as, if the schema represents the structure of the entities then information model formally represents the schema.

c. *To store the structured policies into a Policy Repository*

The Policy Model layer should support a policy repository where policies, which are modelled by the *information model*, can be stored and can be retrieved later for policy analysis and distribution of policies across network(s). The policies represented by an information model need to be mapped to a structured specification such as XML or LDAP format. The choice of XML or LDAP is up to the network administrators and the current support in their network.

In different research approaches towards PBNM models, the term *network* and *organisation* has been used interchangeably. We understand that the term *organisation* is more relevant when policies are specified for managing "a group of people who work together" such as business places, schools, hospital, etc., while the term *network* is more relevant when policies are specified for managing network elements, such as routers, firewalls, etc. However, a *network* can be treated as an *organisation* of network elements together with the users of the network services. We have identified that users and devices are tightly bound together and referred to as *entities*, while the business policies, entities and relationships between entities is collectively referred to as an *environment*.

As discussed in the requirements for a policy-managed network, *the proposed framework should be able to view the network and users as a set of entities rather than individual devices i.e. clustering*. This requires a formal structural definition of the environment to be managed. In the following section, we will define *Policy Schema*, which represents the schema of a managed network in our proposed policy-managed system.

4.2 Policy Schema

A Policy Schema models the schema of the organisation. It represents the overall structure of the organisation by grouping entities into different hierarchies to reflect the relationships between them. The advantage of defining a Policy Schema is that it provides a structure of an organisation where different rules can be applied to a set of entities rather than particular individuals based on their hierarchy and relationships i.e. *clustering*.

A simple example of a Policy Schema (Figure 4-2) can be where an *administrator* manages *team leaders*, each *team leader* supervises some *team members* and each *team member* is allocated a number *clients*. Hence, when a policy is applied to a *team member*, it is implicitly applied to all the users who are *team members*.

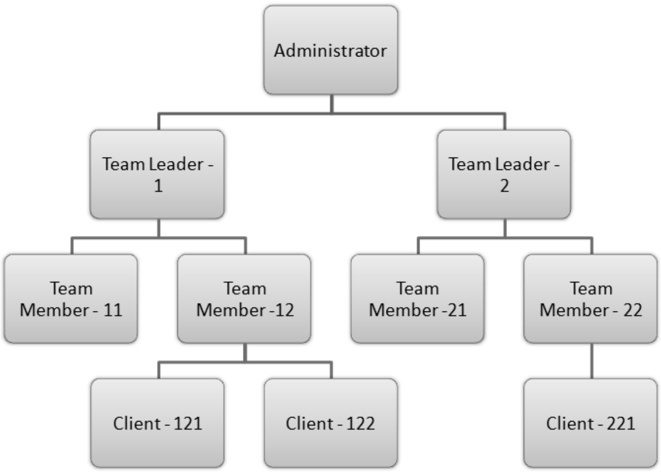


Figure 4-2 Simple example of Policy Schema

Even though the IETF PBNM model does not define such structuring of an organisation, there have been some efforts related to defining a Policy Schema. The following section discusses some of these efforts:

4.2.1 Ponder

Lupu et. al [104] proposed a management framework where the main components are: (i) domains for grouping objects, (ii) a policy service to support the specification and storage of policies, and (iii) roles to reflect the organisational structure. The concept of *subject*, *target* and *domains* are also introduced. This framework was later ported to Ponder language [41] (discussed in section 2.2.2).

The *subject* of a policy specifies the human or automated manager to which policies apply. The *target* of the policy specifies the objects on which *actions* are to be performed. Domains are groupings of objects and are defined to be similar to the file system directories. The subject or target of a policy is expressed as a domain of objects and the policies are applied to all the objects in the domain, so that a single policy can be specified for a group of objects [105]. The authors further suggest that this helps to cater for large-scale systems as it is not necessary to define separate policies for individual objects in the systems, but rather for groups of objects.

The introduction of the concept of domains is of our particular interest. Domains provide a flexible means of partitioning the objects in a large system according to geographical boundaries, object type, management functionality and authority or for the convenience of human managers. In our definition, domains are used to group objects in order to apply a common policy to a set of objects e.g., in a department within a company, or to a pool of users. A domain does not encapsulate the object it contains but merely holds references to the object interfaces. A membership of a domain is explicit and not defined in terms of a predicate on object attributes. A domain is thus very similar in concept to a file system directory but may hold references to any type of object including a person (i.e. an entity).

4.2.2 **DCCM**

Another prominent approach for defining an overall schema of the organisation was initiated by Dynamic Cryptographic Context Management (DCCM) [106]. It defined an organisation as a *secure group* and identified the *group* to be a collection of *participants* authorised to access a set of *information.*

DCCM is a Defence Advanced Research Projects Agency (DARPA) funded project to develop and demonstrate an efficient technique to provide security for very large, dynamically changing groups of participants. The “large” is defined as groups with a number of members typically ranging from 10,000 to 100,000 or more. “Dynamic” means that as new members may be added to the group at any time and existing members may be evicted from the group, thereby requiring immediate changes to some of the services and authorisation. These two characteristics of the DCCM project are of particular interest in our research work.

DCCM has defined an overall schema of the organisation as a *secure group* which is a collection of members who are authorised to access a set of information. Secure group mechanisms enable the members, and only the members, to access the information. It recognises and supports a range of schemas for the organisation, from a broad flat model to a strict hierarchical model. A hierarchical group is based on the inherent structure of the members forming a group.

The policies and supporting infrastructure for the group also take advantage of existing structure and relationships between members. For example, there may be multiple authentication policies and servers for a group, based on the existing policies within the hierarchy or across multiple hierarchies.

The most basic entity in a secure group is the *participant*. Typically a participant represents a person, but it can also represent a device, or a piece of software. A group of participants sharing a secure group communication mechanism for a specified period of time for a common purpose is defined as *session*. A session implies that all of the participant share the same security mechanisms and share a common security configuration. For example, a single broadcast for a group would be a session. A *project* is a set of sessions occurring over time. All the sessions within a project use the same policies to support the same set of participants. The project is the unit of administration for access control.

This access control framework administers the list of participants in a project and enforces an access control policy between project members and non-members. Multiple sessions can occur simultaneously within a project, and a participant can join more than one of the sessions. The highest level in organisational schema is defined as a *system*. A system is the supporting infrastructure for a set of related participants that transcends individual projects. The system maintains a single authentication database that is used across multiple projects.

4.2.3 **RBAC model**

The NIST RBAC standard is composed of two parts: RBAC *Reference Model* and RBAC *System and Administrative Functional Specification.* As discussed in section 2.3, the RBAC reference model has defined four components: Core RBAC, Hierarchical RBAC and Constrained RBAC (Static Separation of Duty and Dynamic Separation of Duty) (figure 2.7).

The reference model define sets of basic RBAC elements and relations, such as a set of roles, a set of users, a set of permissions, and relationships between users, roles and permissions. The essence of RBAC lies with the notion of *roles* as an intermediary between conventional access control subjects and objects, i.e. permissions (e.g. allow, prohibit, etc.) are assigned to roles (e.g. gold user, silver user, etc.) instead of being assigned to users (e.g. John). It is primarily a non-discretionary access control [107] model where users are not directly associated with permissions.

RBAC models have matured to the point where they are now being prescribed as a generalized approach to access control involving number of heterogeneous users (home users, guest users, mobile users, etc.) working with the system under different permissions. For instance, various RBAC models are now being embedded in commercial-off-the-shelf software-based products and in the government sectors [108]. The idea of organising the reference model in components is to permit vendors to partially implement RBAC features in their products. The existing implementations of RBAC are focused either towards operating systems as a module to provide access control, or towards database management systems [109]. Furthermore, RBAC has been specialized for use with component-based message passing architectures, similar to the proposed management framework in this thesis.

Following the Core RBAC model, we have included sets of five elements in our proposed Policy Schema namely users (USERS), roles (ROLES), objects (OBS), operations (OPS), and permission (PRMS). Furthermore, we have followed the DCCM approach to add a set of operations which can be performed on these elements. The operations (discussed in section 4.2.4.3) are defined as Add, Remove, Freeze, Thaw and Resync.

4.2.4 **Proposed Policy Schema**

We have adopted the RBAC reference model for defining our Policy Schema. At this layer, it is not important that RBAC is mainly used to represent access control policies. The RBAC ‘type’ models offer a structured layout of an organisation or a managed network. They define users and associated roles, permissions to those roles and a set of operations that can be performed by those roles.

The choice of RBAC model is also supported by the characteristic of RBAC models as both abstract and general. It is abstract as the properties which are not currently relevant to a management framework are not included, and it is general as different designs can be valid interpretations of the model. Thus, the model allows design decisions to be postponed, and is usable as a basis for the design of a variety of networked systems. The goal in choosing the model is to provide a concise and usable notation as possible so that the properties of the model are not obscured by excessive notational details.

The following section discusses the proposed schema and the proposed extensions to the RBAC model. The proof-of-concept implementation of Policy Schema is presented in section 8.3.1.

4.2.4.1 **Role Modelling in Managed Networks**

The introduction of roles greatly simplifies the management of networks as it separates the dynamic associations between users and roles from the relatively static association between roles and permissions. The concept of role is defined as “the set of rights and duties associated with a position, which are assigned to the person who occupies that position”. The use of roles rests upon the observation that most organisations’ policy decisions are based upon a role that the individual is acting in, rather than their identity.

RBAC model also enriches the Policy Schema by introducing the concepts of role hierarchies and association inheritance constraints derived from the object-oriented paradigm.

4.2.4.2 **Proposed Extensions to RBAC Model**

In particular, the Policy Schema can be used to perform the following two tasks:

1. Develop an *abstract* organisational schema that verifies the *correctness* of the policies based on it.
2. Specifying individual properties of the entities such as users, roles, objects and their associations by asserting authoritative and prohibitive constraints, and then check the *consistency* between the existing schema and the new entities that have been introduced, and the constraints associated with them.

We have observed that the RBAC standard performs the first task of the Policy Schema, i.e. to define an abstract organisational schema. This observation, and the RBAC approach that originates from it, allows us to take advantage of several benefits in specification of a policy-based system, such as:

- A single policy applies to all members of a role, rather than defining policies for each individual entity – *clustering,*
- When an individual entity leaves or joins a role, there is no necessity to change the policies associated with the role,
- Policies that are common to many large organisations, such as separation of duty, can be implemented conveniently through roles by declaring separate roles with constraints upon individual entities concurrently activating these roles.

However, for the second task, Core RBAC is not sufficient. The NIST RBAC has also proposed a Hierarchical RBAC model (Figure 4-3) by introducing role hierarchies where roles are organised in a hierarchy. The hierarchical structure of roles allow defining *propagation policies*, i.e. a junior role inherits or overrides permissions from senior roles, but not vice versa.

This concept for policy inheritance means that policies specified for one role may be inherited by other roles based on the role-hierarchy. For example, if a gold user is not allowed to start multimedia services, then a silver user cannot start multimedia services either, but if a gold user is allowed to start multimedia services, a silver user may inherit the permission or override it to not allow multimedia services. Hence, this model ensures that every individual policy satisfies the constraints of the parent hierarchy also referred to as *propagation policies*.

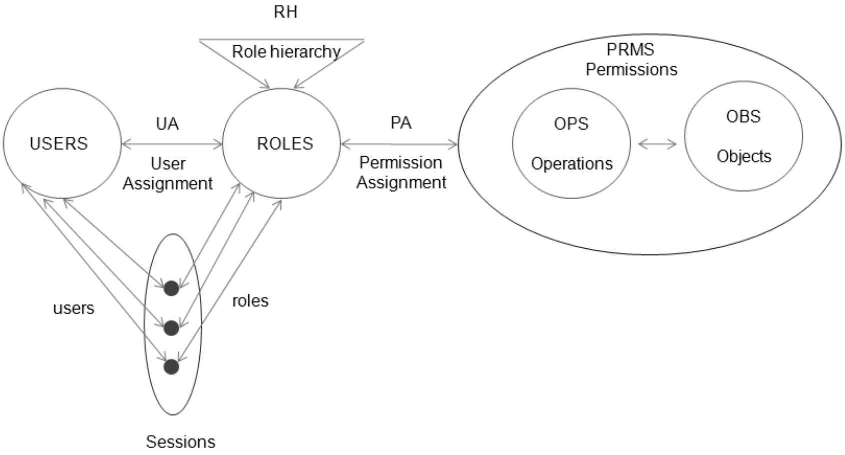


Figure 4-3 Hierarchical RBAC with Static and Dynamic Separation of Duty

Constrained RBAC has further introduced constraints to check the consistency of the overall schema. The Static and Dynamic Separation of Duty (SSD and DSD) introduced constraints to the User Assignment (UA) relationships by excluding the possibility of a user assuming conflicting roles. A maximum cardinality value is associated with each role, where the maximum combination of roles should be lesser or equal to its cardinal value. For example, for constraining a user to assume the roles *biller* and *auditor*, a set {biller, auditor} with cardinality 2 is defined (the user can only assume cardinality-1 roles in the set). In other words, SSD prevents the policy conflict where a user can assume two conflicting roles e.g. a user *John* cannot be assigned roles of both *biller* and an *auditor*.

However, there are two limitations in the Constrained RBAC model:

(i) The SSD constraints are applied only to the activation of roles without considering other components in RBAC model. This constraint mainly reflect that RBAC model offer simplest separation of duty properties and do not consider conflicting Permission Assignment (PA) i.e. possibility of a role assuming conflicting permissions.

(ii) In RBAC terms, permission is an approval to perform an action (e.g. use, start, stop, etc.) on one or more targets (e.g. file, software application, service, etc.).

Permissions are defined as a mapping (Cartesian product) between objects and operations.

PRMS :: OBJECTS × OPERATIONS

However, the RBAC model defines permission for individual objects and individually allowed operations and hence more fine grained constraints cannot be defined adequately. It is also evident that the Chinese-Wall security policy [110], where conflicting operations are restricted on the same object, cannot be implemented in RBAC model.

To eliminate these limitations, we have extended the Constrained RBAC model by introducing *objects hierarchy* and *action hierarchy* (Figure 4-4), i.e. representing OBS in an object hierarchy and representing OPS in an action hierarchy to provide flexibility in defining policy targets and help maintain a hierarchical control of permitted actions on a group of protected objects. For example, a permission to *use* a multimedia service inherently allows users to perform *start* and *stop* operations. Our proposal can express a variety of separation of duty properties and address the Chinese-Wall constraint (discussed in section 4.2.5).

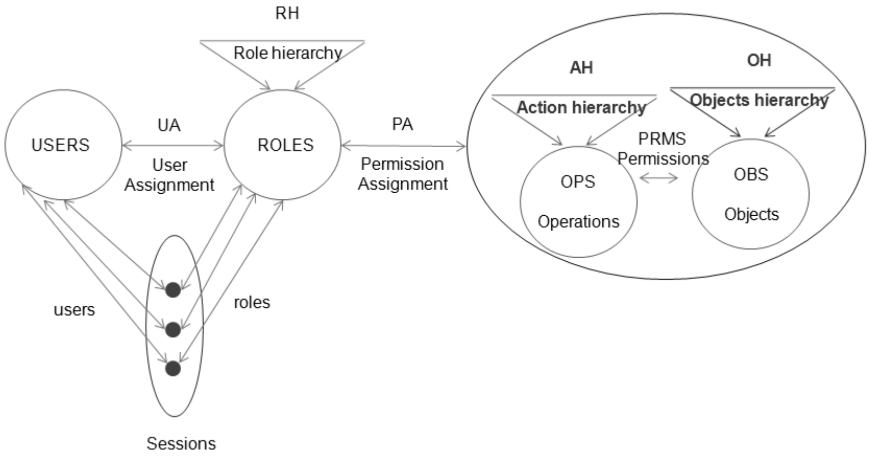


Figure 4-4 Proposed extension to Constrained RBAC model

The proposed extensions provide a conceptually similar concept to the directory structure of a file system in modern operating systems, where policies inherited from a directory are applicable to all the files included within. Regarding the second task of the Policy Schema, i.e. when individual entities such as users, roles, objects and their associations have been defined by asserting constraints, it is also important to check the *consistency* between the existing Policy Schema and the new entities that are introduced. We have used the following *group operations* for an entity in the Policy Schema.

4.2.4.3 **Group Operations**

Following the DCCM approach for group hierarchies, the Policy Schema operations have been defined as:

- **Add**: Roles, Objects and Actions can be added to a schema.
  There are three security relevant aspects associated with such an introduction. Firstly, the administrator of the schema needs to apply an access control policy to determine if the role, object or action can be added. Secondly, the schema policy must specify whether or not new roles can have access to other active roles. Finally, a negotiation policy must specify how to negotiate and ensure that the schema context is complaint with all the policies of the schema. The policies may require a change that is mutually exclusive with an existing object hierarchy.
- **Remove**: Roles, objects and actions can be removed from a domain. A remove is used when a role, object or action is no longer a valid member of that schema or a sub-hierarchy. If a role is to be moved from one sub-hierarchy to another while being active in the same hierarchy level, then it has to be removed from the current sub-hierarchy and added to the new sub-hierarchy. Our proposed model does not support moving of entities within hierarchy levels.
- **Freeze**: Freezing an entity is a special case of removing an entity. When an entity is frozen, it is not removed from the domain, but any new policies added into the schema do not affect the entity.
- **Thaw**: Thawing an entity is a reverse of freeze operation. When an entity is thawed, all the policies relating to that entity are checked again for any possible conflicts.
- **Resync**: Resync operation is used when an external event requests to update the status of an entity and all the related policies need to be checked again.

4.2.5 Analysis of proposed RBAC extensions

In terms of providing ABC services and our management framework's requirement of *clustering*, our proposed objects hierarchy and actions hierarchy offer a more structured format to the Policy Schema. With the proposed extension to the Constrained RBAC model, our Policy Schema now contains: role hierarchy, objects hierarchy and actions hierarchy in comparison to RBAC's support for only role hierarchy (Figure 4-5).

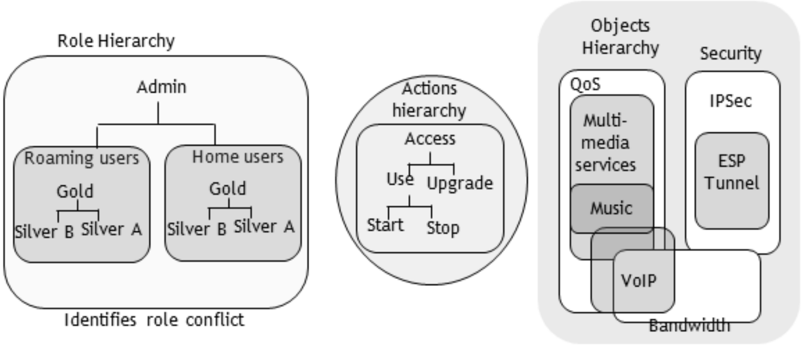


Figure 4-5 Proposed Policy Schema

In addition to providing a more formal structure to the Constrained RBAC model, our proposed hierarchical structure helps identify potential policy conflicts. Conflict detection based *only* on the Policy Schema is described below.

4.2.5.1 Conflict Detection by proposed RBAC extensions

Policy conflict occurs when the objectives of two or more policies cannot be simultaneously met. For example, a *GoldVoIPWeekdayAccess* policy can be defined as "*Gold user is allowed to access VoIP services between 9am and 5pm*" would be in conflict with a *GoldVoIPWeekendAccess* policy that defines that "*Gold user is not granted access to the VoIP services on weekends*". Since, if both these policies are enforced, the Gold user is both granted (authorised) and refused (prohibited) permission between 9am and 5pm at weekends.

Conflicting policies:

GoldVoIPWeekdayAccess: Gold user is allowed to access VoIP services between 9am and 5pm

GoldVoIPWeekendAccess: Gold user is not granted access to the VoIP services on weekends

To detect inherent conflicts especially for large and distributed environment is a complex task and requires specialised approaches and methodologies. A detailed discussion on conflict detection is presented in section 5.4. In this section, we present two constraints which help in detecting and preventing policy conflicts when our proposed extended RBAC model is employed.

a. **Separation of Duty constraint**

The Separation of Duty constraint ensures that a user should not be assigned to two roles which are in conflict with each other. It defines a mutually exclusive relation between two conflicting roles assumed by a single user. An example is to separate the roles of *auditor* and *biller* or *controller* and *buyer*, which cannot be assigned to a single user simultaneously.

While defining the Policy Schema based on role hierarchy, a Separation of Duty constraint can be achieved by controlling membership in, activation of, and use of roles as well as permission assignment. The Separation of Duty constraint is supported by Hierarchical and Constrained RBAC models. Our proposed Policy Schema has extended the Constrained RBAC model and hence inherently supports the Separation of Duty constraint.

b. **Chinese Wall Security Model**

The introduced hierarchical structure of objects and operations help in developing a Chinese-Wall security model [110] where conflicting operations are restricted on the same object. Specifically, if a role has permission for an action on an object, and if the action is performed once, then the same role cannot perform the same action on other conflicting objects.

For example,

ChineseWallConstraint: If an auditor A has audited company X's account, then a restriction is needed to be imposed for same auditor to audit company Y's accounts, if companies X and Y are in competition.

Following the *ChineseWallConstraint* policy, an objects hierarchy "Competition" and actions hierarchy "Audit" can be defined.

a. Role Hierarchy
Auditor :: {Team Leader}
Team Leader :: {Member A, Member B}
b. Objects Hierarchy
Competition :: {Company X, Company Y, Company Z}
c. Action Hierarchy
Audit :: {Access, Edit}
Access :: {Read, Print}

As illustrated in Figure 4-6, if "Team Leader" has "Audited" "Company X's" accounts, then following *propagation* policies, every role in the hierarchy i.e. "Member A" and "Member B" do not have "Audit" permission for the objects hierarchy "Competition". Hence, when a "Member A" (in *role hierarchy* of "Auditors") wants to "Access" (in *actions hierarchy* of "Audit") - is not permitted for "Competition" *objects hierarchy.*

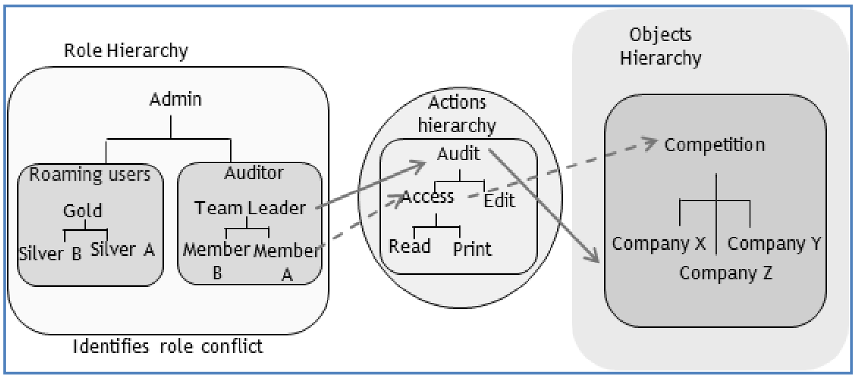


Figure 4-6 Implementing Chinese-Wall Security Policy through Proposed RBAC extensions

4.3 **Information Model**

In the previous section, we have defined a Policy Schema based on the extended Constrained RBAC model to represent the structure of an organisation. Every Policy Schema requires a formal method to describe this representation. An Information Model provides such representation. It provides a common language in which different types of management entities can be represented. In this section, we will discuss the Information Model proposed by the IETF and the extensions required to represent our proposed Policy Schema.

We have used an object-oriented information model to represent various entities in a managed environment. An entity can be a person, a computer, a router, or even a protocol message – that needs a uniform and consistent representation for configuration and management. The information model of the policy domain is required to be able to describe the role of the components and the relations between specific entities. The standardisation of these models is required to enable the consistent exchange of information between systems provided by different vendors.

4.3.1 **IETF Information Model**

The IETF have proposed standards for the information model of the policy system called Policy Core Information Model (PCIM) [31]. PCIM is an object-oriented model for representing policy information. The model is generic and can be used to specify a wide range of policies and different ways to implement the information model.

4.3.1.1 **PCIM**

PCIM is based on CIM version 2.5 of DMTF. The IETF has chosen a rule-based policy representation in its specification. This model (Figure 4-7) is an abstraction and representation of the entities in a managed environment and is independent of any specific repository, application, protocol, or platform. The policy classes and associations defined in this model are sufficiently generic to allow them to represent a wide range of policies related to different areas (e.g. QoS, IPSec, etc).

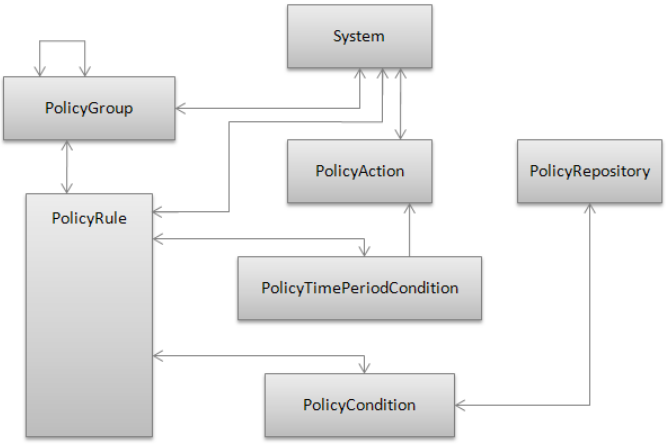


Figure 4-7 Overview of Core Policy Classes and Relationships

The PCIM model consists of two hierarchies of object classes – one represents policy information and the control of the policies, while the other represents the associations that relate policies to one another.

This approach of policy specification treats the policy-based system as a state machine in which policies determine the manner in which state transitions occur. Following this model, policies are defined as rules that relate to a set of conditions to a set of actions whilst also changing the state of the system. Figure 4-8 shows a simplified class model that represents the structure of a policy rule in the PCIM specification.

In the PCIM approach, a policy is defined as a set of policy rules (*PolicyRule* class). Each policy rule consists of a set of conditions (*PolicyCondition* class) and a set of actions (*PolicyAction* class) and are of the form:

```
if (set of conditions) then (perform set of actions)
```

If the set of policy conditions described by the class *PolicyCondition* evaluates to be true, then a set of actions described by the class *PolicyAction* must be executed. Because the PCIM specification is intended to be generic, the *PolicyAction* and *PolicyCondition* classes are defined to be abstract. A vendor implementing a policy-based system using this model must define their own, vendor specific versions of actions and conditions that can be used by the policy rules.

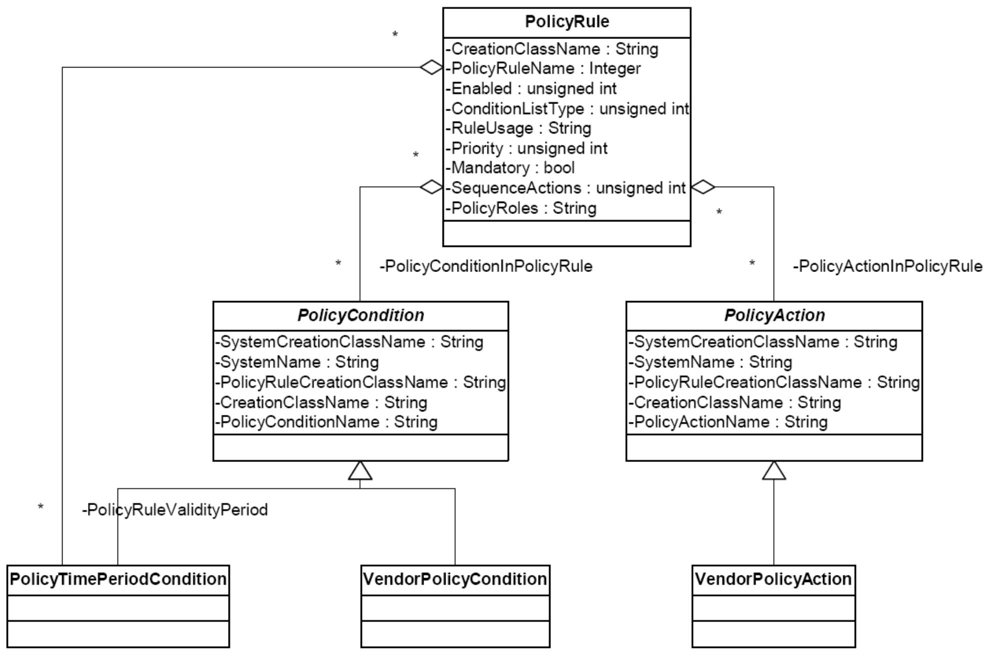

PolicyRule
-CreationClassName : String
-PolicyRuleName : Integer
-Enabled : unsigned int
-ConditionListType : unsigned int
-RuleUsage : String
-Priority : unsigned int
-Mandatory : bool
-SequenceActions : unsigned int
-PolicyRoles : String
*
*
*
*
*
-PolicyConditionInPolicyRule
-PolicyActionInPolicyRule
PolicyCondition
-SystemCreationClassName : String
-SystemName : String
-PolicyRuleCreationClassName : String
-CreationClassName : String
-PolicyConditionName : String
PolicyAction
-SystemCreationClassName : String
-SystemName : String
-PolicyRuleCreationClassName : String
-CreationClassName : String
-PolicyActionName : String
*
-PolicyRuleValidityPeriod
PolicyTimePeriodCondition
VendorPolicyCondition
VendorPolicyAction

Figure 4-8 PCIM Specification

The *PolicyAction*, *PolicyCondition* and *PolicyRule* classes are examples of classes from the structural model of PCIM. As illustrated, the association classes relate structural classes to one another. Examples of these association classes include *PolicyActionInPolicyRule* and *PolicyConditionInPolicyRule*.

A policy rule may also be associated with one or more policy time periods (*PolicyTimePeriodCondition* class), indicating the schedule according to which the policy rule is active or inactive. However, since time-based conditions are assumed to be an essential component of any policy-based system, the *PolicyTimePeriodCondition* class is defined as part of the core specification.

Policy rules may be aggregated into policy groups (*PolicyGroup* class) and these groups may be nested to represent a hierarchy of policies. In a *PolicyRule*, rule conditions can be grouped by two different ways: DNF (Disjunctive Normal Form) or CNF (Conjunctive Normal Form). In DNF, conditions within the same group are ANDed and groups are ORed. In CNF, conditions within the same group are ORed and groups are ANDed. The way of grouping policy conditions is defined by the attribute *ConditionListType* in the *PolicyRule* class. Additionally, the attributes

GroupNumber and *ConditionNegated* in the association class *PolicyConditionInPolicyRule* help to create condition expressions.

In order to illustrate this approach, suppose there are five conditions C_i (GroupNumber, ConditionNegated) as follows: C_1(1, false), C_2(1, true), C_3(1, false), C_4(2, true) and C_5(2, false). Then, the overall condition for the *PolicyRule* will be defined as:

if ConditionListType = DNF, then condition =
(C_1 AND (NOT C_2) AND C_3) OR (C_4 AND C_5)

if ConditionListType = CNF, then condition =
(C_1 OR (NOT C_2) OR C_3) AND (C_4 OR C_5)

Although PCIM specification provides a *PolicyKeywords* property that is enumerated to allow policies to be classified into security policies, management policies, error and event policies, it is necessary to specify specific condition classes necessary to support these different types.

There are several commercial products that implement parts of the PCIM specification to provide QoS management. Cisco QoS Policy Manager [56] and Allot Communications NetEnforcer [111] are examples of such products. These tools do not support a policy specification language but instead provide a graphical interface that allows an administrator to locate specific policies and configure them in an appropriate manner.

In summary, the PCIM specification provides an abstract model for defining the structure of policies and relationships between policy objects. It can be combined with the CIM to provide a complete specification of a policy managed system. The ability to nest policy rules and form sub-rules is important for manageability and scalability, as it enables complex policy rules to be constructed from multiple simpler policy rules.

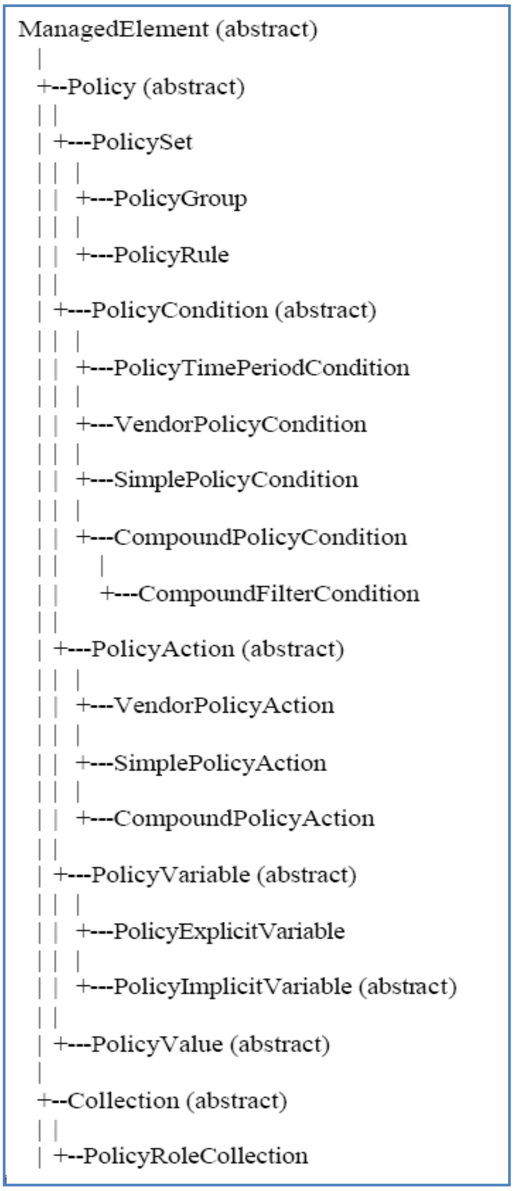
4.3.1.2 **PCIMe**

RFC 3460 [32] proposed several modifications in the original PCIM standard and are referred to as Policy Core Information Model Extensions (PCIMe). PCIMe solve many practical issues raised after the original PCIM publication. For example, *PolicyCondition* has been extended in order to support a straightforward way for representing conditions by combining variables and values. This extension is called *SimplePolicyCondition.* Another modification is the introduction of the class *PolicySet* as a new option for the nesting of policy rules and new classes for supporting an ordered sequence of actions. The modifications introduced by PCIM have backward compatibility with the implementations that follows the original PCIM standard.

The roles of PCIM/PCIMe's major classes are defined as (Figure 4-9):

- *PolicyGroup*: the container class for the set of associated *PolicyRules* or the set of associated *PolicyGroups.*
- *PolicyRule*: the class to represent <if Condition then Action> semantics.
- *PolicyCondition*: the class to represent the policy condition in Policy Rule.
- *PolicyAction*: the class to represent the policy action in Policy Rule.
- *PolicyTimePeriodCondition*: the class to serve the function that activates or inactivates policy rules according to time schedules.
- *PolicyRepository*: the container class to manage information associated with the policy.

Furthermore, the introduced *PolicySet* provides an abstraction for a set of rules. It is derived from *Policy*, and inserted into the inheritance hierarchy above both *PolicyGroup* and *PolicyRule*. This reflects the additional structural flexibility and semantic capability of both subclasses.



```
ManagedElement (abstract)
  |
  +--Policy (abstract)
  | |
  | +---PolicySet
  | |  |
  | |  +---PolicyGroup
  | |  |
  | |  +---PolicyRule
  | |
  | +---PolicyCondition (abstract)
  | |  |
  | |  +---PolicyTimePeriodCondition
  | |  |
  | |  +---VendorPolicyCondition
  | |  |
  | |  +---SimplePolicyCondition
  | |  |
  | |  +---CompoundPolicyCondition
  | |      |
  | |      +---CompoundFilterCondition
  | |
  | +---PolicyAction (abstract)
  | |  |
  | |  +---VendorPolicyAction
  | |  |
  | |  +---SimplePolicyAction
  | |  |
  | |  +---CompoundPolicyAction
  | |
  | +---PolicyVariable (abstract)
  | |  |
  | |  +---PolicyExplicitVariable
  | |  |
  | |  +---PolicyImplicitVariable (abstract)
  | |
  | +---PolicyValue (abstract)
  |
  +--Collection (abstract)
  | |
  | +--PolicyRoleCollection
```

Figure 4-9 PCIMe Specification

The strategy defined by *SimplePolicyCondition* is to build a condition as a Boolean expression evaluated as:

```
does <variable> MATCH <value>
```

Variables are created as instances of specialisations of *PolicyVariable* and values are defined by instances of specialisations of *PolicyValue.* The MATCH element is implicit in the model. PCIMe defines two types of variables: explicit (*PolicyExplicitVariable*) and implicit (*PolicyImplictVariable*).

Considering *GoldSecurity* policy: *If a gold user requests a VPN tunnel during weekdays, provide ESP 3DES encryption*; and user *John* is assigned as *Gold user*. The MATCH element is referred as:

> *User.Username* MATCH "John"

Explicit variables are used to build conditions that refer to objects stored in CIM repository. For example, for the following condition:
User.Username refers to the *Username* attribute of the class *User* in the CIM model. This condition is expressed as

> *PolicyExplicitVariable.ModelClass* = "User"
> *PolicyExplicitVaribale.Property* = "Username"

Because *User.Username* is a string, the *PolicyStringValue* subclass must be used in this condition, i.e. *PolicyStringValue.StringList = "John"*. The explicit variables allow reusing information stored in the repository for PBM tools. Implicit variables are used to represent objects that are not stored in the repository. They are especially useful for defining rules with conditions based on a predefined attribute. For example, service types or protocol headers.

The *PolicyImplicitVariable* then can be defined based on an organisation's specific requirements, such as *PolicyServiceTypeVariable or PolicyProtocolHeaderVariable*. These specialisations have no properties. For example,

> "service type" MATCH "Gold service" would be represented using the class
> *PolicyServiceTypeVariable* and *PolicyServiceTypeValue.StringList* = "Gold"

4.3.2 **RBPIM**

The Role-based Policy Information Model (RBPIM) [112] is a PCIM extension for supporting RBAC policies. RBPIM adopts the RBAC model, but some extensions have been introduced in order to provide a more flexible method for mapping users to roles and describing permissions and also for establishing network topology-based and time-based permission constraints. Figure 4-10 shows the class hierarchy. The gray classes were introduced by the RBPIM model.

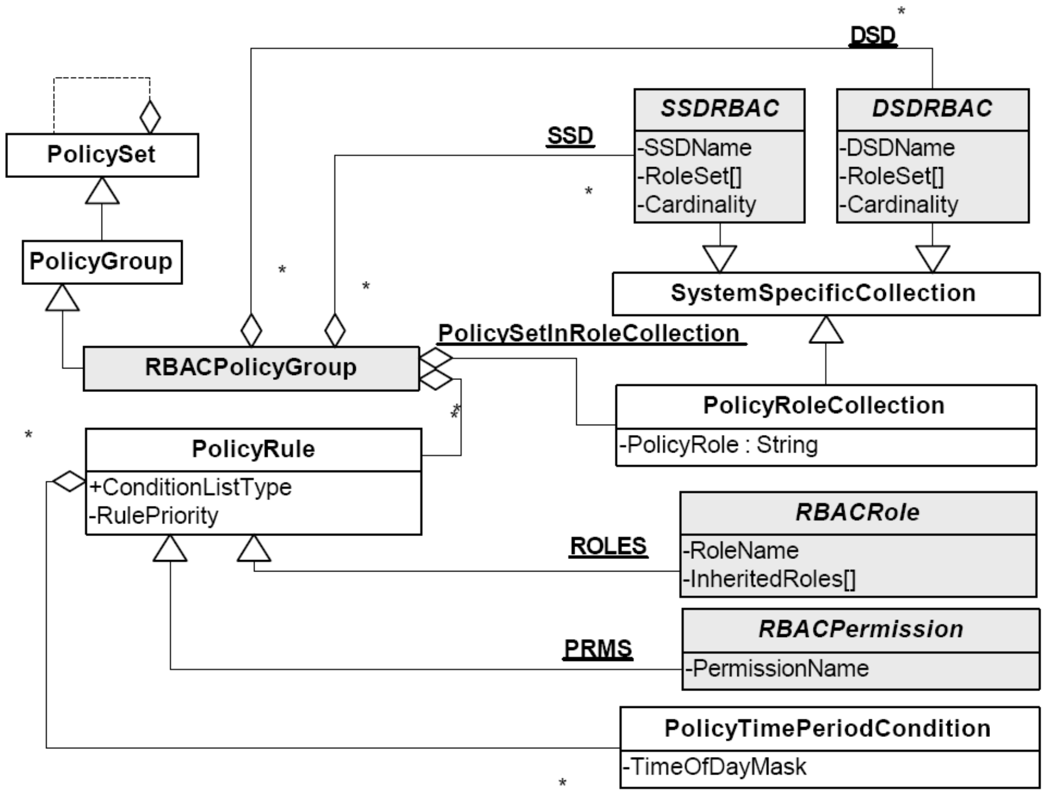


Figure 4-10 Role based Policy Information Model (RBPIM)

The following classes have been introduced: *RBACPermission* and *RBACRole* (specialisations of *PolicyRule*), *AssignerPermission* and *AssignerOperation* (specialisations of *PolicyAction*), *and DSDRBAC* and *SSDRBAC* (specialisations of *Policy*). The *RBACPolicyGroup* class (specialisations of *PolicyGroup*) is used to group the information of the constrained RBAC model.

4.3.3 **Proposed Extended RBAC PIM**

To define our policy model, we have followed PCIM/PCIMe specifications for the following reasons:

a. policies can be tagged with different roles and profiles
b. policies can be prioritised and structured into hierarchal policy groups, useful for conflict resolution, and
c. policies can be translated to a standard format such as XML/LDAP to achieve interoperability in policy exchange.

In the Policy Model Layer, policies are expressed and transmitted in XML format. The *PolicyRule* extensions, named *RBACRole* defines an element for a set of ROLES and *RBACPermisson* defines an element for a set PRMS.

```
RBACRole ∈ ROLES
RBACPermission ∈ PRMS
```

The *RBACRole* can be associated to the lists of *SimplePolicyCondition*, *AssignerRBACPermission* and *PolicyTimePeriodCondition* instances (Figure 4-11).

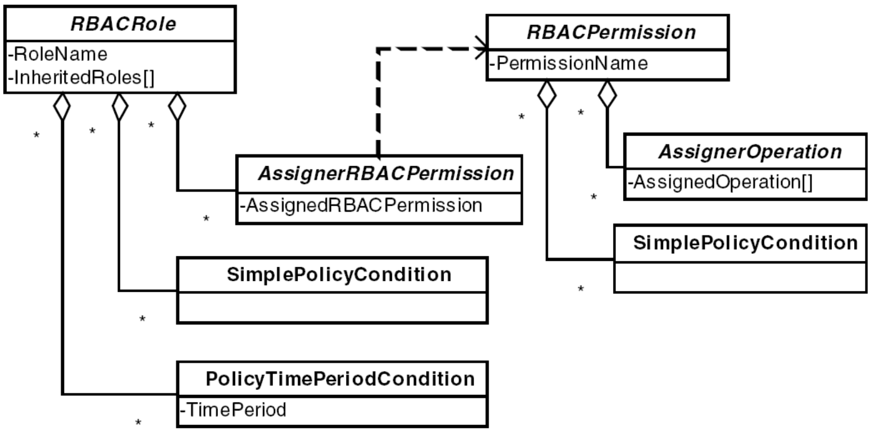


Figure 4-11
RBACRole and RBACPermission

- The instances of *SimplePolicyCondition* are used to express the conditions for a user to be assigned to a role (UA relationship),
- The instances of *AssignerRBACPermission* are used to express the permissions associated to a role (PA relationship),
- The instances of *PolicyTimePeriodCondition* define the periods of time a user can activate a role,
- The instances of *SimplePolicyCondition* are used to describe the RBAC objects, and
- The instances of *AssignerOperation* are used to describe approved operations on these objects. The *RBACPermission* can be associated to a list of *SimplePolicyCondition* and *AssignerOperation* classes

Figure 4-12 shows the PCIM model, RBPIM extensions [112] and our proposed extensions for supporting RBAC policies. The classes marked (*) were introduced in

RBPIM and classes marked (**) are introduced for our proposed extended RBAC PIM.

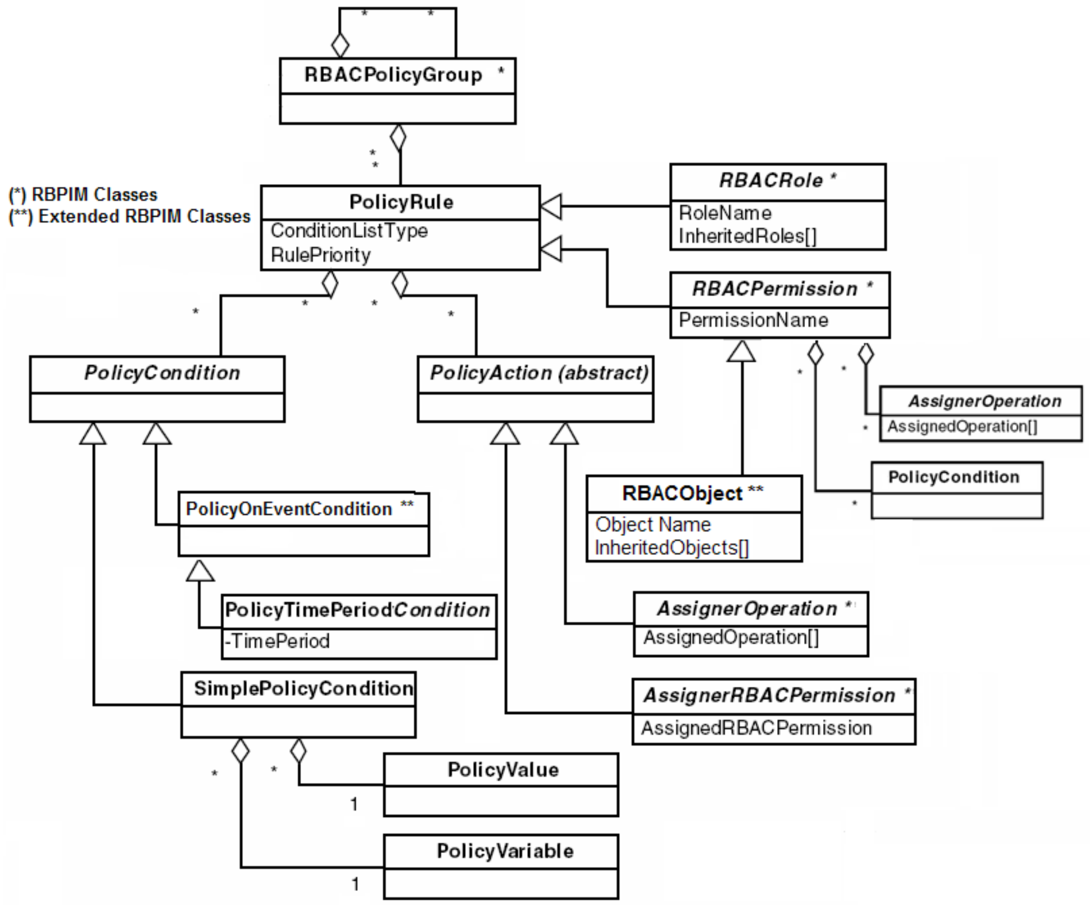


Figure 4-12 Proposed Extended RBPIM

We have introduced a *PolicyOnEventCondition* class and inherited a *PolicyTimePeriodCondition* class. This allows a flexible declaration of specific *events* in which a policy may be (de)activated. Hence, Time-based policy validity becomes an instance of event-based policy activation class. The *RBACObject* class is also introduced to provide a Chinese-Wall security model to restrict conflicting operations on same objects. Note that Chinese-Wall constraints are imposed to the *RBACPolicyGroup* and they could not be represented as rule conditions.

Hence, the *GoldSecurity* policy, will be structured in extended RBPIM as shown in Figure 4-13 (for simplicity, some of the classes are not included in the figure). The attribute *InheritedRoles* is used for expressing role hierarchy in the Constrained RBAC model, and attribute *InheritedObjects* is used for expressing proposed objects hierarchy.

RBACRole:	Gold
InheritedRoles:	RoamingUsers
RBACPermission:	Allow_GoldSecurity
RBACObject:	VPN
InheritedObjects:	SecurityComp.
AssignerOperation:	Create

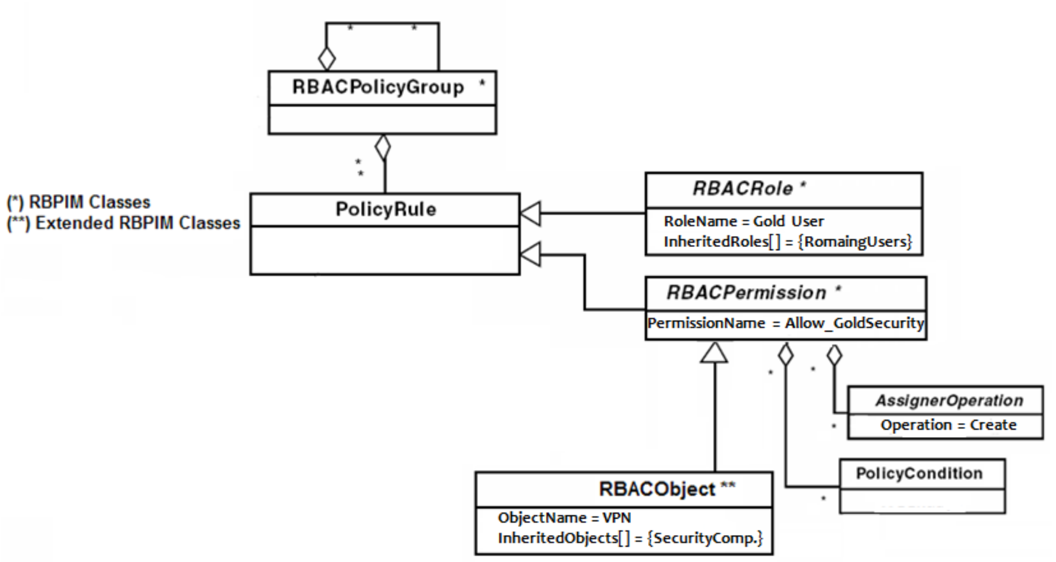


Figure 4-13 Policy representation in Extended RBPIM

In summary, following the PCIM approach, a policy is defined as a set of policy rules (*PolicyRule* class). Each policy rule consists of a set of conditions (*PolicyCondition* class) and a set of actions (*PolicyAction* class). If the set of conditions described by the class *PolicyCondition* evaluates to true, then a set of actions described by the class *PolicyAction* must be executed. Using the *PolicyRule* semantics defined by PCIM, a *RBACRole* instance can still express the following rules: "*if conditions are satisfied then assign the RBACRole permission(s) to the user(s)*". The users (elements of USERS) are represented by a *CompoundPolicyCondition* extension called *UACompoundPolicyCondition.*

The use of the *CompoundPolicyCondition* semantics simplifies the process of assigning a role to a user (UA) because the assignment can be implemented with predefined Policy Schema about the users and organisations. For example, when user *John* logs in as a *Gold* user and requests a *VPN tunnel* during *weekdays*, then the *GoldSecurity* policy (permission) should allow the request (Figure 4-14). In this case, *GoldSecurity* policy will be retrieved from the Policy Storage and implemented by the Syntax Layer (discussed in chapter 6). The Policy Storage scheme is dependent on the choice of the organisation.

RBACRole:	Gold
InheritedRoles:	RoamingUsers
RBACPermission:	Allow_GoldSecurity
RBACObject:	VPN
InheritedObjects:	SecurityComp.
AssignerOperation:	Create
PolicyVariable:	UserName
PolicyValue:	John
TimePeriod:	WeekDays

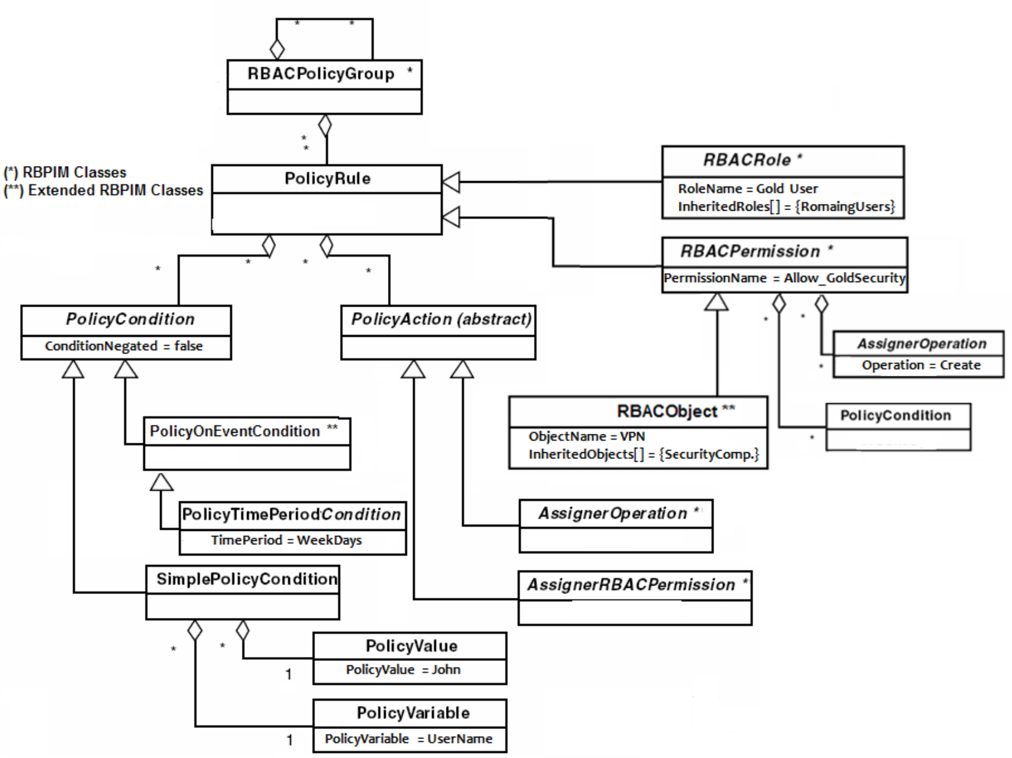


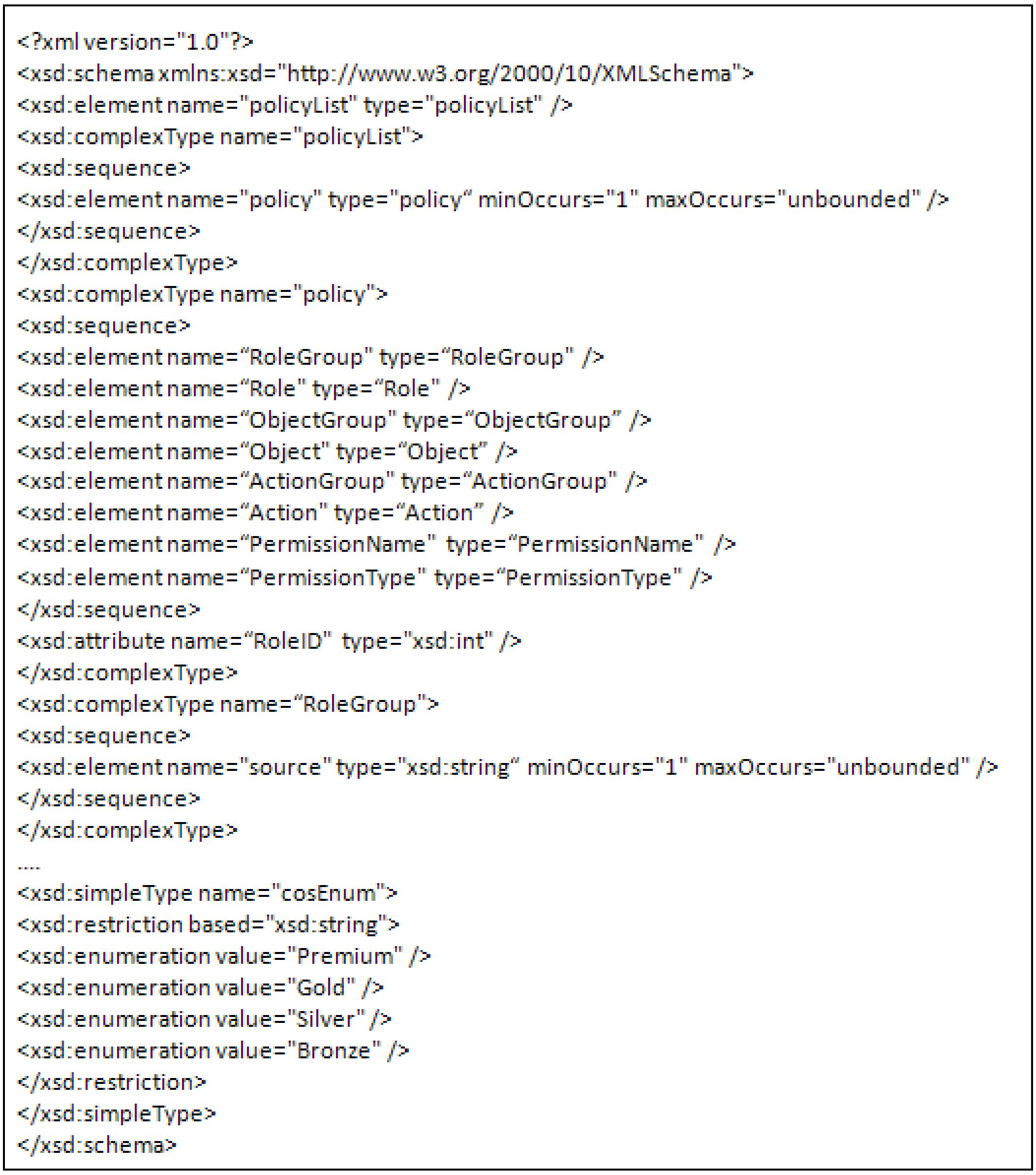
Figure 4-14 Policy implementation from Extended RBPIM

4.4 Policy Storage

An advantage of information modelling for representing high-level policies is that the policies can be easily mapped to structured specifications such as XML or LDAP format which can then be used for policy analysis as well as distribution of policies across networks.

The mapping of CIM standards to XML is already undertaken within the DMTF Web-Based Enterprise Management (WBEM), a set of management and Internet standards technologies [96]. IETF have also defined a mapping of the PCIM to a form that can be implemented in a directory that uses LDAP as its access protocol [113]. In other approaches, such as in the Ponder toolkit domains have been implemented as directories in an extended LDAP Service [114], while Clemente et. al. [115] has proposed the definition of an XML PIB supporting XML-encoding.

In our proposed work, we have used XML Schema definition language proposed by the World Wide Web Consortium (W3C) XML Schema Working Group to represent structured policies [116]. The XML Schema (Listing 4-1) is used to express a schema: a set of rules to which an XML document must conform in order to be considered "valid" according to that schema. In our case, every defined policy should conform to the Policy Schema. An XML Schema instance is an XML Schema Definition (XSD). The choice of either technology (i.e. LDAP/XML) is applicable for the proposed Policy Model layer and supports the *modular* characteristic of the proposed management framework.



```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2000/10/XMLSchema">
<xsd:element name="policyList" type="policyList" />
<xsd:complexType name="policyList">
<xsd:sequence>
<xsd:element name="policy" type="policy“ minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="policy">
<xsd:sequence>
<xsd:element name=“RoleGroup" type=“RoleGroup" />
<xsd:element name=“Role" type=“Role" />
<xsd:element name=“ObjectGroup" type=“ObjectGroup” />
<xsd:element name=“Object" type=“Object” />
<xsd:element name=“ActionGroup" type=“ActionGroup" />
<xsd:element name=“Action" type=“Action” />
<xsd:element name=“PermissionName" type=“PermissionName" />
<xsd:element name=“PermissionType" type=“PermissionType" />
</xsd:sequence>
<xsd:attribute name=“RoleID" type="xsd:int" />
</xsd:complexType>
<xsd:complexType name=“RoleGroup">
<xsd:sequence>
<xsd:element name="source" type="xsd:string“ minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
....
<xsd:simpleType name="cosEnum">
<xsd:restriction based="xsd:string">
<xsd:enumeration value="Premium" />
<xsd:enumeration value="Gold" />
<xsd:enumeration value="Silver" />
<xsd:enumeration value="Bronze" />
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

Listing 4-1 XML Schema

4.5 Analysis of Policy Model layer

The proposed Policy Model layer provides a unified model for representing policies, users and resources, and provides a generic and conceptual framework to represent the overall structure of the organisation. It also realises the *clustering* characteristic of our proposed management framework and helps administrators by defining a formal structural definition of the managed network.

We have proposed an extended Constrained RBAC model to define a Policy Schema and extended the IETF proposed PCIM Information Model to represent the schema. The policies are then translated in generic XML Schema to be stored in the repository.

The proposed *Policy Schema* represents the schema of a managed network in our proposed policy-managed framework. The Policy Schema represented in an Information Model verifies the overall Schema, which then can be stored in Policy Repository in either XML or LDAP formats. The proposed layer is generic and organisations can choose any method to achieve these objectives.

The use of the RBAC model has come naturally to our proposed research. To provide ABC services, the mobile users need to be authenticated and then authorised to perform certain actions in the network and consume certain services. Hence, the need for access control is one of the most important aspects in our management scenario.

The need for access control is also present in several components of a distributed system. In some cases, access control refers to the right of managing network devices, such as gateways and firewalls. In other cases, the access control policies restrict the access of users to resources and define application level services. RBAC provides a generic information model that can be used for representing several types of access control policies. The proposed Extended RBAC model offers a range of combinations where roles, objects and actions are categorised into different hierarchies. Additionally, the proposed extended RBAC model allows defining rules for constraining inherent conflicts in the policies by applying constraints such as static Separation of duty, Dynamic Separation of Duty and Chinese-Wall security policy.

Chapter 5
Semantic Layer

In the previous chapters, we have discussed that policy-based methodology is one of the most effective approaches for managing networks as they facilitate defining high-level business policies without the need for a detailed specification of the environment where the policies are applied. Since policies drive the behaviour of each system component, the presence of any inconsistencies may lead the system to unknown states or errors where either authorised users cannot access their permitted services or unauthorised users gain access to prohibited services. As the overall structure of the access control model and policy description languages become more complex, it is increasingly difficult for a system administrator to verify that the policies are valid and meet business and application requirements.

The RBAC model has been used as a means of grouping policies and using inheritance to simplify this specification. However, NIST RBAC has limitations in that it supports only a role-based hierarchical structure for subjects. In general, an access control policy is defined in terms of three elements - subject, object, and action. In section 4.2.4, we have introduced extensions to the NIST RBAC model where hierarchies of *Actions* and *Objects* were supported. We have also defined an extended RBAC Policy Information Model (RBPIM) to formalise the proposed extensions.

This chapter introduces the concept of a Semantic layer for the proposed PBM framework. This Semantic layer, loosely based on the proposed Policy Schema, introduces a formal discipline, where policies are understood and analysed for any inherent conflicts. There are several approaches to policy conflict detection in the literature. However, following the modular approach of our proposed layered model, organisations and network administrators may choose the *best suited* methodology for such analysis. We have also introduced a new conflict detection approach based on first-order logic and composite mapping (Appendix B).

5.1 Introduction

The focal point in a PBM system is the notion of *policy*. In order for policies to be active in a network, an administrator needs to define them, and devices need to enforce them. For example, for a computer network offering QoS to its users, a policy is a statement defining the rules related to handling different types of traffic within the network. An example of such a policy can be defined as: “All development engineers are permitted to access the development servers which support applications they are working on as a team member and no one else”, while another policy might state that “Email traffic is only allowed from outside the company’s servers and only from a special mail gateway”.

Since policies drive the behaviour of each system component, the presence of any inherent conflict(s) may lead the system to unknown states or errors and hence conflict detection is a critical problem in any PBM system. The IETF Network Working Group has identified policy conflict detection as a *must* requirement for any PBM system and requires the implementation of a suitable conflict detection method [25]. We have also identified conflict detection as a characteristic requirement in the proposed PBM system (discussed in section 3.3). Since conflicts cannot be prevented in policy systems, they need to be *detected* and *resolved* to ensure that the system behaves deterministically and correctly. However, the policy definitions related to configuration of access control devices, network security and quality management, are often device/vendor dependent, hence making the task of creating conflict-free policies very tedious and error prone.

The proposed Semantic layer introduces a formal discipline, where policies are understood and analysed. The policies may be represented as a special grouping of rules, with a particular dependency on a conceptual managed entity. We have defined the role of the Semantic layer as follows: *define a semantic notation to provide a common ontology/formal methodology to understand the contents of the policies and to detect and resolve conflicts*. The semantically analysed policies do not need to be in an implementable state, though they are required to be in a consistent state and be unambiguous. Two policies are defined to be unambiguous if their actions do not contradict each other when both of their conditions are met simultaneously [31].

5.2 Policy Definition

In the IETF approach, policy is defined as a condition-action tuple [31].

$$policy := \ (condition, action)$$

The *policy condition* is defined as a representation of the necessary state and/or prerequisites that define whether a policy's actions should be performed. This representation need not be completely specified, but may be implicitly provided in an implementation or a protocol [25]. In RFC 3060 [31], the policy condition is defined to be expressed as either an ORed set of ANDed sets of statements (disjunctive normal form, DNF), or an ANDed set of ORed sets of statements (conjunctive normal form, CNF).

$$\begin{aligned} DNF\ condition &:= \ (A\ or\ B)\ and\ (C\ or\ D) \\ &:= \ (A\ \cup B)\ \cap (C\ \cup D)\ set\ theory \\ &:= (A \vee B) \wedge (C \vee D)\ Boolean\ login \end{aligned}$$

$$\begin{aligned} CNF\ condition &:= \ (A\ and\ B)\ or\ (C\ and\ D) \\ &:= \ (A\ \cap B)\ \cup (C\ \cap D)\ set\ theory \\ &:= (A \wedge B) \vee (C \wedge D)\ Boolean\ login \end{aligned}$$

The *policy action* defines what is to be done to enforce the rule specified in the policy when the conditions are met. Policy actions may result in the execution of one or more operations and the rule's actions may be ordered.

The TeleManagement Forum (TMF) definition of a policy differs from the IETF proposition. The TMF defines that a policy is a triple event-condition-action tuple [117].

$$policy := \ (event, condition, action)$$

It can be interpreted as: if an *event* took place then evaluate *policy condition(s)* and if condition(s) are satisfied then perform a *policy action*. This approach improves the performance of PBM systems in the case of a large number of defined policies, which can be grouped on the basis of events to which they are assigned [118].

In the area of academia research, Sloman [119] defined policies as "*rules governing the choices in behaviour of a system"*. On the basis of this definition, Damianou [19] proposed a more descriptive form as "*policy is a persistent declarative specification, derived from management goals, of a rule defining choices in the behaviour of a system"*. These definitions identify different properties of a policy:

- Persistent in the sense that a one-off command to perform an action is not a policy. In addition, policies are relatively static compared to the states of the managed system.
- Declarative means that policies define choices in behaviour in terms of the conditions under which predefined *policy actions* can be invoked,
- Derived from management goals, as policies are viewed as being derived from business logic, service level agreements or trust relationships. Management goals are sometimes referred to as high-level or abstract policies.

On the other hand, Strassner has provided a formal definition of a PBM system as "*the usage of policy rules to manage the configuration of one or more entities"* [120]. Strassner elaborated that, PBM is a methodology for managing systems. It does this by modelling different entities in the environment to be managed as a set of entities, and describe different applications that manage one or more systems according to a set of rules. These rules take the form of policies that are applied to the components of the system to better and more efficiently manage those components.

Although different approaches have varying opinions about the definition of policies, we understand that an important purpose of a policy is to simplify the task of administration and management for different disciplines. Additionally, while analysing different management solutions and the types of the policies, it is apparent that policies are varied based on their application areas. For example, *network security policies* generally specify network related information such as IP addresses of systems to be protected by firewalls, whereas *access control policies* describe the precise format of the allowed password structures, and *QoS policies* define the required bandwidth for each service offered by the network service provider. Our analysis is in line with Strassner’s definition that different types of policies are required to govern different managed objects.

Since, different types of policies are required for different management areas, different solutions and/or methods can be employed for conflict detection (and resolution) even within a single PBM system. By following the modular approach of our proposed PBM model, organisations and/or network administrators can make use of any such approaches, provided the characteristics of the system being managed are represented in as much detail as required, and thereafter, conflict-free policies can be defined that govern each state of the managed object.

Another requirement of our modular approach is that the Semantic layer should be able to *read* policies from the Policy Schema and *write* back, without changing the structure (i.e. entities involved and their relationship) of the policies. The sole purpose of the Semantic layer is to detect conflicts and provide conflict resolution methods which may or may not require changes to the Policy Schema.

In general, irrespective of any specific conflict detection approach, the Semantic layer should follow a policy language/ontology to represent the policies such that conflicts within policies are feasibly detectable. Such a policy language/ontology should fulfil the following requirements:

- Expressiveness – to handle a wide range of policy requirements arising in the system(s) being managed;
- Simplicity – to ease the policy definition tasks for administrators with different degrees of expertise;
- Enforceability – to ensure a mapping of policy specification into implementable policies for various platforms;
- Scalability – to ensure performance when conflict detection is performed in a mesh of systems, or when a new subsystem with local policies is added and a conflict detection is required within;
- Analysability – to allow a logic supporting the reasoning of policies.

In the following sections, we will identify different type of policies, and the types of conflicts which may occur to them.

5.3 Type of Policies

There are mainly three modes of policies identified [119] as defined in a standard deontic logic [121] and are widely accepted in the literature [59, 122]. These policy modes can be described as follows [89]:

(i) *Permission* is "the action of permitting or giving leave; allowance; liberty of a licence granted to do something".

(ii) *Prohibition* is "an unambiguous statement or rule, regulating *forbidden* behaviour in a system".

(iii) *Obligation* is "an agreement, enforceable by law, whereby a person or persons become bound to a particular action or performance of some duty by a contract containing such an agreement".

Based on these modes, the following types of policies have been generally used [123].

i. ***Authorisation Policies (Auth+):*** are used to define what services or resources an entity can access, and permits the actions that a subject (user) can perform on the objects in the target domain. It can be described as an access control policy to protect the resources and services from unauthorised access, and are implemented on the target host by an access control component.

ii. ***Prohibition Policies (Auth-):*** These policies define the actions that subjects must not perform on target objects even though they may be actually permitted to perform the particular action. For instance, in general an engineer is allowed to open a company's human resources policy files but not files of the accounts department.

iii. ***Obligation Policies (Obli+):*** Obligation policies specify actions that must be performed within a system when certain events occur and provides an ability to respond to the changing circumstances. They are event-triggered, condition-based rules that can be used to define actions. Hence, these policies define conditions for performing a wide range of management actions, such as, “change QoS when any mobile user demands” or “create a log when a new user is registered in the network”; and define the actions that subjects must perform on the target domain.

iv. ***Propagation Policies (Prop [Up/Down]):*** The hierarchies, i.e. role hierarchy introduced in RBAC model, and object and action hierarchies introduced in proposed Policy Schema, simplifies policy specification by allowing propagation policies (discussed in section 4.2.4.2). In general, if a certain subject role r is allowed to perform a particular action, then a role higher than r should also be allowed to perform the action. Conversely, if roles higher than r are prohibited from performing an action, then r should also be prohibited from performing the same action.

The propagation policies greatly simply management by allowing inheritance of authorisation and prohibition based on the parents in the hierarchy. For example, all the files in a directory can be set to read-only by setting the parent directory as read-only, in comparison to setting the read-only property on individual files.

5.4 Conflict Detection

In RFC 3198, a *policy conflict* is defined to occur when conditions of two policies are satisfied simultaneously but their actions contradict each other. Hence, the entity implementing the policy would not be able to determine which action to perform. For example, a policy that defines that “Gold user is allowed to access VoIP services between 9am and 5pm” would be in conflict with aother policy that defines that “Gold user is not granted access to VoIP services on weekends”. Since, if both these policies are enforced, Gold user is both granted (authorised) and refused (prohibited) permission between 9am and 5pm on weekends. A *policy conflict* is different from a *policy error*, which occurs when attempts to enforce a policy action fail, either due to a temporary state or a permanent mismatch between policy actions and the device enforcement capabilities.

These conflicts can arise due to omissions or errors, or conflicting requirements while specifying the policies. The following section presents a classification of the conflicts and discusses the need for both dynamic and static conflict analysis.

5.4.1 Conflict Specification

Dunlop et. al. [123] have classified the conflicts into four broad categories. Each category of a conflict may present itself either statically or dynamically.

a. *Internal Policy Conflict* occurs when the policies assigned to a single role are deemed to be incompatible with each other. For example, a subject is both authorised and prohibited to perform an action under same set of conditions. Detection of internal policy conflict is required when a new policy is added to the role specification either when the role is initially defined or sometime during the lifetime of the system as the objectives of the role evolves.

b. *External Policy Conflict* occurs when an administrator combines roles, which in isolation of each other do not conflict, but contain policies which in co-existence are in conflict. For example, a policy authorises a subject to download a file anytime while another policy denies all downloads when a server backup is in progress. External policy conflicts may be detected when a new user is assigned to a role (this may occur at run-time) and/or when a new policy is assigned to a role.

c. *Policy Space Conflict* occurs when two or more policy spaces manage the same set of subjects and attempt to enforce different and conflicting policies over them. A *policy space* is defined as a description of entities and action about which a policy is written (a similar concept to Policy Schema discussed in this thesis).

   An example of a policy space conflict might occur when a subject is allowed to both bid and sell the same object in an auction service. Detection of a policy conflict needs to occur both when the policies are initially identified, and at run-time when a new policy is assigned to a role (which may be in conflict with another policy assigned by a different policy).

d. *Role Conflict* occurs when a subject obtains a set of incompatible role assignments. The detection and subsequent resolution of role conflict is required to ensure that the subject does not operate with a union of privileges determined to be incompatible. For example, obtaining the roles of a *banker* and an *auditor* would likely be considered a role conflict. Role conflicts need to be detected when users are initially assigned to roles and when users acquire roles at run-time.

5.4.2 **Classification of Conflicts**

Based on the identified conflict categories, we have further classified the conflicts as follows:

a. ***Conflict caused by Authorisation policies***

Also termed as *modality conflicts*, these are inconsistencies in the policy specification which may arise when two or more policies with modalities of positive and negative authorisation ($Auth+$ and $Auth-$) are defined with same actors, targets and actions [104].

In RBAC terms, modality conflict deals with general conflict such as “no role can be both authorised and denied for the same object and the same action.” Modality conflicts arise from overlapping domains and when there is a triple overlap between sets of subjects, targets and actions (Figure 5-1), however it is impractical to prevent these overlaps.

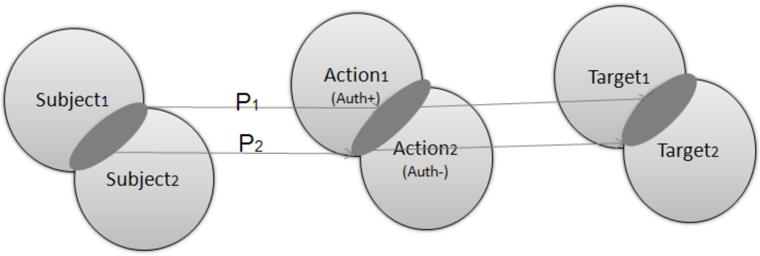


Figure 5-1 Modality Conflict

b. ***Conflict caused by Propagation policies***

An implicit definition of propagation policies may result in unforeseen conflicts. Propagation policies inherently transfer the authorisation and prohibition control, based on the child-parent mappings, which may lead to side effects in access control. Additionally, sometimes an administrator may need to define the role structure “upside down” in some situations, and this will result in propagation conflict in which only lower role users are permitted to do something. For example, a situation in which the rank of member status is decided by how many points the member has purchased. In this case, members with a role lower than $Gold$ should be permitted to access the service to purchase the points and a positive authorisation would be expected to propagate down [91]. Upside-down cases are also common in revoking or cancellation services where lower roles are allowed to perform actions while higher roles are prohibited to do so.

c. ***Conflict caused by Action Composition***

RFC 3060 has defined the purpose of a policy action is to execute one or more operations and actions which may need to be ordered. An action composition implies a set of actions which need to be completed in phases and sometimes in order. For example, to access a secured VPN connection, a user needs a digital certificate and then a password.

The respective actions in the action composition do not need to be related to a single target/service or in a particular order. For instance, the W3C working group [124] has defined case models for web services. A sample case scenario might occur when an online travel booking is made. The user may need authorisation for booking air travel and a hotel reservation. This means that the user should be authorised for both actions (which may be managed by different policies). Any missing authorisation causes prohibition for user's access to authorised services.

d. ***Conflict caused by Constraints***

In our analysis of the proposed RBAC extensions in section 4.2.5, we have demonstrated that the inherent nature of the hierarchical structure of the Policy Schema helps in detecting logical conflicts by introducing constraints such as Separation of Duty and Chinese Wall Security model. Even though these constraints help in implicitly preventing some conflicts, a formal policy definition is required to properly implement these constraints. Any ambiguity in applying the constraints may result in policy conflicts.

We have also introduced a Time constraint, which is not inherently implemented by the Policy Schema. The following discusses policies based on the constraints:

i. ***Separation of Duty (SOD) Policy***

Separation of Duty policy is a constraint policy imposed on existing policies which map users (e.g. John) to roles (e.g. Banker and Auditor) to prevent any *role conflict*s. It reduces the possibility for significant errors by partitioning tasks and associated privileges such that the cooperation of multiple users is required to complete sensitive tasks. The SOD constraints have been identified as [125]:

I. The *static* SOD policy ensures that a user should not be assigned to two roles which are in conflict with each other. In other words, it means that conflicting roles cannot have common users.

II. A *dynamic* SOD property with respect to the roles activated by the users requires that no user can activate two conflicting roles simultaneously. In other words, conflicting roles may have common users but users can not simultaneously activate roles that are in conflict with each other.

ii. ***Chinese Wall Security Policy***

The basis of the Chinese-Wall security policy is to check that the roles are allowed access to a target which is not held in conflict with any other target that they already possess.

The Chinese-Wall security policy is a specialised form of *dynamic* SOD that prevents a role from performing the same action on conflicting objects. Initially, a role has permission for an action on an object. However, once the action is performed, the role cannot perform the same action on a conflicting object. Hence, the Chinese Wall policy constraint is a subtle combination of free choice and mandatory control [110].

iii. ***Time Constraint***

A time constraint policy can be used to specify the period during which an authorisation policy is valid. For example, a *Gold* user can *play* a *movie* on *weekdays*. Another policy specifies that a *Guest* user *cannot play movies* within *business hours*. A time constraint itself does not cause a policy conflict. Policy conflicts can happen only if the time periods specified in various policies overlap, which, as in case of authorisation policies, is impractical to prevent.

5.4.3 Static and Dynamic Conflict Detection

The preference for static and dynamic conflict detection identifies as to when and where the conflict analysis is performed, since the process of conflict detection (and resolution) can be a computationally intensive, time-consuming, and hence an expensive task. Furthermore, there is a probability of several possible sources of conflicts that may occur in a large and dynamically changing network environment. Hence, the choice of when to perform a policy analysis is crucial.

a. Static Conflict Detection

Static conflict detection aims to detect all types of potential conflicts (possible or definite) which clearly could cause conflicts from the policy specification. Hence, it is *preferable* to analyse the policies statically at compile-time, especially when the policies are defined for the first time or whenever the policies have been modified [126]. This analysis is performed offline as a standalone process and any identified static conflict requires immediate attention and resolution, as it will *most* certainly result in a conflict at some time.

The scope of static conflict detection is dependent on the preference of whether to detect conflicts which are clearly specified in the policy specification (*predicted potential conflict*), or to also include conflicts which are not yet identified from the policy specification, but may lead to the conflicts if one or more entities are in the policy space at a given context (*unpredicted potential conflicts*). While including unpredicted potential conflicts will certainly speed up the performance in responding to the user's requests at the runtime (since all the possible conflicts has been detected), the major drawback is the requirement of system resources, since conflicts have to be detected based on all the possible combinations of entities (subject and targets) and their actions, especially when the policy analysis is to be performed for a large and dynamic environment.

Our proposed conflict detection approach will detect static conflicts of all the types identified in section 5.4.2, i.e. conflicts caused by authorisation policies, propagation policies, action composition policies, static and dynamic Separation of Duty polices, Chinese Wall security policies and time constrained policies.

b. **Dynamic Conflict Detection**

Unlike static conflict detection, dynamic conflict detection is performed at run time by dynamically detecting all unpredicted potential conflicts between the entities of the Policy Schema in a given context. As it is performed at run time, the system needs to decide on when to trigger the detection module. There are three different strategies proposed on when to dynamically detect a conflict [126]:

i. *Reactive model* is where dynamic conflict detection is only triggered when there is an explicit request from users, for example, when a user performs any action (start, stop, etc.) from a mobile device to request a service. The system then collects entities' relevant context information and reactively detects the conflicts between other entities in the given context. This technique is more suitable in a situation with only a few requests from the entities. Furthermore, the detection is only limited to the current location, day, and time, which are related to the requested service.

ii. *Proactive model* implicitly and automatically detect the conflict by sensing a user's current context i.e. when the user moves in or out of a service area or a geographic location (for example, an office room). The proactive model is considered as a pessimistic conflict detection approach where it detects and caches all the potential conflicts that may occur in the given context assuming that there *will* be a conflict between entities.

   However, this technique is considered useful only if the participating entities are in the same context (i.e. same room). Even if *one* of the entities has moved to different location, the predicted potential conflict may not be an actual conflict. The model gets even more complex due to an inconsistent state when a new entity (i.e. a user) moves into the same room while the process of conflict detection is in progress.

iii. *Predictive model* detects the conflicts based on user's history file. By analysing the user's usage/movement history, the model can predict the user's movement and employ a proactive model for conflict detection. For example, from the history file, user A is always going to room X to meet user B on Monday at 1pm. Based on this information, the system will proactively compute conflict detection between user A and B in room X at 1pm on Monday. However, if the user's movement and activity are not anticipated by the system, there will be a delay in responding since the system will need to re-detect the conflict based on the user's current context (i.e. current location, time and place).

It is evident from the overview of dynamic conflict detection strategies that a mathematical representation of such models as a single solution is not practical, especially in our specific scenario of providing ABC services, where the number of roaming mobile users is large and dynamic. Furthermore, any dynamic conflict is still a potential and quite unpredictable, in that it may, or may not, proceed to an actual conflict and the inconsistency between policies may be exposed temporarily, or indeed not at all.

Therefore, to mathematically represent the dynamic context of users in policies, we have used a combination of reactive and proactive models by introducing *event* and *time* constraints. An event identifies an occurrence of a set of circumstances which can trigger a specific policy and defines the context of the user. A time constraint is a subset of an event, however due to its ubiquitous nature, time is treated as a separate constraint. Every policy defined in the Policy Schema inherently holds a time constraint. If any time period is not explicitly defined, the policy is applicable for all times. Any explicit request from a roaming mobile user, such as a request for higher bandwidth for a small duration, is treated by reactive approach i.e. only triggered when there is an explicit request from users.

Hence, the goal of policy analysis proposed for the Semantic layer can be summarised as:

- to identify *actual* conflict that has occurred and can be resolved statically, at compile-time;

- to *predict* that a conflict, may, occur in the future (and more specifically, exactly what circumstances will expose that conflict);
- to *monitor* identified potential conflicts; and,
- to *communicate* the actual or potential conflict to a resolution process or in some cases, a human operator, for assistance in the *resolving* the conflict situation.

There are also sub-goals of conflict detection (Figure 5-2):

(a) To group the conflicts based on its type i.e., a possible potential conflict or a definite potential conflict. This is useful to decide on when to resolve the conflict.

(b) To investigate the best technique for conflict detection based on the sources and types of the conflict.

In the following section, we will discuss different techniques proposed by researchers for detecting different types of conflicts, followed by our proposed approach.

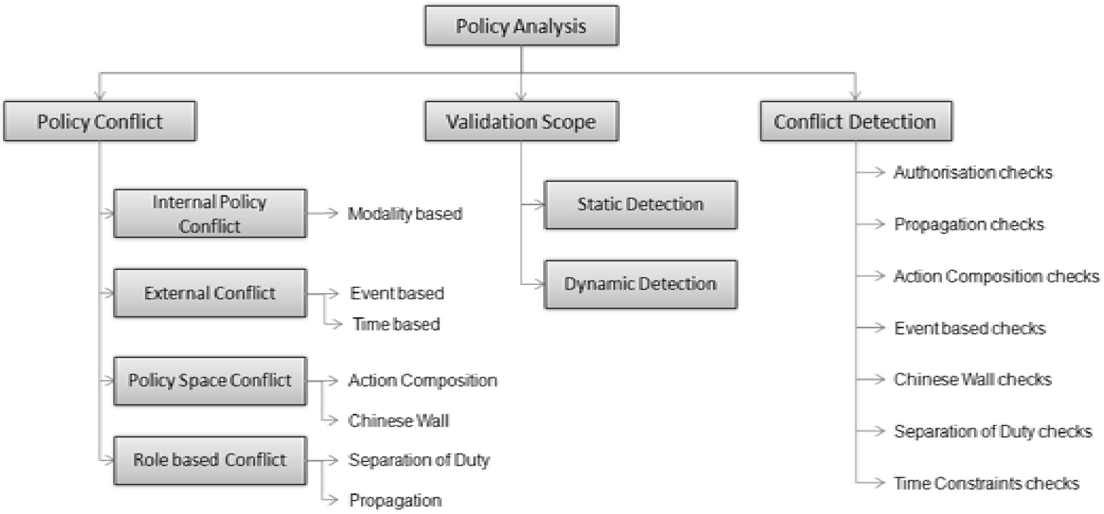


Figure 5-2 Policy Analysis

5.5 Current Trends

Instead of defining a policy notation using a specification language and its underlying grammar, the IETF policy working group proposed using UML-like generic object oriented modelling notations, such as PCIM and PCIMe (discussed in section 4.3.1). Even though this approach constitutes a good choice from the point of view of low-level policy distribution where policy notation is an integrated part of a whole system, however, the IETF approach is considered *too slow* and *too verbose* as every component of a policy is identified to be a full class [21]. Hence, if a policy is of the form:

$$"if\ condition\ (C1\ and\ C2\ and\ C3)\ or\ (C4\ and\ C5) \\ \rightarrow (then\ perform\ action)\ A1, A2, A3\ "$$

it needs to be modelled in at least ten classes: five classes for the conditions, fours for actions, and one for the policy itself, which is in addition to the classes resulting from instantiations of different objects and their relationship classes.

Researchers have since proposed different methods/approaches for policy specification, ranging from formal policy languages that a computer can directly process and interpret, to rule-based policy notations using an if-then-else format, to the representation of policies as entries in a relational database management model managing multiple attributes. An analysis of current approaches to policy specification (semantic based) has outlined two main research approaches moving in opposite directions [127].

On one side, a purely ontology-based approach is followed which relies on the expressive features of Description Logic (DL) languages [128] and uses logic to describe contexts and associated policies at a high level of abstraction, in a form that allows their classification and comparison. This feature is essential in order to detect conflicts between policies before they are actually enforced (i.e. static conflict detection), thus granting interoperability among entities belonging to different domains that adopt different policies. By means of preliminary analysis of policy typologies in different domains, the required policies can be compared and harmonized, if needed, avoiding the cost of failures due to conflicts arising in the enforcement phase. Another interesting application of ontology-based approach lies in the possibility of exploiting policy description to facilitate policy negotiations. Since, in a dynamic and large environment, entities may wish to interact with potentially un-trusted entities, hence, negotiating policy disclosure may help interacting parties to reach an agreement about their mutual behaviour without imposing heavy limitation to their privacy.

On the other side, a rule-based approach is followed to enable evaluation and reasoning about context and policy instance by encoding policies as Logic Programming (LP) rules [129] and relies on the features of logic programming languages [130], such as Prolog. In fact, from the enforcement point of view, policies can be considered as “instructions” to be executed, provided that their activating conditions are evaluated to be true. This perspective suggests that policies should be evaluated in a clear, concise and expressive way to facilitate their evaluation and enforcement, similar to the code of a programming language that needs to be complied or interpreted. For example, the language should allow for the definition of policies over dynamically determined constraints, including run time variables.

Policy languages based on semantic language, such as Resource Description Framework (RDF), DARPA Agent Markup Language (DAML), Web Ontology Language (OWL) and their successor DAML+Ontology Interchange Language (DAML+OIL) [131] have been proposed. Other semantically rich languages, such as KAoS [132] and Rei [133] represent intermediate approaches between the two extremes, i.e. Rule-based and Ontology-based approaches.

Other initiatives define a completely new family of conflict detection methods based on object-oriented modelling. One of the well known policy language of this class is the Ponder language [134], which is a declarative, object-oriented language that can be used to specify both security and management policies. An extension of the Ponder language, Alloy [135], deals with the delegation of obligation policies, where the main issues are the balance between authorization and obligation policies, the source of obligations and the reasons for delegation, and meta-policies for controlling the delegation of obligations. The Chisel policy language [136] deals with mobile-aware dynamic changes in the behaviour of various services of the middleware and allows unanticipated behaviour at run time using behaviours as meta-types.

Furthermore, various domain specific policy languages have been proposed to define policies for specific domains, such as, eXtensible Access Control Markup Language (XACML) [137] for access control, Path-based Policy Language (PPL) [138] for QoS domain and P3P Preference Exchange Language [139] for privacy policies. In addition, policy languages based on graphical schematics have been proposed such as based on graph theory [140].

As discussed in conflict classification that policy combinations may introduce policy conflicts, several languages dealing with policy combination and policy groups have also been proposed. Policy combination methods for many mutually dependent policies on specific purposes are formalized in [141]. Another concept called Policy of Policies [142] has been proposed to orchestrate the deployment of dependent policies following a Policy Finite State Machine-based lifecycle, especially designed for time-sensitive configuration policies.

Since the Policy Schema proposed in this research is based on the RBAC model, conflict detection methods based on the RBAC concept are of our particular interest. Graham et al. [143] proposed a method to detect a modality conflict in the RBAC model by using a decision table. Strembeck [144] presented a method to detect a static separation of duty conflict caused by propagation. However, these methods do not address conflict caused by the structure of actions and they are very specific to a particular policy model.

Kamoda et al presented a static method for conflict detection based on Free-Variable Tableaux (FVT) [91]. The FVT method is a sound and complete theorem prover which can be built based on abductive reasoning. Detection of a conflict effectively requires that a contradiction is derived from a collection of policies. The simplicity of the first-order logic allows a concise policy representation and a faster detection of a conflict. This method also infers the cause of the conflict [145].

Since, the FVT approach also assumes an access control model in which subjects, targets, and actions all have some structure [146] together with the simplicity of first-order logic, this approach is well suited for our proposed conflict detection method. However, we have proposed composite mapping for policy analysis as discussed in the following section.

5.6 Proposed Approach

As discussed, policy analysis refers to the process of checking specifications of the PBM system to ensure that the consistency requirements are met. Ideally, analysis should be performed *before* new policies are deployed so that an administrator has the confidence that new policy will not cause any failures.

There are many types of use case models for policies for providing services [124], and we assume the “aggregation services model”. This model is mainly used for services, such as, for access control and authorisation to services offered by the network/service provider. The management server provides an access to the roaming guests and home users. Based on the Policy Schema and authorisation policies of a user, the management server then checks whether the user should be granted access to the requested service. If it is granted, then the request is transferred to an appropriate *services server* to serve the request.

We have proposed a new technique for policy analysis that uses a formal representation of policies based on first-order logic, together with the composite mapping and reasoning techniques based on set theory to allow administrators to check a range of consistency properties. This technique has significant advantages over previous work on policy analysis since it does not require information about the run-time state of the system in order to detect inconsistencies (ASL [147], Rei), based on complete theorem prover and accounts for the effects of enforcing propagation policies, action composition policies and evaluates separation of duty, Chinese Wall and time constraints.

Since, performing a static policy analysis at the client side (i.e. roaming mobile device) can be more expensive due to constraints such as limited resources, power and processing speed on the mobile device, we propose to perform the conflict detection only at the server side and the relevant conflict free policies are then downloaded to the mobile client. The proposed conflict detection is based on the Policy Schema introduced in section 4.2.4.

5.6.1 Policy Schema

In our proposed approach, we have extended the NIST RBAC model with hierarchies of *Object* (the term Target is usually used by conflict detection approaches) and *Actions*. For demonstration purposes, we have followed a Policy Schema for a hierarchy of users as in Listing 5-1.

The policies are defined by a collection of policy rules governing whether the action is permitted or prohibited. We assume that the authorisation policies needed for checking the request are defined in terms of subject and target role structures [148], [149]. Policies can propagate up or down the role structure. Furthermore, an authorisation policy may be defined in terms of composite actions, which can result in conflicts if separate policies are defined for the various sub-actions. We also assume that we can define different kinds of constraint policy, including the Chinese Wall security policy, Separation of Duty policy and time constraint policies (as discussed in section 5.4.2).

Assuming the organisation has 120 users, and characterised as:

Platinum Users (S_1...S_5)

CEO	S_1
GM Administrative	S_2
GM Marketing	S_3
GM Software	S_4
GM HR Dept.	S_5

Gold Users (S_6..S_{10})

Directors	S_6
Administrative	S_7
Director Marketing	S_8
Director Software	S_9
Director HR Dept.	S_{10}

Silver_I	Users (S_{11}...S_{20})	Programmers
Silver_II	Users (S_{21}...S_{40})	Sales
Bronze_I	Users (S_{41}...S_{60})	Representative
Bronze_II	Users (S_{61}...S_{80})	Customer Service
Bronze_III	Users (S_{81}...S_{100})	Contractors
Guest	User (S_{101}...S_{110})	
VIP Guest	User (S_{111}...S_{120})	

Listing 5-1 User distribution in Policy Schema

5.6.2 **Composite Mapping**

In our approach, a policy is a composite function (Figure 5-3) denoted by $(g \circ f)$ which is mapped from a Subject set (consists of users in roles) to a Target set (objects such as printer, media player, network services, etc.) and then mapped into an Action set (print, play, enable, disable, etc.) depending on the context of subject subset. A partial order relation is defined among elements of Subject set and the graph representation of the relation defines a *role hierarchy structure.*

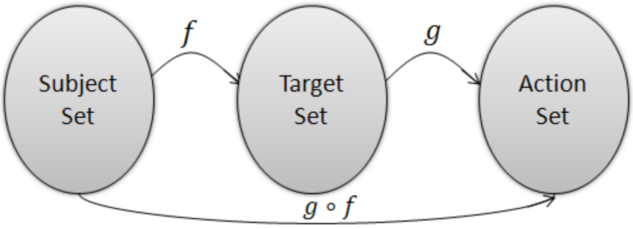


Figure 5-3 Composite Function (S→T→A)

Mathematically,

If, f is a mapping from Subject set S to a Target set T i.e. $f: S \rightarrow T$, and
g is a mapping from Target set T to Action set A i.e. $g: T \rightarrow A$, then
composite mapping is defined from S to A and is denoted by $(g \circ f)$

$Elements\ of\ S\ are\ denoted\ by\ S_i\ where\ S_i = S_1\ to\ S_{120}$

$Elements\ of\ T\ are\ denoted\ by\ t_i\ and\ \overline{t_i}\ where$

$$t_i \cap \overline{t_i} = \emptyset \qquad and,$$

$$t_i \cup \overline{t_i} = T$$

$Elements\ of\ A\ are\ denoted\ by\ a_i\ and\ \overline{a_i}\ where$

$$a_i \cap \overline{a_i} = \emptyset \qquad and,$$

$$a_i \cup \overline{a_i} = A$$

We have defined policy rules in terms of composite mapping, such as

$Policy\ r: gof\ (User = S_i: S_1 \leq S_i \leq S_{120}): Auth+/Auth-, Obli+/Obli-(S,T,A)$

$$\begin{aligned} Policy\ r &= r(g \circ f) \\ &= Auth + (g \circ f)user \\ &= Auth + (g \circ f)x_i \\ &= Auth + \{g \circ (f(user))\} \\ &= Auth + \{g(target)\} \\ &= Auth + \{(Action)\} \end{aligned}$$

$where\ x_i\ is\ the\ user\ or\ entity\ of\ subject\ set$

5.6.3 **Conflict Detection**

A conflict can be detected in this representation by visualising that a (role-based) subject is performing action(s) on same targets when two different policies are applied simultaneously on that particular role. Similarly, there can be situations where two or more roles wish to perform actions on the same target, in which case priorities can be imposed.

In this section, we will review the examples presented in FVT analysis [91] and compare it with our proposed composite mapping scheme. Finally, we will discuss different case scenarios which prove that our presented conflict detection method offers a better practical solution compared to the FVT approach for administrators to define conflict-free policies for a PBM system.

A. **Authorisation Policies**

This is the most basic policy defined in the management server which defines the authorisations between a subject and a target. Policy $r1$ specifies that the subject role *Bronze_II* is allowed to perform an action *play* on the target *movie* and policy $r2$ specifies that the subject role *Gold* is *prohibited* to perform the action *play* on the target role *movie*. The policies $r1$ and $r2$ appear to define authorizations for different subject roles so there should be no problems. However, if these policies are compared with respect to the role structure, then a conflict occurs.

$$Policy\ r1: \qquad Auth + \left(User_{Bronze\ \_II}, movie, play\right)$$

$$Policy\ r2: \qquad Auth - (User_{Gold}\ , movie, play)$$

Policy $r1$ and $r2$ are defined as follows:

$$\begin{aligned} Policy\ r1: \quad & g_1 \circ f_1\left(User_{Bronze\ \_II} = S_i : S_6 \leq S_i \leq S_{80}\right) : Auth + (S,T,A) \\ & = g_1(t_1) \\ & = a_1 \qquad \qquad \ldots\ldots\ldots\ldots\ldots(1) \end{aligned}$$

$$\begin{aligned} Policy\ r2: \quad & g_2 \circ f_2(User_{Gold} = S_i : S_6 \leq S_i \leq S_{10}) : Auth - (S,T,A) \\ & = g_2(\overline{t_1}) \\ & = (\overline{a_1}) \qquad \qquad \ldots\ldots\ldots\ldots\ldots(2) \end{aligned}$$

$where, t_1 \in T\ represents\ movie\ device\ preset\ in\ Target\ Set\ T,$

$\overline{t_1}$ *indicates movie device not present in Target Set T*,

$a_1 \in A$ *represents Action* Play *to the movie device present in Target Set T*,

$\overline{a_1}$ *indicates Action* not to play*on the movie device present in Target Set T*

B. Propagation Policy

The hierarchy structure of Subject, Target, Action, simplifies policy specification by allowing propagation policies. In general, if a subject element S_i is allowed to perform a particular action, then roles higher than $\mathcal{R}$ should also be allowed to perform the action. Conversely, if roles higher than $\mathcal{R}$ are not permitted to perform the action, then $\mathcal{R}$ should not be permitted to perform the action. These propagation policies are specified as follows:

Policy *r3* specifies $Auth+$ policy defined for a role hierarchy (RH) where $\mathcal{R} \in RH$ propagates upwards through roles and policy *r4* specifies $Auth-$ policy defined for RH propagates downward through roles.

$$Policy\ r3: \quad prop(Auth+, \mathcal{R} \in RH, Up)$$

$$Policy\ r4: \quad prop(Auth-, \mathcal{R} \in RH, Down)$$

Policy $r3$ and $r4$ are defined as follows:

$$Policy\ r3: \quad g_3 \circ f_3\big(Users_{All}\ above\ than\ User_{Bronz\ e_{II}} = S_1 \leq S_i \leq S_{60}\big):$$

$$Auth + (S, T, A)$$

$$= g_3(\ t_1)$$

$$= (a_1) \qquad \text{................(3)}$$

From Equations (2) and (3), $User_{Gold}\,(S_6 \leq S_i \leq S_{10})$ is allowed to play the movie by equation (3) and is not allowed to play the movie by equation (2). This is a contradiction in policy $r2$ and $r3$.

$$Policy\ r4: \quad g_4 \circ f_4(User_{All}\ below\ than\ User_{Gold} = S_i : S_{11} \leq S_i \leq S_{120}):$$

$$Auth - (S, T, A)$$

$$= g_4(\overline{t_1})$$

$$= (\overline{a_1}) \qquad \text{................(4)}$$

From Equation (1) and (4), $User_{Bronze\ \_II}(S_{61}\ \leq S_i \leq S_{80})$ is authorised to play the movie by equation (1) and is not allowed to play the movie by equation (4). This is a contradiction in policy $r1$ and $r4$ by the theory of Boolean algebra (e.g. $x = 5\ and\ \overline{x} = 5$ cannot happen).

C. **Action Composition**

The W3C working group has defined a case scenario [124] for booking a travel package online where a user needs authorisation for booking air travel and a hotel reservation. This means that the user should be authorised for both actions. Any missing authorisation causes prohibition for user's access to authorised services.

Policy $r5$ specifies that the subject role *Bronze_II* is authorised for travel package reservation. Policy $r6$ specifies that the subject role *Bronze_II* is authorised for air ticket reservation while policy $r7$ specifies that the subject role *Bronze_II* is not authorised for hotel reservation. Policy $r8$ requires both air ticket and hotel reservation to book a travel package.

$$Policy\ r5: Auth + \left(User_{Bronze\ \_II}, TR, rsv_{travel}\right)$$
$$Policy\ r6: Auth + \left(User_{Bronze\ \_II}, TR, rsv_{air}\right)$$
$$Policy\ r7: Auth - \left(User_{Bronze\ \_II}, TR, rsv_{hotel}\right)$$
$$Policy\ r8: rsv_{travel}\ = rsv_{air}\ \wedge rsv_{hotel}$$

$where, rsv_{travel}\ , rsv_{air}\ and\ rsv_{hotel}\ mean\ send\ a\ request\ for\ some\ holiday\ abroad, to$
$reserver\ an\ airline\ ticket, respectively.$
$TR\ indicates\ a\ Web\ Service\ that\ provides\ travel\ reservation\ services$

Policy $r5$, $r6$, $r7$, $r8$ are defined as follows:

$$Policy\ r5: \quad g_5 \circ f_5\left(User_{Bronze\ \_II} = S_i : S_{61} \leq S_i \leq S_{80}\right): Auth + (S, T, A)$$
$$= g_5(rsv_{air} : at_5) \cap g_5(rsv_{hotel} : ht_5)$$
$$= (airTicket_{reserved} : a_5) \cap (hotelAccomodation_{reserved} : h_5)$$
$$= a_5 . h_5 \qquad \text{................}(5)$$

$where \quad at_5\ represents\ reservation\ in\ airplane\ present\ in\ Target\ set\ T$
$and \qquad ht_5\ represents\ reservation\ available\ in\ hotel\ in\ Target\ set\ T$

$Policy\ r6$: $g_6 \circ f_6\big(User_{Bronze\ \_II} = S_i\!: S_{61} \le S_i \le S_{80}\big)\!: Auth - (S, T, A)$

$= g_6(rsv_{air} : at_5)$

$= (airTicket_{reserved} : a_5)$

$= \ a_5$ ....(6)

$where \quad at_5\ represents\ reservation\ in\ airplane\ present\ in\ Target\ set\ T$

$Policy\ r7$: $g_7 of_7\ \big(User_{Bronze\ \_II} = S_i\!: S_{61} \le S_i \le S_{80}\big)\!: Auth - (S, T, A)$

$= g_7\big(\overline{rsv_{hotel} : ht_5}\big)$

$= \big(\overline{hotelAccomodation_{reserved} : h_5}\big)$

$= \ \overline{h_5}$ ....(7)

$ht_5\ represents\ reservation\ available\ in\ hotel\ in\ Target\ set\ T$

$Policy\ r8$: $g_8 of_8\ (User_{All} = S_i\!: S_1 \le S_i \le S_{120})\!: (S, T, A)$

$rsv_{travel} \ = rsv_{air} \cap rsv_{hotel}$

$= a_5 . h_5$ ....(8)

For $User_{Bronze\ \_II}(S_i\!: S_{61} \le S_i \le S_{80})$, the policies $r5$, $r6$ and $r7$ are conflicting when combined in action composition policy $r8$. Since, as per equation (5) $User_{Bronze\ \_II}$ is authorised for air ticket as well as hotel accommodation reservation, while as per equation (7) is not authorised for reservation of hotel accommodation indicating a static conflict between policies. This conflict can also be proven by the help of Set theory, and is discussed as follows:

By Set theory:

$Let,$

$$U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, \mathrm{a}_5 = \{1, 3, 5, 6,7\}, h_5 = \{3, 5, 7, 8, 9\}$$

$Then,$

$$\overline{a_5} = \{0, 2, 4, 8,9\},\ \overline{h_5} = \{0\ ,1\ ,2, 4, 6\} \quad and,$$

$$a_5 . h_5 = a_5 \ \cap h_5 \ = \ \{3, 5, 7\} \quad and,$$

$$\overline{a_5 . h_5} \ = \ \overline{a_5 \ \cap h_5} \ = \ \overline{a_5} \ \cup \ \overline{h_5} \ = \{0, 1, 2, 4, 6, 8, 9\}$$

As per Boolean algebra, policies $r5$, $r6$ and $r7$ state that:

$$a_5 . h_5 \ = \ \overline{h_5}$$

$Multiply\ both\ sides\ by\ \overline{a_5}$

$$\overline{a_5}.(a_5.h_5) \;=\; \overline{a_5}\,.\overline{h_5}$$

$$\emptyset \;=\; \overline{a_5}\,.\overline{h_5}$$

$$=\; \overline{a_5 + h_5}$$

$$\neq \emptyset$$

D. Event based policy

To represent the dynamic context of users in policies, we have introduced *event* and *time* constraints. An event identifies an occurrence of a set of circumstances which can trigger a specific policy and defines the context of the user.

Let us consider a policy $r9$ which specifies that a $Gold$ user is allowed to access (use) VoIP services between 9am and 5pm, while policy $r10$ specifies that no user is allowed to access VoIP services during weekends. When the policies are combined based on events (weekdays and weekends), $Gold$ user is both authorised and prohibited access to VoIP services between 9am and 5pm at weekends. This conflict can be detected as follows:

$$policy\, r9{:}\, A + \left(OnEvent_{all\_days}\,, User_{Gold}\,, Service_{VoIP}\,, Acess_{use}\right)$$

$$policy\, r10{:}\, A - \left(OnEvent_{weekends}\;, User_{All}\,, Service_{VoIP}\,, Acess_{use}\right)$$

Now, we define an Event based policy, where an Event set E is defined consisting of elements $e_1, e_2, \dots, e_i, \dots, e_n$ such that

$$E = \{e_1, e_2, \dots, e_i, \dots, e_n\}$$

A mapping function c is defined as a condition based mapping which maps from Event set E to Subject S, such that

$$c{:}\, E \;\to S$$

$$s_i = c(e_i), where\ s_i \in S$$

Diagrammatically, a composite mapping based on condition-event can be represented as (Figure 5-4):

$$(g \circ f \circ\; c){:}\, E \;\to A$$

$$(g \circ f \circ c)\;(e_i) = a_i; \quad where\ e_i \in E,\; a_i \in A$$

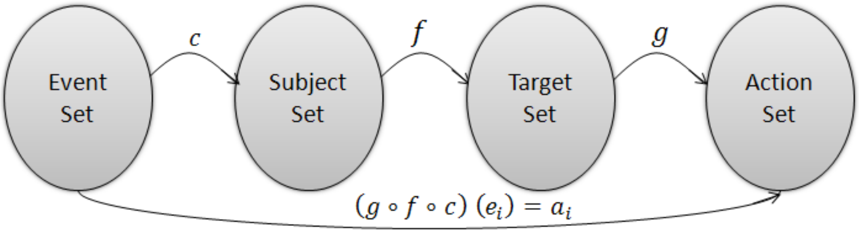


Figure 5-4 Composite Function (E→S→T→A)

Policy $r9$ and $r10$ are defined as follows:

$$policy\,r9\colon\ g_9 \circ f_9 \circ c_9\big(e_i\colon e_1 \le e_i \le e_7,\ e_i = OnEvent_{all\_days}, User_{Gold} = S_i\colon S_6 \le\ \ S_i \le S_{10}, t_i = Service_{video}, T,\ a_i \in A\big)\colon Auth + (E,S,T,A)$$

$$= Auth + (g_9 \circ f_9)\ c_9(e_i); where\ c_9(e_i) =\ S_i\colon S_6 \le S_i \le S_{10}$$

$$= Auth + (g_9 \circ f_9)\ (S_i\colon S_6 \le S_i \le S_{10})$$

$$= Auth + g_9(t_i); where\ f_9(S_i) = t_i$$

$$= a_i \qquad \text{................}(9)$$

$$policy\,r10\colon\ g_{10} \circ f_{10} \circ c_{10}(e_i\colon e_6 \le e_i \le e_7,\ e_i = OnEvent_{weekends}, User_{All} =\ S_i\colon S_1 \le\ \ S_i \le S_{120}, t_i = Service_{video}, T,\ a_i \in A)\colon Auth - (E,S,T,A)$$

$$= Auth - (g_{10} \circ f_{10})\ c_{10}(e_i); where\ c_{10}(e_i) =\ S_i\colon S_1 \le S_i \le S_{120}$$

$$= Auth - (g_{10} \circ f_{10})\ (S_i\colon S_1 \le S_i \le S_{120})$$

$$= Auth - g_{10}(t_i); where\ f_{10}(S_i) = t_i$$

$$= \overline{a_i} \qquad \text{................}(10)$$

Any conflicts between policies $r9$ and $r10$ can be detected by combining equations (9) and (10).

On computation,

$$a_i = Auth + g_9(t_i)$$

$$= Auth + (g_9 \circ f_9)\ (S_i\colon S_6 \le S_i \le S_{10})$$

$$\overline{a_i} = Auth -\ g_{10}(t_i);\ \ where\ f_{10}(S_i) = t_i$$

$$= Auth - (g_{10} \circ f_{10})\ (S_i\colon S_1 \le S_i \le S_{120})$$

Hence,

$$a_i \cap \overline{a_i} = S_i\!: S_6 \leq S_i \leq S_{10}$$
$$\neq \emptyset$$
$$\text{................(10A)}$$

which contradicts with the fact: $a_i \cap \overline{a_i} = \emptyset$ ....(10B)

E. Chinese Wall and Separation of Duty Policy

A Chinese Wall security policy and Separation of Duty policy defines the constraints for targets and actions respectively. Policy $r11$ specifies that subject role $Guest$ is permitted to view accounts of exactly one of the targets i.e. either $Bank_A$ or $Bank_B$ (Chinese Wall security policy:CW). Policy $r12$ specifies that subject role $Bronze\_I$ is permitted to either $sell$ or buy items through an auction site, but not $sell$ and buy a same item (Separation of Duty policy:SoD).

$$Policy\ r11\!: CW(User_{Guest}, \{Bank_A, Bank_B\}, view_{account})$$
$$Policy\ r12\!: SoD\big(User_{Bronze\ \_I}, Auction, \{sell_A, buy_A\}\big)$$

Chinese Wall constraint

The Chinese Wall constraint derived from policy $r11$ implicitly define two positive authorisation policies $r13$ and $r14$, such that

$$Policy\ r13\!: Auth + (User_{Guest}, Bank_A, view_{account})$$
$$Policy\ r14\!: Auth + (User_{Guest}, Bank_B, view_{account})$$

Policy $r11$ is defined as follows:

$$\begin{aligned} Policy\ r11\!: \quad & g_{11} \circ f_{11}(User_{Guest} = S_i\!: S_{101} \leq S_i \leq S_{120}, \\ & \quad t_1 = Bank_A,\ t_2 = Bank_B,\ t_1, t_2 \in T, \\ & \quad a_{11} = viewAccount_A, a_{14} = viewAccount_B,\ a_{11}, a_{14} \in A) \\ & Auth + (S, T, A) \\ & = g_{11}(t_1).\, g_{11}(t_2) \\ & = a_{11}.\, a_{14} \\ & = (viewAccount_A).\,(viewAccount_B) \qquad \text{................(11)} \end{aligned}$$

$where\ t_1, t_2 represents\ Bank_A, Bank_B\ in\ Target\ set\ T\ and$

$a_{11}, a_{14}\ represents\ viewAccount_A, viewAccount_B in\ Action\ set\ A$ (Figure 5-5)

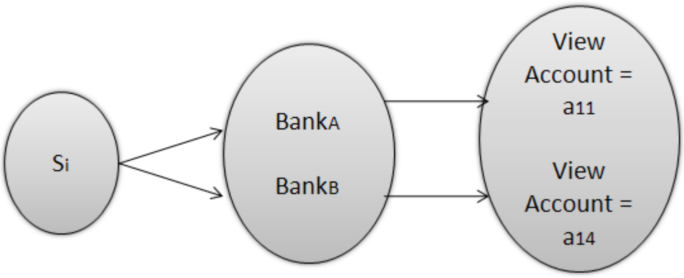


Figure 5-5 Representation of Chinese Wall policy

Separation of Duty constraint

Policy $r12$ specifies that subject role *Bronze_I* of auction services is permitted to either $sell$ or buy items through the auction services, but not $sell$ and buy the same item.

$$Policy\ r12: SoD(User_{Bronze\ \_I}, Auction, \{sell_A, buy_A\})$$

Our Representation

$Policy\ r12$: $g_{12} \circ f_{12}(User_{Bronze\ \_I} = S_i: S_{41} \leq S_i \leq S_{61},$

$t_1 = Auction, t_1 \in T$

$a_{12} = sell, b_{12} = buy; a_{12}, b_{12} \in A):$

$Auth +$

$(S, T, A =$

$either\ sell\ or\ buy\ but\ not\ buy\ and\ sell\ simultaneously)$

$= g_{12}(t_1)$

$= \overline{a_{12}}\, b_{12} + (or\ \cup)\, a_{12}\overline{b_{12}}$ ....(12)

$Here, \overline{a_{12}}, \overline{b_{12}} \in A = Action\ set;$

$\overline{a_{12}}\ represent\ not\ to\ sell\ and\ \overline{b_{12}}\ represent\ not\ to\ buy$

$Policy\ r13$: $g_{13} \circ f_{13}(User_{Guest} = S_i: S_{101} \leq S_i \leq S_{120}, t_1 = Bank_A,\ t_1 \in T;$

$a_{11} = viewAccount_A, a \in A): Auth + (S, T, A)$

$= g_{13}(t_1 = Bank_A)$

$= a_{11} \Longrightarrow viewAccount_A$ ....(13)

$Policy\ r14$: $g_{14} \circ f_{14}(User_{Guest} = S_i:\ S_{101} \in S_i \leq S_{120},\ t_2 = Bank_B,\ t_2 \in T;$

$a_{14} = viewAccount_B,\ a_{14} \in A): Auth + (S, T, A)$

$= g_{14}(t_2 = Bank_B)$

$= a_{14} \Longrightarrow (viewAccount_B)$ ....(14)

These constraint based policies may also lead to other types of policy conflicts. For example, when the two positive authorization policies *r13* and *r14* are combined, a conflict with Policy *r11* results. We have also shown the occurrence of a policy conflict by Venn Diagram (Figure 5-6) and its interpretation by Boolean theory as under:

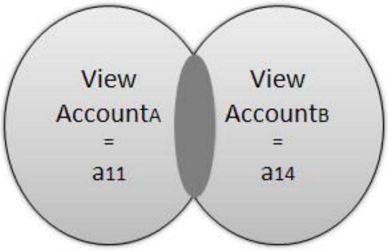


Figure 5-6 Action Set for $User_{Guest}$

Proof of conflict detection

$From\ equation\ (11)$

$$(viewAccount_A).(viewAccount_B) = a_{11}.a_{14} \qquad \text{................} (11)$$

$From\ equation\ (13)$

$$viewAccount_A = a_{11} \qquad \text{................} (13)$$

$From\ equation\ (14)$

$$viewAccount_B = a_{14} \qquad \text{................} (14)$$

$Therefore,$

$$a_{11}.a_{14} = a_{11} \qquad \text{................} (15)$$

$On\ multiplying\ (15)\ by\ \overline{a_{14}}\ both\ sides, we\ get$

$$(a_{11}.a_{14}).\overline{a_{14}} = a_{11}.\overline{a_{14}}$$

$$\emptyset = a_{11}.\overline{a_{14}} \qquad \text{.............} (15.1)$$

$$Now, \quad a_{11}.a_{14} = a_{14} \qquad \text{................} (16)$$

$On\ multiplying\ (16)\ by\ \overline{a_{11}}\ both\ sides, we\ get$

$$\overline{a_{11}}.(a_{11}.a_{14}) = \overline{a_{11}}.a_{14}$$

$$\emptyset = \overline{a_{11}}.a_{14} \qquad \text{.............} (16.1)$$

By Venn diagram of equation (15.1) and equation (16.1) this is true only when $a_{11} = a_{14},$ which contradicts with our hypothesis that $Bank_A$ and $Bank_B$ are two different banks. Similarly, Boolean algebra deduces the same conclusion. An example by set theory has also been presented.

By Set theory,

$Let,$

$$A = \{1, 3, 5, 7, 9\}, B = \{2, 3, 5, 6, 8\}\ and\ U = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$Then,$

$$\overline{A} = \{0, 2, 4, 6, 8\}\ and\ \overline{B} = \{0\ ,1\ ,4, 7, 9\}\ AB = \{3, 5\}, \overline{A}B =$$
$$\{2, 6, 8\}\ and\ A\overline{B} = \{1, 7, 9\}$$

Hence both, $\overline{A}B$ and $A\overline{B}$ are non-empty sets whereas they should be empty sets as per equations (15.1) and (16.1).

F. **Time Constraint Policy**

A time constraint policy defines the time or period during which a policy becomes valid. This constraint is defined in each authorization policy. A set of policy is defined as follows to demonstrate the conflicts due to time constraints. A time constraint policy itself does not cause a policy conflict. Policy conflicts can happen only if the time periods specified in various policies with positive and negative authorisations overlap.

Policy $r15$ specifies that subject role $Gold$ can play a movie for 24 hours (i.e. at any time). Policy $r16$ specifies that subject role named $Guest$ cannot play music between 9:00 to 17:00. In general, a temporal logic is best suited to formalize the time constraint policy.

$$Policy\ r15{:}\ Auth + (Gold, movie, play, [00{:}\,00, 24{:}\,00])$$
$$Policy\ r16{:}\ \ Auth - (Guest, music, play, [09{:}\,00, 17{:}\,00])$$

In the FVT approach, the policies are formalised using first order logic as follows:

$Let, I_1, I_2, \ldots, I_n be\ a\ set\ of\ points\ that\ is\ defined\ on\ a\ time\ axis\ T,\quad where\ I_1 < I_2 < \cdots < I_n$

Then, a time constraint for an authorization policy is specified by:

$$period\ [I_a, I_b], a\ \leq b$$
$$Auth +/Auth{-}{:}\ (S_1, T_1, A_1, [I_a, I_b])$$

$Auth + specifies\ that\ during\ the\ time\ period\ [I_a, I_b],$

$the\ subject\ role\ S_1 is\ permitted\ to\ perform\ the\ acion\ A_1 on\ target\ T_1$

$Auth - specifies\ that\ during\ the\ time\ period\ [I_a, I_b],$

$$the\ subject\ role\ S_1 is\ prohibited\ to\ perform\ acion\ A_1 on\ target\ role\ T_1$$

Hence, a conflict caused by a time constraint policy can be detected using the FVT method as follows, where policies $r15$ and $r16$ is defined as:

$$Policy\ r15 : Auth + (S, T, A, [I_1, I_3])$$
$$Policy\ r16 : Auth - (S, T, A, [I_2, I_4])$$

These policies conflict with each other since the time periods $[I_1, I_3]\ and\ [I_2, I_4]$ are overlapping for the same subject role, target role and action. After such an analysis, the FVT method suggests that to eliminate any overlapping time periods from the conflicting policies, it is concluded that *"to resolve the conflict we need to eliminate the overlapping period [I_2, I_3] from the Policies"*.

We have analysed that the solution presented by FVT method does not provide a practical solution for conflict analysis where an administrator is required to eliminate overlapping time periods from each policy defined in the Policy Schema. The composite mapping method proposed in this thesis suggests a more feasible solution.

Policy $r17$ specifies that subject role $Gold$ can play a movie for [00: 00, 24: 00]) i.e. at any time. Policy $r18$ specifies that subject role $Guest$ cannot play music between 9:00 to 17:00.

Policy $r17$ and $r18$ are defined as follows:

$$\begin{aligned}
Policy\ r17:\quad & g_{17} \circ f_{17}(User_{Gold} = S_i : S_6 \leq S_i \leq S_{11}, t_1 = movie, t_1 \in T;\\
& a_{i,\ i+1} = Play\ within\ the\ time\ interval\ I_i to\ I_{i+1},\\
& a_{i,\ i+1} = [I_i,\ I_{i+1}] = [00{:}00, 24{:}00]; a_{i,\ i+1} \in A):\\
Auth + (S, T, A;\ \ & A = reprsents\ action\ play\ within\ time\ interva[I_i,\ I_{i+1}])\\
& = g_{17}(t_1)\\
& = a_{i,\ i+1}\\
& = [I_i,\ I_{i+1}] = [00{:}00, 24{:}00] \qquad \ldots\ldots\ldots\ldots\ (17)
\end{aligned}$$

$where\ t_1\ represent\ movie\ in\ the\ Target\ set\ T;\ a_{i\ ,\ \ i+1}\ reprsents\ Action\ play\ within\ time\ interval\ [I_i,\ I_{i+1}] = [00{:}00, 24{:}00];\ a_{i\ ,i+1} \in A, which\ are\ present\ in\ Action\ set\ A$

$$\begin{aligned}
&Policy\, r18{:}\quad g_{18} \circ f_{18}(User_{Guest} = S_i{:}S_{101} \le S_i \le S_{120},\, t_1 = movie, t_1 \in T;\\
&\qquad a_{s,\ s+1} = Play\ within\ the\ time\ interval\ I_s\ to\ I_{s+1},\\
&\qquad a_{s,\ s+1} = [I_s,\ I_{s+1}] = [09{:}00, 17{:}00]; a_{s,\ s+1} \in A):\\
&Auth - (S, T, A;\ \ A = reprsents\ action\ play\ within\ time\ interva[I_s,\ I_{s+1}])\\
&\qquad = g_{18}(t_1)\\
&\qquad = a_{s,\ s+1}\\
&\qquad = [I_s, I_{s+1}] = [09{:}00, 17{:}00] \qquad \text{................ (18)}
\end{aligned}$$

$where\ t_1 represent\ movie\ in\ the\ Target\ Set\ T;\ a_{s,\ s+1} represents$
$Action\ play\ within\ time\ interval\ [I_s,\ I_{s+1}] = [09{:}00, 17{:}00];\ a_{s,\ s+1} \in A,$
$which\ are\ present\ in\ the\ Action\ set\ A$

Hence, from above, it is evident that $User_{Guest}$ will not be getting any chance to play the movie since all the allotted time for $User_{Guest}$ is overlapping with $User_{Gold}$. Therefore, implementing the solution presented by the FVT method is not practical which requires elimination of any overlapping time periods.

We propose a better solution with options to optimise utilisation of the service/device present in the Target set. For this purpose, we redefine policy $r17$ and policy $r18$ in a generalised way as policy $17A$ and policy $18A$ respectively. Here, negative authorisation ($Auth\ -$) is to be transformed into positive authorisation ($Auth\ +$) by complimenting (i.e. inverting) the time intervals using first-order logic. Various modality conflicts such as ($Auth\ +/\ Auth\ -$), ($Auth\ -/\ Auth\ +$), ($Auth\ -/\ Auth\ -$) can also be analysed.

Firstly, the policies are transformed into ($Auth\ +$) for detecting the conflicts, as follows:

$$\begin{aligned}
&Policy\, r17\ A{:}\quad g_{17} \circ f_{17}(User_{Gold} = S_i{:}S_6 \le S_i \le S_{11}, t_1 = movie, t_1 \in T;\\
&\qquad a_{i,\ i+j} = Play\ within\ the\ time\ interval\ I_i to\ I_{i+j},\\
&\qquad a_{i,\ i+j} = \left[I_i,\ I_{i+j}\right] = [08{:}00, 18{:}00]; a_{i,\ i+j} \in A):\\
&Auth + \left(S, T, A;\ \ A = reprsents\ action\ play\ within\ time\ interva\left[I_i,\ I_{i+j}\right]\right)\\
&\qquad = g_{17}(t_1)\\
&\qquad = a_{i,\ i+j}\\
&\qquad = \left[I_i,\ I_{i+j}\right] = [08{:}00, 18{:}00] \qquad \text{................ (17A)}
\end{aligned}$$

$where\ t_1 represent\ movie\ in\ the\ Target\ Set\ T;\ a_{i,\ \ i+j} represents\ Action\ play\ within\ time\ interval\ [I_i,\ I_{i+j}] = [08:00, 18:00];\ a_{i,\ \ i+j} \in A,$
$which\ are\ present\ in\ the\ Action\ set\ A$

$$\begin{aligned}
&Policy\ r18\ A: \quad g_{18} \circ f_{18}(User_{Guest} = S_i : S_{101} \leq S_i \leq S_{120}, t_1 = movie, t_1 \in T;\\
&\qquad a_{s,\ \ s+k} = Play\ within\ \ time\ interval\ I_s\ to\ I_{s+k},\\
&\qquad a_{s,\ \ s+k} = [I_s,\ I_{s+k}] = [09:00, 17:00]; a_{s,\ \ s+k} \in A):\\
&Auth + (S, T, A;\ \ A = reprsents\ action\ play\ within\ time\ interva[I_s,\ I_{s+k}])\\
&\qquad = g_{18}(t_1)\\
&\qquad = a_{s,\ \ s+k}\\
&\qquad = [I_s,\ I_{s+k}] = [09:00, 17:00] \qquad \ldots\ldots\ldots\ldots\ldots \text{(18A)}
\end{aligned}$$

$where\ t_1 represent\ movie\ in\ the\ Target\ Set\ T;\ a_{s,\ \ s+k} represents\ Action\ play\ within\ time\ interval\ [I_s,\ I_{s+k}] = [09:00, 17:00];\ a_{s,\ \ s+k} \in$
$A, which\ are\ present\ in\ the\ Action\ set\ A$

$Let,$ Gold $user\ is\ authorised\ to\ play\ movie\ for\ a\ time\ interval\ I_i to\ I_{i+j} \quad and$
Guest $user\ is\ authorised\ for\ a\ time\ interval\ I_s\ to\ I_{s+k}$

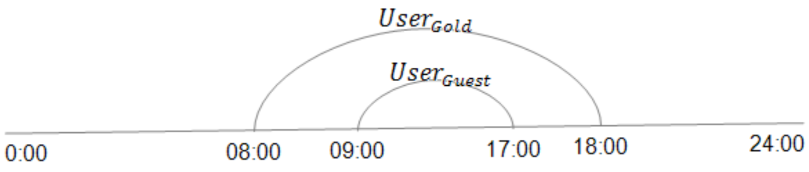
Then, conflicts may occur in four ways:

$Case\ I: when\ i \leq s\ and\ k\ \ \leq j, for\ example,$
Gold $user\ is\ authorised\ from$ 08:00 to
18:00 $hrs\ and$ Guest $user\ is\ authorised\ from$ 09:00 to 17:00 hrs

$As\ per$ policy 17A $Gold\ users\ are\ authorised\ to\ play\ movie\ within\ time\ interval$
$[I_i,\ I_{i+j}] = [00:08, 18:00]$ $and\ as\ per$ policy 18A $Guest\ users\ are\ authorised\ to\ play\ \ movie\ within\ time\ interval\ \ [I_s,\ I_{s+k}] = [09:00, 17:00], where$

$$[I_i, \qquad I_{i+j}] = [08:00, 18:00]$$
$$[I_s, \qquad I_{s+k}] = [09:00, 17:00]$$

$Here, \quad I_i = 08:00, I_s = 09:00\ hours, i\ \leq s\ and$
$I_{i+j} = 18:00\ and\ I_{s+k} = 17:00 \quad and$
$k\ \leq j\ i.e. j = 10, k = 8$



$Case\ II\!: when\ i \geq s\ and\ k\ \leq$
$j, for\ example,$ Gold $user\ is\ authorised\ from$ $10{:}\,00$ to
$19{:}\,00$ $hrs\ and$ Guest $user\ is\ authorised\ from$ $09{:}\,00$ to $17{:}\,00$ hrs

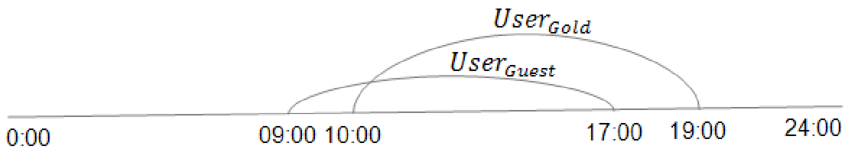
$Policy\ r17\ B\!:\quad g_{17} \circ f_{17}(User_{Gold}\ = S_i\!: S_6 \leq S_i \leq S_{11}, t_1 = movie, t_1 \in T;$
$a_{i,\ \ i+j} = Play\ within\ the\ time\ interval\ I_i to\ I_{i+j},$
$a_{i,\ \ i+j} = \left[I_i,\ I_{i+j}\right] = [10{:}\,00, 19{:}\,00]; a_{i,\ \ i+j} \in A)\!:$
$Auth + \left(S, T, A;\ \ A = reprsents\ action\ play\ within\ time\ interva\left[I_i,\ I_{i+j}\right]\right)$
$= g_{17}(t_1)$
$= a_{i,\ \ i+j}$
$= \left[I_i,\ I_{i+j}\right] = [10{:}\,00, 19{:}\,00]$ …………….(17B)

$Policy\ r18\ B\!:\quad g_{18} \circ f_{18}(User_{Guest}\ = S_i\!: S_{101} \leq S_i \leq S_{120}, t_1 = movie, t_1 \in T;$
$a_{s,\ \ s+k} = Play\ within\ \ time\ interval\ I_s to\ I_{s+k},$
$a_{s,\ \ s+k} = [I_s,\ I_{s+k}] = [09{:}\,00, 17{:}\,00]; a_{s,s+k} \in A)\!:$
$Auth + (S, T, A;\ \ A = reprsents\ action\ play\ within\ time\ interva[I_s,\ I_{s+k}])$
$= g_{18}(t_1)$
$= a_{s,\ \ s+k}$
$= [I_s,\ I_{s+k}] = [09{:}\,00, 17{:}\,00]$ …………….(18B)

$As\ per$ policy 17B $Gold\ users\ are\ authorised\ to\ play\ movie\ within\ time\ interval$ $\left[I_i,\ I_{i+j}\right] =$
$[10{:}\,00, 19{:}\,00]$ $and\ as\ per$ policy 18B $Guest\ users\ are\ authorised\ to\ play$ $movie\ within\ time\ interval\ \ [I_s,\ I_{s+k}] = [09{:}\,00, 17{:}\,00], where$

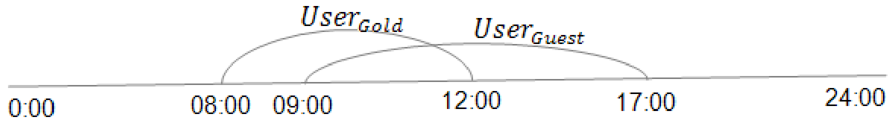
$$\left[I_i,\qquad I_{i+j}\right] = [10{:}\,00, 19{:}\,00]$$
$$[I_s,\qquad I_{s+k}] = [09{:}\,00, 17{:}\,00]$$

$Here, I_i = 10{:}\,00, I_s = 09{:}\,00\ hours, i\ \ \geq s\ and\ I_{i+j} = 19{:}\,00\ and\ I_{s+k} = 17{:}\,00$
$and\ k\ \ \leq j\ i.e. j = 9, k = 8$

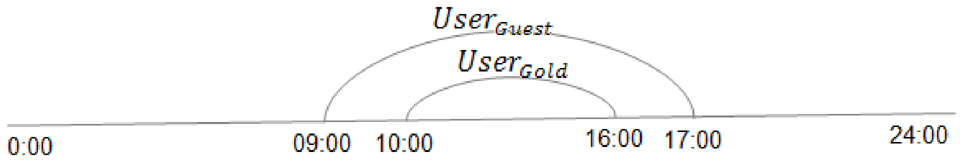


Similarly we can define case III and case IV.

Case III: when i≤ s and k ≥ j, for example, Gold user is authorized from 08:00 to 12:00 hrs. and Guest is authorized from 09:00 to 17:00 hrs.



Case IV: when i≥ s and k ≥ j, for example, Gold user is authorized from 10:00 to 16:00 hrs and Guest is authorized from 09:00 to 17:00 hrs.



Summary of the Cases

1. When i≤ s and k ≤ j, for example, and Gold user is authorized from 08:00 to 18:00 hrs and Guest is authorized from 09:00 to 17:00 hrs.
2. When i ≥ s and k ≤ j, for example, and Gold user is authorized from 10:00 to 19:00 hrs and Guest is authorized from 09:00 to 17:00 hrs.
3. When i≤ s and k ≥ j, for example, Gold user is authorized from 08:00 to 12:00 hrs and Guest is authorized from 09:00 to 17:00 hrs.
4. When i≥ s and k ≥ j, for example, Gold user is authorized from 10:00 to 16:00 hrs and Guest is authorized from 09:00 to 17:00 hrs.

5.6.4 Analysis of Proposed Conflict Detection Approach

It is possible to enumerate all policies derived implicitly by propagation and action composition policies and then to detect an implicit conflict by comparing the original and derived policies. However, this would be computationally expensive and it is still hard to identify the original policies that cause any conflict. We have proposed a novel conflict detection approach using composite mapping based on our proposed Policy Schema.

Our proposed method can statically detect conflicts typically arising from explicit positive and negative authorisations, propagation and action composition policies referring to the sets of subject, target or action. The traditional rule-based access control policies only operate at the network layer, whereas our approach supports more sophisticated application-oriented access control policies defined in terms of subject and target role structures. In addition, policy conflicts with application constraints such as requirements for the Separation of Duty, Chinese Wall and time constraints are also detected. Event based policies to analyse conflicts based on a user's current context is also supported for dynamic conflict detection.

The proposed approach is based on first-order logic for representing policies, the same approach as the FVT method, and is a complete theorem prover and infers the cause of the conflict using Boolean logic and Set theory. We have also presented analysis of different case scenarios where it is evident that the time constrained policies can produce conflict with the overlapping subject, target and action. The FVT method concluded that to resolve this conflict, the overlapping period from the policies needs to be eliminated. This may not be a practical solution in a large and dynamic environment. The proposed composite mapping method offers greater flexibility in terms of detecting conflicts in comparison with the FVT method with options to optimise utilization of the service/device present in the Target set.

5.7 Conflict Resolution

After the policy conflicts have been identified, resolution of those policies is equally important. The resolution of policies usually requires a manual/human intervention. We have identified the following conflict resolution techniques to harmonise policy conflicts that may occur in a PBM system:

i. **Prioritisation**

   Prioritisation among policies has emerged as a basic and a natural mechanism for resolving policy conflicts. Generally, priorities are assigned to rules that have the potential for conflicts and when conflict occurs, it is resolved by choosing policies whose priority satisfy some criterion. However, defining numeric priority has scalability problems [150]. Priorities can be applied on sets of hierarchies and associated priorities assigned to respective sets.

ii. **Time Period/Validity Period**

In the proposed conflict detection technique based on composite mapping, we have identified a better approach for detecting any time period/validity period conflict. Hence, a time constraint which specifies the period during which an authorisation policy is valid is a realistic approach for conflict resolution.

iii. **Default Policies**

There may be some default policies which are applied consistently to the overall PBM system. Default policies are specialised cases of prioritisation, where pre-defined priorities can be applied to conflicting policies. For example:

(i) *prohibition holds precedence over authorisation* i.e. if a policy prohibits a role to perform an action while another policy authorises the same action, then prohibition takes precedence and the role is denied to perform the action.

(ii) *role hierarchy holds precedence over policy* i.e. if a conflict occur between users who have different roles, then a user with a higher role can override the policy that belongs to the user with a lower role;

(iii) *context holds precedence over user* i.e. if there is a conflict between a user and a location, then location takes precedence. For example, an engineer is allowed to access software code from a development server, but will be denied access if connected from outside office premises.

5.8 Analysis of Semantic layer

Policies derive the behaviour of each component in a PBM system. The decision as to whether a user can access a service is based on policies which must be flexible enough to cope with the changes in the network topology and interactions between multiple service providers involved in offering ABC services. Moreover, fine-grained access control is needed as the number of offered services and users grow. Hence, presence of any inherent ambiguity within policies may lead the system to unpredictable or unknown states and errors where authorisation of users to access services may be compromised.

The IETF Network Working Group has identified policy conflict detection as a *must* requirement for any PBM system and requires implementing a suitable conflict detection method. We have also identified conflict detection as a characteristic requirement in our proposed PBM system. Furthermore, since conflicts cannot be prevented in policy systems, they need to be detected and resolved to ensure that the system behaves deterministically and correctly. The primary goal of detecting a conflict is to investigate possible sources of conflicts and conflict types that may occur within the system. By knowing that there is a potential conflict would allow the system to accommodate the conflict resolution earlier. Hence, by the time it occurs, the system will already have resolved the conflict.

In the previous sections, we have observed that different semantic analysis approaches have been proposed by various researchers. Policy specifications vary in expressiveness and complexity. Some policy specification languages have mechanisms to detect and resolve some kinds of conflicts, and others leave the task of conflict handling to additional tools. As implied by Strassner, the characteristics of the system being managed must be represented in as much detail as required. Then, policies can be defined to govern each state of the managed object.

The proposed Semantic layer introduces a formal discipline, where policies are understood and analysed. We have defined the role of the Semantic layer as: *define a semantic notation to provide a common ontology/formal methodology to understand the contents of the policies and to detect and resolve conflicts*. The semantically analysed policies are not needed be in an implementable state, though they are required to be in a consistent state and be unambiguous. For example, while using rule-based approaches, which provide a concise *instruction* which network elements can understand and operate directly, may not require the Syntax layer. However, if an organisation prefers to employ Ontology-based approaches, then the Syntax layer is used for policy distribution.

Following the modular approach of the proposed PBM framework, a Semantic layer can employ any of the semantic analysis approaches based on the preferences of the administrator and the requirements imposed by the particular needs of the network/organisation. Furthermore, since different types of policies are required for different management areas, different solutions and/or methods can be employed for conflict detection (and resolution) even within a single PBM system. There are two main requirements identified for choosing any conflict detection technique. Firstly, the characteristics of the system being managed must be understood and represented in policies in as much detail as required. Secondly, the Semantic layer should be able to read and write policies from the Policy Schema without changing the structure (i.e. entities involved and their relationship) of the policies. The sole purpose of the Semantic layer is to detect conflicts and provide conflict resolution methods which may or may not require changes to the Policy Schema.

Chapter 6
Syntax Layer

Policies are defined as rules, governing the allocation of network resources to certain applications and users. Once these high-level policies have been defined and semantically analysed to ensure for integrity, they need to be distributed to various network devices to be interpreted correctly and enforced.

In the IETF PBNM model, the task of interpretation and distribution of policies is performed by the PDP while enforcement of these policies is done by the PEP. Hence, the role of PDP augments as to interpret (i.e. understand the high-level policies and possibly detect any conflicts) and to distribute the policies to PEPs. This two-fold responsibility of PDP adds complexities in the definition of the PDP. On one side, PDP needs to decide on consistent policies to achieve desired business goals, while on the other side, it needs to represent the policies in a structural format so as to be distributed to "less-intelligent" PEPs for enforcement.

In our proposed model, we have delegated these two major responsibilities for the PDP to two separate layers, i.e. Semantic layer and Syntax layer, where each layer is responsible for policy interpretation and policy distribution respectively.

The Syntax layer provides a modular approach to support a consistent format for the distribution of policies and focuses on formal policy structure and policy negotiations between intra- and inter-domains. This chapter discusses the proposed Syntax layer and different approaches considered for policy distributions.

6.1 Introduction

While identifying the requirements of a management framework, we highlighted that for defining policies for distributed, large enterprises, two important issues must be considered: (i) the model (or language) adopted for representing policies, (ii) the approach adopted for interpreting, distributing and enforcing the policies.

Several extensions to traditional approaches have been proposed by various working groups (e.g. SNMPConf, COPS), which could provide necessary capabilities required for the distribution of policies and monitoring of current state of network, as an integrated approach to the management of PBNM framework. On the other hand, the standardisations and continuous advancements in the development of XML-based languages have opened a new paradigm for distributing policies which includes interoperability and mutual agreeable features. The use of XML is primarily motivated by the vast heterogeneity of entities in a distributed environment, though these languages have limited semantic expressability (i.e. unable to detect inherent conflicts).

In the following sections, we will closely look at the traditional approaches standardised by the IETF working groups and newer approaches of XML-based languages from our perspective for providing ABC services.

6.2 IETF Proposal

In the definition of the IETF PBNM model, PEP acts as a component of a network node (e.g. a router, switch, or hub) where the policy decisions are actually enforced and the PDP is where the network policy decisions are made. The PDP make policy decisions using information retrieved from the policy repositories. Since PEPs can potentially be from multiple vendors, a common policy language and a communication protocol is needed to support the distribution of policy information to these devices. Examples of such communication protocols, based on request/reply format, are Common Open Policy Service (COPS) [36] protocol and Simple Network Management Protocol (SNMP) [151]. In the following section we will focus on the COPS protocol while the SNMP protocol is discussed in section 7.3.1.

6.2.1 COPS

The COPS protocol is an attempt by the IETF to standardize the communication between a PDP and PEPs. The Resource Allocation Protocol (RAP) working group, who developed COPS, identified its purpose as to "establish a scalable policy control model for RSVP Protocol" [152] . However, COPS soon received significant attention from research groups, within and the outside IETF. RAP also defined the COPS Usage for Policy Provisioning (COPR-PR) as "a scalable policy control model for DiffServ policy provisioning" [153]. COPS was also utilised in several other management areas such as accounting [154], IP filtering [155], security [156] and load balancing [157].

RFC 3198 [25] defined COPS as a query and response TCP-based protocol that can be used to exchange policy information between a PDP and its client PEPs. The base model is defined as:

i. The *outsourcing model* (Figure 6-1) [152] assumes that there is a signalling request from the managed device that must be authorised based on a policy criteria. Signalling requests are typically associated with an end-to-end signalling protocol (such as RSVP). The outsourcing model is also referred to as *pull* or *reactive* mode, as the managed device pulls decisions from the PBNM system, and the PBNM system reacts to those events.

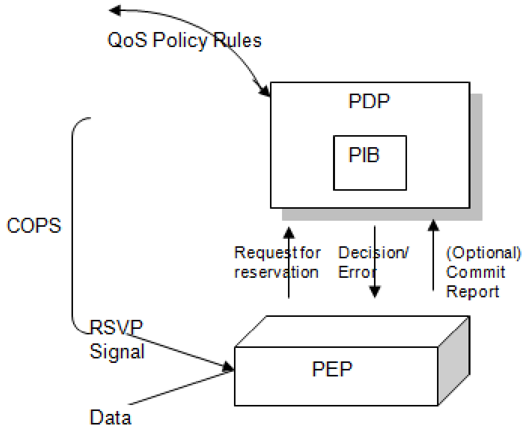


Figure 6-1 PBNM Outsourcing model

A sample scenario could be:

- PEP on a RSVP router receives a signal to reserve bandwidth for a multimedia application at a specified time for approved users.
- PEP then outsources the decision to the PDP to decide whether the request is valid.

- PDP returns the "approve" decision message and PEP enforces the policy.
- PDP maintains the state and receives a report from the PEP.

ii. The *provisioning model* (Figure 6-2) [153] is almost the reverse of the *outsourcing model*, where the PBNM system predicts future configuration needs and proactively pre-provisions new rules to the managed device. In spite of responding to the device requests, a PBNM system prepares and *pushes* configuration to the devices, as a result of an external event, such as a change of applicable policy, time of day, expiration of account quota, or as a result of a third party signalling. The provisioning mode is most commonly used for controlling network policy for non-signalled protocols, such as DiffServ, or configuring devices for particular services (such as VPNs or VoIP).

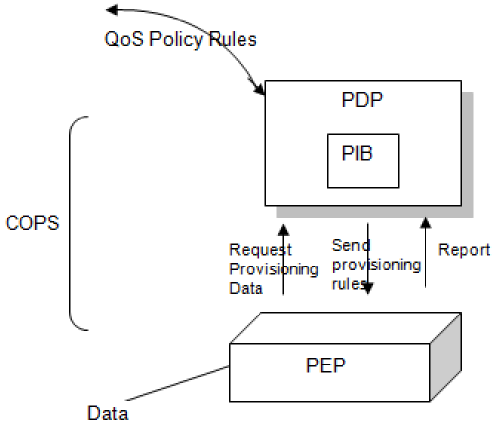


Figure 6-2 PBNM Provisioning model

A sample scenario could be:

- A network manager enters a new policy for a multimedia application usage or a PEP on a network node requests provisioning data that has not resulted from QoS signalling.
- PDP processes and translates the policy into rules understood by the PEP.
- PDP then sends the provisioning rules.
- PEP makes required configuration changes and updates the rules.

The provisioning approach is generic and can be explored in several management domains. The IETF has explored the provisioning approach for distributing *DiffServ* configuration [158] and *IPSec* configuration [35]. Other potential target domains for future standardisation are Multiprotocol Label Switching (MPLS), Access Control and 3GPP UMTS [159].

6.2.2 **Policy Information Base**

To deploy policies across a network, a management system requires a generic format to represent the information contained in the policy and a protocol to carry the information. The IETF has defined Policy Information Base (PIB) [93] to represent policy information, and COPS protocol to carry the information. A PIB is a data definition which allows a PEP to inform its capabilities to the PDP and carry policy configuration information to PEPs. A PEP can also send reports (e.g. status, errors) to the PDP which can update its policies as required. The information in the PIB must be compatible with the policy schema.

Generally speaking, a PIB is a local database for policy information. It uses a hierarchical tree-like structure where the branches of the tree represent structures of data or Provisioning Classes (PRCs), while the leaves represent various instantiations of Provisioning Instances (PRIs). A PRC can be used to both *notify* and receive *install* information to and from PDP. A PRC can also *install-notify*, hence providing bidirectional information exchange between a PEP and a PDP [160]. PRCs and PRIs are uniquely identified by PRI Identifiers (PRIDs). A PRID in a COPS-PR message provides all the information necessary to the PEP to decode and process an entire policy rule in the message (Figure 6-3).

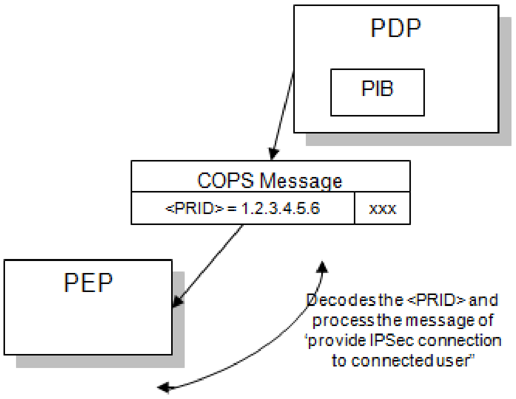

PDP
PIB
COPS Message
<PRID> = 1.2.3.4.5.6
xxx
PEP
Decodes the <PRID> and process the message of 'provide IPSec connection to connected user"

Figure 6-3 PEP processing of PRID

PRIDs have a hierarchical structure of the form 1.3.4.2.7, where the first part identifies the PRC (i.e. 1.3.4) and the last part identifies the instance (i.e. 2.7).

Hence, a *DropAnonymousWebsitePacket* policy: "Drop all packets coming from anonymous website", can be defined as "If source IP address == 202.132.89.92 then drop packet" and is represented in PIB as in Figure 6-4:

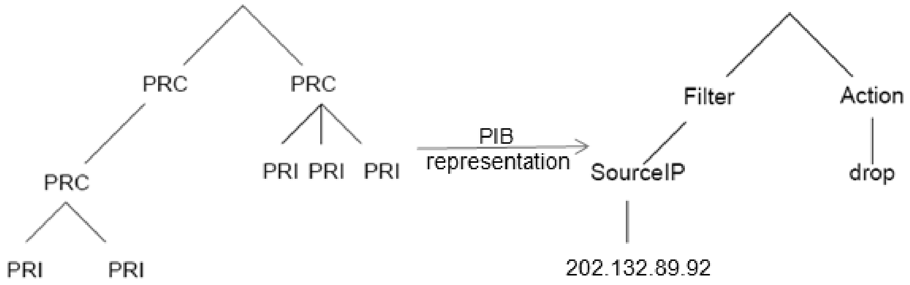


Figure 6-4 Policy Information Base tree structure

The hierarchical structure of a PRID allows new policy rule classes to be implemented by extending PRC branches and defining new PRIs. Hence, to support any new policy rule no changes to the COPS protocol or any modifications to previously defined PIB variables are required. This allows organisations to introduce new services and policy rules without altering previously defined policy rules. For example, a new policy rule for VoIP services may be introduced by extending the PIB with a unique PRID to a managed environment already supporting QoS and IPSec services.

6.2.3 **Analysis of IETF Approach**

The original specification of the COPS protocol was intended for use within the context of admission control with signalling protocols, such as RSVP, and was not designed to distribute policies from a central site to multiple routers. However, various modifications were proposed to allow a PDP to send configuration and provisioning information to routers of other disciplines, such as DiffServ [161] and IPSec [162]. The types of messages that are exchanged remains the same as in the original specifications of the COPS protocol, though, a new set of objects were defined to carry the provisioning information related to new services back to the routers.

The main characteristics of the COPS protocol are:

- The protocol employs a client/server model in which a PEP send requests, updates, and deletes to the remote PDP. The PDP returns decision back to the PEP.
- Individual PEP clients are required to initiate a handshake with their PDP prior to the exchange of any policy data (Figure 6-5). This is an improvement over traditional network management systems, where the server deploys configuration data by initiating communication with the clients.

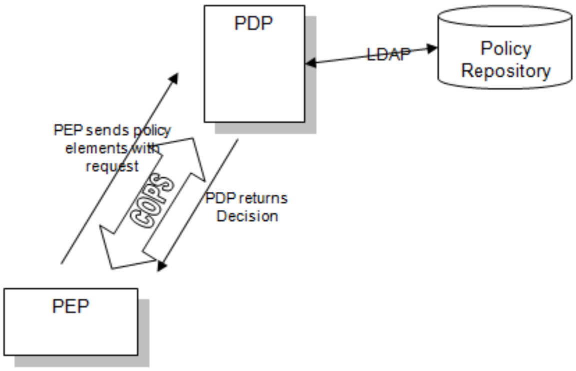


Figure 6-5 COPS base model

- The PEP uses (reliable) TCP connections to send requests to and receive decisions from the remote PDP, as compared to traditional network management protocols which usually operate over an (unreliable) UDP connection. Hence, COPS has increased the reliability and responsiveness of communication between clients and servers and establishes a ‘stateful’ request/decision exchange.

- *Fail-over* can be achieved by deploying multiple PDPs. For example, a PEP (router) could establish a connection to an alternative PDP (alternative network monitoring server) if the current PDP crashes.
- COPS protocol provides a secure connection by employing authentication, replay protection and message integrity. COPS can also use existing protocols such as IPSec to authenticate and secure the communication channel between PEPs and a PDP.

In the previous discussions, we have outlined that a PIB provides unique flexibility and extensibility in managing provisioning policies in the COPS-PR model. New policy-rule classes can be implemented by simply extending the PIB and no changes to the COPS protocol are necessary, as well as no modifications to previously defined PIB variables are required. However, the COPS protocol does not provide policy negotiation, while interoperability is harder to achieve as most PIB implementations are proprietary based with vendor supporting only specific features of their network devices [163].

On the other side of the spectrum, Web Services [92] are becoming a de facto for managing access control in a request/response format. Although limited in semantic expressability, web services allow a service provider to publish its policies and to inform roaming mobile users regarding the offered services. Other benefits include determining mutual agreeable policies between service providers or users, and interoperability [164]. These approaches are mainly used for access control and authorisation purposes; though other forms of policy transfer is also possible.

In our proposed model for network management, the semantic analysis of policies is already performed at the Semantic layer and the sole purpose of the Syntax layer is to map the abstract policies to an easily distributable format. The distribution of policies needs to be interoperable with different vendor implementations. Furthermore, most of the policies in a network management environment (as in our case) are based on access control [165]. These characteristics of XML-based languages make them an ideal candidate for consideration in the proposed Syntax layer.

6.3 XML

The use of Extensible Markup Language (XML) is motivated by vast heterogeneity of entities in a distributed environment. Various functional units connected via multiple access technologies, within an environment and among different service providers, need to be linked together to provide ABC services. Hence, an *interoperable* mechanism to efficiently express and enforce the policies is required.

Using XML as a language for expressing policies has several advantages [166]. XML is ideal for transferring information between heterogeneous platforms as XML parsers are available for many platforms. Furthermore, XML policy documents can be structurally validated using a XML policy schema. This is possible because XML documents can reference to their schema, instead of carrying the schema itself. This structural validation *normalises* the policy into different tokens based on the Policy Schema (discussed in section 4.1)

However, using XML as a standard for policy expression has some limitations. XML can define a structure of the policies and its semantic validity is mostly implicit-meaning and conveyed on the basis of a shared understanding derived from human consensus. Implicit semantics are ambiguous and promote fragmentation into incompatible representation variations, and hence require extra (manual) work that a richer representation could eliminate. Though, if an implementation requires the use of XML approach for policy distribution, Semantic Web-based policy representations may be used by applying contextual information [22].

XML-based access control policies and Web Services are produced by statically or dynamically integrating independent web systems using a set of XML standards such as Simple Object Access Protocol (SOAP) [167], Universal Description, Discovery and Integration (UDDI) [168] and Web Services Description Language (WSDL) [169]. Additionally, by using standard policy description languages such as WS-Policy [170], Web Services Policy Language (WSPL) [171] and eXtensible Access

Control Markup Language (XACML) [94], it is possible to realise extensive access control for services offered by the network. This enables advanced and sophisticated services to be provided enabling users to perform several procedures simultaneously, resulting in a better overall service [172].

The following section takes a closer look at different policy description languages. Our key focus is to consider the generic nature of XACML to represent different types of policies (including access control) and its negotiating capabilities.

6.3.1 **XACML**

XACML is a declarative access control policy language, designed around and written in XML, to encode data exchanges to express and enforce access control policies in a variety of environments. It is an Organisation for the Advancement of Structured Information Standards (OASIS) standard specification and a replacement for IBM's XML Access Control Language (XACL) [173].

XACML presents a complete solution for modelling, storing and distributing access control policies and adopts a generic access control model, based on the concepts of policies, rules and targets. A XACML Target is a triple formed by subject, resource (object, in our case) and action. Targets are used for selecting policies which must be considered to evaluate a decision request and, for determining if a request is permitted or denied.

XACML also provides a XML schema for a general policy language, which is used to protect the resources/services and make access decisions over these resources, together with a processing environment model to manage the policies and to conclude the access decisions. XACML Context specifies a request/response protocol that an application environment can use to communicate with the decision point entity.

6.3.1.1 **XACML Processing Environment**

The XACML profile specifies five main entities to handle policy decisions:

a. Policy Enforcement Point (PEP): PEP receives the access requests and evaluates them with the help of the other actors and permits or denies access to the resource.

b. Policy Administration Point (PAP): PAP is the repository for the policies and provides policies to the Policy Decision Point.

c. Policy Decision Point (PDP): PDP collects all the necessary information from different components and concludes a decision.

d. Context Handler: sends a decision request to the PDP and translates PDP response context to the native response format of the PEP.

e. Policy Information Point (PIP): PIP is where the necessary attributes for the policy evaluation are retrieved from several external or internal components. These attributes can be retrieved from the resource to be accessed together with an environment (e.g. time), a subject, and so forth.

Figure 6-6 [137] illustrate these components and an information flow. An access request is received at a PEP which then sends it to the Context Handler. The Context Handler maps the request and the attributes to the XACML Request context and sends the request to the PDP. While evaluating the request, if the PDP require some attributes, it sends the attribute queries to the Context Handler. The Context Handler then collects these attributes with the help of the PIP from the resources, subject and the environment. After evaluation, the PDP sends the XACML Response to the PEP via a Context Handler. The PEP then applies the authorisation decisions.

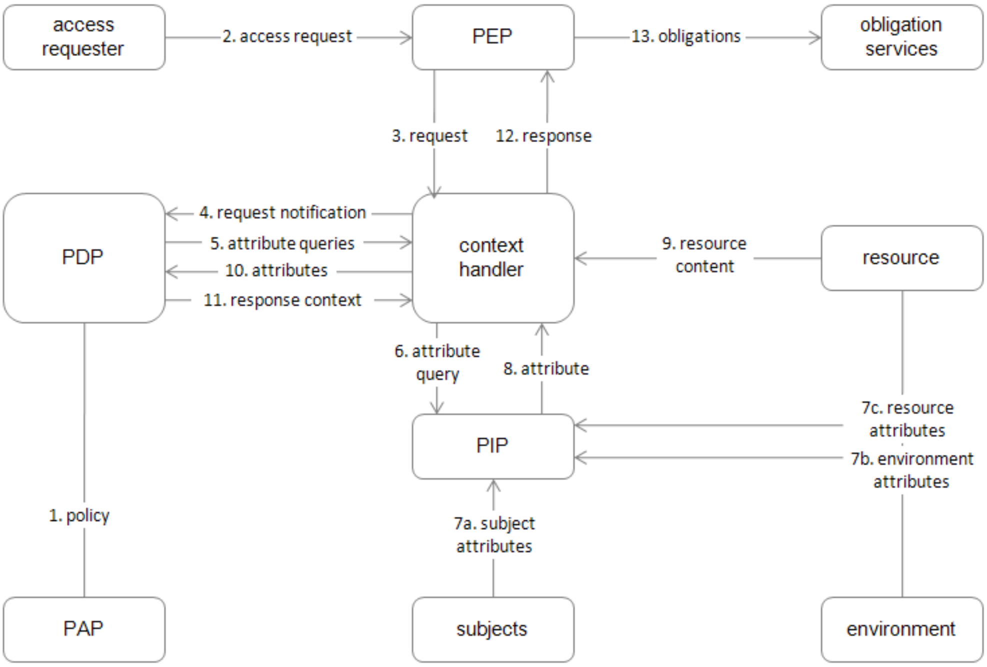


Figure 6-6
XACML Environment - Similar to IETF PBNM framework

6.3.1.2 **XACML Model**

The language model is composed of Rules, Policies and PolicySets, and are defined as follows:

(i) A *Rule* element is the basic element of the policy. It defines the target elements to which the rule is applied and defines conditions to apply the rule. It has three components, namely target, effect, and condition. The target element defines the resources, subjects, actions and the environment to which the rule is applied. Hence, if a *GoldWeekdayVoIPAccess* policy states that "if a gold user is accessing VoIP services during the weekdays, initiate SIP session", then a XACML normalisation will identify *VoIP service* as a resource target data element, *initiating SIP sessions* as a target action and *weekdays* as an environment.

GoldWeekdayVoIPAccess policy

i. High level

If a Gold user is accessing VoIP services during weekdays, initiate SIP session

ii. XAMCL Normalisation

Target → VoIP Service
Action → Initiate SIP session
Environment → Weekdays

(ii) *XACML Policies* are a set of rules which are combined with some algorithms. The algorithms can be "Permit Override" which states (i) allow the policy to evaluate "Permit" if any rule in the policy evaluates to be "Permit" (ii) if a rule evaluates to "Deny" and all other policy evaluations are "Not Applicable" then the result is "Deny", (iii) if the rules evaluates to "Not Applicable" then the policy is "Not Applicable."

(iii) *PolicySet* is a set of policies derived from a policy-combining algorithm as in the policy. PolicySet also defines two components: *target* and *obligations*. Target indicates where the policy rule is applied based on the subjects, resources, actions, and environment. For example, if a policy's target is a user role, it indicates that the policy restricts the access rights related with that user role. An Obligation indicates a requirement of actions to be performed when either a role is assumed or based on some attributes such as time. For example, *UserPasswordChange* policy can require "every user to change their passwords at least once a month", or *RemoteMonthlyBackup* policy requires "backup operators to backup entire system image in remote location every first day of the month".

Hence, it is clear that the management framework of XACML is similar to the IETF PBNM framework making it feasible for deploying XACML in policy-managed frameworks. The XACML also provides a *profile* which manages optional features like privacy, authorisation, security and authentication. Hence, this profile can be used as WS-Security, WS-Trust, and provides a binding of elements of WSDL and SOAP in order to facilitate distribution by these standards. Furthermore, the request and response context used by the PDP are also defined in the standard allowing organisations and network operators to use other representations, such as, Security Assertion Markup Language (SAML) [53].

6.3.2 **Other Languages**

Many Web Services standards are emerging to make web services secure and privacy protected. They can be deployed as access control policies to verify the users before they are provided access to the network services. Web Services also allow service providers to advertise their services so that roaming mobile entities can gain knowledge about the services offered. This allows mobile entities to able to make informed decisions for best available offerings, hence fulfilling the basic idea of ABC services. The following section enlists some of the XML-based languages used in specific areas:

a. **P3P**

The Platform for Privacy Preference Project (P3P) [139] enable network operators to express their privacy practises in a standard format that can be retrieved automatically and interpreted easily by user agents. The P3P user agents allow users to be informed of service practices (especially for access control to resources offered from a web-based portal) in both machine- and human-readable formats. The P3P also employs procedures to automate decision-making based on these practices when appropriate. A data-centric relation can be defined to express relational semantics of P3P where semantic analysis of P3P policies is performed using the Resource Description Framework (RDF) and modelled as a relational database [174].

b. **WS-Policy**

WS-Policy [175] provides a general-purpose model and syntax to describe and communicate the policies of a service through Web Services. It specifies a base set of constructs that can be used and extended to other Web Service specifications to describe a broad range of service requirements and capabilities. WS-Policy also introduces a simple and extensible grammar for expressing policies and a processing model to interpret them. The policy assertions are expressed using XML and the grammar itself is specified with XML Schema.

By using Ontology Web Language (OWL), the expressiveness of the WS-Policy representation can be broadened which simplifies the interaction between WS-Policy and any newly introduced protocols [176]. The OWL representation allows a semantic analysis of the policies while it does not require focusing on their implementation (discussed in section 5.5).

c. **SAML**

SAML is a XML-based framework for communicating user authentication, entitlement and attribute information. SAML handles user authentication and carries attribute information for authorisation (access control). For example, once the identity of a user is confirmed by the system using SAML, the access rights of the user can be identified by using XACML mechanisms.

d. **WSPL**

Web Services Policy Language [177], developed by Sun Microsystems, is suitable for specifying a wide range of policies, including authorisation, QoS, quality-of-protection, reliable messaging, privacy and application-specific service options. WSPL is interesting in several aspects as a Syntax layer negotiating protocol. It supports merging two policies, resulting in a single policy that satisfies the requirements of both policies, assuming such a policy exists. Policies can be based on comparisons other than equality, allowing policies to depend on fine grained attributes such as time of day, cost, or network subnet address. By using standard data types and functions for expressing policy parameters, a standard policy engine can support any policy. The syntax of WSPL is a strict subset of the XACML standards and is under consideration as a standard policy language for use in access control through web-based admission [178].

6.4 Analysis of different approaches

As discussed in section 6.3.1.1, the OASIS XACML proposal also follows the IETF PBNM PDP/PEP approach. However, OASIS does not make a distinction between outsourcing and provisioning models, neither defines a standard protocol for supporting the communication. An analysis of the XACML indicates that it was primarily conceived for supporting the outsourcing approach [179].

As an important difference between the approaches adopted by OASIS and IETF relate as to how the policies are represented and stored. OASIS proposes XACML as a particular model for access-control and policies are represented and stored as XML documents. On the other hand, IETF defines PCIM as a generic model, independent from the way the policies are represented and stored. However, the PCIM model is abstract, and needs to be extended in order to support particular areas of management, such as QoS [1].

Furthermore, the policies defined in XACML format need to be in terms of a <PolicySet> and not in terms of a single <Policy> element as discussed in PIB (section 6.2.2). Since from XACML version 1.1, the <Obligations> element is mapped to <Policy> or <PolicySet>, but it cannot be mapped to Rules i.e. all <Rules> in a policy define the same <Obligations>. Therefore, distinct services cannot be represented in a single policy. Additionally, even though the representation of policies in an IETF PIB is proposed to be performed by a PCIM information model, however, the mere representation loses the conflict detection. Further, the IETF model does not provide interoperability and negotiation features.

Hence, it is clear that XACML and COPS-PIB cannot be tightly integrated nor are they adequate to be solely used in policy-managed network offering ABC services in a heterogeneous environment. However, both approaches have key advantages in specific areas and their applicability in different scenarios, such as for intra- and inter-domain policy distribution is a feasible option.

6.5 Our Approach

We have identified the primary purpose of the Syntax layer as to provide:

a. A distribution bridge from semantically correct policies to distributable policies which can be translated by the Enforcement layer to be implemented in network devices,

b. An interoperable channel for distribution of policies with different vendor implementations, and

c. A framework where policy negotiations is possible. For providing ABC services, it is important for the service providers to exchange information regarding the class of service which can be provided to the roaming mobile entity. In case the network is not capable of offering such services, it needs to renegotiate the respective Service Level Specifications (SLS)[1].

We have analysed that both COPS protocol and XML-based languages (XACML, in our case) has advantages and limitations. The COPS protocol is well supported in the network management paradigm together with extensibility in the PIB model. For each new discipline for which device provisioning is needed, a new PIB can be defined. Furthermore, the COPS protocol is a fail-over and stateful protocol but it does not provide interoperability and does not cater for roaming mobile users dynamic needs (even though the COPS outsourcing model supports RSVP-type requests, however, it requires a COPS-PIB implementation in mobile nodes).

The XML-based languages provide access control policies to authenticate and authorise users before any network resources are allocated. Furthermore, being XML based, these languages support interoperability (a reference to the Schema is required) and if the services are deployed as web services, then mobile users may send dynamic requests to use specific services, hence, catering for runtime requirements of mobile users. Furthermore, they offer policy negotiation.

This provides a unique opportunity to utilise traditionally employed COPS-type protocols for intra-domain management with feedback and monitoring elements, while XACML-type languages can be used to distribute policies for inter-domain management and service negotiations between service providers. Any roaming users' runtime requests are categorised as an inter-domain policy request. Hence, our proposed Syntax layer supports interoperability and policy negotiations, while considering run-time requirements of mobile users to provide ABC services.

[1] A Service Level Agreement (SLA) is defined as a service contract between a service provider and their customer (including other service providers) that defines providers responsibilities in terms of network levels and times of availability, method of measurement, consequences if service levels are not met. The Service Level Specification (SLS) is a subset of an SLA that describes the operational characteristics of the SLA. SLS may consist of expected throughput, drop probability, latency, constraints on the traffic and traffic profiles 180. SLS, "Service Level Specification, The QoS Forum," Cited Aug 2008; http://www.qosforum.com/.

Both the proposals for extending the COPS-PIB [181] and deployment of Web Services to cater for run time requirements of a mobile user have been realised in our proof-of-concept implementation of policy-managed network and policy-managed mobile client, discussed in sections 8.3 and 9.5 respectively.

6.5.1 **Analysis of Syntax Layer**

The Syntax layer provides a generic framework for representing the policies for distribution in a distributed heterogeneous environment.

While defining the characteristics of our proposed model in section 3.3, we have identified that to offer ABC services, a management framework must exhibit characteristics such as: modular, flexible, adaptable, clustering, scalable, conflict detection and interoperable. Even though these characteristics were defined in providing ABC services, they are equally valid for any distributed network. Two of the shortcomings of the IETF PBNM model related to distribution of policies are scalability and interoperability.

In our proposed model, we have solved these concerns by separating the conflict detection and policy distribution processes into separate layers. This provides two advantages: firstly, at the Semantic layer, conflict detection can be performed solely as an independent process and best methods suitable to the organisation may be employed. Similarly, the Syntax layer allows syntactic representation of policies and is only concerned with distributing the policies for enforcement. Hence, we have *extended* the IETF PBNM model by representing the PDP as a set of two components, i.e. Semantic layer and Syntax layer.

Secondly, based on our modular approach where any preferred method may be applied for semantic analysis of policies, the same applies to the Syntax layer where different approaches may be employed for the policy distribution. Organisations are able to choose a best policy distribution mechanism, such as using the COPS protocol for intra-domain management and, XACML for inter-domain management and service negotiations among service providers.

Organisations may follow conventional COPS or SNMP protocol or use XML-based languages (e.g. XACML, WSPL, SAML, etc.) depending on their particular requirements. Specially designed policy languages can have significant usability advantages over generic rule specification languages, but the ultimate choice should be driven by careful consideration of the target user groups, current network infrastructure support and their projected preferences.

Chapter 7

Enforcement Layer

The main idea behind any network management framework is to simplify network operations and network provisioning. We have discussed that this simplification can be achieved by providing a high-level abstraction to network administrators to provide them with a business-level view of the entire network rather than them managing individual devices.

As discussed in previous chapters, these abstractions can be translated to different representations for semantic and syntax analysis. It is also necessary to ensure that the policies are in harmony with one another. However, eventually, these policies need to be implemented by network components, and for the proper management of network, need to be translated to a format understandable by individual networking elements.

In our management framework, the Enforcement layer is proposed for this purpose. The Enforcement layer translates polices from Syntax notation to configuration instructions which can then be enforced directly by the network devices. This chapter presents a brief overview of the common approaches that have been proposed for the enforcement of the policies.

7.1 Introduction

The need of management protocols for device management from a centralised location has been required by the networking community from the time networks emerged. Many existing solutions have been designed specifically to address this concern. These solutions are usually proprietary and are referred to as “management frameworks”. Even though these management frameworks vary from vendor to vendor, they share many common features, such as; each framework is typically deployed in an environment with a management console and several agents. An agent is located on each of the devices that are being managed and is responsible for enforcing the commands from the management console together with sending responses (such as reports, alarms, and alerts) back to the management console.

7.2 Conventional Approaches

There are different approaches to managing networks. Conventionally, Command Line Interfaces (CLI), scripts and code generators are used to configure devices. In these cases, a network operator manually inputs commands for different network devices, assuming the knowledge of scripts and command lines. However, as the complexity of networks and the number of network devices from different vendors grew, more formal approaches were needed. The following sections present an overview of some of the conventional approaches, followed by some of the formal approaches.

7.2.1 Command Line Script

One of the most conventional approaches to configure a network element (e.g. router) is for a network operator to write a set of commands using a particular CLI of the network element that can perform an action required to implement the policy.

For example, Cisco’s Modular QoS CLI [182] is used to enforce QoS policies based on traffic class. The steps required are: (i) defining a traffic class with a *class-map* (ii) using a *policy-map* to create a service policy by associating the traffic class with one or more QoS policies, and (iii) attaching the service policy to the interface with the *service-policy* command.

The *class-map* command is used to define a traffic class. A traffic class contains three major elements: a name, a series of *match* commands, and an instruction on how to evaluate these match commands. The syntax of a class-map command is:

```
class-map [match-any | match-all] class-name
no class-map [match-any | match-all] class-name
```

To configure a service policy, a policy-map command is used to specify the service policy name and configuration commands used by class-map, together with one or more QoS policies. The syntax of policy-map command is:

```
policy-map policy-name
no policy-map policy-name
```

Hence, for a policy to configure a router to offer a minimum guaranteed bandwidth for VoIP services can be defined as: “Allot bandwidth to respective class traffic as Voice = 1 Mbps, Video = 5 Mbps, Other traffic = Best Effort”. This policy, in terms of configuration parameters, can be enforced as (Listing 7-1):

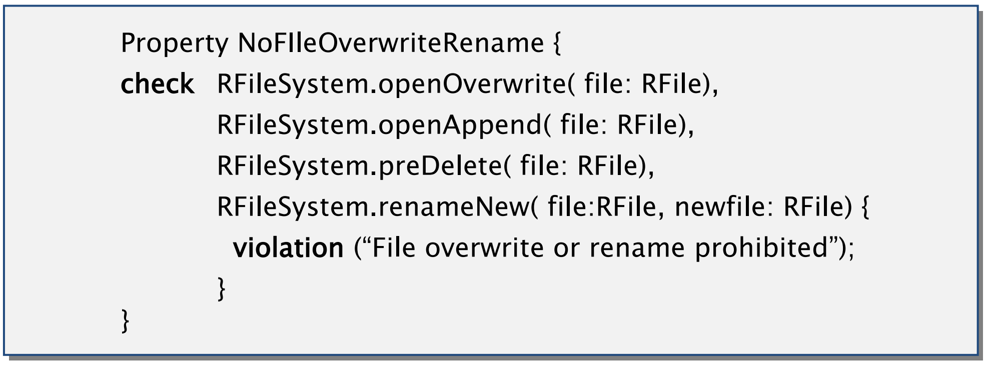
```
Configuration:
        Router(config)# policy-map CHILD
        Router(config-pmap)#   class VOICE
        Router(config-pmap-c)# priority 1000
        Router(config-pmap-c)# class MCA
        Router(config-pmap-c)# bandwidth 2000
        Router(config-pmap-c)# class VIDEO
        Router(config-pmap-c)# bandwidth 5000
        Router(config)#   policy-map PARENT
        Router(config-pmap)#   class class-default
        Router(config-pmap-c)# shape average 10000000
        Router(config-pmap-c)# service-policy CHILD
```

Listing 7-1 Sample of Cisco's Modular QoS CLI

7.2.2 Complied Code

To overcome the dependency of CLI for each network element, compiled code provides a platform independent architecture for flexible policy-directed code. It expresses policies at an abstract level and provides a tool that generates the wrappers needed to enforce a policy on a particular platform. This mechanism allows enforcement of access control policies over compiled code by running enforcement code on network elements for which policies are written.

Naccio [183] proposed a code rewriting system which defined two components – a *policy generator* and an *application transformer*. The policy generator produces a policy-enforcing platform library to check the code necessary to enforce the policy and produces a policy description file that contains transformation rules required to enforce the policy. This approach can also enforce security on the system by adding a security component to the network element. The application transformer is run when a user elects to enforce a particular policy on an application. It reads a policy description file and the program performs the directed transformation to produce a wrapper, which adds access control checks to the network element. For example, a *NoFileOverwriteRename* policy defined as “Prevent deletion of a file by restricting any overwriting of its content or renaming another file to its name” is coded as (Listing 7-2):



```
Property NoFIleOverwriteRename {
check   RFileSystem.openOverwrite( file: RFile),
        RFileSystem.openAppend( file: RFile),
        RFileSystem.preDelete( file: RFile),
        RFileSystem.renameNew( file:RFile, newfile: RFile) {
          violation (“File overwrite or rename prohibited”);
        }
}
```

Listing 7-2 Simple example of Policy Directed Code Safety

However, in order to support extensible security policies, it must recompile policy definitions, recreate library wrappers and re-modify programs to use the new wrappers.

7.3 Formal Approaches

The problem of using conventional approaches is that the management tools need to be aware of different types of scripts and command lines that are applicable to different types of devices in the network. For example, in order to distribute DiffServ QoS policies within a network that include routers manufactured by Cisco, IBM and Lucent, the management tool must be aware of the configuration scripts supported by all these different manufacturers. In many cases, the configuration commands also differ for different models (even belonging to the same manufacturer), and the management framework must be aware of these differences too.

Hence, a standardised approach was required to state the policies in a standard format. The generic idea was that each device or the proxy acting on their behalf will download the policies in a standard format and convert them to required configuration rules. The following section presents an overview of these formats and protocols.

7.3.1 **SNMP**

Simple Network Management Protocol (SNMP) [184] is firmly established in the networking community as a proven management protocol to manage routers and hosts. Furthermore, whenever a new protocol or device is introduced, invariably a related Management Information Base (MIB) [185] is also introduced. MIB is essentially an ASCII text file that describes SNMP network elements as a list of data objects and allows configuration and management of the new service or technology. It acts as a dictionary of the SNMP language, where every object referred to in an SNMP message must be listed in the MIB. The MIB is conceptually similar to the Policy Information Base (PIB) of the COPS protocol [186] (discussed in section 6.2.2).

Given the ongoing support of MIBs, several academic and commercial organisations apply SNMP as a protocol-of-choice for policy enforcement [187] and other network services [188]. In order to employ the SNMP protocol for device-level management, a process and a tool is required to convert the policies into appropriate MIB values that will conform to the specific configuration requirements of the device. There are two generalised approaches to use SNMP to enforce policies.

a. To configure the policies using MIB definition for the *specific* policy discipline. For example, to distribute DiffServ QoS policies, MIB definition of DiffServ is required. However, using discipline-specific MIB configuration does not provide a generic approach to enforce policies and configure devices.
b. A second approach was proposed by the SNMPConf working group [38]. The group redefined MIB definitions to allow a generic policy-based configuration where a generic policy is customised for each discipline by a discipline-specific policy MIB [189]. The advantage of this approach is that proposed common policy filters can be shared among disciplines. For example, IPSec and DiffServ filters can be coordinated through a common policy MIB.

> However, the MIB defined for a discipline in this manner differs from the MIB defined specifically for the discipline. In other words, the DiffServ policy MIB defined by the SNMPConf working group is different from the DiffServ MIB defined by the DiffServ working group [190].

The main advantage of using a SNMP-based approach for policy enforcement is the universal acceptance of the SNMP protocol and core MIB which is implemented in most of the networking devices. There are many well-defined proprietary MIB modules developed by network device vendors to support their management products. As new services evolve, eventually their MIB standards are defined. SNMP also works well for device monitoring and is useful for statistical feedback, status polling, alarm detection, and root cause analysis.

However, SNMP has been criticised on certain aspects. Since, SNMP was initially designed as a programmatic interface between management applications and devices, its usage without management applications or smart tools appears to be more complicated. Furthermore, standardised MIB modules often lack writable MIB objects, which are used for configuration, and most often lead to situations where writable objects exist only in proprietary solutions. The SNMP transactional model and the protocol constraints also make it more complex to implement MIBs compared with the implementation of commands for a CLI interpreter.

As such, the security characteristics of SNMP are relatively weak. These security problems are often compounded by the requirement to allow SNMP traffic through the firewalls. This restricts the use of SNMP-based policy distribution in many applications of security policies, especially, when supporting VPN and its related policies. Another aspect is its inefficiency is where each MIB data element must be written using a separate request process between individual devices and the management tool. Hence, SNMP-based MIB configurations are relatively slow.

7.3.2 **LDAP**

The Lightweight Directory Access Protocol (LDAP) [191] is being used for an increasing number of directory applications including network databases for storing network configuration information, service policy rules [192] and authentication rules [193].

In order to enforce policies using LDAP, a management framework needs to convert an abstract policy into the entries that can be populated into a LDAP directory. After the LDAP entries are written into a directory, any LDAP client running on each of the devices can access the directory to read entries and determine appropriate policies to implement them locally.

To distribute the policies using LDAP, an agreement is required among the management frameworks and the agents (LDAP clients) to define a common format to specify policies stored in the directory. This common format is determined by defining a *LDAP schema* [113], which classify the objects that are created within the directory and the relationships between objects are represented. The *LDAP schema* for LDAP directory definition is conceptually similar to the *Policy Schema* proposed in this research.

However, LDAP schema definition strictly depends on the discipline for which low-level policies are defined. For example, the LDAP schema implemented by IBM servers to support QoS policy configuration consist of two classes, *ServicePolicyRule* and *ServiceCategory* [194]. Figure 7-1 shows the two classes together with their cardinal relation.

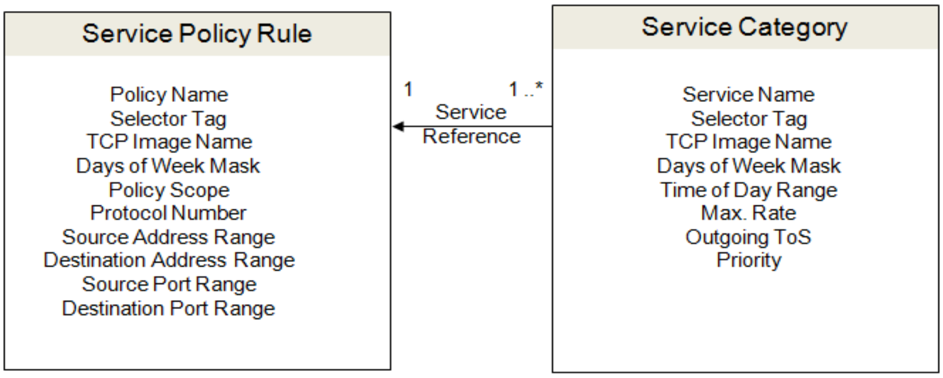


Figure 7-1 Object Model of IBM QoS Schema

The *ServiceCategory* class defines attributes of the DiffServ traffic class that may be supported within a device. The parameters of such a class include definitions such as outgoing Type of Service marking and maximum bandwidth to be used for the traffic class. The *ServicePolicyRule* specify the “flows” which are mapped onto each of the service classes and can be mapped to LDAP Data Interchange Format (LDIF) [195], which is a common way to represent LDAP directory information. Hence, an *OutgoingAsteriskVoIP* policy defined as “All outgoing packets that originate from local ports and use transport protocol 5060 are to be marked with Expedite Forwarding with a DiffServ value of 101110” is represented in LDIF as shown in Listing 7-3. Since, the LDAP directories are designed mostly to optimise access and lookup and not for rapid updates, the policies that need to be changed frequently are inappropriate for storing in LDAP.

```
dn: cn=rule1, loc=policy, o=canterbury, c=nz
objectClass: ServicePolicyRule
SelectorTage: TelecomServer
PolicyName: VoIP1
Direction: Outgoing
ProtocolNumber: 5060
SourcePortRange: 1024-65535
ServiceReference: EF
```

Listing 7-3 Sample LDIF entry for IBM QoS Schema

7.3.3 **COPS**

The IETF working groups have defined different protocols (e.g. DiffServ [196], Integrated Services [197], and RSVP [198]) to provide QoS in a network. When a QoS enabled router receives a QoS signalling message, it needs to decide whether to accept the connection request. Some of the conditions used for accepting connections can easily be determined by the router. For example, it can determine whether it has enough resources to meet the performance level desired by the incoming request, as in the case of SNMP. However, more complex conditions for accepting connections which are based on information such as traffic type requirements, security considerations, identity of users and applications, needs communication among several network interfaces.

To overcome this problem the IETF Resource Allocation Protocol (RAP) working group have defined the Common Open Policy Service (COPS) protocol (discussed in section 6.2.1). The standard COPS protocol operates by maintaining a TCP connection between a router and a policy server. The router connects to the policy server by issuing a “client-open” message that describes it capabilities and the type of policy decisions it can enforce. The policy server responds with a “client-accept” message that contains parameters for maintaining their connection. Periodic keep-alive messages are exchanged in order to ensure that the router and the policy server are maintaining a consistent state with each other.

As discussed in section 6.2.2, a COPS PIB is defined to provide a generic way to specify policy information. A PIB is a collection of policy rules that are implemented and supported at a “COPS-aware” router and contains a set of objects that carry the policy information. This core information model (discussed in section 4.3) provides a formal description of the types of classes that would be needed to describe any type of policy. For each new discipline for which device provisioning is needed, a new PIB can be defined.

The distinctive features of the COPS protocol can be summarised as [199]:

a. Both policy server and router states are completely synchronised with one another at all times. If there is a failure in communication, the state is resynchronised when the network is operating properly again and the router’s network configuration is valid. Also, if there is any failure in a transaction, that specific failure is reported back and local configuration rolls-back to the last “good” transaction state.
b. COPS non-overlapping instance namespace ensures that no other manager can corrupt a specific configuration. All the transactions are required to be executed in order. Furthermore, at any point in time, only a single manager is allowed to have control for a given subject category on a device. This single manager assumption simplifies the protocol as it makes it easier to maintain a shared state.
c. Similar to SNMP, COPS was also initially designed as a programmatic interface between management applications and devices (to support policy control over QoS signalling protocols) and views the network as a data-centric. Hence, mapping from a data-centric view to task-oriented view and vice-versa has complexities in common with SNMP.

Table 7-1 presents a comparison of the presented standards based protocols [200]:

Criteria	COPS	LDAP	SNMP
Console complexity	Low	Low	Low
Agent complexity	Medium	Medium	Medium
Error control	High	OK	High
Delay	Low	Medium	Low
Central repository	No	Yes	No
Maturity	Low	High	High

Table 7-1 Comparison of standard-based protocols for Enforcement layer

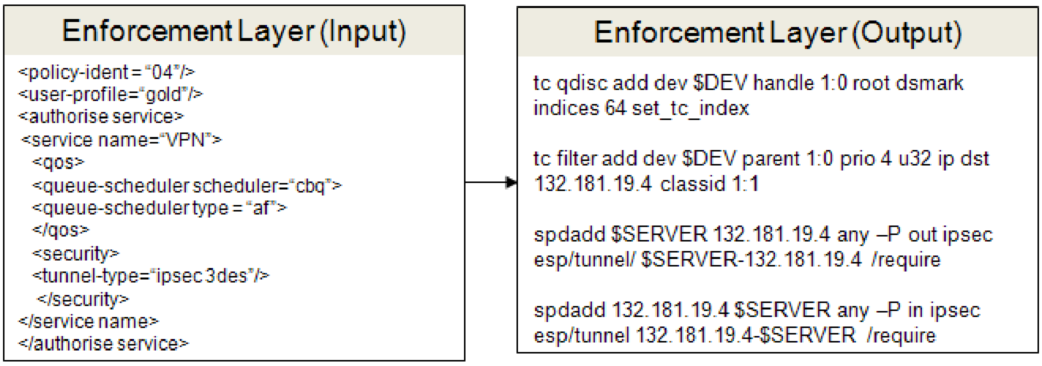
7.4 Summary

The previous sections have demonstrated that (i) there are different methods for enforcing policies to individual network elements and, (ii) each approach has some advantages and limitations. A practical network is likely to deploy a combination of enforcement methods (both standards-based and proprietary solutions) based on the specific nature of network elements deployed. Even the IETF PBNM and other working groups have proposed that a PDP can retrieve the policies using a LDAP protocol from a LDAP repository, while communication between PDP and PEPs uses COPS protocol [25].

Verma in [200] suggested that a standard approach where all the server platforms are supported will require complete interoperability, where three objectives need to be standardised:

- A standard protocol must be used by the devices and the management framework to retrieve and extract policies.
- A standard format must be used to specify policies.
- A standard convention must be used for each device to determine how to obtain policies that are relevant to that device.

The last requirement identifies the need for a proposed Enforcement layer to provide a translation bridge where syntax-level policies can be translated to specific device commands. Hence, a *GoldSecurity* policy (discussed in section 3.5.1) can be enforced from a Syntax layer representation to a Linux-based router supporting DiffServ QoS as shown in Figure 7-2.



Enforcement Layer (Input)

```
<policy-ident = "04"/>
<user-profile="gold"/>
<authorise service>
 <service name="VPN">
   <qos>
   <queue-scheduler scheduler="cbq">
   <queue-scheduler type = "af">
   </qos>
   <security>
   <tunnel-type="ipsec 3des"/>
    </security>
</service name>
</authorise service>
```

Enforcement Layer (Output)

```
tc qdisc add dev $DEV handle 1:0 root dsmark
indices 64 set_tc_index

tc filter add dev $DEV parent 1:0 prio 4 u32 ip dst
132.181.19.4 classid 1:1

spdadd $SERVER 132.181.19.4 any –P out ipsec
esp/tunnel/ $SERVER-132.181.19.4 /require

spdadd 132.181.19.4 $SERVER any –P in ipsec
esp/tunnel 132.181.19.4-$SERVER /require
```

Figure 7-2 Example of Enforcement layer

If a router supports a COPS-PIB or supports SNMP-MIB, then COPS or SNMP protocols can, respectively, be employed to enforce the policies. If a network device does not support either protocols then a proxy using command-line scripts may be used for configuration. As the device commands are very specific to the networks and the installed network elements, it is not possible to either restrict or recommend any particular approach, though existence of such a layer is important. This restriction is valid for any device-dependent message-exchange model where there is a direct dependency on the type of network interfaces, similar to the requirements imposed by an operating system for specific device drivers to communicate with respective devices.

Chapter 8
Proposed Architecture

In the previous chapters, we have identified that the main problem with offering ABC services to the roaming mobile users is that different access networks are not particularly integrated. In most cases, users are forced to manually interact with the system when switching between networks, which even requires restarting of running applications. This constraint contradicts with one of the basic requirements of ABC service i.e. to provide seamless mobility.

We further identified that to provide access to *any* roaming mobile user exposes a network to unnecessary congestion, stealing of resources and other security concerns. Furthermore, service providers need to dynamically manage their networks and to be able to adapt to current network conditions to provide agreed services to their users (both home and guest users). Hence, a *management framework* is required to deploy the services offered by the network.

However, we found that the IETF PBNM model falls short of such requirements. In Chapter 3, we proposed a new layered model as an extension to the IETF PBNM model to provide ABC services. In previous chapters, we have discussed each layer in detail and examined, recommended, extended or proposed new approaches to best suit an organisation’s requirements. This chapter is building on the previous chapters and defines the detailed architecture of the proposed PBM model. The public literature talks mostly about the simplistic IETF model at a very high level. This chapter proposes a more realistic PBM architecture by considering real life scenarios while offering ABC services. This motivates the need for additional components and architectural changes to be made to the IETF model. In this chapter, we will propose and implement an architectural framework of the generic model proposed in Chapter 3.

8.1 Introduction

Figure 8-1 presents the proposed architecture of the policy-managed network. The solution is presented as a set of interacting components realising the system's functionality. A different set of components structured in a different way could have also served our purpose as long as they fulfilled the same functionality. Nevertheless, the components presented enclose the system's functionality in a structured and a logical way.

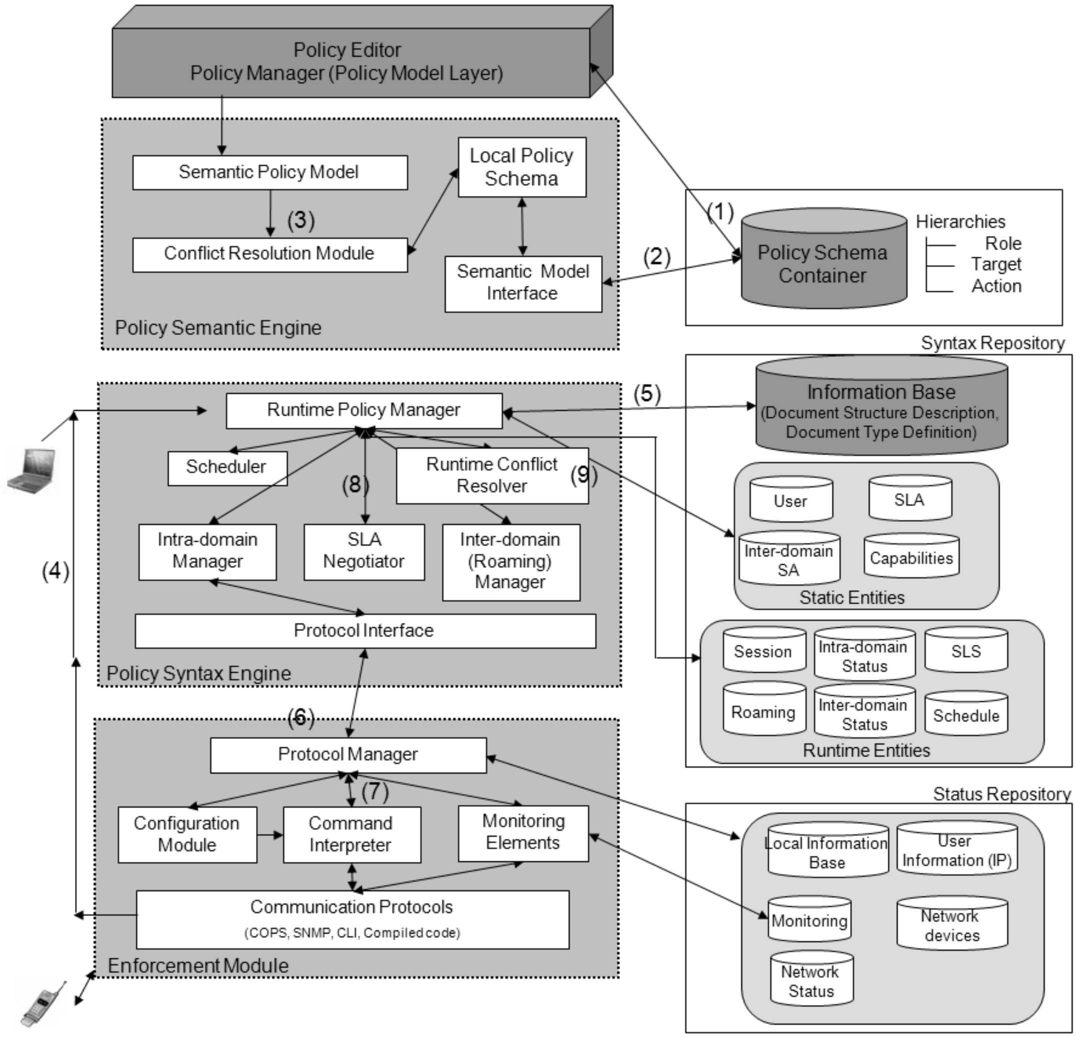


Figure 8-1 Proposed architecture of policy-managed network

Considering the policy, P1: *If a gold user requests a VPN tunnel during weekdays, provide QoS (i.e. Assured Forwarding) with security (i.e. ESP-3DES encryption), as long as user's traffic does not exceed the profile-bandwidth limit, or any network router fails. In either case, degrade the services to best effort with same encryption.*

The typical sequence of events relates to the inter-working of different components in the proposed architecture to provide agreed-upon services to user *John*, when he logs-in as a *gold* user.

Initially, the Policy Schema (hierarchical structure of the organisation) is stored in the Policy Schema Container in XML/LDAP format (1). When the Policy Semantic Engine is initiated, it stores a local copy of the Policy Schema translated by the Semantic-Model Interface (2). When the administrator inputs a new policy (e.g. P1) via the Policy Model Layer, the Semantic Policy Model generates an information model (discussed in section 4.3) of P1 for detecting and resolving any conflicts (discussed in section 5.6) using the Conflict Resolution Module (3).

When the user John connects with the managed network, either directly with a Runtime Policy Manager (RPM, if the mobile device is policy aware), or with any managed router (below the Enforcement Module), it requests an initial policy (profile) provisioning (4). The profile-request message is a combination of "roles + capabilities" which is used to select a subset of policies required to manage John. On receiving the request, the RPM generates an Information Base for the Syntax Repository (5). The Information Base follows a similar concept used by the IETF in defining a PIB (discussed in section 6.2.2) to associate the policies with the selected user.

The policies of John are then downloaded via the Protocol Interface and transferred to the Protocol Manager in the Enforcement Module via the COPS (or an alternative) protocol (6). The Protocol Manager communicates and monitors the network entities (e.g. router, firewall, etc.) via Monitoring Elements, while a Command Interpreter translate COPS instructions into device commands while referencing network parameters from the Configuration Module (e.g. the network parameters defined in Configuration Module for *Gold* service: `bandwidth=1.2Mbps`, `delay=120ms` are translated to device command information: `iptables -R ALLOW` (7). The network values in the Configuration Module could be altered at run time depending upon the current network condition as long as they operate within the prescribed values.

Any change in the network state e.g. router failure or out-of-profile user request, is reported to the RPM which then requires the Intra-domain Manager to manage the home network, while if the user’s request or present situation affects other network domains, then the RPM ensures that the Inter-domain Manager honours the SLA with other service providers by initiating the SLA Negotiator if required (8). Depending on the present status of the network, RPM may accept, upgrade, downgrade or even reject a user’s/other network’s request which is managed by the Runtime Conflict Resolver (9).

We will now discuss the introduced components of the architecture in detail, starting from the Repositories, the Policy Model Layer, the Policy Semantic Engine, the Policy Syntax Engine, and the components of the Enforcement Module.

8.2 Components of the proposed Architecture

While defining the properties of our proposed PBMS model, we have identified two key requirements: Modularity and Scalability (section 3.3). The scalability requirement can be achieved if the components of a management system can operate in a distributed environment and increased or decreased in number without any (or minor) changes required in the core architecture. The modularity requirement states that the management system should be able to choose their preferred subsystems. The proposed architecture attempts to fulfil these two key requirements together with the requirements.

In the following section, we will discuss the functional characteristics of each component in the proposed architecture and analyse whether modularity and scalability is achieved. The section focuses on explaining the services offered by each component rather than their specific implementation requirements.

8.2.1 Repositories

There are a number of Repositories identified in the proposed architecture. The Repository is generally assumed to be a passive store. In other words, it is used to simply store data and not used to process or act on the data. These Repositories or *Containers* store all the information of the organisation, including:

a. The Policy Schema which represents the conceptual structure of the organisation in a hierarchy of roles, objects and actions.
b. The static information such as user subscriptions, SLA, capabilities of network elements, etc.
c. The dynamic or runtime information such as a user's current security associations, intra-domain and inter-domain status, current sessions with roaming mobile users, etc.
d. The network-based information such as current bindings of IP address with users, current network status and network load on each networking element, etc.

Since the Repositories maintain all the information of the organisation and the information is of varied types applied at different levels of management, the PBM system places a variety of requirements on the Repository.

8.2.1.1 **Requirements**

The simple fact is that different applications make use of different management information. Each type of management information has its own storage, access, and other requirements. Hence, it is impossible to design a single repository that can simultaneously meet all of the different needs. This problem is solved by using a combination of a common information model and a set of standard model mappings. The role of the common information model is to define the information in a technology neutral fashion, so that different model mappings can be used to translate these data according to the specific needs of a particular repository in an optimal manner. Hence, a common information model and mapping translators are used together to implement the Repository (Figure 8-2).

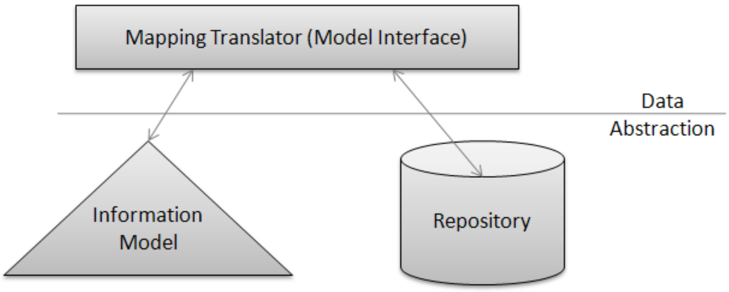


Figure 8-2 Abstraction by Repository

This approach places a unique requirement on the PBM system based on the fact that a wide variety of management data that are of interest to a PBM system requires multiple types of data Repositories. This is because no one Repository can have all of the necessary features to facilitate the storage and retrieval of varied types of data required by different components. This concept is similar to the fact that no single type of policy can be implemented throughout the network. For example, the business view of policies is different from the network view. Similarly, Repository storing policy information related to the business view will be different from Repository storing network information.

Another benefit of defining separate Repositories is the fact that policies are used to maintain and/or change an object's state by allowing or prohibiting certain actions. Hence, where a RPM maintain Inter-domain security associations and user's subscription database, it should not be allowed to change (or in some cases even view) the Policy Schema. Similarly, a Protocol Manager can maintain a Status Repository of runtime information of the network and its currently associated users, however, it should not be allowed to alter the user's SLA and subscription information. Hence, any lower configurations which are usually based on runtime environment should not affect the high-level policies which are usually static in nature.

8.2.1.2 **Repository at different layers**

We have proposed different Repositories applicable at different levels of the PBM system and are discussed as follows:

a. ***Policy Schema Container***

The Policy Schema Container holds the Policy Schema of the organisation. A Policy Schema represents the overall structure of the organisation by grouping roles (and users associated with roles), objects and actions into different hierarchies to reflect the relationships between them. Note that although the Repositories (or *Containers*) are distributed within the network, there is only one Policy Schema. Hence, the proposed architecture is a centralised management system with distributed entities.

If an operation of a network manager requires any restructuring of the Policy Schema e.g. adding a new role, then the Policy Schema needs to be verified.

This has to be done before the component(s) interested in using the policy have started the processing of the policy.

b. ***Syntax Repository***

The Syntax Repository contains the data associated with the Policy Syntax Engine and is discussed in detail in section 8.2.4.1. It maintains three different components:

i. The *Information Base* describes a model to define a database used to manage the devices in a network. It comprises a structured collection of objects or a schema of language for XML to manage entities in a network. Conceptually, defining an Information Base for a network is similar to defining a language for describing valid documents.

ii. The *Static Entities* is a collection of databases used to store static information such as user identification information, SLA definitions, inter-domain security associations, etc.

iii. The *Runtime Entities* represent a collection of databases to store dynamic information of the network and maintain current session information of current home and guest users, inter-domain and intra-domain status, etc.

c. ***Status Repository***

The *Status Repository* (discussed in detail in section 8.2.5.1) maintains a collection of databases to store runtime information of the network and the associated users at the device level. The network information such as a list of active routers, current network load, round trip delays, etc and, user information such as current IP address assigned to the user is maintained.

8.2.1.3 **Analysis of Repositories**

The approach of different Repository for different layers provides scalability and modularity to the proposed architecture. Since the Repositories maintain a set of databases providing a single functional component, the number of Repositories can be increased or decreased based on the requirements of the network or when the implementation is geographically selective. Hence, the Repositories do not need to be tightly integrated with each other or to a geographic area. Furthermore, a Repository can be replaced, altered or reused based on the organisation's or network's requirements (Figure 8-3). This approach for separate Repository for separate layers and the features of scalability and modularity provides the feasibility of the proposed architecture to be applicable to large and distributed environments.

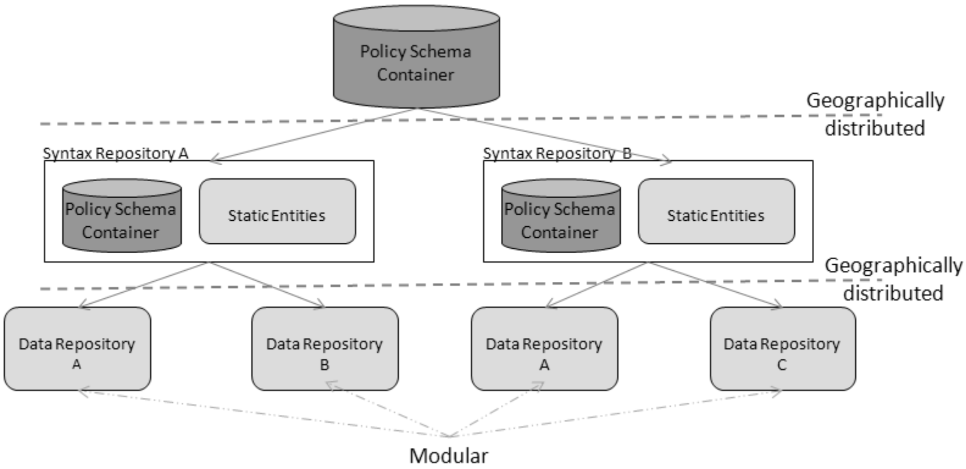


Figure 8-3 Multiple Repositories in a single PBM system

8.2.2 **Policy Model Layer**

The Policy Model Layer provides an architectural implementation of the *Policy Model Layer* discussed in Chapter 4 and is characterised by two components: a Policy Editor and a Policy Manager (Figure 8-4).

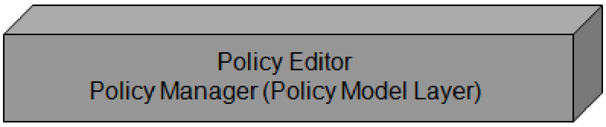


Figure 8-4 Policy Model Layer

The *Policy Editor* provides an environment for network managers to specify business policies while the *Policy Manager* includes a parser for policy normalisation and translates the policies into an object-oriented model to be, (if it exists), analysed by the Semantic layer and then, stored in the policy repository. The Policy Manager only represents the normalised structure of the policy and does not derive any meanings i.e. semantics of the policy.

The Policy Editor is the component used by the network manager to maintain policies and related information in the Policy Schema Container. It provides a logically centralised interface to the outside world for storing, managing, retrieving, and searching policies and policy information. In general, a PBM system can be interacted with using a GUI, an Application Programming Interface (API), or both (Figure 8-5). The policies need to be specified in the Policy Editor in a format that is familiar to the network manager.

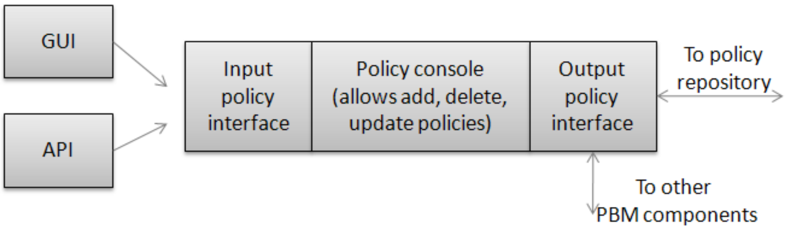


Figure 8-5 Policy Editor

The Policy Manager offers two main functionalities:

a. Firstly, it validates that the network manager has permission to perform the desired operation on the policy. Since, in most cases with the current management software and tools, a complete set of permissions is assigned to any user who is assigned the role of an *administrator*. However, in a large environment, most networks are designed as a set of domains, each of which is owned and run by different users. Therefore, assigning the same privileges to each administrator enables an administrator to manage policies, which in turn might manage policies operating on devices that they might not own.

   Hence, based on our proposed Policy Schema, a *user* is assigned to a *role*, and a *role* is authorised to perform an *action* on *objects*. Any network manager (user) who wants to specify business policies, needs to assume a *role* which should have an authorisation to perform related create, read, update and delete actions on the policy object. This role-based authorisation can be further supplemented with external information, such as the time of the day.

b. After authenticating and authorising a network manager, the Policy Manager normalises the inputted policies into tokens based on the Policy Schema. The normalised policy is then translated into an object-oriented model. This model provides a uniform and a consistent representation of the policy which is in accordance with the entities defined in the Policy Schema. The policy is then mapped to a structured specification such as in XML or LDAP format to be either (i) sent to the Semantic Policy Model for policy analysis in terms of detecting and resolving conflicts or (ii) stored directly in the Policy Schema Container if the organisation does not support any semantic analysis.

8.2.3 Policy Semantic Engine

The Policy Semantic Engine defines a set of components for the semantic analysis of the policies. The components introduced are (Figure 8-6):

a. The *Local Policy Schema* stores a local copy of the Policy Schema to ensure that the introduced policy is consistent with the overall Policy Schema of the organisation.

b. The *Semantic Policy Model* generates a semantic mapping of an inputted policy for conflict analysis.

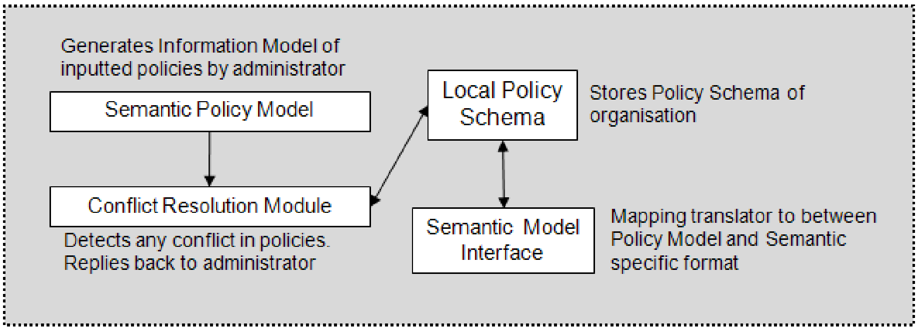

Generates Information Model of inputted policies by administrator
Semantic Policy Model
Conflict Resolution Module
Detects any conflict in policies. Replies back to administrator
Local Policy Schema
Stores Policy Schema of organisation
Semantic Model Interface
Mapping translator to between Policy Model and Semantic specific format

Figure 8-6 Policy Semantic Engine

c. Any inconsistency arising due to the introduced policy is detected and further resolved by the *Conflict Resolution Module*.

d. The *Semantic-Model interface* acts as a Mapping Translator (discussed in section 3.5.5) to provide a mapping of the policy from and to the Policy Schema stored in the Policy Schema Container and the preferred format used by the Semantic Engine.

We will further discuss each component in detail.

8.2.3.1 **Local Policy Schema**

The Local Policy Schema stores a copy of the Policy Schema of the organisation locally. It may either load an exact copy of the structure of the Policy Schema or may redefine the structure. The concept of hierarchies in Policy Schema comes as a natural choice to depict the organisational structure. However organisations may not prefer a hierarchy, or specifically, a hierarchy in a tree structure. A tree structure is where there is only one root node and every other node is either a parent or a child. Every child can have only one parent node and no child node can have another child node. Sometimes, a similar structure or a graph where partial ordering between nodes is not defined and a child node can have more than one parent, may represent the organisations structure more logically (Figure 8-7). However, the tree structure is optimum for detecting inherent conflicts between roles [201].

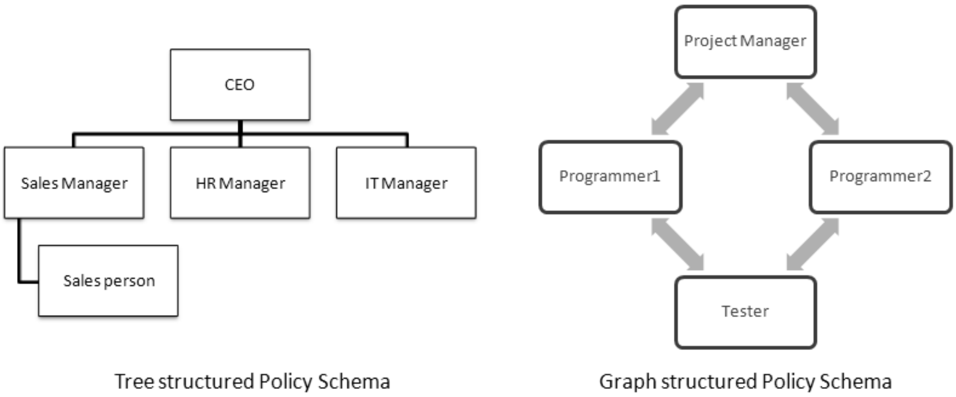


Figure 8-7 Different structure for Policy Schema

Based on the Policy Schema, different policies are defined which provide authorisation to (or prohibition of) performing certain actions on certain targets. When a policy is added, deleted or updated by a network manager, the structure of the Policy Schema is changed. This may also have side effects on the policies already defined and further needs to be analysed for any conflicts inherent within the structure of the Policy Schema. For example, if a parent node is deleted then every child node needs to be assigned a new parent for every hierarchical structure (i.e. a tree or a graph).

8.2.3.2 **Semantic Model Interface**

The Semantic Model interface generates a model of the policy being updated by the network manager in accordance with the approach used for semantic analysis. In this case, either a new policy is added or an old policy is updated or deleted.

As discussed in section 5.5, numerous approaches for policy analysis have been proposed, where almost every approach requires a policy to be presented in a specific format. For example, the IETF PBNM policies are represented in an if-condition-then-action format whereas policies represented in a rule-based approach on Logic Programming rules are represented in a Prolog type structure.

8.2.3.3 **Conflict Resolution Module**

The subject of conflict detection and resolution is an open area of research. There are different types of policy conflicts and equally, different types of approaches to detect those policy conflicts. The Conflict Resolution Module (CRM) is an integral part of the Policy Semantic Engine and performs validity checks on whether the updated policy and resultant Policy Schema are in accordance with the overall schema of the organisation. Hence, it checks for the satisfiability and feasibility of a policy in the organisation where it is applicable.

For any CRM to be an efficient tool, an explicit separation between the capabilities of a role and its permitted actions on different objects is required. As discussed previously, this means that the characteristics of the system being managed must be understood and represented in policies in as much detail as required. If this is not done, the CRM will be forced to try and compare the sets of roles, actions and objects with different combinations and different features.

Following the modular approach of the proposed PBM framework, a CRM can employ any of the conflict detection approaches based on the preferences of the network manager and the requirements imposed by the particular needs of the network/organisation. Furthermore, since different types of policies are required for different management areas, different solutions and/or methods can be employed for conflict detection and resolution. The CRM is only responsible for detecting policy conflicts which are employed in a static environment, or in an environment which is not affected by the current state of the network, such as conflicts caused by Role-based policies, Authorisation policies, Propagation policies and Action Composition, and conflicts caused by constraints such as Separation of Duty policy and Chinese Wall Security policy (discussed in section 5.4.2). Any runtime policy conflict is resolved by the *Runtime Conflict Resolver* in the Policy Syntax Engine.

We have also proposed our conflict detection technique based on first-order logic and composite mapping (section 5.6) where we have extended the usage of additional policy attributes for resolving conflicts in static and dynamic environments including Action Composition policies, Propagation policies and On-event conditions.

8.2.3.4 **Semantic Model Interface**

The Semantic Model interface acts as a Mapping Translator to map the Policy Schema stored in the Policy Schema Container to the Local Policy Schema and vice versa. We assume that when the Policy Schema is retrieved from the Container, it is in a consistent state. Furthermore, it is required that when the Policy Schema is translated back to be stored in the Policy Schema Container, it needs to be in a consistent state. The CRM will inform the network manager of any inconsistencies. A typical example of a Semantic Model Interface can be where a first-order logic statement is converted to XML format where both the Policy Semantic Engine and the Policy Schema share a common XML Schema.

Since the Semantic Model interface is a Mapping Translator, the key functional requirement is that the mapping needs to be unambiguous to preserve the meaning of the policies (i.e. semantics). Additionally, to support the modular design, the Policy Semantic Engine needs all the input and output of the interface to be translated in a XML format with a predefined XML Schema.

8.2.3.5 **Analysis of Policy Semantic Engine**

The proposed Policy Semantic Engine is built on a modular and a scalable concept where a dedicated Policy Semantic Engine is required for different conflict detection and resolution approaches preferred by the organisation (Figure 8-8). This requirement is validated by the fact that the sole purpose of this engine is to detect and resolve any conflict inherent in the definition of policies and hence, all the components introduced in the engine, such as the Semantic Policy Model, the Semantic Model interface, the Local Policy Schema and the CRM are specific to the particular approach being incorporated.

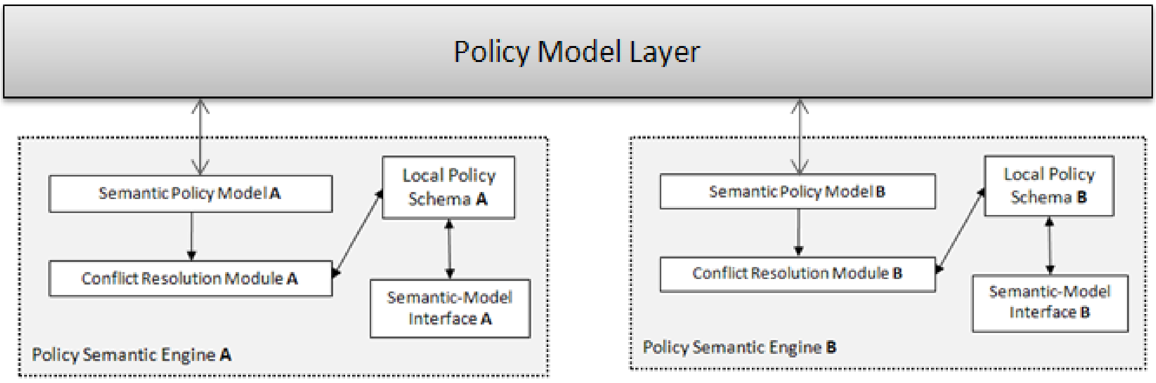


Figure 8-8 Multiple Policy Semantic Engines in a single PBM system

Hence, multiple Policy Semantic Engines can be integrated into a management system based on the requirements of either:

(i) Using multiple specialised policy conflict detection techniques for better and exhaustive conflict detection, or

(ii) Geographically dispersed network where network managers are not in the same place where the networking components are installed, or

(iii) Different conflict detection approach is employed for different domain.

A natural derivative of this approach is to view a mesh of Policy Semantic Engines in a hierarchical structure where a specific domain is managed by specific engines and the Local Policy Schema only downloads the entities related to that particular domain.

8.2.4 **Policy Syntax Engine**

The Policy Syntax Engine represents a functional block for the distribution of policies. It is an architectural implementation of the Syntax layer (discussed in Chapter 6) in our proposed management framework. In the IETF terms, together with the Syntax Repository, this engine can be regarded as a PEP with a local PDP.

The Policy Syntax Engine can facilitate:

- Dynamic runtime requirements of the network elements and users,
- Interoperability between different networks and policy negotiations, and

- Providing an interface between high-level domain-wide policy definitions and instance-specific information required for the device management.

However, the engine does not define the operational commands or substitute any network parameter values. For example, a policy P1 will be represented as, *If user==‘gold’ and user_status==‘allowed’ and network_status==‘ok’ then mark ESP packets as Assured Forwarding*. This policy has not yet been assigned to any specific device or network element, nor has it specified how the Assured Forwarding mechanism will be implemented.

The functional components of the Policy Syntax Engine are defined as follows (Figure 8-9):

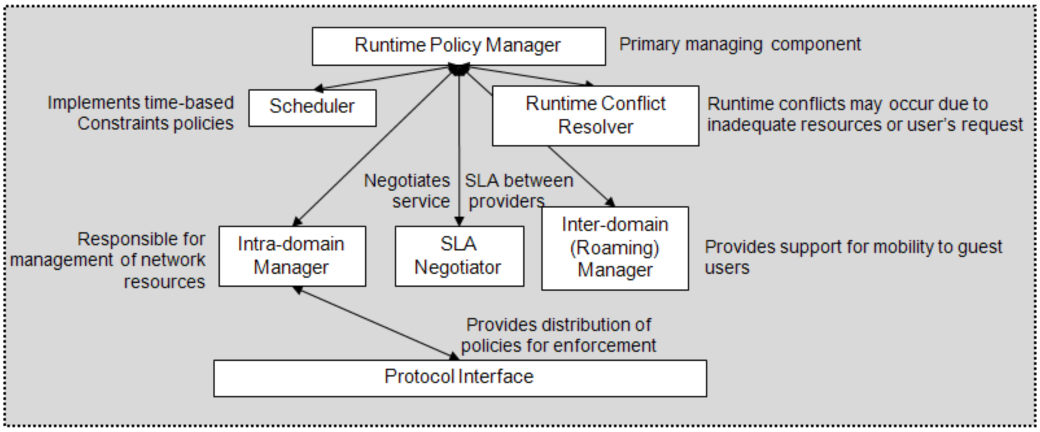

Runtime Policy Manager
Primary managing component
Implements time-based Constraints policies
Scheduler
Runtime Conflict Resolver
Runtime conflicts may occur due to inadequate resources or user's request
Negotiates service
SLA between providers
Responsible for management of network resources
Intra-domain Manager
SLA Negotiator
Inter-domain (Roaming) Manager
Provides support for mobility to guest users
Provides distribution of policies for enforcement
Protocol Interface

Figure 8-9 Policy Syntax Engine

- The *Runtime Policy Manager* is the primary managing component of the Policy Syntax Engine and coordinates with all other components of the engine.
- Any runtime conflict may occur while the network is operational especially when the resources are inadequate to honour the SLA with users or when a user makes an out-of-bounds request. In these cases, the *Runtime Conflict Resolver* provides a solution of either accepting or rejecting those requests.
- The *Scheduler* implements the time-based constraint and identifies the time period for which a policy is valid.
- The *Intra-domain Manager* manages the dynamic network configuration to meet the requirements of home users.

- The *Inter-domain Manager* supports the mobility and dynamic network configuration in order to meet the requirements of roaming mobile users.
- The *SLA Negotiator* supports policy negotiation for inter-domain communication between different service providers.
- The *Protocol Interface* supports the distribution of policies to manage network resources using management protocols such as Web Services or the COPS protocol.

We will further discuss each component in detail including the Syntax Repository.

8.2.4.1 **Syntax Repository**

The Syntax Repository stores the information required by the Policy Syntax Engine. It incorporates three different components, Information Base, Static Entities and Runtime Entities (Figure 8-10):

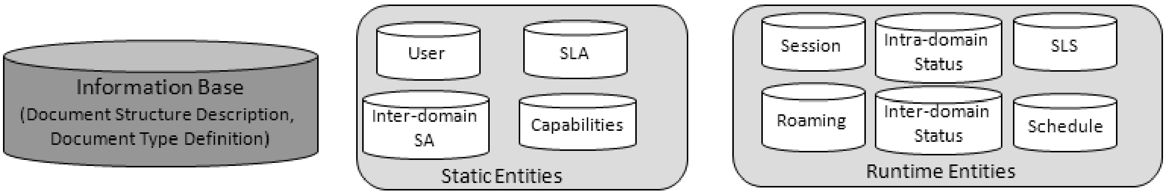


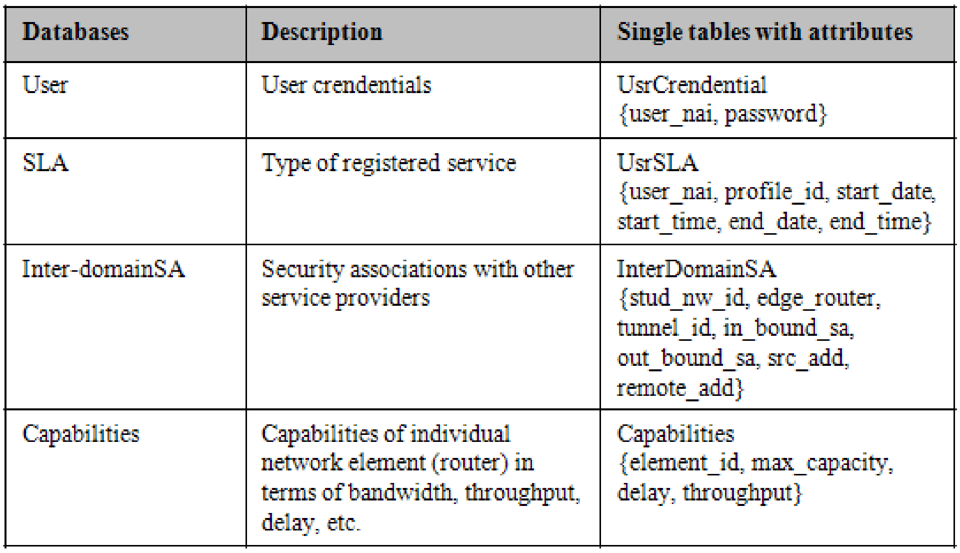
Figure 8-10 Syntax Respository

- As mentioned previously, the *Information Base* describes a model to define a database used to manage the devices in a network. It is conceptually similar to a Domain Specific Document (DSD) [202] or a Document Type Definition (DTD) [203] for XML defining schema where the entities are grouped and the entity groups are organised in some structure which can be browsed. That is, from any point in the structure at least some other related parts of the structure can be identified and moved to. Typically, a hierarchy is used for defining a structure and entities are grouped based on common attributes, for example an object-oriented structure in a PIB (discussed in section 6.2.2) or as a tree-like structure in XML Schema [116]. The entities can be grouped based on the hierarchies of roles, objects and actions. Hence, an Information Base can be considered as a practical implementation of the Policy Schema.

- The *Static Entities* is a collection of databases used to store static information which applies to instances which do not change with the runtime or dynamic behaviour of the network or its users. The organisations can define a set of database and relative table structure as per requirements. For simplicity and demonstration purposes, we have shown a set of four databases each with a single table (Table 8-1).

  a. The *User* database has a single table *UsrCredential* which maintains user's Network Access Identifier (NAI) [204] e.g. john@networkA.com and password. The NAI is an ID submitted by the user (home or guest) during authentication. For guest users, the purpose of the NAI is to assist in the routing of the authentication request to the service provider with which the user is currently registered. In cellular registrations, the NAI is of the format `mobilenumber@serviceprovider`

  b. The *SLA* database maintains the user's profile (e.g. gold, silver, etc.) and the subscription start and end date-time.



Databases	Description	Single tables with attributes
User	User crendentials	UsrCrendential {user_nai, password}
SLA	Type of registered service	UsrSLA {user_nai, profile_id, start_date, start_time, end_date, end_time}
Inter-domainSA	Security associations with other service providers	InterDomainSA {stud_nw_id, edge_router, tunnel_id, in_bound_sa, out_bound_sa, src_add, remote_add}
Capabilities	Capabilities of individual network element (router) in terms of bandwidth, throughput, delay, etc.	Capabilities {element_id, max_capacity, delay, throughput}

Table 8-1 Static Entities of Syntax Repository

  c. The *Inter-domainSA* database maintains security association information when the network needs to communicate with other networks. The security association identifies the edge router which is connecting with the other network together with inboud and outbound security rules such as certificate exchange, IPSec tunnels or simply network_id-password attributes.

   d. The *Capabilities* database maintains information of individual network element. For example, a network router's information such as bandwidth, throughput, delay, etc.

- The *Runtime Entities* represent a collection of databases to store dynamic information of the network and users. For demonstration purposes, we have shown a set of six databases:
   a. The *Session* database maintains session information of the home users who are currently logged-on to the network.
   b. The *Roaming* database maintains session information about the guest users who are currently logged-on to the network.
   c. The *Intra-domain Status* database maintains status information of network elements offering services to the home users. It is referenced by the *Intra-domain Manager* to provide SLA service to a network's home users.
   d. The *Inter-domain Status* database maintains status information of network elements offering services to the guest users. It is referenced by the *Inter-domain Manager* to honour a SLA with guest users of other networks. A SLA is negotiated between two networks to offer their subscribed users a level of service when they roam to the other network. This service can be defined in terms of times of availability, methods of measurements whether the services are offered as per agreement, defined traffic levels if the user's request exceeds its profile, other costs involved, etc.
   e. The *SLS* database maintains operational characteristics of a SLA which may consist of information such as expected throughput, drop probability, latency, constraints on inbound and outbound traffic, traffic profiles, and marking and shaping of data packets.
   f. The *Schedule* database maintains scheduling information, such as current timeout for a guest user's session, SA tunnel timeout, etc. This information is referenced by the *Scheduler* of the Policy Syntax Engine for time-based authorisation of policies.

8.2.4.2 **Runtime Policy Manager**

The Runtime Policy Manager (RPM) is the primary managing component of the Policy Syntax Engine. It is a coordinating agent of the engine and focuses on the Intra- and Inter-domain management together with inter-operability with other networks/service providers and runtime requirements of its home and guest users.

The RPM is a “front-end” of the network where the users (home and guest) get authenticated, authorised and eventually connect to the network to access the offered services. In the case of devices which are not “policy-aware” i.e. in IETF PBNM terms, which are not PEP, or does not support direct connection with the RPM, they are connected to the *Enforcement Module* and the authentication information is propagated upwards. We have also proposed an architecture of a policy-managed mobile client discussed in Chapter 9.

The functionalities offered by the RPM are summarised as follows:

- To provide network access to the users. When a user logs in to the network-domain, after being authenticated using NAI, a role is (or a set of roles are) associated by initiating a time-based session. The roles assigned with the policy model may be either static or dynamic. The membership of the static roles initiates a *session* for the selected entities within the organisation, whereas membership of dynamic roles is evaluated at the runtime according to some pre-defined predicate (e.g. lowest workload server, active backing server, etc).
- To communicate with the *Information Base* of the Syntax Repository to retrieve policies in response to service requests from a user.
- To maintain and monitor the information in *Static Entities* and *Runtime Entities* such as user subscription information, current SLA, capabilities of network components, current time-based sessions with users, and Intra- and Inter-domain status information.
- To cater for the runtime requirements of users. The RPM manages any out-of-bounds request from the user. For example, John, a gold user requests additional bandwidth for the next hour for an encrypted video conferencing service, which is normally only available to Platinum profile users. The RPM

requests the *Runtime Conflict Resolver* to check the current network load based on the *Static* and *Runtime Entities* from the *Syntax Repository* to either accept or reject, or alter the offered services.

- To coordinate with *Intra-domain Manager* for efficient utilisation of network resources and honour the SLA with home users.
- To coordinate with *Inter-domain Manager* to provide services to roaming mobile users and honour the SLA with guest users.
- To coordinate with the *SLA Negotiator* in case of a network element failure which affects services offered to roaming mobile users subscribed with other service providers.

8.2.4.3 **Runtime Conflict Resolver**

In a profile-based architecture, particular traffic receives a predefined treatment based on the predefined policies. This treatment can be interpreted as a particular Per Hob Behaviour (PHB) [205] where user's traffic receives treatment according to its profile and the SLA related to the profile. To honour the SLA, networks usually allocate resources which, based on specific SLS, are usually static and can lead to bandwidth wasting and starving users.

In the static approach any out of bounds traffic is simply dropped, remarked or assigned a new profile [206]. However, this decision is static and is taken once for all, i.e. when the network element is configured. For example, the *Out of bound* policy defied as *if user traffic gets more than the allotted profile, drop all out of bounds packets* will be applied (Figure 8-11) to all the packets regardless of whether the network is capable of processing the packet(s).

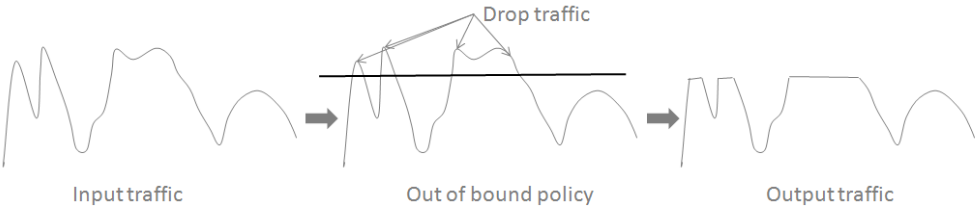


Figure 8-11 Static Policy Decision for Runtime Conflict

Another scenario can be where a network element (e.g. an edge router) fails and cannot provide any services. In this case, a dynamic relocation of current connections (session and services) to some other router is required.

These scenarios are treated as a dynamic or a runtime conflict in our proposed architecture. In case of a runtime conflict, the RPM requests the Runtime Conflict Resolver to resolve the issue based on the Runtime Entities from the Syntax Repository. The Runtime Conflict Resolver may accept, reject or modify the request.

8.2.4.4 **Scheduler**

The Scheduler component implements a time-based constraint on all the entities and components of the architecture including users, Repositories, engines and the managers. For example, it keeps a track of the individual user session timeouts, SA tunnel timeouts, response time from Monitoring elements, etc. Specifically, the Scheduler maintains time-based constraints during which a policy is valid (discussed in section 5.4.2).

In section 5.6.3(F), we further concluded that a usual approach of eliminating the overlapping time periods from the policies is not a practical solution especially in a large and dynamic environment. Our proposed composite mapping method offers greater flexibility in terms of optimising the time overlaps between policies.

8.2.4.5 **Intra-domain and Inter-domain Manager**

The *Intra-domain Manager* is responsible for the management and efficient utilisation of network resources and operates to honour agreed-upon SLAs with its home users. The *Inter-domain Manager* is responsible for the management of network resources to honour agreed-upon SLAs with guest users subscribed with other networks with which this network maintains security associations. Both the managers also interact with the monitoring services to gather the current status of the network.

The Inter-domain manager communicates with the SLA Negotiator allowing guest users to use services provided by the network when they are away from their home network without having to buy a new subscription. To achieve this goal, roaming contracts must be established between the service providers on per-bandwidth or per-user approaches [207].

Hence, the main characteristics of these managers are defined as:

- To employ a client/server model in which Protocol Manager can send requests and updates.
- To employ a reliable (i.e. TCP-based) management protocol for communication.
- To be able to deploy the policies and maintain a state of successful distribution of policies.
- To be able to monitor the network elements and generate a status report.

To offer such management, both managers are required to support a management protocol together with a set of databases (i.e. Syntax Repository). In Chapter 6, we extensively discussed the feasibility of optimal management protocols (both object-oriented and XML-based) for intra-domain and inter-domain policy distribution.

In section 6.4, we proposed our approach of deploying the COPS protocol for intra-domain management and XACML for inter-domain policy negotiations. Furthermore, the COPS protocol supports proprietary and vendor-dependent extensions which can be fine-tuned based on the specific requirements of networks and for the efficient utilisation of message exchanges. The XACML offered inter-operability in message exchanges between service providers sharing a common Schema and without exposing their internal structure. XACML also supports policy negotiations.

Similar to the OSI network management model [208], the proposed Intra-domain and Inter-domain Managers together with the Static Entities, Runtime Entities, Enforcement Module and Status Repository focus on five areas of functions offered to the network. These are sometime referred to as the FCAPS model.

- ***Fault***

  The goal of fault management is to recognise, isolate, correct and log faults. Fault management is concerned with monitoring network elements, detecting any network faults, logging information, issuing warnings and if possible, fixing the problem. A common fault management technique is to implement a management protocol such as, in our case COPS protocol, to collect information about network devices.

As discussed in section 6.5 of our proposed approach for policy distribution, the Intra-domain Manager monitors the network elements using the COPS protocol which provide services to home users, while Inter-domain Manager monitors network elements allotted for guest users and negotiates SLA with other service provider/networks using XACML.

- ***Configuration***

  The goals of the configuration management are to gather, set and track configuration of the devices. The Configuration management is concerned with monitoring system configuration information, and any changes that takes place. This area is especially important, since many network issues arise as a direct result of changes made to the configuration of devices.

  In our proposed architecture, the “conflict-free” policies are distributed by the Intra-domain Manager for such changes. However, the type and level of information contained in these policies is dependent on where the policy is currently deployed to.

- ***Accounting***

  The accounting management is concerned with tracking network utilisation information, such that users (or networks to whom guest users are subscribed to) can be appropriately billed or charged for accounting purposes.

  As a user is authenticated by the RPM, the Intra-domain Manager initiates a user session including their connection initiation and termination times stored in the Syntax Repository. This information can be used for billing purposes.

- ***Performance***

  The performance management is focussed on ensuring that network performance remains at acceptable levels. This area is concerned with gathering regular network performance data such as network response times, packet loss rate, etc. The goal is to prepare the network for dynamic changes to user’s requirements and an optimal utilisation of network resources.

  The Intra-domain Manager together with the information collected from the Enforcement Module keeps a track on the usage of network resources. In case of any runtime requests from a user, the RPM requests the Runtime Conflict Resolver to make a decision on the request. This decision is forwarded to the Intra-domain and Inter-domain Manager, as applicable, which in turn updates relevant information in the Syntax Repository.

- ***Security***
  The goal of security management is to control access to the network resources including user authentication and authorisation and, management of security associations with other service providers.
  In the proposed architecture, a user is authenticated by the RPM (based on user's NAI and credentials) and security association with service providers is maintained in the Syntax Repository.

8.2.4.6 **SLA Negotiator**

The SLA Negotiator is a module to support policy negotiations between two service providers for inter-domain management of roaming guest users to honour the agreed upon SLA.

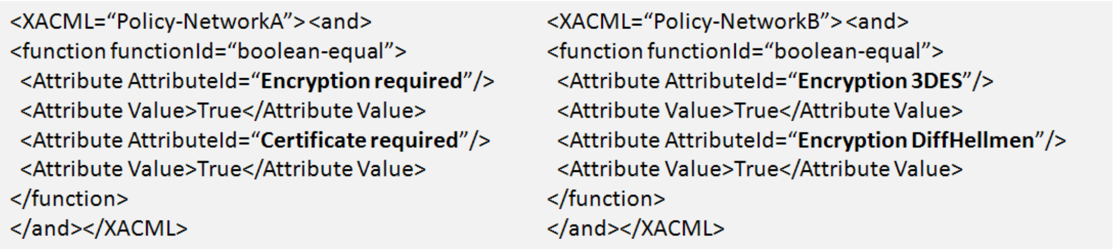
The SLA Negotiator is required in the cases:

- When the SLA parameters are below agreed upon values. The SLS parameters may degrade due to network-link failure, dynamic changes in topology or sporadic heavy load on the network.
- When a guest user requests a specific service (or service level) which is generally not offered by the network. For example, a request for higher-level security protocol suite or QoS assurance which is generally not supported by the network.

As discussed in section 6.3.1, a policy negotiation is inherently supported by XACML, which is used as a management protocol for inter-domain communications. It derives an intersected policy from two policy sources which is not only based on equality matching of attributes but also allows for the defining of fine-grained parameters such as protocol suite to apply or any changes to the time, cost or network address. The SLA Negotiator focuses only on the policy negotiation and does not provide conflict detection since the policies derived from different service providers are rightly assumed to be conflict-free.

Figure 8-12 portrays a scenario when user *John* roams from the home network A to a guest network B, where his subscribed *gold* profile requires *VPN security with either 3DES or certificate-based encryption*. However, the guest network B does not provide security to its home and guest users.

After a successful authentication and profile download from home network A, when *John* connects with the RPM of network B, he requests for an *encrypted VPN tunnel.* The RPM in turn requests the Runtime Conflict Resolver which then determines that the network (B) supports some encryption protocols. However it is not available by default and the RPM requests SLA Negotiator to negotiate an optimal solution.



```
<XACML="Policy-NetworkA"><and>
<function functionId="boolean-equal">
 <Attribute AttributeId="Encryption required"/>
 <Attribute Value>True</Attribute Value>
 <Attribute AttributeId="Certificate required"/>
 <Attribute Value>True</Attribute Value>
</function>
</and></XACML>
```

```
<XACML="Policy-NetworkB"><and>
<function functionId="boolean-equal">
 <Attribute AttributeId="Encryption 3DES"/>
 <Attribute Value>True</Attribute Value>
 <Attribute AttributeId="Encryption DiffHellmen"/>
 <Attribute Value>True</Attribute Value>
</function>
</and></XACML>
```

Figure 8-12 Policy negotiation between service providers

Using XACML, network A negotiates for either *Encryption required* or a *Certificate Required*. The network B fulfils the support requirement by choosing *Encryption required* and offers to support *3DES encryption*.

8.2.4.7 **Protocol Interface**

The Protocol Interface is an optional component which provides a bridge of communication between the Policy Syntax Engine and the Enforcement Module. This bridge is only required when the distribution protocol is different from the enforcement protocol. For example, the Policy Syntax Engine supports the COPS protocol while the *Protocol Manager* of the *Enforcement Module* supports SNMP. Hence, the Protocol Interface needs to map the COPS-PIB policy structure to the SNMP-MIB. For the Intra-domain management, we have proposed to use COPS as a common protocol between Intra-domain Manager and the Protocol Manager.

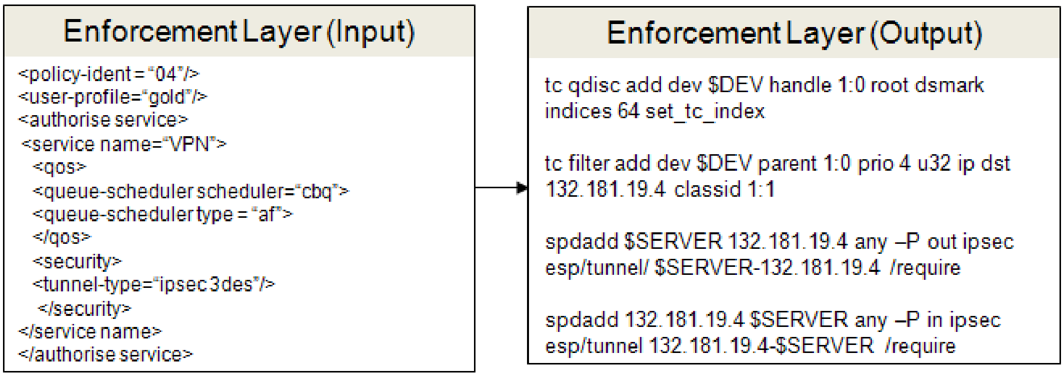
8.2.4.8 **Analysis of Policy Syntax Engine**

Similar to the proposed Policy Semantic Engine, the Policy Syntax Engine is built on a modular and scalable concept where different Policy Syntax Engines can be incorporated for network management based on per-management-domain requirements or geographic requirements. However, unlike the Policy Semantic Engine where a dedicated engine was required for different conflict detection and resolution approaches, a Policy Syntax Engine can support different management protocols, for example, COPS for intra-domain and XACML for inter-domain management. A dedicated Syntax Repository is required with every Policy Syntax Engine deployment.

8.2.5 **Enforcement Module**

The Enforcement Module is an architectural implementation of the Enforcement layer (discussed in Chapter 7). It is the point in the PBM system that enforces the policies to configure network devices through hardware and software means, as appropriate, and provides an interface between the Syntax representation of policies and the device-specific configuration commands used to finally deploy the policies.

The configuration commands are characterised by specific network parameters to be used to implement the policy. Hence, the policy P1 (*if a gold user requests a VPN tunnel during weekdays, provide Assured Forwarding with ESP-3DES encryption*) expressed at this level would specify Class Based Queuing to realise Assured Forwarding services in a DiffServ network, and mark the packets arriving from `IP address=132.181.19.4; port=5554` with `DSCP=0x64` together with adding an IPSec tunnel (Figure 8-13).



```
<policy-ident = "04"/>
<user-profile="gold"/>
<authorise service>
 <service name="VPN">
   <qos>
   <queue-scheduler scheduler="cbq">
   <queue-scheduler type = "af">
   </qos>
   <security>
   <tunnel-type="ipsec 3des"/>
    </security>
</service name>
</authorise service>
```

```
tc qdisc add dev $DEV handle 1:0 root dsmark
indices 64 set_tc_index

tc filter add dev $DEV parent 1:0 prio 4 u32 ip dst
132.181.19.4 classid 1:1

spdadd $SERVER 132.181.19.4 any –P out ipsec
esp/tunnel/ $SERVER-132.181.19.4 /require

spdadd 132.181.19.4 $SERVER any –P in ipsec
esp/tunnel 132.181.19.4-$SERVER /require
```

Figure 8-13 Policy mapping by Enforcement layer

8.2.5.1 **Status Repository**

The *Status Repository* maintains a set of databases to store runtime information of the network and the associated users at the device level. It incorporates the following databases (Figure 8-14):

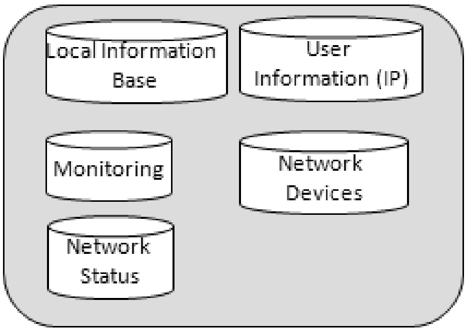


Figure 8-14 Status Repository

- The *Local Information Base* is similar in concept and structure to the *Information Base* and is used to store the policies sent by the Policy Syntax Engine. In the proposed architecture, the Policy Syntax Engine and the Enforcement Module share a common management protocol (i.e. COPS) and hence a common information base (i.e. PIB).
- The *Network Devices* maintain a list of all the network devices (for example routers, firewalls, etc.) managed by the Enforcement Module. This database maintains extensive information regarding specific details of the network device as offered by the vendor or determined by the network administrator. The information includes, for example, error rates in network device due to heavy network load, efficient operating temperature, supported protocols, etc. It also maintains a brief summary of all the operational commands been implemented in the device. Hence, the information is usually static since it does not maintain any active or runtime information about the network.
- The *Monitoring* database maintains a list of all the active network devices together with other information such as their current network load, current round trip time, error rates, etc and keeps a track of their current status by polling. This polling is performed by a management protocol such as SNMP (discussed in section 7.3.1).
- The *Network Status* database maintains a generalised status of the network such as total bandwidth available, total number of users connected, total

number of active routers, etc. In case of any runtime request, this information is collected from all the Enforcement Modules currently active in the network. The Protocol Manager forwards this information to the Policy Syntax Engine for a "fish-eye" view so as to decide as which Enforcement Module is optimal to provide the services.

The following components of the Enforcement Module are identified (Figure 8-15).

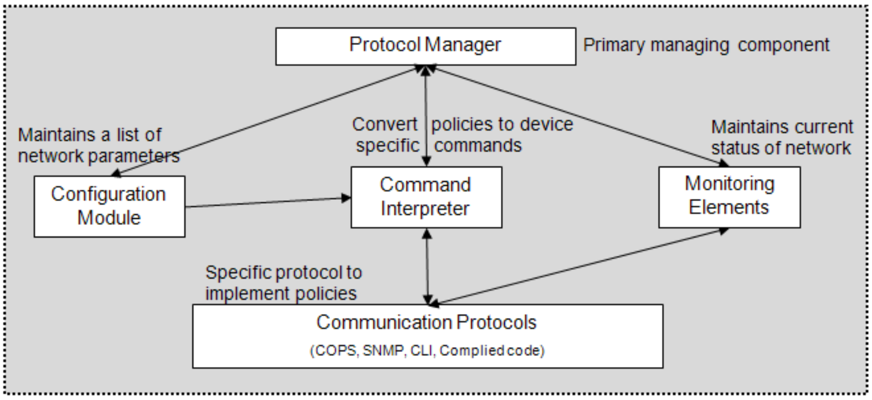

Protocol Manager
Primary managing component
Maintains a list of network parameters
Convert specific
policies to device commands
Maintains current status of network
Configuration Module
Command Interpreter
Monitoring Elements
Specific protocol to implement policies
Communication Protocols
(COPS, SNMP, CLI, Complied code)

Figure 8-15 Enforcement Module of proposed architecture

8.2.5.2 **Protocol Manager**

The Protocol Manager is a primary management entity for translating syntax-based rules into device-specific commands. It communicates with the Status Repository and coordinates with the Configuration Module, Monitoring elements and Command Interpreter to monitor, analyse and report the present state of its managed entities to the RPM.

The Protocol Manager maintains user-programmable features of the Enforcement layer which are represented by capabilities. This abstraction enables comparison between different types of network elements that use different features to perform the same type of operation. For example, different types of network elements support different types of QoS mechanisms. These mechanisms can be available either in software or in hardware. Therefore, the Protocol Manager must be able to abstract enough information from policies to allow for those differences while remaining specific enough to deploy the policy to the networking devices.

However, the main obstacle to overcome is that different network devices often use different programming models (i.e. protocol and command to implement a function), even if they are manufactured by the same vendor. Hence, many user-programmable features are often directly related to the environment in which they are used.

In this scenario, keeping a track of individual capabilities of each network element is required. For example, a QoS router that is able to perform advanced traffic conditioning functions for a lightly loaded network may not be able to perform the same functions for a heavily loaded network. Hence, the device knowledge not only incorporates the device capabilities as specified by the vendor, but also includes inter-device dependencies, device-service dependencies and other environmental constraints (such as a device cannot work at very high room temperature, etc.)

8.2.5.3 **Configuration Module**

The Configuration Module maintains a list of the network parameters and translates the high-level service requirements to network related values. For example, the gold service is defined as `bandwidth=1.2Mbps`, `delay=120ms`. The network values in the Configuration Module could be altered at runtime depending upon the current network conditions as long as they operate within prescribed values.

8.2.5.4 **Command Interpreter**

The Command Interpreter provides the most network specific expression of a policy. These resultant policies are enforced at the edge network elements (e.g. routers, firewall) via *Communication Protocols,* such as CLI, Complied code, SNMP, or user-define commands (discussed in Chapter 7). Communication Protocols are specific to particular network elements, however, these network elements do not need to be policy aware.

8.2.5.5 **Monitoring Elements**

The Monitoring Elements maintain information regarding the number of active components available in the network. They indicate real time information regarding the capability of the network (based on the number of active interfaces – such as routers). The Protocol Manager then utilises this information to define network parameters to be used by the Configuration Module.

The monitoring capabilities are summarised in order to:

- Automatically discover applications running across the network,
- Perform real-time and historical monitoring of application traffic,
- Monitor compliance of devices to provide and support the services,
- Feedback whether the policies have been applied successfully.

8.2.5.6 **Communication Protocols**

The Communication Protocols component acts as a container for the policies which are now ready to be deployed to the network devices. It represents a set of protocols supported by the Enforcement Module such as COPS, SNMP, CLI or Compiled code (discussed in Chapter 7).

8.2.5.7 **Analysis of Enforcement Module**

Since the deployment of the Enforcement Module is specific to the type of network, it is assumed that there are a number of Enforcement Modules across a large network managing a set of clustered network elements (or devices). This cluster is based on either:

- Type of network devices (for example, all Linux routers supporting DiffServ QoS),
- Geographical area (for example all the routers in Block C of Computer Science department), or
- Entities for which policies are deployed (for example, routers managing traffic of all Gold users).

8.3 Proof of Concept Implementation

The policies are sometimes thought as being *static* or *dynamic*. This notion is however incorrect. A policy is always a static entity. It may be parameterised, but it is still static since the policy must be invariant in order to produce predictable behaviour. However, if the policy is dynamic, it is impossible to predict what will happen when it is applied. On the other hand, network environments are inherently dynamic. This is in fact one important reason why policies are needed. Since the network environment can change frequently, it is desired to use a policy to ensure that a standard response can be provided. Hence, the policies defined in a management system cannot be dynamic while the management system requires a dynamic implementation of the policies to resolve any static and dynamic conflicts.

However, some of the policy interactions can be captured at the design or definition phase, while others can be captured only at runtime. Policies may depend on contextual data that may not be available at the design time, and testing all the possible combinations is not feasible. Also, at the design time, one might not be aware of all the policies that could exist in a system. In this proof-of-concept implementation, we will be focusing on static conflict detection. Any dynamic run-time requirement is handled by the Runtime Conflict Resolver (as shown in section 8.3.4 for negotiation of services).

This section presents our proof-of-concept implementation of the proposed architecture. Our implementation does not implement all the components identified in the architecture and is focused on four main aspects of the management framework:

i. To define a Policy Schema of the organisation with role, object and action hierarchies;
ii. To ensure the integrity of the Policy Schema when any new policy is introduced. The Policy Semantic Engine is required to check for any conflicts caused by the Separation of Duty and Chinese Wall Security policy;
iii. Distribution of policies from the Policy Syntax Engine to Enforcement Layer via COPS protocol; and,
iv. SLA negotiation between service providers.

In the following section, we will discuss the implementation of each of the aspects identified above:

8.3.1 Define Policy Schema

The primary objective of the Policy Manager is to define a Policy Schema of the organisation. The Policy Schema is maintained in a XML format that can be parsed and data structures can be created using XML processors. Different APIs, such as, Document Object Model (DOM) [209] and Simple API for XML (SAX) [210] have been standardised as a language-independent approach for representing and interacting objects in XML Schema i.e. to create, manipulate and access XML-based data structures.

The Java API for XML Processing (JAXP) [211] is one of the Java XML programming APIs [212] and provides the capability of validating and parsing a XML document. The JAXP provides access to DOM and SAX interfaces are defined as:

- The DOM interface parses an entire XML document and constructs a tree of nodes and maintains complete in-memory representation of the Schema.
- The SAX interface does not create a default tree and traverses the XML document on an event-basis (i.e. when a node is selected, the tree is traversed).

The JAXP DOM interface is preferred in our implementation. To generate a DOM tree representation of a XML Schema, a DOM-based XML processor is invoked to parse the required document. Objects under the DOM may be specified and addressed according to the syntax and the rules of the programming language (i.e. Java, in our case) used to manipulate them.

The extracted data may be used in the application logic by accessing the contents of the nodes of the DOM tree using DOM API methods. The steps for implementing Extended RBAC Model using XML tools are defined as:

1. Representation of RBAC data in an XML Schema
    a. Define a Data Type Definition (DTD) to represent the schema of the RBAC model, allowing expressiveness, flexibility and document readability;
    b. Create a XML document that captures the RBAC data;
    c. Validate the XML document for conformance to the defined DTD
2. Use the XML Schema content to implement the RBAC model

For the implementation purposes, we have defined the role hierarchy as shown in Figure 8-16.

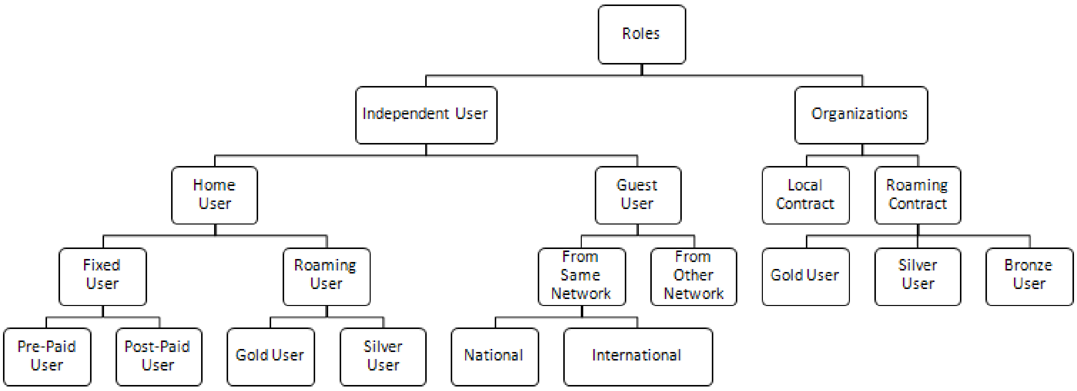
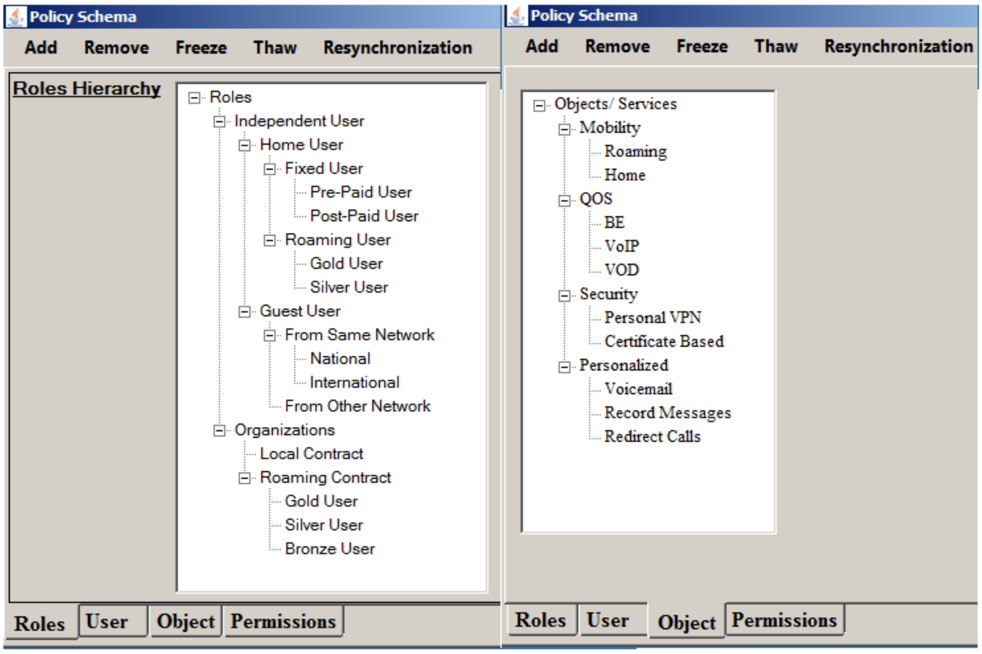


Figure 8-16 Conceptual Role Hierarchy

The Roles hierarchy consists of *Independent User* and *Organization.* The *Independent User* hierarchy maintains *Home User* and *Guest User*. The *Home User* hierarchy is divided into *Fixed User* and *Roaming User.* The *Fixed User* is categorised into *Pre-Paid User* and *Post-Paid User*; while *Roaming User* is categorised into *Gold User* and *Silver User.* The Objects hierarchy consists of services such as *Mobility*, *QoS, Security* and *Personalised*, which are further classified to *Roaming*, *Home* for *Mobility*, *Best Effort*, *VoIP*, *VOD* for *QoS*, etc. The implementation is shown in Figure 8-17.



- Roles
  - Independent User
    - Home User
      - Fixed User
        - Pre-Paid User
        - Post-Paid User
      - Roaming User
        - Gold User
        - Silver User
    - Guest User
      - From Same Network
        - National
        - International
      - From Other Network
  - Organizations
    - Local Contract
    - Roaming Contract
      - Gold User
      - Silver User
      - Bronze User

- Objects/ Services
  - Mobility
    - Roaming
    - Home
  - QOS
    - BE
    - VoIP
    - VOD
  - Security
    - Personal VPN
    - Certificate Based
  - Personalized
    - Voicemail
    - Record Messages
    - Redirect Calls

Figure 8-17 Roles and Object hierarchy

8.3.2 **Integrity Checks on Policy Schema**

Our proposed Extended Policy Schema has defined five *group operations* which can be performed on the Policy Schema, i.e. Add, Remove, Freeze, Thaw and Resync (section 4.2.4.3). The group operations will be initiated when a new role R is added. The sequence of steps is:

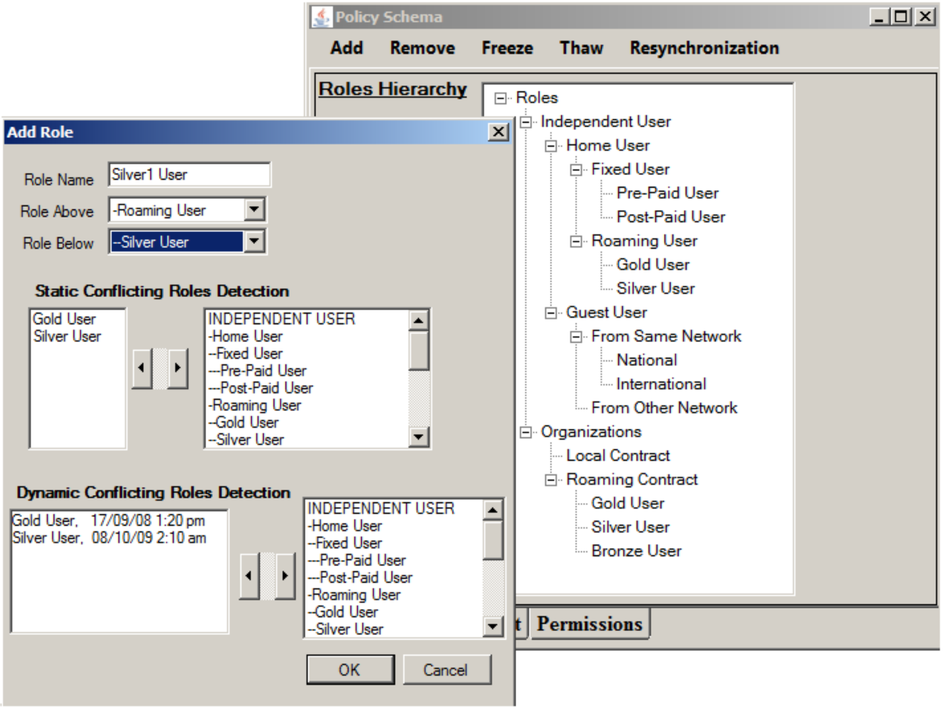
i. An *add* operation is initiated.

ii. Since any changes to the role structure will affect the Policy Schema, a *freeze* operation is initiated to prevent any changes to the Policy Schema by some other network manager.

iii. A *thaw* operation is performed where a consistency check is performed by the CRM to detect any conflicts in the role, action or objects hierarchy and to ensure that the resultant Policy Schema is in a consistent state. A resolution policy or a set of resolution policies are executed to eventually solve undesired feature interactions (e.g. Separation of Duty, Chinese Wall Security policy).

iv. Finally, a *resync* operation is performed to update the status of the Policy Schema and to notify all the related components.

Figure 8-18 shows when a Role *Silver1 User* is added the role hierarchy of the Policy Schema whose status is below *Silver User* and above *Roaming User*.



```
Roles
  Independent User
    Home User
      Fixed User
        Pre-Paid User
        Post-Paid User
      Roaming User
        Gold User
        Silver User
    Guest User
      From Same Network
        National
        International
      From Other Network
  Organizations
    Local Contract
    Roaming Contract
      Gold User
      Silver User
      Bronze User
```

Role Name: Silver1 User

Role Above: -Roaming User

Role Below: --Silver User

Static Conflicting Roles Detection

```
Gold User
Silver User
```

```
INDEPENDENT USER
-Home User
--Fixed User
---Pre-Paid User
---Post-Paid User
-Roaming User
--Gold User
--Silver User
```

Dynamic Conflicting Roles Detection

```
Gold User,  17/09/08 1:20 pm
Silver User,  08/10/09 2:10 am
```

```
INDEPENDENT USER
-Home User
--Fixed User
---Pre-Paid User
---Post-Paid User
-Roaming User
--Gold User
--Silver User
```

Figure 8-18 New role added in the Policy Schema

Each role in the role hierarchy maintains a cardinality and level in the hierarchy. Any addition of a new role, initiates a *Resync* operation on the Policy Schema and the resultant role hierarchy is compared with the RBAC DTD. Any inconsistency or conflicts are detected by the conflict detection technique and reported back to the administrator.

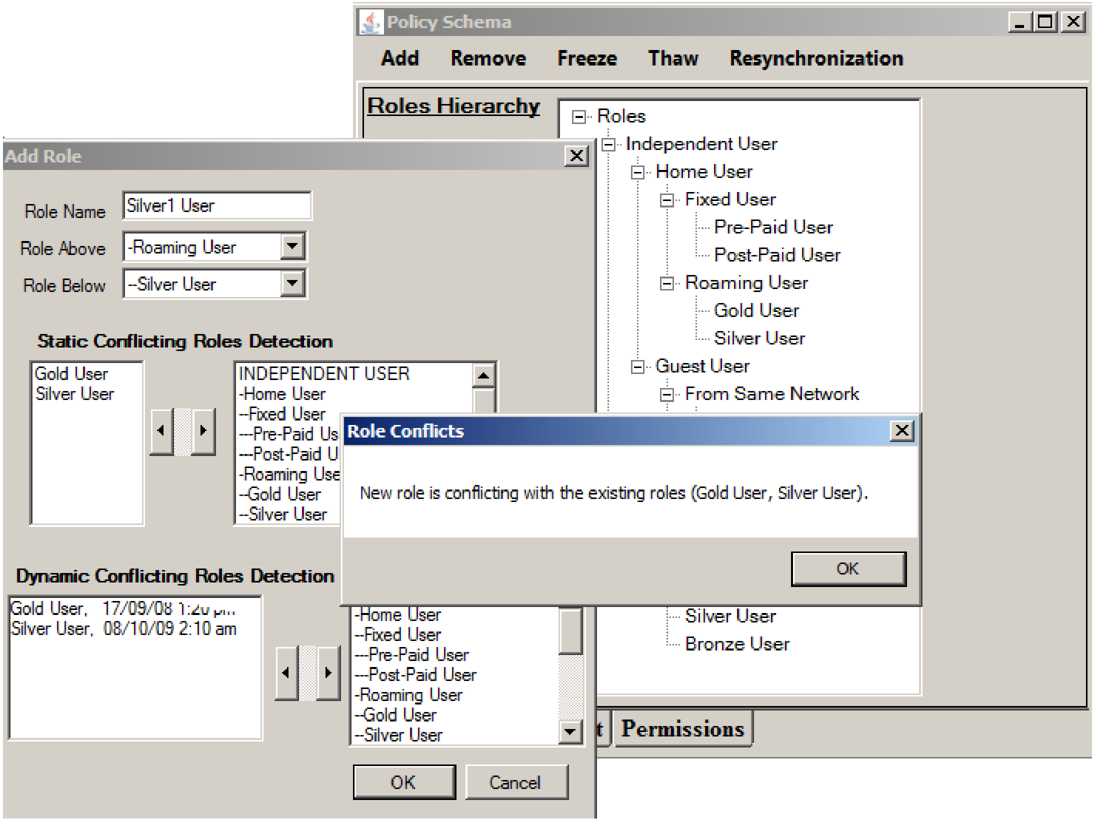
Listing 8-1 illustrates a sample RBAC DTD for defining role hierarchies [201].

```
<!ELEMENT Role_Graph (Application , (role )* )*>
<!ELEMENT Application (DB_Name , Server )>
<!ELEMENT DB_Name (#PCDATA )>
<!ELEMENT Server (#PCDATA )>
<!ELEMENT role (Name , Cardinality? , (Parent_Role?)* , (Child_Role?)* ,
                    (SSD_Role?)* , (DSD_Role?)* )>
<!ELEMENT Name (#PCDATA )>
<!ELEMENT Cardinality (#PCDATA )>
<!ELEMENT Parent_Role (#PCDATA )>
<!ELEMENT Child_Role (#PCDATA )>
<!ELEMENT SSD_Role (#PCDATA )>
<!ELEMENT DSD_Role (#PCDATA )>
```

Listing 8-1 RABC DTD

When an administrator attempts to upload a new policy, in principle this should be checked not only against other policies for the same user, but also against policies that the user might be subject to, e.g., due to their role in the enterprise, or due to the contractual obligations. We can however only check for static interactions, i.e., those that are inherent to the policies, independent of changing contextual data. This suggests the use of offline detection methods and filtering techniques. Any inconsistency detected needs to be reported to the administrator.

Figure 8-19 illustrates that a role *Silver1User* cannot be in the sub-tree of *Roaming User* and *Silver User*. More information on different types of conflict detection is presented in detail in section 5.6.3 including conflict detection in Authorisation, Propagation, Action Composition, Chinese Wall, Separation of Duty and Time Constraint policies.



Role Name: Silver1 User
Role Above: -Roaming User
Role Below: --Silver User

New role is conflicting with the existing roles (Gold User, Silver User).

Figure 8-19 Static conflict role detection

8.3.3 **Distribution of Policies**

After the policies have been defined, a distribution mechanism (discussed in detail in Chapter 7) is required to distribute the policies. The policies require an information base and a management protocol to carry the policies.

8.3.3.1 **Information Base (RBAC-PIB)**

Following our proposed work on extended RBPIM (Role-Based Policy Information Model – section 4.3.3) we will use a PIB defined to carry RBAC policies between the Policy Syntax Engine and the Enforcement Module. In the extended RBPIM model, we have extended the Constrained RBAC model to add *objects hierarchy* and *actions hierarchy*.

The framework is inspired by the IETF standards related to both policy representation and policy distribution by adopting a provisioning approach. The provisioning approach is based on three main elements [25].

a. A device independent policy information model, used for representing policies that can be reused across different devices (i.e. RBPIM);
b. A PIB which represents the policy assigned to a specific device. The PIB is generated from the device-independent policy model by a policy translation process. This translation takes into account the device capabilities, i.e. the mechanisms that the specific device supports for enforcing the policy;
c. A protocol (i.e. COPS-PR) specifically designed for supporting policy provisioning using the PIB structure for negotiating capabilities, transporting and installing the PIB into the device.

The framework is defined by introducing a device-independent RBAC information model and a RBAC-PIB. The mapping approach followed is based on XML-Schema specification. All the classes (e.g. *PolicyConditionVariable*, *PolicyActionValue*, *SimplePolicyCondition*, *SimplePoliceAction*, etc. subclasses), and even values are in conformity with our information model (section 4.3.3). Since we have extended the Constrained RBAC model, the RBAC-PIB comes as a natural choice for representing the information transferred from the Syntax Policy Engine to the Enforcement

Module, which is in IETF PBNM terms, from the PDP to the PEP during the provisioning process.

Listing 8-2 illustrates our PIB implementation based on the implementation of bandwidth broker [213] in Java.

```
public PRC getPRC(byte[] prcIndex, int offset, int length) {
    String key = new String(prcIndex, offset, length);
    return (PRC) this.prcs.get(key);
  }

  public void putPRC(byte[] prcIndex, PRC prc) {
    putPRC(prcIndex, 0, prcIndex.length, prc);
  }

  public void putPRC(byte[] prcIndex, int offset, int length, PRC prc) {
    String key = new String(prcIndex, offset, length);
    this.prcs.put(key, prc);
  }

  public PRI getPRI(byte[] prid) {
    if (prid == null) return null;
    PRC prc = getPRC(prid, 0, prid.length - 1);
    if (prc == null) return null;
    return prc.getPRI(prid[prid.length - 1]);
  }
```

.

Listing 8-2 COPS PIB

8.3.4 Service negotiation between service providers

The Inter-domain Manager interacts with the monitoring services to gather present status of the network elements which are providing services to the guest users, and communicates with the SLA Negotiator. The roaming contracts established between service providers allow them to publish and share their offered services based on the Web Services.

For implementation purpose, any request generated from the network operator is encapsulated using the SOAP which carry XACML attributes to initiate a request/response dialog (as discussed in Section 8.2.4.6). Listing 8-3 SOAP HandlerListing 8-3 shows a SLA request from TelecomDomainA to allocate total bandwidth of 1Mbps for a time period of 24 hours (i.e. 2pm 22 September 2007 to 5pm 22 September 2007). Listing 8-4 shows the Java implementation to process the SOAP request.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
      <SOAP-ENV:Body>
            <stationName>TelecomDoaminA</stationName>
            <stationID>1</stationID>
            <CACert>TelecomCA</CACert>
            <slaRequest>123</slaRequest>
            <serviceType>EF</serviceType>
            <totalBW>10000</totalBW>
            <requestBW>1000</requestBW>
            <start>122204880000</start> <!-2007-09-22 14:00:00->
            <end>1222016400000</end>  <!-2007-09-22 17:00:00 ->
      </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 8-3 SOAP Handler

```
public boolean handleRequest(MessageContext context) {
        try{
            SOAPMessageContext smc = (SOAPMessageContext) context;
            SOAPMessage msg = smc.getMessage();
            SOAPPart sp = msg.getSOAPPart();
            SOAPEnvelope se = sp.getEnvelope();
            SOAPHeader shd = se.getHeader();

            SOAPBody sb = se.getBody();
            java.util.Iterator childElems = sb.getChildElements();
            SOAPElement child;
            StringBuffer message = new StringBuffer();
            while (childElems.hasNext()) {
                child = (SOAPElement) childElems.next();
                message.append(new Date().toString() + "--");
                formLogMessage(child, message);
            }

            System.out.println("Log message: " +
message.toString());
        } catch(Exception e){
            e.printStackTrace();
        }
        return true;
    }
```

Listing 8-4 SOAP Request

8.4 Analysis of Proposed Architecture

In section 3.3, we have proposed a set of characteristics (or requirements) which must be exhibited by any PBM system in order to provide ABC services. We will discuss each of proposed requirements and analyse whether the proposed architecture fulfil those criteria.

i. ***Modular***

In the proposed layered model, we have discussed that each layer (namely, Policy Model layer, Semantic layer, Syntax layer and Enforcement layer) needs to be configurable to allow organisations to *choose* the best available approach/technology.

Similarly, the proposed architecture is modular by design where the inner workings of each functional component (i.e. Policy Model layer, Policy Semantic Engine, Policy Syntax Engine and Enforcement Module) is derived based on the functionalities required and the methodology implemented.

ii. ***Flexible***

The Modularity and Flexibility properties are related. The modularity requires the PBM system to allow organisations to select optimally suited approach or technology. The flexibility property requires that the PBM system should be able to cope with different underlying devices, technologies and services, and should be adjustable to the changes in protocols, devices and modifications. However, whereas modularity indicates a replacement of any specific functional component, flexibility indicates a coexistence of both the functional components.

As discussed in the analysis of the Policy Semantic Engine (section 8.2.3.5), multiple Policy Semantic Engines can be deployed in a single PBM system. It is based on the requirements of either choosing specific methodology for detecting conflicts for specific management areas or to "double-check" policies by running different conflict detection methods on an entire set of policies.

iii. ***Adaptable***

The adaptable functionality requires that the PBM system should be able to make decisions associated with the user's current context and should be able to adapt based on the current status of the network. Furthermore, a request/reply methodology should be supported, for in case any active network fails, then the management system should be able to detect and resolve the issue.

In the proposed architecture, the Monitoring Elements in the Enforcement Module maintains the current status of the network elements and updates the Status Repository. In case of any network element failure, the Monitoring Elements respond back to the Protocol Manager using management protocols such as COPS or SNMP. The Runtime Conflict Resolver together with Intra-domain and Inter-domain Managers, and SLA Negotiator allow home and guest users to send out-of-bounds requests based on their current context.

iv. ***Clustering***

The clustering requires that the network elements and users should be represented as a set of entities rather than individuals, since it is impractical in large-scale and dynamic environments to define policies for individual entities.

The proposed architecture supports the extended RBAC Policy Schema as a XML Schema to represent the hierarchical structure of the organisation.

v. ***Scalable***

The PBM system should be scalable to cater for organisation (enterprises or service providers) which is growing in size either conceptually or physically. As discussed in modularity, the proposed architecture is scalable by allowing functional components to be increased or decreased in number. Furthermore, each functional component maintains its own Repository. For example, the number of Policy Syntax Engines can be altered, where each engine maintains a dedicated Syntax Repository. The modularity and scalability is also achieved for each Repository as discussed in sections 8.2.1.3, 8.2.3.5, 8.2.4.8 and 8.2.5.7.

vi. ***Conflict Detection***

The PBM system should support a policy conflict detection mechanism. The proposed architecture has introduced Policy Syntax Engine to detect and resolve any conflicts inherent in the definition of the Policy Schema. Furthermore, any runtime conflict (such as user's out-of-bounds request and any network node failure) is managed by the Runtime Conflict Resolver in the Policy Syntax Engine.

vii. ***Interoperable***

The PBM system should be interoperable to be able to communicate with other service providers to authenticate and authorise users, and to negotiate different service levels.

Interoperability is an important aspect in providing ABC services and has been offered in the proposed architecture in following manner:

- A technology and implementation neutral XACML language is used for inter-domain communications. The service providers will be able to communicate with each other if they share a common XML Schema.
- The services offered by the service provider together with any authentication and authorisation messages are communicated via Web Services. Hence, no proprietary based API is required. The RPM exposes the services offered and simply reply to any requests while hiding all the internal details as how the request was processed internally.
- In case a service provider is not able to honour the SLA, the SLA Negotiator tries to devise a mutual agreeable policy which is acceptable to both the service providers.

Hence, it is clear from this analysis that the proposed architecture (and the proposed model on which it is built upon), fulfils all the requirements of a PBM system in order to provide ABC services.

In our implementation, we followed our proposed Extended RBAC model for Policy Schema, our proposed conflict detection technique based on First-order logic and Composite Mapping for Semantic analysis, while using existing COPS and XACML protocols for distribution of intra- and inter-domain policies respectively. The interoperability is achieved by using XACML based Web Services to support policy negotiations.

Similar to the proposed model, the proposed architecture can also be viewed as an elaborate extension of IETF PBNM architecture. In this case, the Policy Semantic Engine and the Policy Syntax Engine can be combined into a single component to be a PDP and without differentiating roles of any other specific component proposed in this architecture. The Enforcement Module can be treated as a PEP, while all the proposed Repositories can be combined into a single entity of Policy Repository. The Policy Model Layer without a Policy Manager can be defined similar to the IETF proposed Policy Management Tool.

In the following section, we will present a case study of the interworking of messages between the proposed network management framework and a roaming mobile user.

8.5 Case Study

This case study (Figure 8-20) considers a typical scenario illustrating how the service providers may interact to provide ABC services to a roaming mobile user during a handoff. This scenario outlines message exchanges between service providers (home and guest networks) and considers both policy-managed and policy unaware mobile clients (discussed in Chapter 9).

We assume that the home and guest networks have implemented the proposed extended PBM architecture. However, they can choose any approach for semantic analysis (Chapter 5), syntax analysis (Chapter 6) and information model (Chapter 4). The home network in this case study performs semantic checks based on our proposed conflict detection method (section 5.6.3), information model as extended RBPIM (section 4.3.3) while the syntax approach uses the COPS protocol for intra-domain communication and XACML for inter-domain policy negotiations (Chapter 6). SNMPv3 is used for communication protocols (Chapter 7).

We also assume that the mobile user is authenticated by NAI-based user authentication and both the networks support some form of class-based service differentiation (e.g. DiffServ) to mark high priority traffic. All XACML communication between SLA Negotiators is digitally signed and encrypted.

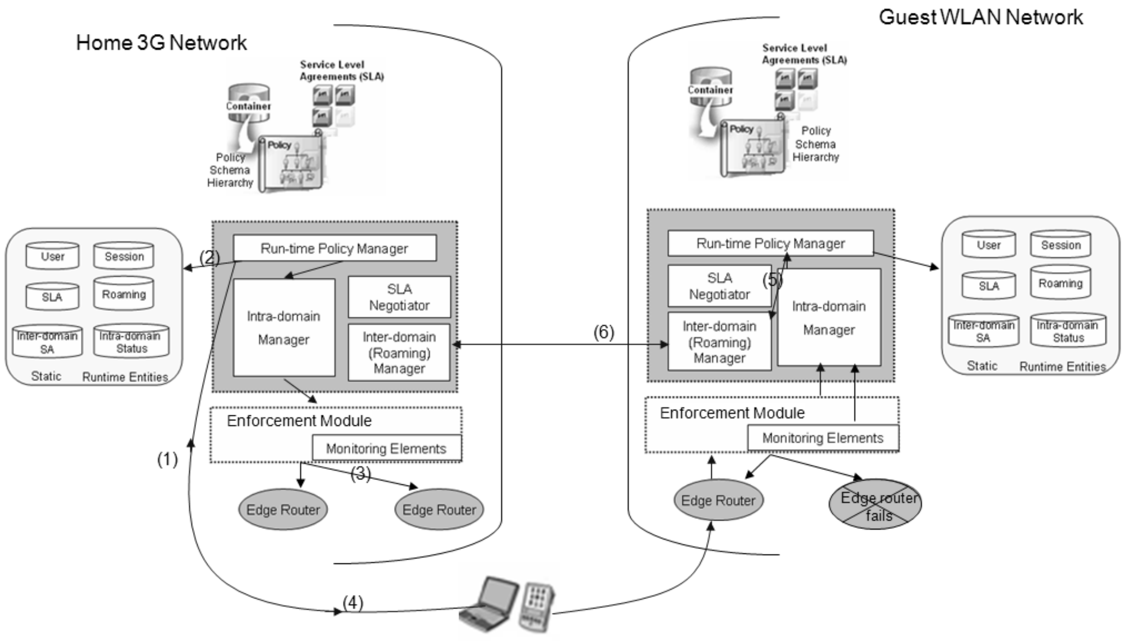


Figure 8-20 Message flow between managed networks to provide ABC services to mobile entity

Initially, the mobile user sends an authentication request (1) to the home RPM using his NAI (e.g. john@networkA.com). As the authentication process is a web-based service, both policy aware and policy unaware mobile clients can be authenticated. The RPM then authenticates the user from User (Static Entities) database, initiates a Session in Runtime Entities (2) and informs the Enforcement Module which then configures the edge routers (3).

The mobile device downloads the user's profile (i.e. gold) and configures itself accordingly (4). In case the mobile device is policy unaware, the managed routers control the data flow from the mobile device and handles any out-of-bounds traffic according to the policies from the RPM.

When the Network Access Devices of the mobile client, for instance, presently connected with the CDMA network, detects a new WLAN guest network, and the Optimum Network Detector selects it as a better access interface, the mobile client starts the process of handoff. When the edge router of the guest WLAN network requests an authentication credential, the mobile device submits its user's NAI (5). Considering the NAI domain (@networkA.com), the guest RPM forwards this credential to its Inter-domain (Roaming) Manager (RM) which is forwarded to the home RM (6). After successful authentication the home RPM changes the user’s Session state and forwards the user’s profile to the guest network.

The SLA database and SLS information in Static and Runtime Entities helps the networks ensure that the user’s profile is honoured at the guest network. To make this case study more dynamic, we assume that the bandwidth negotiated between networks is based on a per-user basis, however in a real environment, a bulk of SLS is negotiated and the RMs utilise available resources in real time. The guest network then configures its edge router to mark and provide assigned priority to the registered roaming user.

In case the user wants a higher profile bandwidth for the next hour (e.g. john requests platinum profile for video conferencing), it sends a request to the home RPM via a Web Service. If this request is accepted, the home Inter-domain Manager then informs guest Inter-domain Manager. If the guest RPM also accepts the request, after considering the Intra-domain Manager and network’s present status, it then configures the edge routers. A successful notification is sent to the home RPM which then notifies the user. The mobile device then downloads an updated profile from its home network.

Consider another case where an edge router of the guest network fails. The Monitoring Elements then inform their RPM and the Intra-domain Manager selects the next best available router and continues providing agreed network services. However, if the guest network is not able to provide agreed services to the roaming guest user, the SLA negotiators interact to provide next best available services.

Chapter 9

Policy Managed Mobile Client

In Chapter 1, we have identified the objective of our research work – to provide ABC service over underlying heterogeneous mobile environments. To achieve this objective, we state that the service providing networks require a *management framework* for optimal utilisation of their network resources, whereas a *policy-managed mobile client* is required to provide ABC services to the mobile users across networks.

In Chapter 3, we further defined the requirements for the network management framework and the policy-managed mobile client, and proposed their respective models. Following which, in previous chapters we have discussed the proposed layered model for centralised management of networks and discussed each layer in detail including the proposed architecture.

In this chapter, we will propose a new architecture for a policy-managed mobile client and discuss its implementation and test results in both Linux and Windows environments. In summary, we will describe an operating environment for the client model proposed in Chapter 3, which is based on Mobile-IP infrastructure and support policy enabling mechanisms including internal and external profiles and mobility decisions based on a user's current context. We have also introduced a new *Infrastructure* (I) parameter to assess the suitability of current service offerings from different service providers to select the best available option. In the real-world simulation of a proposed architecture, we will consider cases where the mobile user is moving between heterogeneous networks (i.e. WLAN and CDMA2000 for testbed purposes) and under different security requirements. We will further present a new network selection algorithm to optimise the overall handoff delays and to select a best access interface to support ABC services from the client side.

9.1 Introduction

Traditional mobile networking applications focus on ensuring continuity of services under the assumption that devices connect to the infrastructure using a single access technology. However, the rapidly growing number of multi-access mobile communication devices (such as laptops, Personal Digital Assistances, cell phones, etc.) and heterogeneous access networks (such as wide-area, local-area and wireless networking technologies) each with different characteristics are required to coexist in a complementary manner. These access technologies vary in bandwidth, delay, communication range, power consumption, security, reliability, implementation complexity, end-to-end cost, end-user cost and several other aspects. The existence of multiple access networks have made it possible to develop new wireless services that can be used anywhere and anytime using any carrier, operator or service provider to allow users to be *always connected.*

The notion of *best* is often based on a number of user and application dependent factors such as personal preferences, device capabilities, current running applications and their QoS needs, current running sessions and their connectivity needs, network resources, and network coverage. We have referred to them collectively as a *user's current context*.

However, since the multi-access devices and plethora of access networks offer users with more flexibility and choice in communications, it also imposes new demands on network management and interoperability. Furthermore, the range of services offered by these networks is geographically selective e.g. high performance and less expensive WLAN services are useable only within small areas while slower and costlier cellular networks provide coverage to much larger areas. Hence, when a user moves between different physical locations, it may become necessary - due to limited coverage or bad network performance - to perform a handoff to another network to be *always connected.* Similarly, if a better network becomes available, a handoff should automatically be initialised to the network offering the *best* price/performance ratio subject to the user's context to provide *always best connected* services. As a result, whether the handoff is intra-network (within a single technology) or inter-network (between different technologies), a mobile node moving from one location to another, needs to change its point of attachment.

9.2 Handoff

A handoff is a process by which the ongoing communications and application's current sessions of a mobile node is transferred from one point of connection in a network to another point of connection in the same or different network. Normally, users are required to take an active part in the handoff process and are often required to manually select which network to use. Moreover, during or immediately after a handoff, it is very common that packet losses and delays occur due to signalling and location updates. For applications based on HTTP this delay is not of vital importance, e.g. waiting one or two seconds extra when downloading a web page is not critical. However, for security sensitive applications, this handoff is fatal as it requires breaking the existing security connections and dropping any current sessions (e.g. download of a file from a secured VPN tunnel needs to be terminated and restarted). Furthermore, creating a new connection generally requires manual input from users and requires a re-exchange of security attributes such as username-passwords, digital certificates, etc. Similarly, for delay sensitive applications such as real-time media streaming, any delays and packet losses are extremely crucial.

9.2.1 Terminologies

In the literature, there are frequent uses of a number of terms defined to identify different types of handoffs, and are outlined here for clarity and simplicity. Generally, the term *handover* is used to describe when a mobile node changes its point of attachment, while the term *handoff* is referred to the entire process of performing a handover. However, these two terms are often used interchangeably and should be assumed as the same. Similarly, there are different types of handoffs defined.

In [214] Vidales identified that a *homogenous handover* occurs when a mobile node moves from one access router to another, where both access routers belong to the same technologies, while a *heterogeneous handover* occurs where access routers belong to different technologies. Furthermore, when the mobile device moves from the access router which belongs to a technology with smaller coverage but more bandwidth than the new access router, it is termed as *upward handover* (e.g. handover from WLAN to CDMA), while in the opposite case, it is termed as *downward handover*. The term *inter-system* handover is usually used when a mobile device moves between two independent systems – controlled by different network operators, while if they are part of the same network operator, the handover is termed as *intra-system* handover (Figure 9-1). In [215] Stemm and Katz introduced terms: *vertical handoff* for handoff between different networks, and *horizontal handoff* refers to the handoff within the same network. A *soft handoff* is defined as when the mobile node listens (i.e. receive packets) on both the interfaces simultaneously while performing the vertical handoff. In contrast, a *hard handoff* occurs when the mobile node listens exclusively on one interface and expects packet losses.

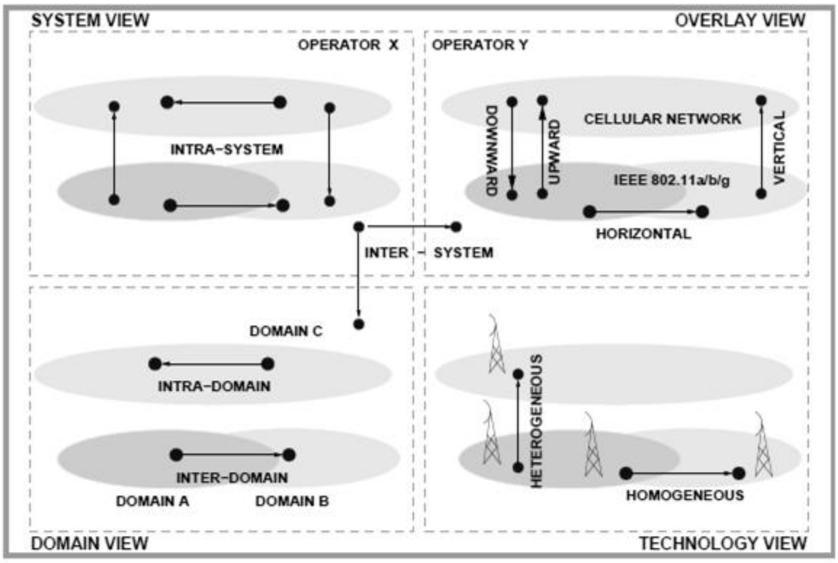


Figure 9-1 Handover taxonomy and views

9.2.2 **Coupling**

The inter-system handover or vertical handoff was considered by the 3GPP Technical Specification Group (3GPP TSG) working group [216]. The group drafted a feasibility study in which they presented different levels of integration between access routers of different networks, according to the component where coupling takes place. The group identified different types of couplings especially between a 3G network and WLAN networks and are discussed as follows:

a. ***Open Coupling***

There is no real integration effort between two or more access technologies. Thus, separate processes for handoff can be used. However the billing system is shared between networks. These models do not enable seamless vertical handoffs and when a mobile node changes its current point of attachment, the ongoing sessions are terminated.

b. ***Tight Coupling***

The key characteristic of this model is to make the general access networks (e.g. WLAN) to appear to the 3G core network as another 3G access network. The WLAN network would emulate functions which are natively available in 3G radio access networks and the WLAN gateway will upstream data to the 3G core network as either a Packet Control Function (PCF) for a CDMA2000 core network, or as a Serving GPRS Support Node (SGSN) for a UMTS network. The WLAN gateway hides the details of a WLAN network from the 3G core, and implements all the 3G protocols (e.g. mobility management, authentication, etc.) required in a 3G radio access network.

Furthermore, the mobile nodes are also required to implement the corresponding 3G protocol stack on top of their standard WLAN network cards, and switch from one physical layer to the other as needed. All the traffic generated by clients in the WLAN network is then injected using 3G protocols in the 3G core. The different networks would share the same authentication, signalling, transport and billing infrastructure, independently from the protocols used at the physical layer on the radio interface.

c. ***Loose Coupling***

Similar to the tight coupling approach, the loose coupling approach introduces a new element in the WLAN network, i.e. a WLAN gateway. However, the gateway does not have any direct links to the 3G network elements such as PCF or SGSN and connects directly to the Internet (Figure 9-2). The WLAN gateway may include mobile users that have signed on locally, as well as mobile users visiting from other networks. This approach is called loose coupling since it completely separates the data path in WLAN and 3G networks, and high speed WLAN data traffic is never injected into the 3G core network while the end user still achieves seamless access [217].

In this approach, different mechanisms and protocols can handle authentication, billing and mobility management in the 3G and WLAN portions of the networks. However, for seamless handoffs, they need to interoperate.

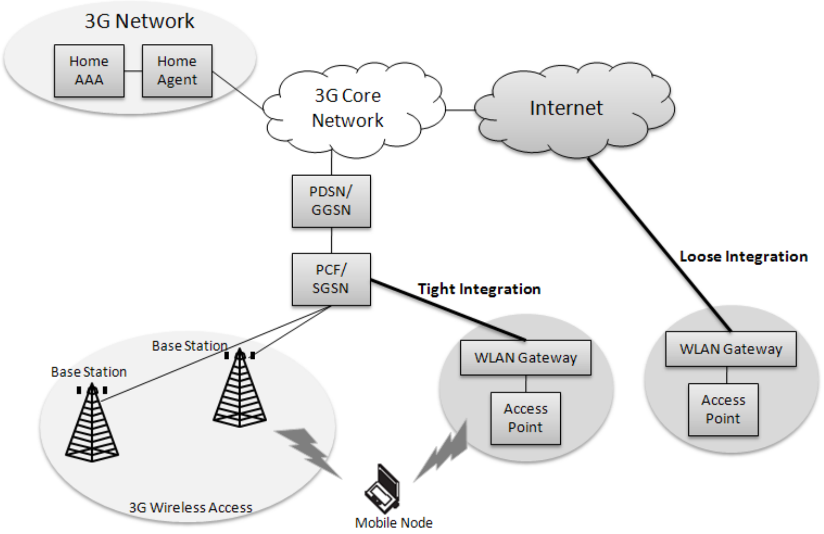


Figure 9-2 Types of couplings for Vertical handoff

It is evident that the tight integration approach presents several disadvantages. Since 3G core networks directly exposes their interfaces to the WLAN network, the same operator must own both the WLAN and 3G parts of the network. Hence, independently operated WLAN islands could not be integrated with 3G networks. By injecting WLAN traffic directly into the 3G core, the configuration and design of 3G network elements such as SGSN, need to be modified to sustain the increased network load. Furthermore, the configuration of mobile client devices such as WLAN network cards needs to be modified to implement the 3G protocol stack and mandate the use of 3G-specific authentication mechanisms. Hence, with the advantages of loose coupling which allows independent deployment and traffic engineering of WLAN and 3G networks together with the choices of authentication and security protocols, it has been our preferred choice for vertical handoffs.

9.3 Mobile-IP

It is in general consent that any handoff (vertical or horizontal) is to be performed at the network layer. The Open Systems Interconnection Reference Model (OSI Reference Model) [218] separates data communication functionality into different layers. The lower two layers (i.e. physical and link layers) are strongly attached to the specific access technology in use and maintain signalling and control messages. Hence, layer-2 can be used for horizontal handoffs where same technologies share the same signalling and control messages and current connections can be transferred from one local point to another. For example, a CMDA voice call can be transferred from one PCF to another.

9.3.1 Network layer and IP Paradox

The higher layers (i.e. transport layer and above) are tightly integrated to the application and focus on presenting the data. The IP layer (or network layer) can facilitate the integration of heterogeneous networks since it includes control and signalling (e.g. addressing, routing, encapsulation of packets, etc.) common to every technology. Although the network layer is network-dependent, however, due to the explosive growth of TCP/IP-based applications and services, most of the technologies have converged on IP. Most of the current heterogeneous radio networks, including WLAN, UMTS, CDMA, etc. use different mappings of IP and hide the signalling from the network layer making it a perfect choice for vertical handoffs. However, the IP layer does not provide mobility.

The essential mobility problem lies in the dual roles of an IP address used as both the identity and the physical location of a host (i.e. a mobile node). An IP address is used to identify a node’s location, and to forward packets to it. However, higher level protocols, such as TCP and UDP, also uses the IP address along with a port number to identify the data streams. This causes a problem for IP mobility as the IP address determines how packets are forwarded and must be changed when a node moves so that the packets continue to be forwarded to it. At the same, the IP address must remain constant, so any open TCP or UDP sessions are not disturbed. This is the IP paradox.

9.3.2 Mobile-IP Protocol

To solve the IP paradox, the IETF Mobile Working group standardised the Mobile-IP protocol [219] working on the IP layer for both IPv4 [220] and IPv6 [221]. It is believed that Mobile-IP is the oldest and probably the most widely known mobility management protocol.

Mobile-IP associates two different IP addresses with the mobile node for different roles. The mobile node has a permanent IP address, called its *home address,* which identifies the mobile node when it is connected to its home network and is used in packets to identify the TCP and UDP data streams. When the mobile node moves away from its home network, it must obtain a new (temporary) IP address at the foreign network so packets can continue to be routed to the mobile node's current physical location. This new IP address, called the *care-of-address*, is either provided by a *foreign agent* in the foreign network or is directly acquired by the mobile node (*collocated care-of address*). In the case of a care-of address provided by the foreign agent, it is the IP address of the foreign agent itself, while in co-located care-of address, it is acquired in some other way (e.g. from a DHCP server at the foreign network). This combination of two IP addresses, one to identify the TCP and UDP streams, and one to identify the point of attachment, solves the IP paradox.

To receive the packets destined for a mobile node to its (co-located) care-of-address, the *home agent* introduces a level of indirection by keeping binding updates of the current (co-located) care-of-address. Any packets received by the home agent on behalf of mobile node are encapsulated by the mobile node's most recent (co-located) care-of address as destination address and sent using IP-in-IP, Generic Routing Encapsulation (GRE) or Minimal tunnelling. In case of care-of address, the foreign agent decapsulates the packets and forwards them to the mobile node, while in co-located care-of address mode, the mobile node serves as the endpoint of the tunnel and performs the decapsulation itself.

Figure 9-3 presents a typical case scenario where a mobile node with a home address 4.4.4.4 is at home network (4.4.4.4/24) and downloading a large file from 20.20.20.20 and maintains a FTP session of 4.4.4.4-20.20.20.20:20 (1). Since, the mobile node is at home, no mobility protocol is required.

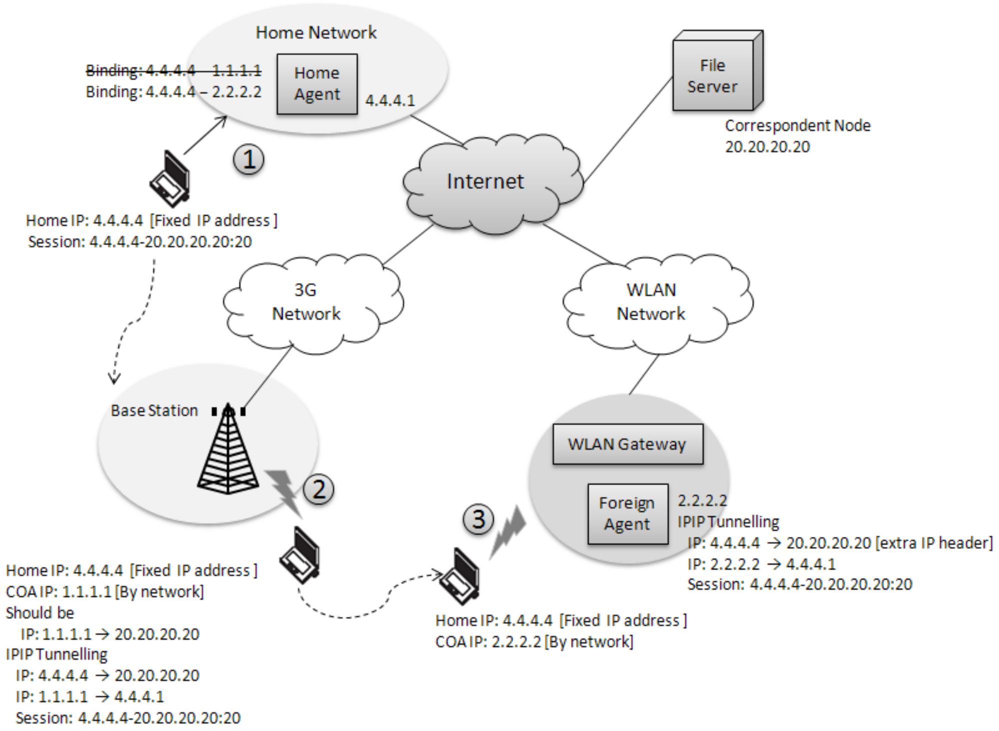

Home Network
~~Binding: 4.4.4.4 – 1.1.1.1~~
Binding: 4.4.4.4 – 2.2.2.2
Home Agent
4.4.4.1
File Server
Correspondent Node
20.20.20.20
Internet
1
Home IP: 4.4.4.4 [Fixed IP address]
Session: 4.4.4.4-20.20.20.20:20
3G Network
WLAN Network
Base Station
2
WLAN Gateway
Foreign Agent
2.2.2.2
IPIP Tunnelling
IP: 4.4.4.4 → 20.20.20.20 [extra IP header]
IP: 2.2.2.2 → 4.4.4.1
Session: 4.4.4.4-20.20.20.20:20
3
Home IP: 4.4.4.4 [Fixed IP address]
COA IP: 1.1.1.1 [By network]
Should be
IP: 1.1.1.1 → 20.20.20.20
IPIP Tunnelling
IP: 4.4.4.4 → 20.20.20.20
IP: 1.1.1.1 → 4.4.4.1
Session: 4.4.4.4-20.20.20.20:20
Home IP: 4.4.4.4 [Fixed IP address]
COA IP: 2.2.2.2 [By network]

Figure 9-3 Mobile-IP Protocol

While downloading, the mobile node moves to a 3G network and is allocated a co-located care-of address of 1.1.1.1 (2). Conventionally, the FTP session should break since the IP address/port number required for the session is changed. However, the Mobile-IP protocol sends a registration request to the home agent (4.4.4.1). The home agent then adds an entry in the binding table as 4.4.4.4-1.1.1.1 and reply successful registration to the mobile node. The home agent then forwards all the packets destined to the mobile node to its current co-located care-of address. Every data packet communicated between the home agent and the mobile node is encapsulated with an extra IP header. The outer IP header (IP-in-IP encapsulation) sent from the mobile node contains the source address as the new co-located care-of address and destination to the home agent (i.e. 1.1.1.1 → 4.4.4.1) while the FTP session is maintained with 4.4.4.4-20.20.20.20:20.

While still download the file, the mobile node is on the move and detects a new WLAN connection available with a foreign agent (2.2.2.2) and sends a registration request to it (3). The foreign agent forwards the registration request to the mobile node’s home agent with its own care-of address (i.e. 2.2.2.2). The home agent then updates the binding entry to 4.4.4.4-2.2.2.2, and sends the registration reply back to the foreign agent. In this case, the encapsulation is performed by the foreign agent while still maintaining the session as 4.4.4.4-20.20.20.20:20.

The Mobile-IP standards documents include the following:

- RFC 2002 [219] and RFC 3344 [220], define the Mobile-IP protocol itself;
- RFC 3775 [221] defined Mobile-IPv6 protocol;
- RFC 2003 (IP-in-IP) [222], RFC 2004 (Minimal Encapsulation) [223] and RFC 1701 (GRE) [224], define different types of tunnelling used in Mobile-IP
- RFC 2005 [225], describes the applicability of Mobile-IP
- RFC 2006 [226], defines Mobile-IP Management Information Base (MIB). The Mobile-IP MIB is a collection of variables within a node which implements Mobile-IP that can be examined or configured by a manager station using SNMPv2 [151].

It is clear that the Mobile-IP protocol maintains active network sessions. However, with every handoff it requires message exchanges and registration between home agent, mobile node and if applicable, foreign agent. Hence, any handoff procedure involves a set of messages to notify all the related entities of a particular connection that a handoff has been performed, and that the connection has to be redefined. However, the architectural issues related to methodology, message exchanges, control signals, and software/hardware elements involved in rerouting the connection becomes more challenging especially if the mobile device is moving between different interfaces, homogenous and heterogeneous architectures, and/or performing *horizontal* and *vertical handoff.* The mobility requirements of mobile users also change with various scenarios. Such users typically want to connect to the public or private networks most convenient to them at the time of connection.

As discussed in section 2.4, there has been considerable research work in the mobility management area to ensure that the handoff is seamless (i.e. minimising the number of dropped packets) and transparent (i.e. without any intervention to the user's current sessions and requiring any input from the user). However, most of the research work has mainly focused on how to preserve the communication and to manage location updates, while the management of handoff to handle security, QoS and the context of a user to cater for runtime requests in making handoff decisions is still a challenging problem.

In the following sections, we will revisit the proposed layered-model introduced in section 3.6.2 and then propose a new architecture based on the proposed model. The implementation details and performance analysis are also presented.

9.4 Client Model

While proposing a model for the policy-managed mobile client, we have identified that to provide a seamless handoff, the handoff latency must be low enough to not disturb a user's currently running applications. In section 3.6.1, we have identified the requirements which *must* be satisfied by the proposed mobile client so as to provide ABC service to the mobile user from the client side. The network support requirements are discussed in section 3.2.

9.4.1 Requirements

As discussed in section 3.6.1, the framework for a policy-managed mobile client should:

1. Support seamless and transparent handoff of user's active sessions i.e. an efficient way for *how-to-handoff,*
2. Be able to detect whether a handoff is required and select the best available interface,
3. Be able to select an optimum time to handoff if any handoff is required, i.e. an efficient way for *when-to-handoff,*
4. Support the protocols of the services to be offered to the mobile user,
5. Support a profile-based management system to allow a mobile user to input personal preferences (such as priority of access networks based on bandwidth, subscription and cost) and security profiles for connection at different locations.
6. Be able to, if supported, request the service providing network for run-time requirements of the mobile user.

Based on the identified requirements we have proposed a new framework for a policy-managed mobile client.

9.4.2 **Proposed Model**

In retrospect, we will briefly discuss the proposed client model (Figure 9-4). The proposed model is a four-layer stack: Policy layer, Mobility layer, Enforcement layer and Network layer.

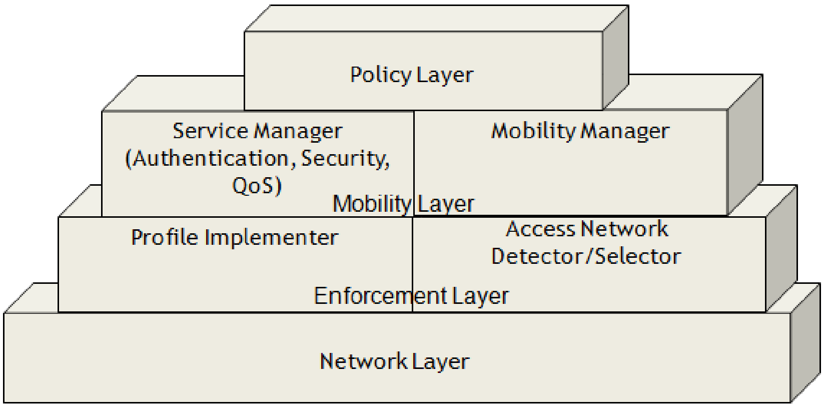


Figure 9-4 Proposed model of Policy-managed Mobile Client

a. The *Policy layer* provides support for profile management. It manages profiles downloaded from external sources (e.g. service providing networks or organisations). In the IETF PBNM terminology, it acts as a PEP where policies from the managed network can be directly installed. However, the mobile device does need to support the COPS protocol.

b. The *Mobility layer* provides seamless mobility support to the mobile node. It contains a Mobility Manager and a Service Manager. The Mobility Manager provides support for mobility protocols (e.g. Mobile-IP), and the Service Manager is a set of *managers* where each manager provides support for a specific service (e.g. QoS, security, VoIP, etc.). To support a service, a manager also needs to support the protocol associated with the service, for example, to support QoS a support for DiffServ or RSVP protocols is required. Similarly, to support VoIP a support for SIP or H.323 protocol is required.

c. The *Enforcement layer* implements the *internal* and *external* profile. The internal profile is created by the mobile user as personal preferences such as prioritising an access network based on cost, while the external profile is downloaded from the network and contains information such as services offered by the network, SLA information and network usage cost, etc.

d. The *Network layer* implements the profile policies into low-level device specific commands. It provides a direct interface with the network adapters of the mobile device. The Network layer also interacts with radio devices to collect information, such as received signal strength, speed of the mobile user, etc., which helps the *Network Detector/Selector* module to make handoff decisions.

Based on the proposed layered model, the following section presents the proposed architecture and its implementation details.

9.5 **Proposed Architecture**

Figure 9-5 represents the architectural framework of the proposed model. To demonstrate the working of the architectural components, we assume a real-world scenario where a ubiquitous coverage of 3G services is available and is provided by a service provider *3GNZ.* A set of hotspot areas is available where WLAN services are provided by a WiFiNZ service provider. We assume that there is no SLA between respective companies and a mobile user is required to buy separate subscriptions. We also assume that each service provider maintains a PBM system and is able to honour the SLA with their users.

The mobile user *John* works in an *ABC* company which supports the Mobile-IP protocol for mobility and maintains its own home agent. Every mobile user is required to initiate a secured VPN connection to connect to the company’s email server and file servers. John is provided a laptop computer with network devices: Lucent wireless card for WLAN services and a 3G CDMA wireless card. John also has a subscription for WiFiNZ as anytime-one-dollar-a-day-for-10MB-of-data plan, and a monthly plan of $60 for 250MB of data in 3G network.

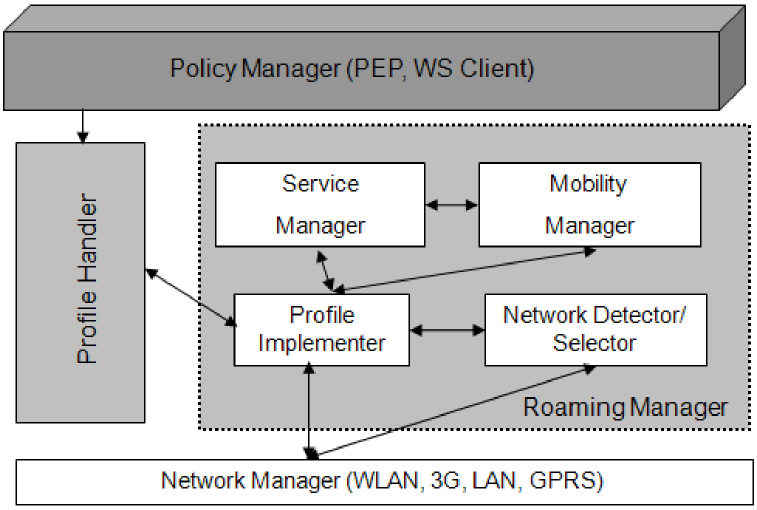

Policy Manager (PEP, WS Client)
Profile Handler
Service Manager
Mobility Manager
Profile Implementer
Network Detector/ Selector
Roaming Manager
Network Manager (WLAN, 3G, LAN, GPRS)

Figure 9-5 Proposed architecture of Policy-managed mobile client

The workings of the proposed architecture will be as follows:

0. The *Policy Manager* allows mobile user (John) to add local preferences and is stored as an internal profile such as the priority of individual access devices, related costs based on purchased subscriptions, support for applications, etc. These profiles are downloaded to the *Profile Handler*, which maintains a local schema of such information.
1. When the mobile node (laptop) starts, the *Roaming Manager* initiates *Profile Implementer* and *Network Detector/Selector* modules. The Profile Implementer then downloads the profiles.
2. The *Network Manager* prepares a list of active interfaces (i.e. WLAN and 3G in this case) and starts scanning for the availability of respective services. This information is forwarded to the Profile Implementer which detects whether the mobile node is at home network or away by the current IP address(es) of its network interfaces.
3. If the mobile node is away from the home network and there are more than two access networks available, the Profile Implementer then forwards the device information together with the profiles to the Network Detector/Selector module.
4. Assuming that current networks available are WLAN and CDMA2000, the Network Detector/Selector selects the best applicable network device (i.e. WLAN) and access network (i.e. WiFiNZ) and informs the Profile Implementer.

5. The Profile Implementer then initiates the *Mobility Manager* for the Mobile-IP protocol and *Service Manager* for the IPSec protocol to create a VPN tunnel from the mobile device to the mobile user's home network.
6. The Network Detector/Selector module continues to monitor the network parameters to identify any need for a handoff. In case of where a handoff is required, it informs the Profile Implementer to initiate a Mobile-IP registration request.

We will now discuss in detail the introduced concepts and components in the proposed architecture.

9.5.1 **Infrastructure Parameter**

We have introduced a new parameter: an Infrastructure (I) parameter to confirm whether the offered infrastructure support from the service provider is acceptable to the mobile node/user. This then assists the mobile node to behave according to the user's current application requirements and network's present conditions. For example, while connecting to a hotspot, a user's internal policy might require either WEP encryption or an IPSec-VPN tunnel support directly to the home network. If the new service provider does not offer or cannot provide any of these features, the user may not accept the connection.

However, while using the Infrastructure parameter, there are no explicit message exchanges between the service provider and the mobile node. Simply stated, it is a Boolean value deciding of whether the available *best* connection should be accepted by the mobile node/user based on the current offerings of the network and the external and internal profiles (i.e. available resources, present network state, supported protocols, user's preferred choices, etc.)

9.5.2 **Proposed Components**

A number of components have been introduced in the architecture to provide a feasible framework to implement the proposed architecture. In summary,

a. The *Policy Manager* provides a policy-management framework for mobile client,

b. The *Roaming Manager* provides seamless mobility and (semi)transparent handoff – based on user's discretionary involvement in handoff decisions.
c. The *Network Manager* provides a platform for communicating with the radio interfaces of the mobile device, and
d. The *Profile Handler* provides a common message bus for transferring policy information between *Policy Manager* and *Policy Implementer*.

The following section discusses each component in detail.

9.5.2.1 **Policy Manager**

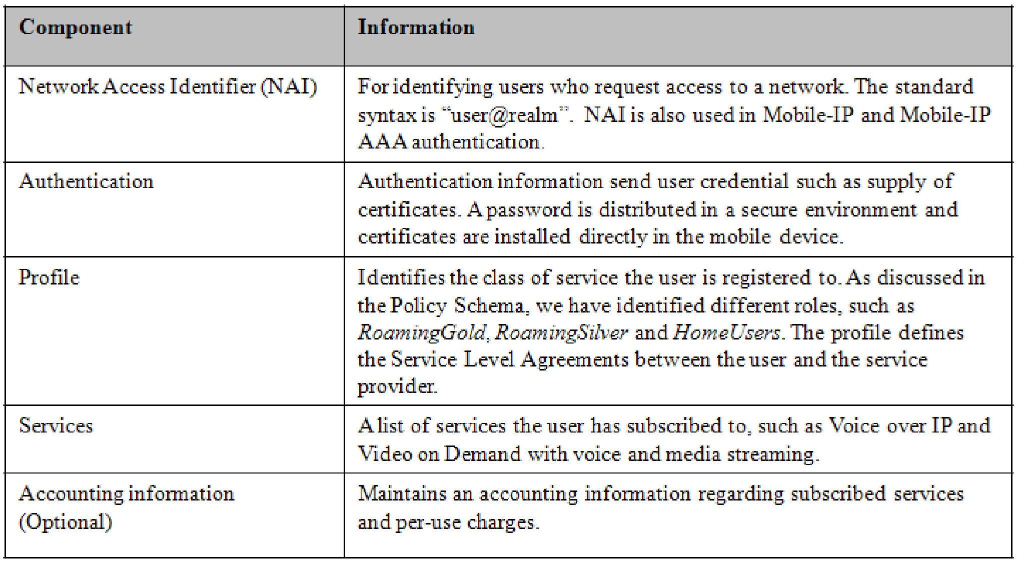
The Policy Manager, which acts as a PEP in the PBNM model, downloads and stores the user profile from its home network. It may communicate with the PBNM defined PDP component using the COPS outsourcing model (discussed in section 6.2.1), where the PEP sends resource requests to the PDP and the decision is carried by the PDP whether to accept or reject the request.

As discussed in the architecture of network management, the *Runtime Policy Manager* in the *Policy Syntax Engine* may also support XML-based languages and Web Services (discussed in section 6.3) and the communication may be based on a request/response model. WSDL based Web Services can be employed for sending a mobile user's runtime requests e.g. request from a *gold* profile user for a *platinum* profile *bandwidth* for the *next hour* for a *video conference*.

The Profile Manager is characterised by:

a. **External Profile**

As discussed in the architecture of a managed network, the network maintains a database of users' *subscriptions.* Hence, each user is registered with a Network Access Identifier (NAI) [204], a profile information (i.e. a class of a service e.g. *gold* profile) and a list of other subscribed services such as VoIP and VoD. The profile may also include accounting records which can be used for billing the user for the subscriptions and services based on the usage time or bandwidth. A typical user profile is shown in Table 9-1



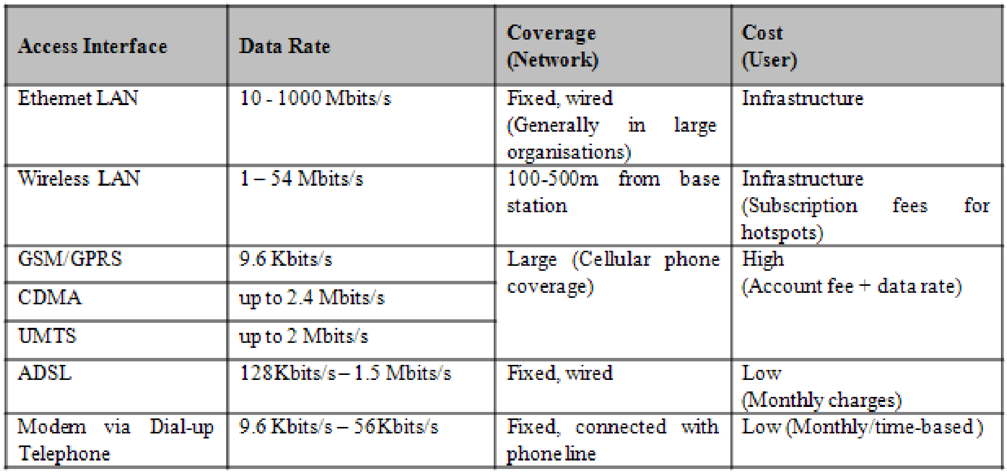
Component	Information
Network Access Identifier (NAI)	For identifying users who request access to a network. The standard syntax is "user@realm". NAI is also used in Mobile-IP and Mobile-IP AAA authentication.
Authentication	Authentication information send user credential such as supply of certificates. A password is distributed in a secure environment and certificates are installed directly in the mobile device.
Profile	Identifies the class of service the user is registered to. As discussed in the Policy Schema, we have identified different roles, such as *RoamingGold*, *RoamingSilver* and *HomeUsers*. The profile defines the Service Level Agreements between the user and the service provider.
Services	A list of services the user has subscribed to, such as Voice over IP and Video on Demand with voice and media streaming.
Accounting information (Optional)	Maintains an accounting information regarding subscribed services and per-use charges.

Table 9-1 User Profile Components

b. Internal Profile

The principle of automation of switching from one network to another is based on the concept of *user involvement with minimal user interaction*. User involvement is required for the policy specification where user's preferences are added for the first time, while minimal user interaction implies automation. It is essential that the policy specification be simple and intuitive to avoid users having to manually configure the network settings.

For example, the Internal Profile specifies a preference list for selection of access networks which may be specified in terms of cost, network condition, power consumption, connection duration time, connection setup time, and others. The *network condition* is collectively defined as a set of dynamic parameters for choosing the best available network, which includes available bandwidth at each reachable network, network latency, and reliability of the reachable networks. The available bandwidth may either be indicated by the network prior to the connection (for example, in a service advertisement) or it may be manually analysed by the mobile device using *round trip time* (for example, a ping request). Table 9-2 presents an overview of the different technologies and their associated attributes which may affect users' personal preferences.



Access Interface	Data Rate	Coverage (Network)	Cost (User)
Ethernet LAN	10 - 1000 Mbits/s	Fixed, wired (Generally in large organisations)	Infrastructure
Wireless LAN	1 – 54 Mbits/s	100-500m from base station	Infrastructure (Subscription fees for hotspots)
GSM/GPRS	9.6 Kbits/s	Large (Cellular phone coverage)	High (Account fee + data rate)
CDMA	up to 2.4 Mbits/s		
UMTS	up to 2 Mbits/s		
ADSL	128Kbits/s – 1.5 Mbits/s	Fixed, wired	Low (Monthly charges)
Modem via Dial-up Telephone	9.6 Kbits/s – 56Kbits/s	Fixed, connected with phone line	Low (Monthly/time-based)

Table 9-2 Parameter values: Network, Technology and Users perspective

9.5.2.2 **Roaming Manager**

The Roaming Manager is a composite component whose primary objective is to provide ABC services i.e. mobility, security and QoS, to the user by selecting and connecting to the best available access interface and the access network. As the handoff decisions and handoff operations are performed at the mobile node, the Roaming Manager periodically collects current network conditions and consults with a policy module to decide the *best* reachable network. If the selected interface is not the one currently in use and if it has been consistently the best available network for a given period of time, then the Roaming Manager performs a handoff to it.

The Roaming Manager consists of Mobility Manager, Service Manager, Profile Implementer and Network Selector/Detector components, and discussed below:

A. **Mobility Manager**

The *Mobility Manager* manages the mobility protocols which involve processes such as routing table manipulation, sending location updates and encapsulating packets. The Mobility Manager also performs the general re-routing of packets during any handoff and periodically sends registration requests to the *home agent* in the case of the Mobile-IP protocol being used for mobility.

It is a general consensus that macro mobility protocols such as Mobile-IP or Hierarchical Mobile IP are more suited for mobility support than micro mobility protocols in heterogeneous network scenarios [217]. We have identified two different scenarios where the mobility requirements of a mobile node changes: (i) when the mobile node is connected in a managed hotspot environment controlled by a *Network Access Server* and (ii) when the mobile node is connected without any supporting infrastructure i.e. *Road Warrior*.

I. Hotspot with Network Access Server

Hotspots are defined as specific geographical locations where an access point provides public wireless broadband network services to mobile visitors through a WLAN connection, and are often located in heavily populated places such as university, organisations, airports, stations, cafés, hotels, and others.

To mange hotspots, a NAS is usually deployed which is characterised by the number of access points (depending on the area covered), foreign agent, home agent (optional), firewall and an AAA (Authentication, Authorisation and Accounting) infrastructure. The NAS supported WLAN infrastructure is common in places where a user's individual subscription can be maintained and a stronger security protocol can be adopted. For example, in universities where a server certificate together with individual staff/student username-password database is maintained. However, NAS are not common in public hotspot areas where the number of users are not monitored and the WLAN security is not an issue.

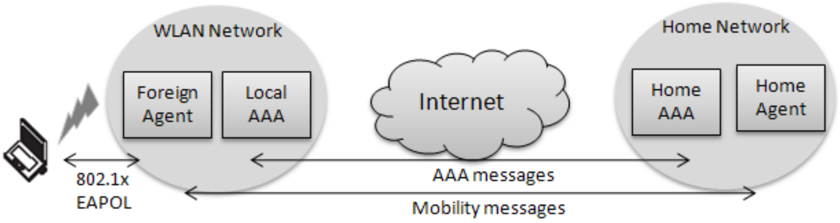


Figure 9-6 Mobility in a Managed Hotspot area

When a user enters a hotspot (Figure 9-6), the NAS blocks all the traffic until the user is authenticated by the NAS AAA server. The Mobility Manger module supports this authentication and offers a NAI retrieved from the *external profile*. After successful authentication, the NAS provides care-of address and some form of security to the mobile device (e.g. 802.1x and WiFi

Protected Access (WPA) or WPA2 access security after the user/device is authenticated by user-password or certificate-based profile). In this case, the NAS acts as a foreign agent in Mobile-IP protocol and decapsulates all the packets destined to the mobile node.

II. **Road Warriors**

The term *Road Warrior* is usually referred to when a mobile user is roaming without any infrastructure support (i.e. without a foreign agent, NAS or AAA infrastructure). In terms of Mobile-IP, in this case, the mobile node needs to operate in a *co-located care-of address* mode, where the end-point of the mobility tunnel is the mobile node itself and it needs to decapsulate the packets received from the home agent. However, this mode of connection exposes the communication between the mobile device and the organisation (i.e. home agent) with which the mobile node is connected to, particularly while using wireless communication. This scenario is common in public hotspot areas where there is no secure tunnel support.

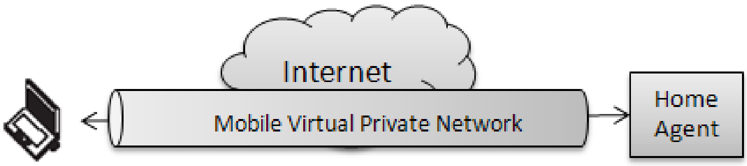
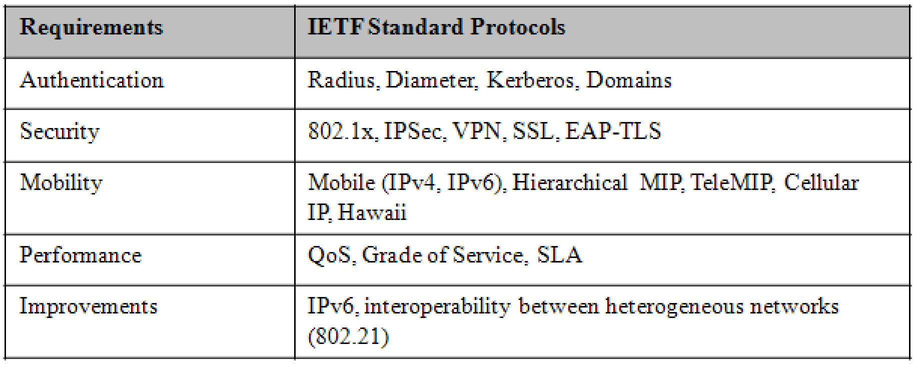


Figure 9-7 Mobility without infrastructure support - Road Warrior

The proposed Mobility Manager supports co-located care-of address mode and initiates a Mobile Virtual Private Network (MVPN) to provide encryption of data in the IP-layer using IPSec in tunnelled mode [227]. This allows an organisation's resources to be accessible to mobile users and provides secure intranet access over an insecure public network. Interestingly, in the 3G networks where home agents are hidden from the public access, a mobile node is *always* connected in a co-located care-of address mode.

B. **Service Manager**

The *Service Manager* provides a framework to support all the protocols of the services been offered to the mobile user. Additionally, it is tightly integrated with the *Mobility Manager* to support protocols for security and authentication integrated together with the mobility protocols. For example, use of IPSec and AAA protocols with the Mobile-IP protocol. Table 9-3 presents a list of standard protocols which need to be supported to offer respective services to the mobile user.



Requirements	IETF Standard Protocols
Authentication	Radius, Diameter, Kerberos, Domains
Security	802.1x, IPSec, VPN, SSL, EAP-TLS
Mobility	Mobile (IPv4, IPv6), Hierarchical MIP, TeleMIP, Cellular IP, Hawaii
Performance	QoS, Grade of Service, SLA
Improvements	IPv6, interoperability between heterogeneous networks (802.21)

Table 9-3 Protocol support by Service Manager

The Service Manager is not a single module but a set of *managers* where each respective *manager* supports a suite of protocols to support the requested services. For instance:

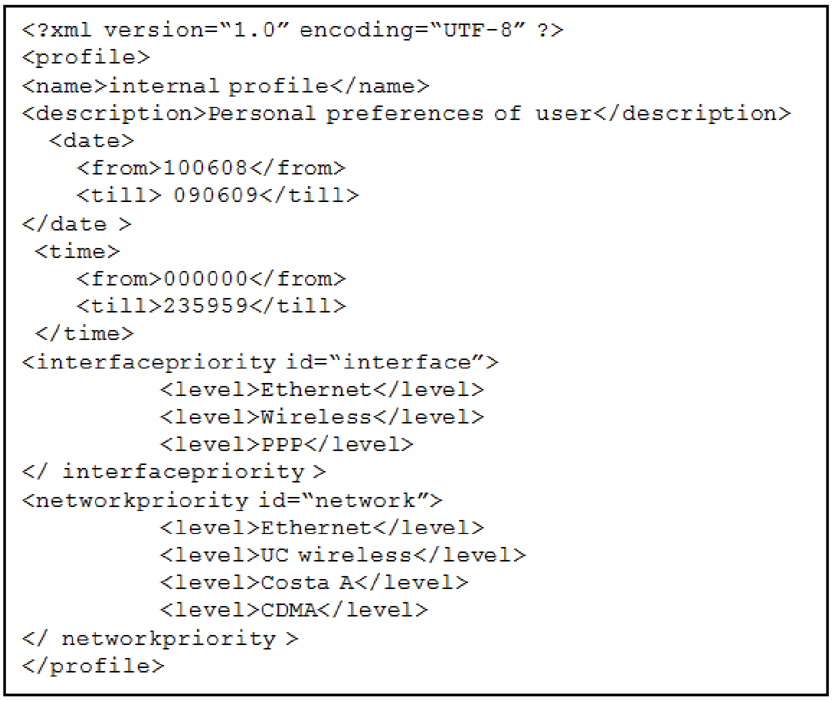
i. An *Authentication Manager* will support a set of protocols to provide authentication to the mobile node. For example, a user connected via the UMTS network requires support for UMTS-AKA as one-pass authentication, while a CDMA network requires the support of the Challenge Handshake Authentication Protocol (CHAP). In the case of a WLAN network, support for WPA and WPA2 protocols is required.

ii. A *Security Manager* will support a set of protocols to provide different types of security to the mobile user/node applicable in different scenarios. For example, a WLAN network may support either no encryption, WEP encryption based on 40, 128 or 256-bit keys, or a security suite of IEEE 802.1x security protocols ranging from username-password to public-private key based certificates. The certificate based Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) is widely supported by WLAN networks for authentication and security [228]. Hence, the implementation of the architecture should provide support for these security protocols.

iii. Any *Additional Services* require a manager to support the protocol and the service. For example, a QoS Manager may need to support SIP for multimedia session control and to offer voice and video streaming services to the mobile user. Although the support of additional services is not required for the Service Manager, it may become an integral part of the service platform for future applications.

C. **Profile Implementer**

The *Profile Implementer* implements the profiles (i.e. internal and external) of the user and translates it to device relevant information. It maintains both the statistical and abstract information of the profile. Statistical information comprises attributes which can measured, such as received signal strength, perceived bandwidth, round trip time, etc., while abstract information are relative based on user's current context. For example, an *internal profile* which states that an interface selection priority list is defined as *Ethernet* > *Wireless* > *CDMA* is only valid if the mobile device is currently supporting Ethernet, Wireless and CDMA network cards. Similarly a network selection priority list is defined as *LAN* > *CostaA(WLAN)* > *CostaB (WLAN)* > *CDMA* provided these network are available at the time a decision is made (Listing 9-1). The profiles are maintained in an XML format.



```
<?xml version=“1.0” encoding=“UTF-8” ?>
<profile>
<name>internal profile</name>
<description>Personal preferences of user</description>
  <date>
    <from>100608</from>
    <till> 090609</till>
</date >
 <time>
    <from>000000</from>
    <till>235959</till>
 </time>
<interfacepriority id=“interface”>
          <level>Ethernet</level>
          <level>Wireless</level>
          <level>PPP</level>
</ interfacepriority >
<networkpriority id=“network”>
          <level>Ethernet</level>
          <level>UC wireless</level>
          <level>Costa A</level>
          <level>CDMA</level>
</ networkpriority >
</profile>
```

Listing 9-1 Internal profile in XML format

D. Network Detector and Selector

The *Network Detector and Selector* module incorporates sophisticated network detection and optimum network selection algorithms. The Network detector part constantly monitors the statistics received from the *Network Manager* and maintains a list of available options for the mobile node, while the *Network Selector* selects the best available network and an optimum time to perform a handoff. Any decision from the *Network Selector* is forwarded to the *Profile Implementer* which then decides on a handoff based on the *internal* and *external* profiles.

We have incorporated several parameters such as weighted priorities, signal threshold, hysteresis margin, dwell timers and link quality to improve the overall handoff timings. The proposed network selection algorithm is discussed in detail in section 9.6

9.5.2.3 **Profile Handler**

The *Profile Handler* interacts with all the components as a common message bus where abstract-level policies from the *Policy Manager* are forwarded to the *Profile Implementer* which are then implemented by the *Network Manager*. These capabilities do not include policy analysis or conflict resolution from the mobile client side.

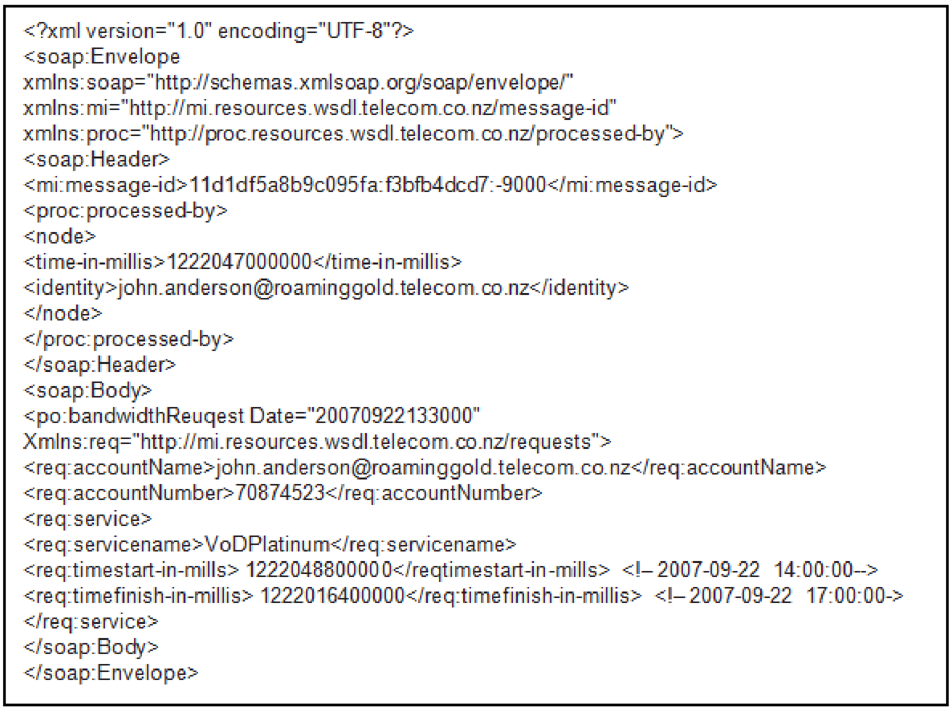
Considering the low-processing power of the mobile nodes and their dependence on battery power, we assume that the policies received by the mobile node, either from the network as an *external profile* or inputted by the user as an *internal profile,* are “conflict-free”. Since a profile is used to identify a priority among the network interfaces and devices, this assumption is relevant in real-world scenarios as well as in semantic analysis of policies, where one of the prominent conflict resolution techniques is through prioritisation [229] (discussed in section 5.7). The Profile Handler also manages the out-of-bound requests of the mobile user.

a. **Out-of-bounds Request**

   In the mobility scenario, it is generally the case where a mobile user wants to use a service which demands network resources beyond a user’s normal profile. For example, user *John* with a *gold* profile wants a higher bandwidth of *platinum profile* for a limited time to perform *Video on Demand* operation.

   As discussed in section 8.2.3, the proposed architecture of network management supports Web Services, where out-of-bound requests are transferred to the *Runtime Policy Manager* which, based on the current network conditions, either accept or reject the request. This concept is similar to the RSVP approach for QoS provisioning [198].

   The message exchanges are performed using WSDL, which is a standardised specification to describe networked XML-based services. It provides a simple way for service providers to describe the basic format of requests to their systems regardless of the underlying protocol (such as SOAP or XML). Our implementation of the architecture uses SOAP for sending the request. For example, a request from user *John* for higher bandwidth to perform Video on Demand can be wrapped in SOAP format as shown in Listing 9-2.



```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mi="http://mi.resources.wsdl.telecom.co.nz/message-id"
xmlns:proc="http://proc.resources.wsdl.telecom.co.nz/processed-by">
<soap:Header>
<mi:message-id>11d1df5a8b9c095fa:f3bfb4dcd7:-9000</mi:message-id>
<proc:processed-by>
<node>
<time-in-millis>1222047000000</time-in-millis>
<identity>john.anderson@roaminggold.telecom.co.nz</identity>
</node>
</proc:processed-by>
</soap:Header>
<soap:Body>
<po:bandwidthReuqest Date="20070922133000"
Xmlns:req="http://mi.resources.wsdl.telecom.co.nz/requests">
<req:accountName>john.anderson@roaminggold.telecom.co.nz</req:accountName>
<req:accountNumber>70874523</req:accountNumber>
<req:service>
<req:servicename>VoDPlatinum</req:servicename>
<req:timestart-in-mills> 1222048800000</reqtimestart-in-mills>  <!– 2007-09-22  14:00:00-->
<req:timefinish-in-millis> 1222016400000</req:timefinish-in-millis>  <!– 2007-09-22  17:00:00->
</req:service>
</soap:Body>
</soap:Envelope>
```

Listing 9-2 SOAP wrapper for out-of-bounds request

9.5.2.4 **Network Manager**

The *Network Manager* component translates and implements the policies into device specific commands. It is an architectural implementation of the *Network layer* of the proposed model.

The Network Manager maintains a direct interface with the network adapters of the mobile device to collect real-time information, such as current signal strength, round trip time, perceived throughput, speed of the mobile device, etc., and stores it in a shared location which is then retrieved by the *Network Detector/Selector* module. The implementation of a Network Manager is specific to the internal hardware and device drivers of the mobile device.

In this section, we have discussed different components introduced in the proposed architecture. We have also described why the decision for a handoff (i.e. selecting an optimum access network and an optimum time to handoff) is extremely important as every change in the point of connection causes a service disruption, and hence we introduced a dedicated Network Detector/Selector component.

The selection of an optimum network may be considered as a straight forward task, since a network priority list is either manually added by the mobile user through an internal profile, or can be extracted by network specific information such as available bandwidth, round trip time, etc. For instance, for a list of available networks, an Ethernet LAN will almost always have higher priority than a WLAN interface which will have higher priority than a CDMA interface. However, selecting an optimum time to handoff is still a challenging task. Based on the criteria of *when-to-handoff*, it may be possible that an active CDMA connection needs to be continued even if a WLAN connection is available, since the mobile user is moving at a speed where handoff to a WLAN network of a small coverage can be detrimental to the overall performance.

9.6 Proposed Network Selection Algorithm

The traditional algorithms for selecting optimum time to handoff usually employ simple intuitive rules to compare the received signal strength from different points of connection and then decide on when to make the handoff. However, the degradation of the signal level is a random process, and simple decision mechanisms based only on current signal strength results in a *ping-pong effect.* A ping-pong effect, similar to router flapping, is described as when a mobile node handoffs to the network whose signal strength is higher in that instantaneous moment and initiates a series of handoffs especially when the mobile node is moving on the perimeter of a WLAN hotspot, hence degrading the overall service level.

9.6.1 Related Work

In order to address the ping-pong problem, one trend focuses on an idea that to make more accurate handoff decisions, location-aided handoff strategies may be an optimum solution [230, 231]. These studies have shown that the user movements can be fairly predicted by using a history of the recorded user movements, current direction and the velocity of the user. Based on this concept other approaches such as hypothesis testing [232], dynamic programming [233] and pattern recognition [234] to predict the behaviour of the mobile user have been proposed. Other techniques based on *learning* and predicting mobile user’s movement have been also proposed, such as neural network [235] which is based on pattern classification and use signal strength measurements for path identification and derive trained samples to define the next move of the user. While other approaches involve a fuzzifier [236], which converts parameters into fuzzy sets that are mapped into a membership value.

However, it has been discussed that mobility prediction algorithms in general are incapable of adapting to new situations and that a small random variation can cause many mobility prediction algorithms to fail [237]. Besides it is unclear if current technologies, for example 802.11b can provide sufficient positioning precision [238] to make handoff decisions fast enough. Furthermore, these state-of-the-art proposals complicate the handoff algorithms, including requiring either a fully trained system before deployment (neural networks) or generating a table showing the outcomes for all possible values of the input criteria and then generating a single number representing the handoff factor (fuzzifier).

Another trend is based on a concept that if the packet loss during a handoff could be avoided completely, it would be possible to perform speculative handoffs and hence without degrading the service quality. To provide lossless handoffs between heterogeneous networks, soft handoff support in layers above the network layer has been proposed. For example, Resilient Mobile Socket scheme [239] provides soft handoff support by allowing simultaneous use of multiple UDP sockets for data communication. However, this makes the handoff process dependable on layers other than the IP layer. Also, in a real-world scenario, it is unlikely that the mobile node will be able to use multiple access connections simultaneously for data communication. For example, if a hotspot area is available, a mobile user may not want to use a 3G network due to cost implications and will only initiate a 3G connection when the current WLAN connection becomes unusable. Hence, hard handoff and vertical handoff usually portrays the real world scenarios more closely.

The related works which have explored vertical handoff decision mainly focus on the traditional issues, such as received signal strength (RSS) and data rate. Some interesting work in this area includes: a fast-Fourier-transform-based signal decay detection scheme [240] which was used to reduce the ping-pong effect, and an adaptive threshold configuration approach was proposed to prolong the time a user stays in WLAN. In [241, 242], a vertical handoff algorithm was proposed which considered signal strength, data rate, and packet loss due to handoff delay for a single service per user. Another handoff system based on computed background noise and RSS was proposed in [242]. Similarly, in [243], an active application-oriented handoff decision algorithm was proposed for multi-interface mobile terminals to reduce the power consumption caused by unnecessary handoffs and other unnecessary interface activation, and in [244], a policy-enabled handoff decision algorithm was proposed along with a cost function that considers several handoff metrics.

9.6.2 **Proposed Approach**

It is clear that each of the research approaches which focused on improving the handoff performance of vertical handoffs, hard handoffs and consider the processing limitations of mobile devices, have primarily considered signal strength, in combination with other parameters such as hysteresis, background noise, dwell timer and others with great success [245-247].

We propose a network selection algorithm to eliminate the ping-pong effect and hence reduced the number of unnecessary handoffs. The proposed network selection focuses on how to trigger handoffs rather than describing how to implement mobility, QoS, security or user aware support, which are managed by specific protocols such as Mobile-IP, 802.1x, internal-external profile, etc. Our assumption considers that mobile devices are of limited processing powers and limited battery capacity. The parameters of the handoff decision algorithm should be selected so that they minimises the use of system resources. The proposed selection algorithm is mainly concerned with the handoff initiation and network selection, and does not address the availability and reservation of resources in the destination network, which is independently managed by the Infrastructure parameter (discussed in section 9.5.1) and the Policy Manager (discussed in section 9.5.2.1).

The first goal of any seamless handoff is low handoff latency, power saving and low bandwidth overhead. The criteria for selecting the initial mode in the mobile device are the signal strength quality, data rate, service type, and the capacity of the respective networks. For seamless vertical handoff, we have included several parameters and collectively referred to them as a *priority* of an interface in an *internal profile*, which includes:

- Received Signal Strength (RSS)
- Bit Error Rates
- Perceived QoS and the QoS requirements for the current application
- Network Coverage
- Cost
- Battery powered requirements to implement the handoff algorithm and its execution, and
- User Preference – user wants to be connected to the cheapest network available regardless of QoS or coverage offered.

An interface selection algorithm processes this information and decides the most optimal network from those available and the optimal time to initiate a handoff.

Unlike any intra-network/microcellular handoff scenarios, where the best possible time to handoff is when the mobile device is at the midway point between the two identical cellular base stations in a 3G network or two access points in a WLAN network, for any heterogeneous handoff, an efficient algorithm will try to use the services of the higher bandwidth (e.g. WLAN access point) as long as possible and perform any handoff to a lower bandwidth network (e.g. a cellular base station) as the last alternative. The WLAN access point has a much higher priority than the cellular base station as the cellular service provides data rate of order of magnitude much smaller than the WLAN service (e.g. transmitting at 11 Mbits/s for 1 second is preferable to transmitting at 144 Kbits/s for 76 seconds). Hence, it is important to closely monitor the signal strength of the higher priority interface and to remain connected with it as long as is feasibly possible and to delay the handoff.

9.6.2.1 **Proposed Algorithm**

The handoff performances achieved by Pollini [244] and Tripathi [248] have shown improvements especially as the algorithmic calculations involved in such algorithms are significantly less. These algorithms employed thresholds to compare the values of metrics, primarily received signal strength (RSS), from different points of attachments. In order to avoid the ping-pong effect, additional parameters such as hysteresis margins have also been employed [249]. A hysteresis margin is added to the currently selected network to delay the handoff decision to another network.

We have proposed and analysed (in section 9.7) an automated algorithm by introducing link quality, hysteresis margin, priority of individual interfaces and dwell timers into the interface selection algorithm to optimise the handoff initiation time. Even though the proposed algorithm employs similar parameters as proposed by other approaches (i.e. signal strength, hysteresis effect, dwell timer, etc.), their usage in the mathematical calculations is significantly different. For instance, we have introduced normalised threshold of signal strength of available networks together with standard deviation of the number of samples collected to reduce the overall random variance which is common in wireless technologies. The following section discusses the proposed algorithm and the proposed formulae.

a. Handoff Initiation

Handoff initiation is the process of monitoring the current network connection, recognising the need for handoff and subsequently initiating it. At any given time, the mobile node selects one of its physical interfaces as its *current* interface and registers with the mobility agent on that interface. To avoid any data loss, it maintains association with the *current* interface while probing for an alternate *better* interface (Figure 9-8).

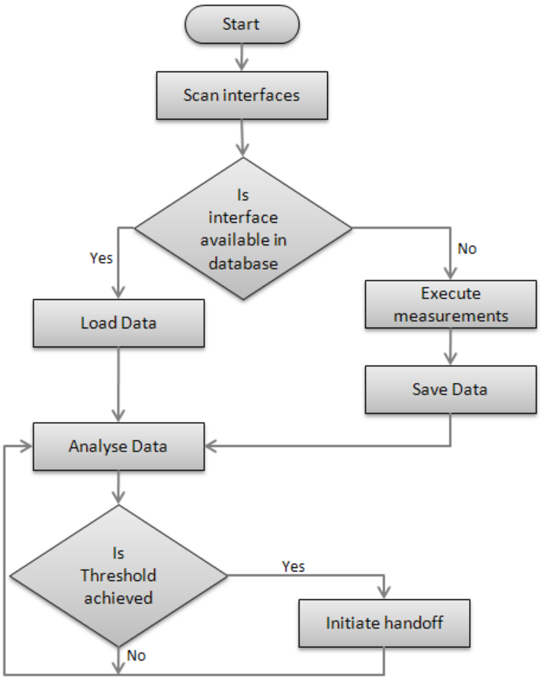


Figure 9-8 Handoff Initiation Flowchart

b. **Network Selection**

The network selection stage is used to select a network connection that can satisfy the requirements of the network provider and the user, such as low cost, signal strength, optimal bandwidth, low network latency and high reliability. For example, considering the local profile of the user for interface priority is defined as Ethernet > WLAN > CDMA.

We have employed a network selection algorithm that uses the current signal strength, threshold levels and weighted priority of respective interfaces to select an active interface over a time period, where the samples are analysed based on their standard deviation.

A standard deviation is a measure of the variability of a set of values. A low standard deviation indicates that the sample data points tend to be very close to the same value (i.e. the average of all the values), whereas high standard deviation indicates that the values of the data set are spread out over a large range of values. Hence, the use of standard deviation eventually solves the basic problem with ping-pong effect, where the received signal strength of a WLAN network constantly fluctuates hence resulting in the mobile node connecting and disconnecting rapidly from the WLAN network.

The number of samples to be collected is determined by the dwell timer. The Received Signal Strength (RSS) is a measurement of the signal/noise ratio present in a received signal.

- RSS: The connection point (CP) whose signal is being received with the highest strength is selected ($choose\ CP_{new}\ if\ RSS_{new}\ >\ RSS_{old}$).

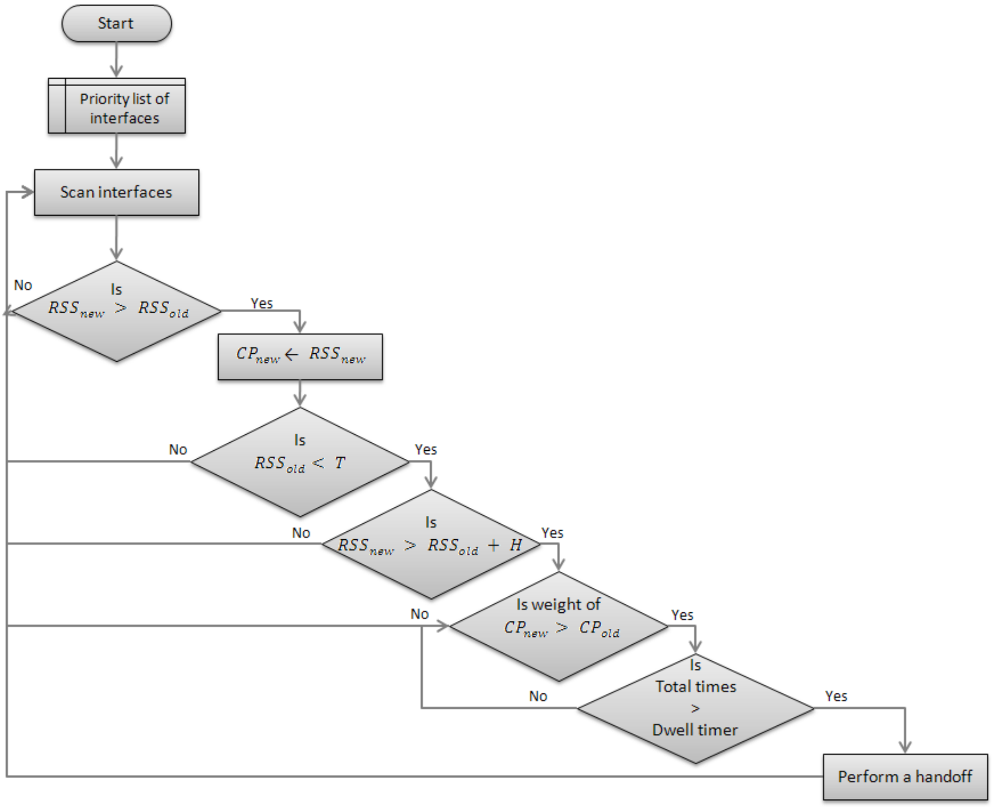
Handoff is made if, $CP_{new}\ >\ CP_{old}$

- RSS plus $Threshold$: the RSS of a new CP exceeds that of the current one and the signal strength of the current CP is below a threshold T (i.e. $if\ RSS_{new}\ >\ RSS_{old}\ and\ RSS_{old}\ <\ T$).
- RSS plus $Hysteresis$: RSS of a new CP is greater than that of the old CP by a hysteresis margin H (i.e. $if\ RSS_{new}\ >\ RSS_{old}\ +\ H$).
- $RSS, Hysteresis, and\ Threshold$: RSS of a new CP exceeds that of the current CP by a $hysteresis\ margin\ H$ and the signal strength of the

current CP is below a threshold T (i.e. $RSS_{new} > RSS_{old} + H \; and \; RSS_{old} < T$).

- $Priority$: the priority P of a new CP is higher than that of an old CP by a factor margin. The priority of any interface is a numerical representation of factors such as network bandwidth, cost, overall throughput, network latency and reliability. A $weight \; \omega$ of individual interfaces is calculated until the number of samples is greater than as defined by the dwell timer.
- $Dwell \, Timer$: a timer value η is defined to collect the number of samples. A standard deviation is calculated on each weight w_i over a time period of η. If the weight of the interface i is still higher, then a handoff is performed.

The flowchart of the proposed algorithm is shown in Figure 9-9.



Start
Priority list of interfaces
Scan interfaces
No
Is $RSS_{new} > RSS_{old}$
Yes
$CP_{new} \leftarrow RSS_{new}$
No
Is $RSS_{old} < T$
Yes
No
Is $RSS_{new} > RSS_{old} + H$
Yes
No
Is weight of $CP_{new} > CP_{old}$
Yes
No
Is Total times > Dwell timer
Yes
Perform a handoff

Figure 9-9 Proposed Network Selection Algorithm Flowchart

9.6.2.2 **Interface Weight**

As depicted in Figure 9-9, the weight of individual interfaces is calculated once the signal strength s_i of a new interface is greater than the signal strength of an old interface. In this case, if the signal strength of the old interface is also lower than the threshold value of individual interface L_i + Hysteresis margin for a specified period (where the number of samples is more than the dwell timer η), then a handoff needs to be performed.

The signal strength of individual interfaces need to be normalised for comparing with other interfaces. The signal strength range (i.e. the lower threshold L_i and the higher threshold H_i) varies based on the type of the interface. For example, WLAN networks are measured in –dB (negative decible), the lower the value, the better connection (-96dB is better signal reception than -40dB) while CDMA signal strength is measured in +dB (positive decibels) where a higher number implies better connection. However, the threshold values of WLAN networks (-40dB to -120dB) and CDMA networks (15dB to 25 dB) varies.

The priority p_i is defined as a single numerical value of the set $\{1, 2, 3\}$. Mathematically,

If parameters are defined as:

$$Normalized\ (s_i)\ =\ (s_i - L_i)\ /\ (H_i - L_i)\ *\ 100,\ for\ range\ [0, 100]$$

$$RSS{::}\ s_i\ \in\ [0{,}100],$$

$$Priority{::}\ p_i\ \in\ \{1, 2, 3\}$$

$$Lower\ Threshold{::}\ L_i\ \in\ [0, 100],$$

$$Higher\ Threshold{::}\ H_i\ \in [0, 100],$$

$$Dwell\ Timer{::}\ \eta$$

Then the weight formula is defined as:

$$if\ i\ is\ the\ current\ interface, then$$

$$\begin{aligned} w_i &= 1000 * p_i +\ 2s_i +\ H \quad && if\ s\ \geq L_i \\ &= 2s_i +\ H && if\ s_i\ < L_i \end{aligned}$$

$$if\ i\ is\ not\ the\ current\ interface, then$$

$$\begin{aligned} w_i &= 1000 * p_i +\ s_i - H \quad && if\ s\ \geq H_i \\ &= s_i -\ H && if\ s_i\ < H_i \end{aligned}$$

$$\omega = \frac{1}{\eta}\sqrt{\left\{\eta\left(\sum_{i=1}^{\eta}({w_i}^2)\right)-\left(\sum_{i=1}^{\eta}({w_i}^2)\right)\right\}}$$

Perform a handoff when $\omega_{new} > \omega_{old}$

9.7 Proof-of-concept Implementation

In order to test the feasibility of our proposed architecture solution, we have designed and implemented a testbed for evaluating various solutions under different handoff scenarios. The presented testbed analyses vertical and hard handoff i.e. a mobile node cannot receive IP packets simultaneously on two or more interfaces from multiple heterogeneous networks (discussed in section 9.2.1) using a common network layer. We have integrated mobility, security and authentication by layering Mobile-IP, IPSec and AAA protocols to provide seamless handoff to mobile users in different scenarios for hotspot users and road warriors. The implementation is both Linux and Windows based. The profile management is performed using WSDL and SOAP based web services by the *RunTime Manager* and is implemented only in the Windows environment.

For implementation of the proposed network selection algorithm, a slightly faster way for computing the standard deviation is being employed noting that samples are taken once per 100msec. We have ignored the round-off error, arithmetic overflow and arithmetic underflow tradeoff for the speed for performing calculations.

To evaluate the proposed architecture in a real world scenario, all the experiments were performed in a real network. We have implemented the testbed to investigate the performance of handoff between 802.11 WLAN (Home and Foreign Network) and CDMA2000 interfaces, paying particular attention to factors such as handoff delay and overall throughput, which influence both the end user and the service providers. However, the proposed model and architecture is independent of the underlying networking technologies. Initially, we have implemented the Mobile-IP protocol in Linux environment in Java programming language based on the Dynamics HUT

Mobile-IP implementation [70] which was later ported to the Windows environment where the mobile client was rewritten to realise the proposed architecture.

9.7.1 Testbed

The testbed (Figure 9-10) consists of three types of access interfaces: a Lucent WaveLan 802.11b wireless card for WLAN access which was later upgraded to Intel WLAN card, a CDMA2000 RTT1 network card for 3G access which was later upgraded to a T3G CDMA EVDO card, and a Broadcom Gigabit Ethernet card for LAN access. The Ethernet network operates at 100Mbits/sec, while the WLAN network operates at 11Mbits/sec. The CDMA2000 RTT/EVDO network operates at 144Kbits/sec-1.4Mbits/sec. These network cards are used to connect to four types of networks:

i. A Home WLAN network consists of an access point, a home agent and an AAA server. One desktop computer was used to run both the home agent and the AAA server. The home agent runs our modified Dynamics HUT Mobile IP [70] and the Racoon Internet Key Exchange (IKE) [250] implementation to support IPSec tunnels. Racoon is an IKE key management daemon used to provide automated IPSec key negotiation functionality and is distributed as part of the IPSec-tools package [251].

ii. A Foreign WLAN network similar to the home agent consists of an access point, a foreign agent and AAA server. One desktop computer was used to run both the foreign agent and AAA server. The foreign agent runs Dynamics HUT Mobile IP implementation.

iii. A CDMA2000 network. The T3G card was provided by Telecom NZ which connects to a real-world CDMA2000 network, and

iv. LAN network provided by the University of Canterbury.

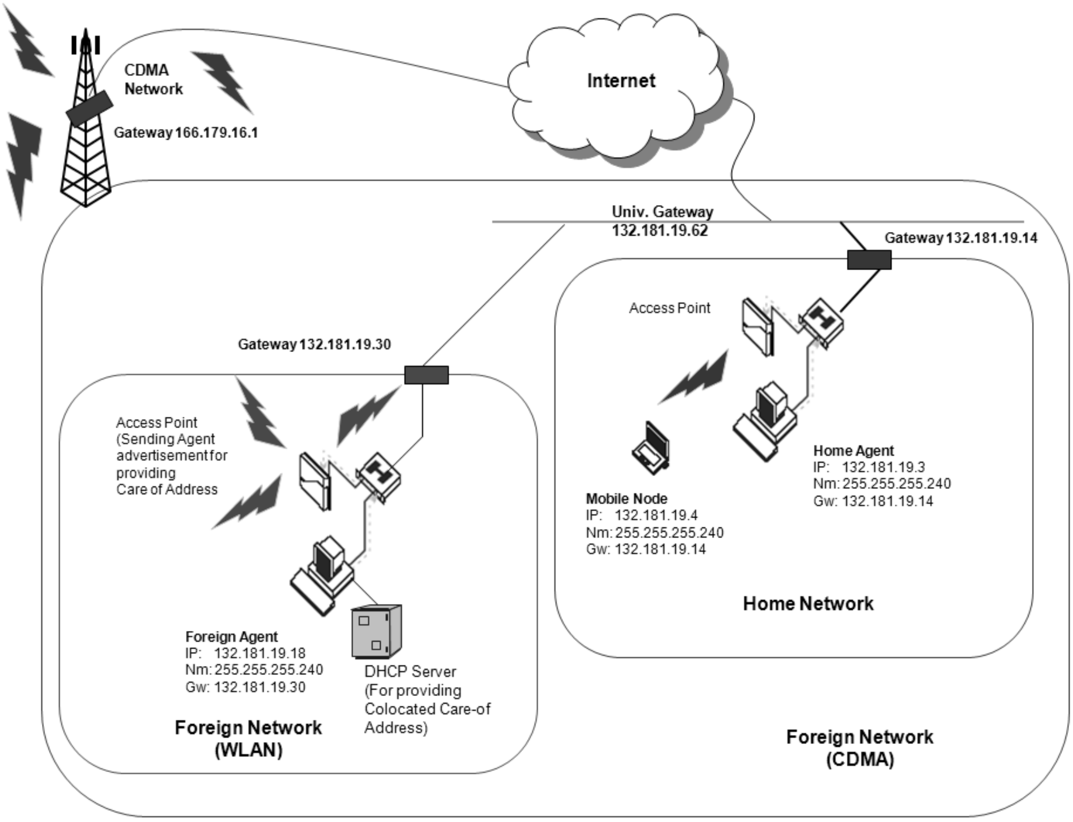
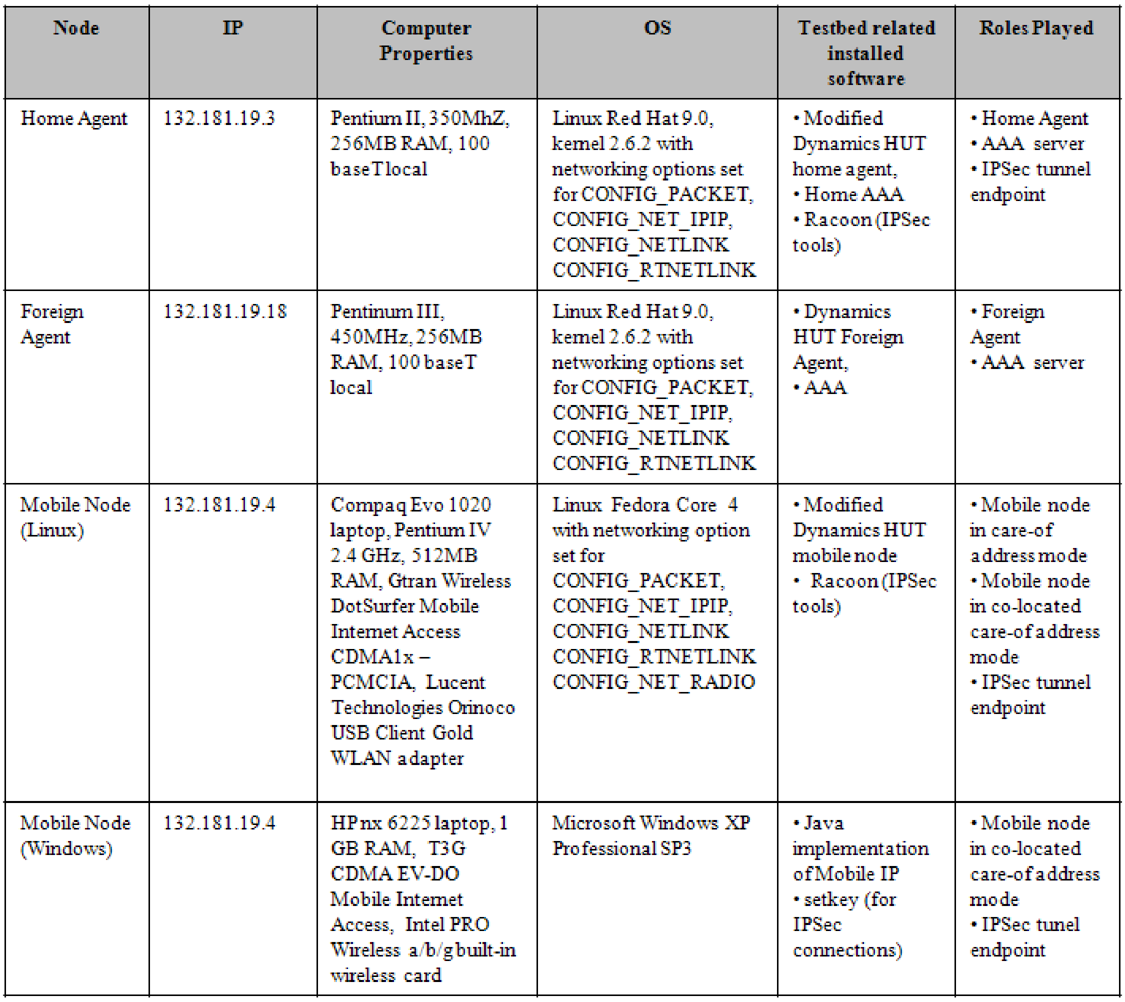


Figure 9-10 Testbed network configuration

The goal of the experiments is to evaluate the impact of mobility on the available throughput. We consider the scenario that:

- Local WLAN Intranet access is commonly preferred in a corporate environment,
- Foreign WLAN access is preferred in hotspot areas,
- Hotspot areas may provide an infrastructure support such as a foreign agent or may not have any infrastructure support, in which case, the mobile node should be able to maintain a Mobile-IP tunnel,
- Large area coverage access through CDMA network, and
- End-user terminals are mobile devices with limited processing power and limited battery lifetime.

Table 9-4 describes the hardware, operating systems and related software used to implement different nodes (or agents) in our testbed.

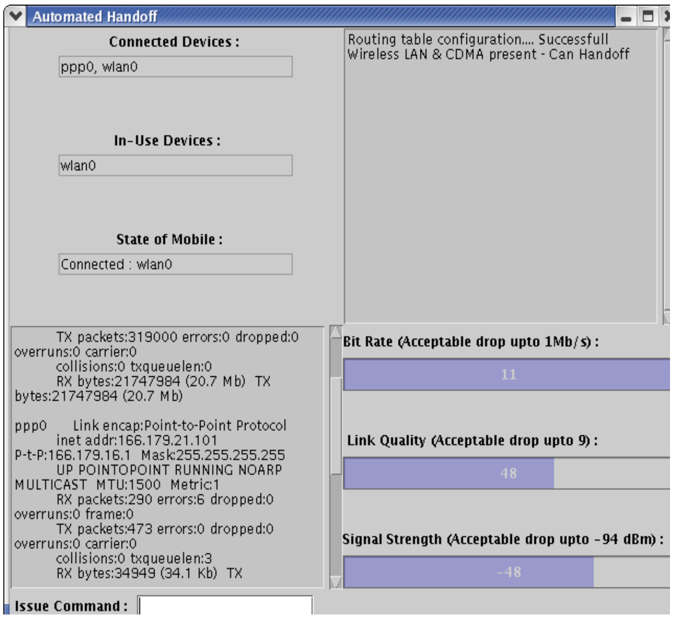


Node	IP	Computer Properties	OS	Testbed related installed software	Roles Played
Home Agent	132.181.19.3	Pentium II, 350MhZ, 256MB RAM, 100 baseTlocal	Linux Red Hat 9.0, kernel 2.6.2 with networking options set for CONFIG_PACKET, CONFIG_NET_IPIP, CONFIG_NETLINK CONFIG_RTNETLINK	• Modified Dynamics HUT home agent, • Home AAA • Racoon (IPSec tools)	• Home Agent • AAA server • IPSec tunnel endpoint
Foreign Agent	132.181.19.18	Pentinum III, 450MHz, 256MB RAM, 100 baseT local	Linux Red Hat 9.0, kernel 2.6.2 with networking options set for CONFIG_PACKET, CONFIG_NET_IPIP, CONFIG_NETLINK CONFIG_RTNETLINK	• Dynamics HUT Foreign Agent, • AAA	• Foreign Agent • AAA server
Mobile Node (Linux)	132.181.19.4	Compaq Evo 1020 laptop, Pentium IV 2.4 GHz, 512MB RAM, Gtran Wireless DotSurfer Mobile Internet Access CDMA1x – PCMCIA, Lucent Technologies Orinoco USB Client Gold WLAN adapter	Linux Fedora Core 4 with networking option set for CONFIG_PACKET, CONFIG_NET_IPIP, CONFIG_NETLINK CONFIG_RTNETLINK CONFIG_NET_RADIO	• Modified Dynamics HUT mobile node • Racoon (IPSec tools)	• Mobile node in care-of address mode • Mobile node in co-located care-of address mode • IPSec tunnel endpoint
Mobile Node (Windows)	132.181.19.4	HP nx 6225 laptop, 1 GB RAM, T3G CDMA EV-DO Mobile Internet Access, Intel PRO Wireless a/b/g built-in wireless card	Microsoft Windows XP Professional SP3	• Java implementation of Mobile IP • setkey (for IPSec connections)	• Mobile node in co-located care-of address mode • IPSec tunel endpoint

Table 9-4 Testbed nodes properties

9.7.2 Linux Implementation

Figure 9-11 presents the implementation of the proposed architecture in Java programming language in Linux environment. The development of the mobile client software was mainly focussed on implementing the proposed interface selection algorithm together with improving the factors of how-to-handoff and when-to-handoff. The mobile client continually observes all the network parameters and performs a handoff whenever it is required to do so. The client implementation also provided user's a facility to execute commands manually to override the handoffs.



Automated Handoff

Connected Devices : ppp0, wlan0

In-Use Devices : wlan0

State of Mobile : Connected : wlan0

```
Routing table configuration.... Successfull
Wireless LAN & CDMA present - Can Handoff
```

```
        TX packets:319000 errors:0 dropped:0
overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:21747984 (20.7 Mb)  TX
bytes:21747984 (20.7 Mb)

ppp0      Link encap:Point-to-Point Protocol
        inet addr:166.179.21.101
P-t-P:166.179.16.1  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP
MULTICAST  MTU:1500  Metric:1
        RX packets:290 errors:6 dropped:0
overruns:0 frame:0
        TX packets:473 errors:0 dropped:0
overruns:0 carrier:0
        collisions:0 txqueuelen:3
        RX bytes:34949 (34.1 Kb)  TX
```

Bit Rate (Acceptable drop upto 1Mb/s) : 11

Link Quality (Acceptable drop upto 9) : 48

Signal Strength (Acceptable drop upto -94 dBm) : -48

Issue Command :

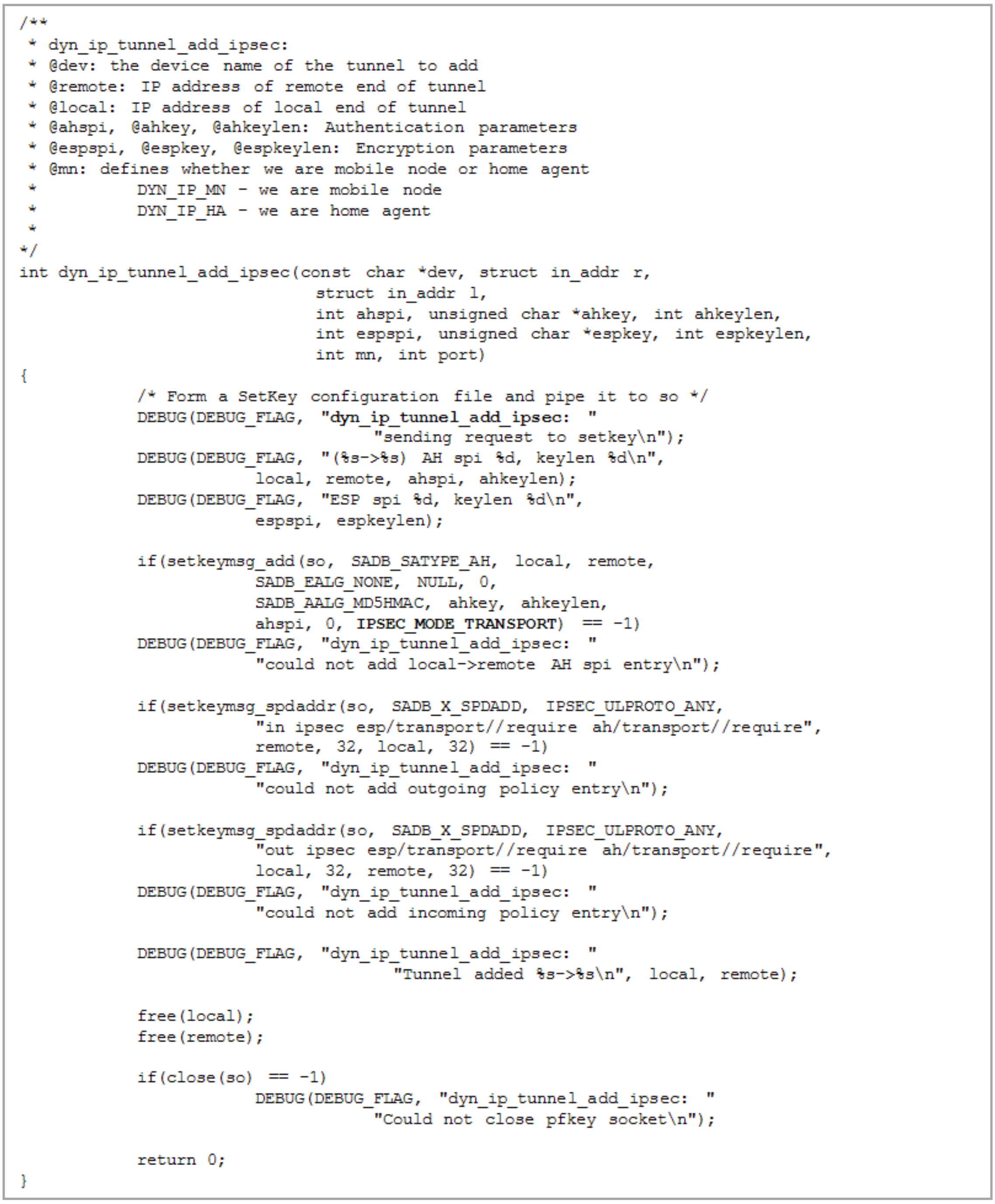
Figure 9-11 Linux implementation

We have considered three interfaces to provide Internet access to the mobile node: WLAN (wlan0), foreign WLAN for hotspots (TULMNA) and CDMA2000 (ppp0), arranged in priority with wlan0 having the highest. This arrangement derives four cases:

1. If all the three interfaces are available and the parameters of wlan0 are within the acceptable range i.e. mobile node is within the home network coverage area then, the connection is made via wlan0 interface. A Boolean variable *can_handoff* is set to true, so that handoff can be performed, for if at any time Boolean variable *should_handoff* becomes true.
2. If only WLAN and CDMA connections are available, i.e. MN is roaming in a hotspot, then priority is given to TUNLMNA
3. If only the CDMA network is available, i.e. either no hotspot is present or MIP software module failed to initialise due to home agent failure or any error in initialisation) then the algorithm switches to the ppp0 interface.
4. If none of the valid interfaces is available, Internet access cannot be provided to the mobile user by making *can_handoff* = false, which is also the case in a simple IP environment.

The Dynamics HUT Mobile-IP implementation provided basic networking functionalities supporting home agent, foreign agent and mobile node. However, there was no support for IPSec protocol to provide security and authentication. We have modified the Dynamics implementation to add IPSec support for both transport mode and for tunnel mode.

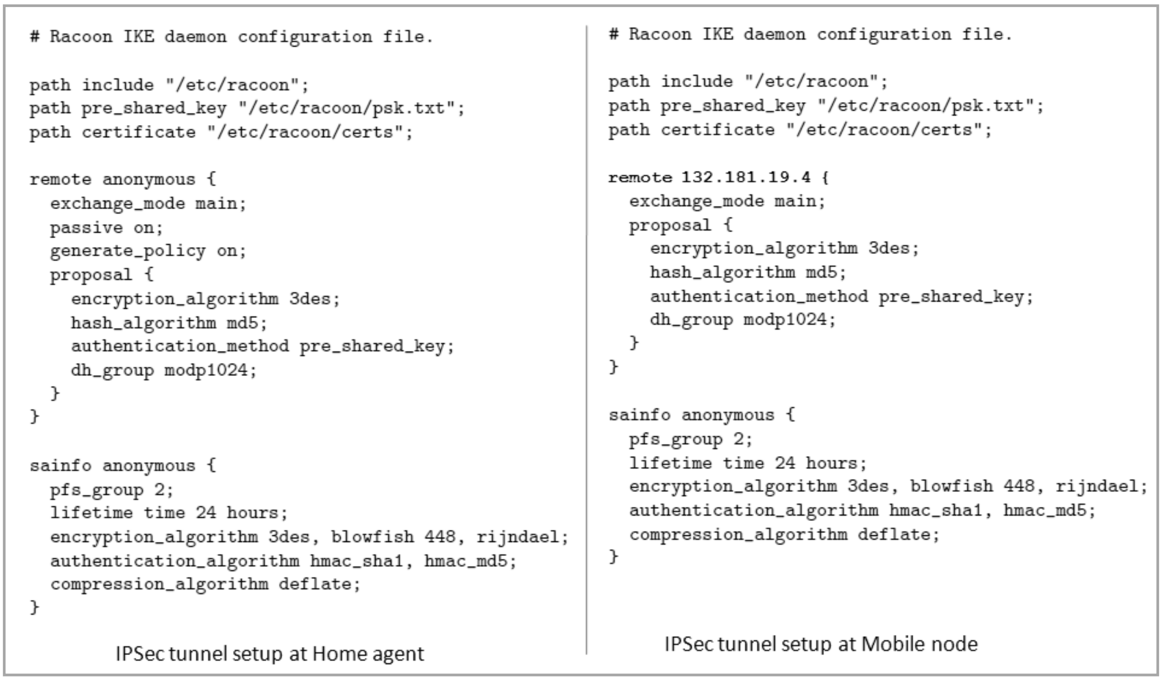
This implementation required additional `libipsec` and `include-glibc` packages to provide IPSec functionality, and needed additional changes to integrate the package into the Dynamics implementation. We have added `dyn_ipsec` and update `dyn_ip` files to achieve the desired tunnel (Listing 9-3).



```
/**
 * dyn_ip_tunnel_add_ipsec:
 * @dev: the device name of the tunnel to add
 * @remote: IP address of remote end of tunnel
 * @local: IP address of local end of tunnel
 * @ahspi, @ahkey, @ahkeylen: Authentication parameters
 * @espspi, @espkey, @espkeylen: Encryption parameters
 * @mn: defines whether we are mobile node or home agent
 *          DYN_IP_MN - we are mobile node
 *          DYN_IP_HA - we are home agent
 *
 */
int dyn_ip_tunnel_add_ipsec(const char *dev, struct in_addr r,
                              struct in_addr l,
                              int ahspi, unsigned char *ahkey, int ahkeylen,
                              int espspi, unsigned char *espkey, int espkeylen,
                              int mn, int port)
{
            /* Form a SetKey configuration file and pipe it to so */
            DEBUG(DEBUG_FLAG, "dyn_ip_tunnel_add_ipsec: "
                                    "sending request to setkey\n");
            DEBUG(DEBUG_FLAG, "(%s->%s) AH spi %d, keylen %d\n",
                        local, remote, ahspi, ahkeylen);
            DEBUG(DEBUG_FLAG, "ESP spi %d, keylen %d\n",
                        espspi, espkeylen);

            if(setkeymsg_add(so, SADB_SATYPE_AH, local, remote,
                        SADB_EALG_NONE, NULL, 0,
                        SADB_AALG_MD5HMAC, ahkey, ahkeylen,
                        ahspi, 0, IPSEC_MODE_TRANSPORT) == -1)
            DEBUG(DEBUG_FLAG, "dyn_ip_tunnel_add_ipsec: "
                        "could not add local->remote AH spi entry\n");

            if(setkeymsg_spdaddr(so, SADB_X_SPDADD, IPSEC_ULPROTO_ANY,
                        "in ipsec esp/transport//require ah/transport//require",
                        remote, 32, local, 32) == -1)
            DEBUG(DEBUG_FLAG, "dyn_ip_tunnel_add_ipsec: "
                        "could not add outgoing policy entry\n");

            if(setkeymsg_spdaddr(so, SADB_X_SPDADD, IPSEC_ULPROTO_ANY,
                        "out ipsec esp/transport//require ah/transport//require",
                        local, 32, remote, 32) == -1)
            DEBUG(DEBUG_FLAG, "dyn_ip_tunnel_add_ipsec: "
                        "could not add incoming policy entry\n");

            DEBUG(DEBUG_FLAG, "dyn_ip_tunnel_add_ipsec: "
                                      "Tunnel added %s->%s\n", local, remote);

            free(local);
            free(remote);

            if(close(so) == -1)
                        DEBUG(DEBUG_FLAG, "dyn_ip_tunnel_add_ipsec: "
                                    "Could not close pfkey socket\n");

            return 0;
}
```

Listing 9-3 Adding IPSec support to Dynamics Mobile-IP

Various other setup scripts (Listing 9-4) were written to configure the basic networking functionalities on the home agent, the foreign agent and the mobile node. On the home agent, an IPSec tunnel script initiates IPSec daemons to accept and setup any IPSec connection.



```
# Racoon IKE daemon configuration file.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

remote anonymous {
  exchange_mode main;
  passive on;
  generate_policy on;
  proposal {
    encryption_algorithm 3des;
    hash_algorithm md5;
    authentication_method pre_shared_key;
    dh_group modp1024;
  }
}

sainfo anonymous {
  pfs_group 2;
  lifetime time 24 hours;
  encryption_algorithm 3des, blowfish 448, rijndael;
  authentication_algorithm hmac_sha1, hmac_md5;
  compression_algorithm deflate;
}
```

IPSec tunnel setup at Home agent

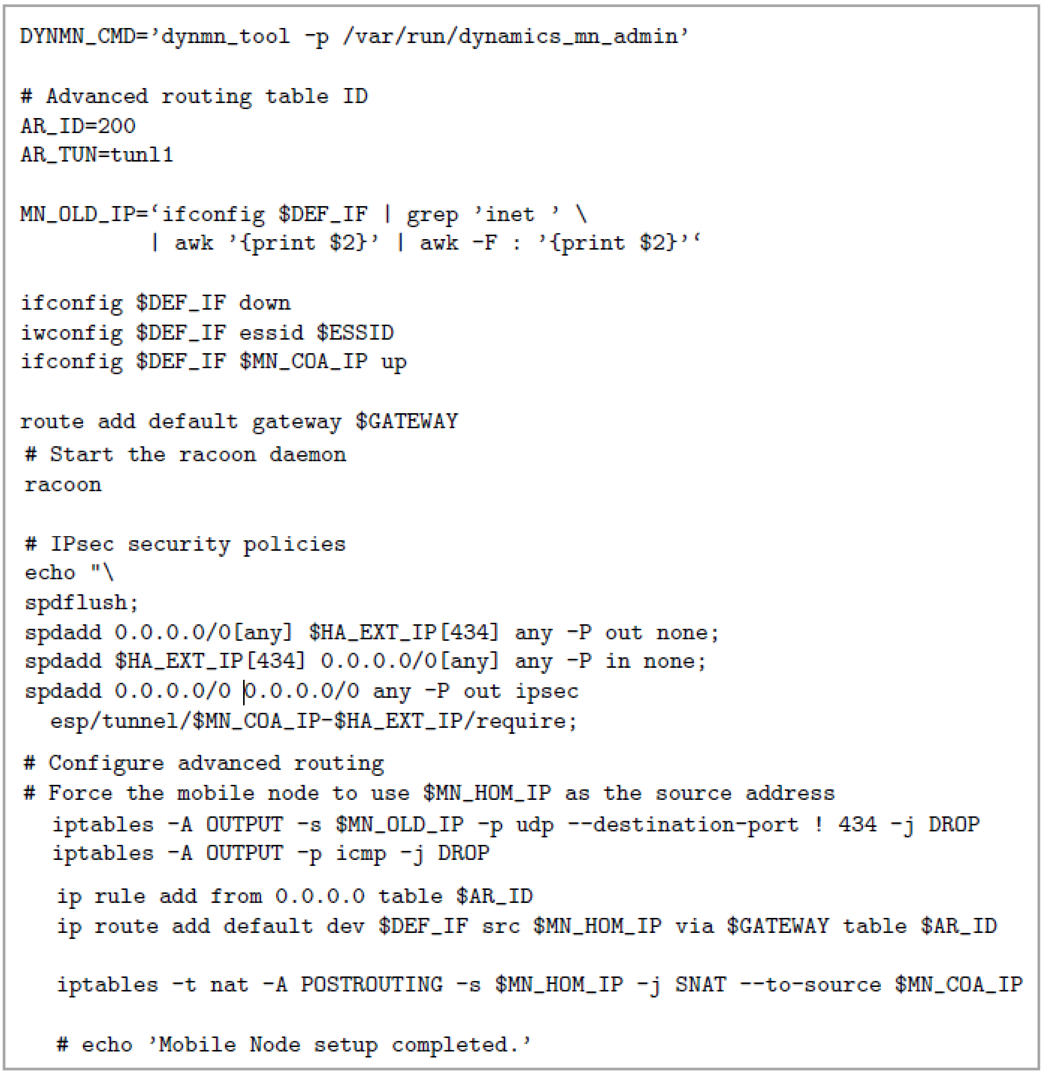
```
# Racoon IKE daemon configuration file.

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";

remote 132.181.19.4 {
  exchange_mode main;
  proposal {
    encryption_algorithm 3des;
    hash_algorithm md5;
    authentication_method pre_shared_key;
    dh_group modp1024;
  }
}

sainfo anonymous {
  pfs_group 2;
  lifetime time 24 hours;
  encryption_algorithm 3des, blowfish 448, rijndael;
  authentication_algorithm hmac_sha1, hmac_md5;
  compression_algorithm deflate;
}
```

IPSec tunnel setup at Mobile node

Listing 9-4 Scripts for IPSec tunnel setup at home agent and mobile node

On the mobile node side, separate scripts were required according to the current handoff condition. The three different scenarios were identified as:

- Setup – when the mobile node moves from the home network into a foreign network, or when the mobile node first requires IPSec services in a foreign network. Since the IPSec tunnels can be setup either in transport mode or in tunnel mode, the foreign agent does not need to support IPSec services. A script used to setup an IPSec tunnel at the mobile node is presented in Listing 9-5.
- Handoff – when the mobile node moves between different networks; and
- Close – When the mobile node moves from a foreign network back to the home network, or when IPSec services are no longer required.



```
DYNMN_CMD='dynmn_tool -p /var/run/dynamics_mn_admin'

# Advanced routing table ID
AR_ID=200
AR_TUN=tunl1

MN_OLD_IP=`ifconfig $DEF_IF | grep 'inet ' \
          | awk '{print $2}' | awk -F : '{print $2}'`

ifconfig $DEF_IF down
iwconfig $DEF_IF essid $ESSID
ifconfig $DEF_IF $MN_COA_IP up

route add default gateway $GATEWAY
# Start the racoon daemon
racoon

# IPsec security policies
echo "\
spdflush;
spdadd 0.0.0.0/0[any] $HA_EXT_IP[434] any -P out none;
spdadd $HA_EXT_IP[434] 0.0.0.0/0[any] any -P in none;
spdadd 0.0.0.0/0 0.0.0.0/0 any -P out ipsec
  esp/tunnel/$MN_COA_IP-$HA_EXT_IP/require;

# Configure advanced routing
# Force the mobile node to use $MN_HOM_IP as the source address
  iptables -A OUTPUT -s $MN_OLD_IP -p udp --destination-port ! 434 -j DROP
  iptables -A OUTPUT -p icmp -j DROP

  ip rule add from 0.0.0.0 table $AR_ID
  ip route add default dev $DEF_IF src $MN_HOM_IP via $GATEWAY table $AR_ID

  iptables -t nat -A POSTROUTING -s $MN_HOM_IP -j SNAT --to-source $MN_COA_IP

  # echo 'Mobile Node setup completed.'
```

Listing 9-5 Script to setup IPSec tunnel from mobile node

Following our initial implementation of the Mobile-IP protocol with IPSec tunnel support, together with the use of scripts and a handoff selection algorithm to optimise the overall handoff performance, a new project was commenced under Technology Assessment Project (TAP) funded by the Technology NZ (TechNZ) in conjunction with Telecom NZ. This *ABC Extension* project was focussed on porting the Linux implementation of our Mobile-IP-IPSec implementation to the Windows environment using the Java programming language. The Windows implementation also included the profile-based handoff.

9.7.3 **Windows Implementation**

The implementation of a policy-managed mobile client was particularly challenging in the Windows operating system especially due to its closed architecture. Other challenges were:

a. The non-server edition of Windows (e.g. Windows XP Professional edition) does not support IP-in-IP tunnelling which is an essential requirement for implementing the Mobile-IP protocol.

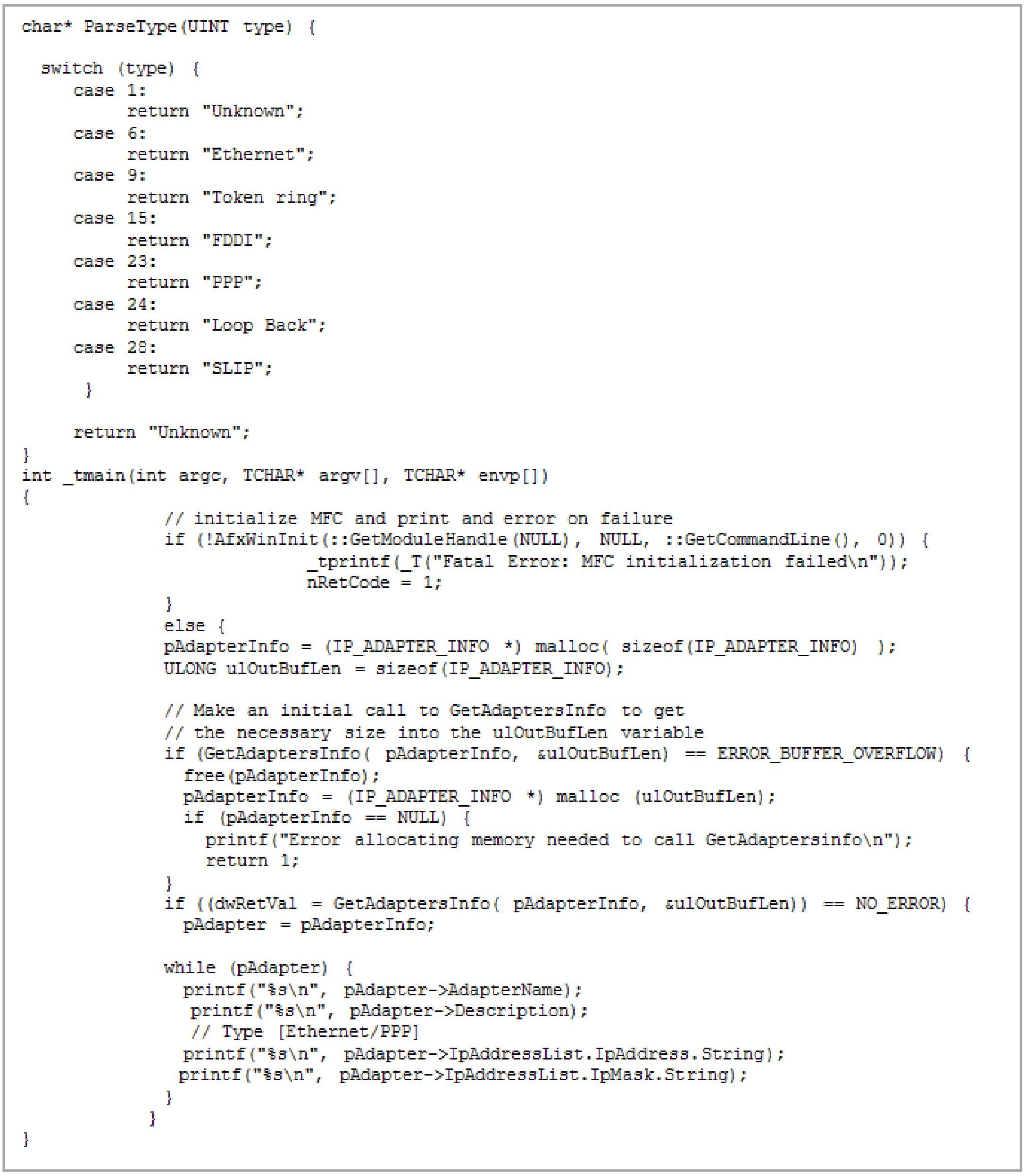
b. Additionally, in Windows, if an interface has a different subnet than the current default gateway address of the mobile node, then the routing entry to the default gateway is not possible. This implies that if the home address of the mobile node is 4.4.4.4 and current care-of address is 1.1.1.1, then the routing entry for 4.4.4.4 to the default gateway (i.e. 1.1.1.1) is not supported.

c. Our preferred choice of language, Java, only recognises TCP/UDP headers and does not support manipulation of raw IP packets.

d. In IPv4, the protocol number is used to configure the firewalls, routers, proxy servers, etc. This protocol number can be found in the *Protocol* field of an IP header. The Linux identifies IP-in-IP encapsulation (protocol number 4) while the Windows identify IP-within-IP encapsulation (protocol number 96) [252]. Hence, IP-in-IP tunnelling between a Windows based mobile node and a Linux based home agent was not possible.

We solved all the problems in the Windows implementation and the mobile client supported mobility in hotspots and Road Warrior scenarios.

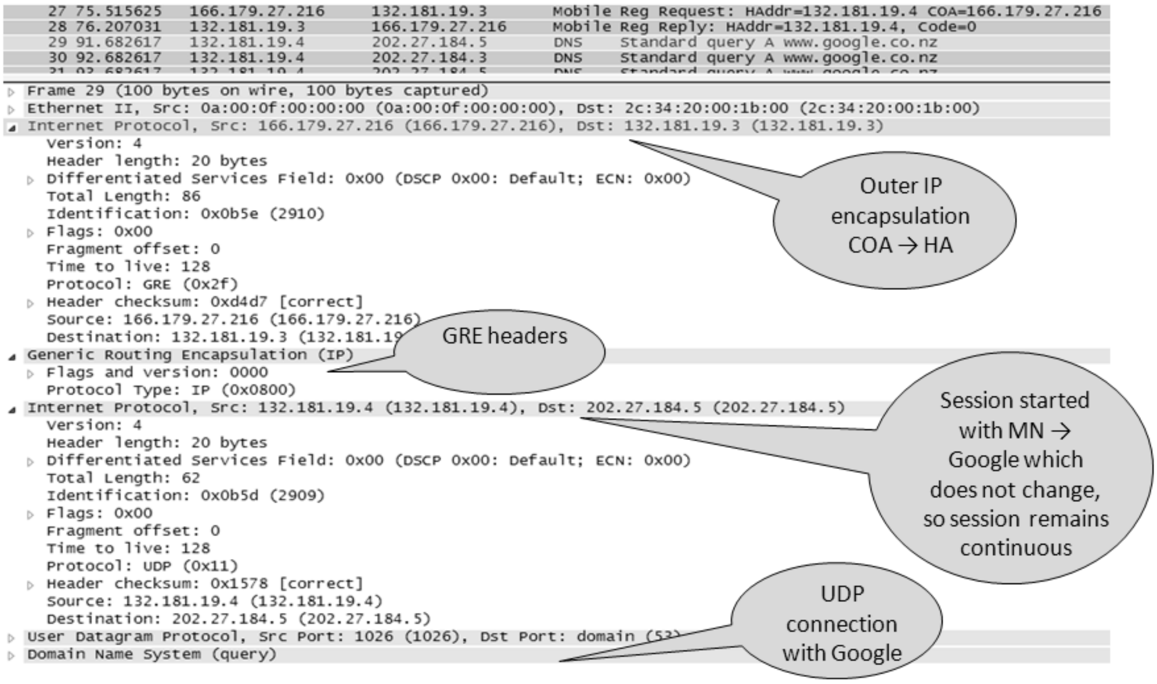
a. We solved the IP-in-IP tunnelling problem by developing a virtual device driver and assigning it the fixed home address of the mobile node. All the packets were managed in kernel mode while separate daemons were used to capture the local packets. The introduced virtual driver provided an interface for direct programming at the kernel level and several device level programs were written such as abcHotspots (to detect available WLAN hotspots), abcInt (to display the interface information) and abcRouting (to override routing table entries to add virtual ARP and virtual gateway entries). Listing 9-6 presents the implementation of abcInt using Microsoft Foundation Class (MFC) Library [253].



```
char* ParseType(UINT type) {

  switch (type) {
     case 1:
          return "Unknown";
     case 6:
          return "Ethernet";
     case 9:
          return "Token ring";
     case 15:
          return "FDDI";
     case 23:
          return "PPP";
     case 24:
          return "Loop Back";
     case 28:
          return "SLIP";
      }

     return "Unknown";
}
int _tmain(int argc, TCHAR* argv[], TCHAR* envp[])
{
             // initialize MFC and print and error on failure
             if (!AfxWinInit(::GetModuleHandle(NULL), NULL, ::GetCommandLine(), 0)) {
                           _tprintf(_T("Fatal Error: MFC initialization failed\n"));
                           nRetCode = 1;
             }
             else {
             pAdapterInfo = (IP_ADAPTER_INFO *) malloc( sizeof(IP_ADAPTER_INFO) );
             ULONG ulOutBufLen = sizeof(IP_ADAPTER_INFO);

             // Make an initial call to GetAdaptersInfo to get
             // the necessary size into the ulOutBufLen variable
             if (GetAdaptersInfo( pAdapterInfo, &ulOutBufLen) == ERROR_BUFFER_OVERFLOW) {
               free(pAdapterInfo);
               pAdapterInfo = (IP_ADAPTER_INFO *) malloc (ulOutBufLen);
               if (pAdapterInfo == NULL) {
                 printf("Error allocating memory needed to call GetAdaptersinfo\n");
                 return 1;
             }
             if ((dwRetVal = GetAdaptersInfo( pAdapterInfo, &ulOutBufLen)) == NO_ERROR) {
               pAdapter = pAdapterInfo;

             while (pAdapter) {
               printf("%s\n", pAdapter->AdapterName);
                printf("%s\n", pAdapter->Description);
                // Type [Ethernet/PPP]
               printf("%s\n", pAdapter->IpAddressList.IpAddress.String);
              printf("%s\n", pAdapter->IpAddressList.IpMask.String);
             }
            }
}
```

Listing 9-6 MFC in Windows for Interface Programming

b. The GRE protocol was implemented instead of IP-in-IP or IP-within-IP which was identified by both Windows and Linux platforms. The GRE protocol increased the payload by 8 bytes. However, we disabled the *key* field which saved 4 bytes in each packet. Figure 9-12 depicts the communication where the mobile node is connected to a CDMA network (with co-located care-of address as 166.179.27.216), and a GRE Mobile-IP tunnel is between the mobile node and home agent to communicate with Google.com (202.27.184.5).



```
   27 75.515625   166.179.27.216     132.181.19.3       Mobile Reg Request: HAddr=132.181.19.4 COA=166.179.27.216
   28 76.207031   132.181.19.3       166.179.27.216     Mobile Reg Reply: HAddr=132.181.19.4, Code=0
   29 91.682617   132.181.19.4       202.27.184.5       DNS    Standard query A www.google.co.nz
   30 92.682617   132.181.19.4       202.27.184.3       DNS    Standard query A www.google.co.nz
   31 93.682617   132.181.19.4       202.27.184.5       DNS    Standard query A www.google.co.nz
▷ Frame 29 (100 bytes on wire, 100 bytes captured)
▷ Ethernet II, Src: 0a:00:0f:00:00:00 (0a:00:0f:00:00:00), Dst: 2c:34:20:00:1b:00 (2c:34:20:00:1b:00)
◢ Internet Protocol, Src: 166.179.27.216 (166.179.27.216), Dst: 132.181.19.3 (132.181.19.3)
    Version: 4
    Header length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 86
    Identification: 0x0b5e (2910)
  ▷ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: GRE (0x2f)
  ▷ Header checksum: 0xd4d7 [correct]
    Source: 166.179.27.216 (166.179.27.216)
    Destination: 132.181.19.3 (132.181.19
◢ Generic Routing Encapsulation (IP)
  ▷ Flags and version: 0000
    Protocol Type: IP (0x0800)
◢ Internet Protocol, Src: 132.181.19.4 (132.181.19.4), Dst: 202.27.184.5 (202.27.184.5)
    Version: 4
    Header length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 62
    Identification: 0x0b5d (2909)
  ▷ Flags: 0x00
    Fragment offset: 0
    Time to live: 128
    Protocol: UDP (0x11)
  ▷ Header checksum: 0x1578 [correct]
    Source: 132.181.19.4 (132.181.19.4)
    Destination: 202.27.184.5 (202.27.184.5)
▷ User Datagram Protocol, Src Port: 1026 (1026), Dst Port: domain (53
▷ Domain Name System (query)
```

Outer IP encapsulation COA → HA

GRE headers

Session started with MN → Google which does not change, so session remains continuous

UDP connection with Google

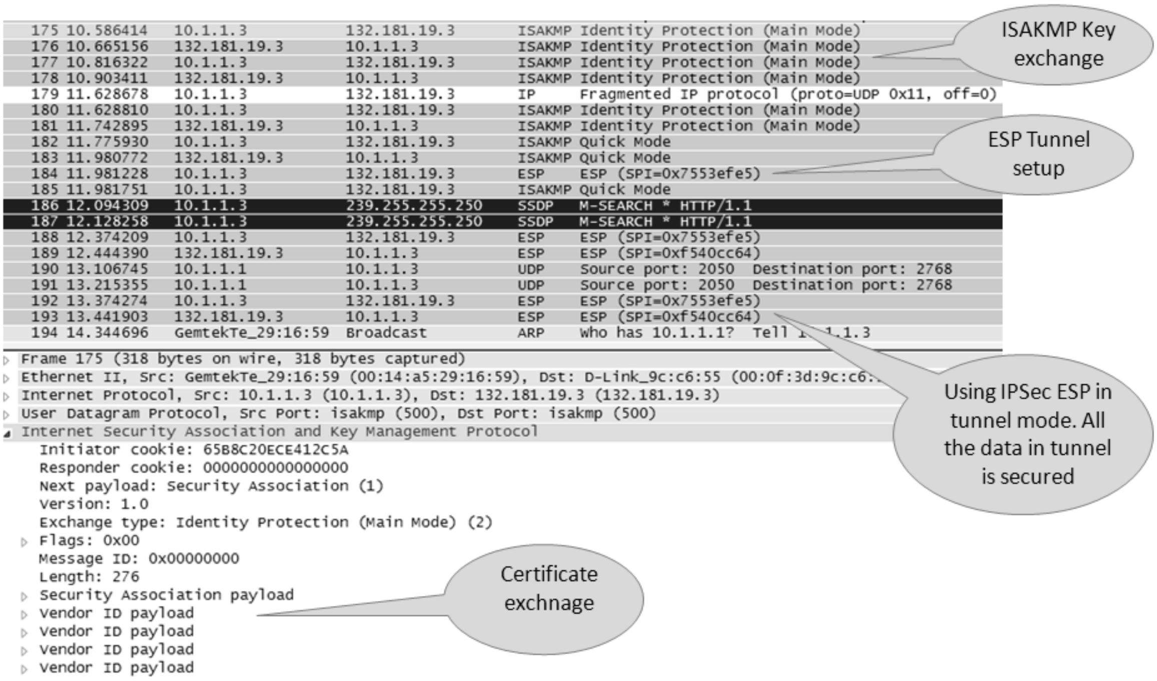
Figure 9-12 GRE protocol for Mobile-IP to support Windows and Linux

Our Windows implementation of the mobile client was focused on real-life scenarios of inter-working WLAN and 3G networks, where in most cases, the foreign agent and NAS do not exists. Additionally, 3GPP standards require mobile nodes to always be in co-located care-of address mode.

Furthermore, the hotspots usually maintain a network that uses private IP address space [254, 255]. The Private IP addresses were originally defined due to the shortage of publicly registered addresses created by the IPv4 standard. These addresses are private since they are not globally assigned and any packets from the network are not routable on the public Internet. To connect these isolated networks to the Internet, Network Address Translation (NAT) [256] gateways are used. However, Mobile-IP tunnelling is incompatible with NAT and new extensions were defined using the home agent UDP port for encapsulated data [257]. We proposed a different approach, since all the handoff should be performed in the network layer, where:

- if a mobile node is connected to a CDMA network, then it operates in a co-located care-of address mode;
- if a mobile node is connected to a WLAN hotspot area with a private address, then it operates in a co-located care-of address mode and maintains an IPSec tunnel between the home agent and the mobile node, hence passing safely

through NAT traversal and without breaking the Mobile-IP-IPSec tunnels. Figure 9-13 shows such a Mobile-IP-IPSec tunnel setup where all the data transferred between a mobile node (whose current co-located care-of address is a private address assigned by a WLAN network 10.1.1.3) and the home agent (i.e. 132.181.19.3) is protected.



```
175 10.586414  10.1.1.3           132.181.19.3     ISAKMP Identity Protection (Main Mode)
176 10.665156  132.181.19.3       10.1.1.3         ISAKMP Identity Protection (Main Mode)
177 10.816322  10.1.1.3           132.181.19.3     ISAKMP Identity Protection (Main Mode)
178 10.903411  132.181.19.3       10.1.1.3         ISAKMP Identity Protection (Main Mode)
179 11.628678  10.1.1.3           132.181.19.3     IP     Fragmented IP protocol (proto=UDP 0x11, off=0)
180 11.628810  10.1.1.3           132.181.19.3     ISAKMP Identity Protection (Main Mode)
181 11.742895  132.181.19.3       10.1.1.3         ISAKMP Identity Protection (Main Mode)
182 11.775930  10.1.1.3           132.181.19.3     ISAKMP Quick Mode
183 11.980772  132.181.19.3       10.1.1.3         ISAKMP Quick Mode
184 11.981228  10.1.1.3           132.181.19.3     ESP    ESP (SPI=0x7553efe5)
185 11.981751  10.1.1.3           132.181.19.3     ISAKMP Quick Mode
186 12.094309  10.1.1.3           239.255.255.250  SSDP   M-SEARCH * HTTP/1.1
187 12.128258  10.1.1.3           239.255.255.250  SSDP   M-SEARCH * HTTP/1.1
188 12.374209  10.1.1.3           132.181.19.3     ESP    ESP (SPI=0x7553efe5)
189 12.444390  132.181.19.3       10.1.1.3         ESP    ESP (SPI=0xf540cc64)
190 13.106745  10.1.1.1           10.1.1.3         UDP    Source port: 2050  Destination port: 2768
191 13.215355  10.1.1.1           10.1.1.3         UDP    Source port: 2050  Destination port: 2768
192 13.374274  10.1.1.3           132.181.19.3     ESP    ESP (SPI=0x7553efe5)
193 13.441903  132.181.19.3       10.1.1.3         ESP    ESP (SPI=0xf540cc64)
194 14.344696  GemtekTe_29:16:59  Broadcast        ARP    Who has 10.1.1.1?  Tell 10.1.1.3

▷ Frame 175 (318 bytes on wire, 318 bytes captured)
▷ Ethernet II, Src: GemtekTe_29:16:59 (00:14:a5:29:16:59), Dst: D-Link_9c:c6:55 (00:0f:3d:9c:c6:..
▷ Internet Protocol, Src: 10.1.1.3 (10.1.1.3), Dst: 132.181.19.3 (132.181.19.3)
▷ User Datagram Protocol, Src Port: isakmp (500), Dst Port: isakmp (500)
◢ Internet Security Association and Key Management Protocol
    Initiator cookie: 65B8C20ECE412C5A
    Responder cookie: 0000000000000000
    Next payload: Security Association (1)
    Version: 1.0
    Exchange type: Identity Protection (Main Mode) (2)
  ▷ Flags: 0x00
    Message ID: 0x00000000
    Length: 276
  ▷ Security Association payload
  ▷ Vendor ID payload
  ▷ Vendor ID payload
  ▷ Vendor ID payload
  ▷ Vendor ID payload
```

ISAKMP Key exchange

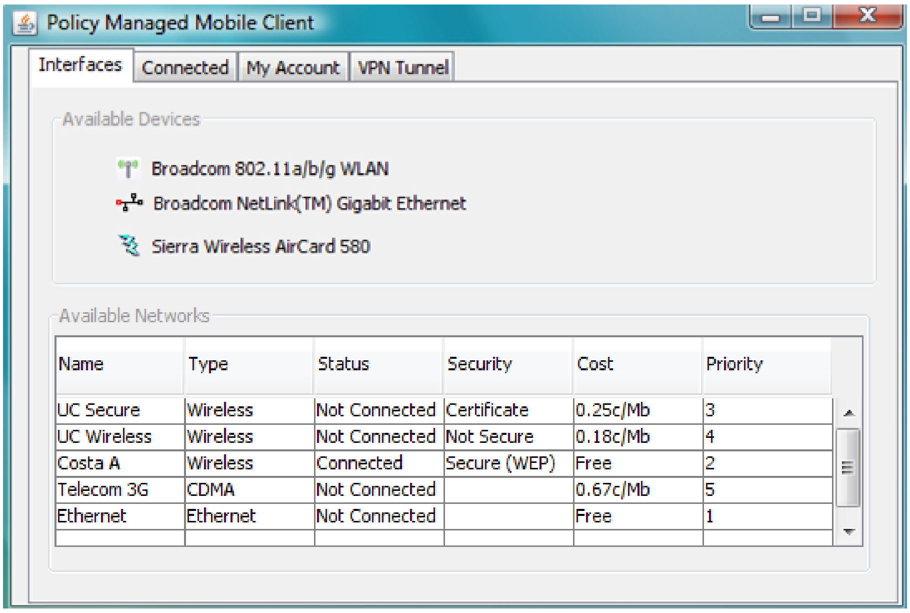
ESP Tunnel setup

Using IPSec ESP in tunnel mode. All the data in tunnel is secured

Certificate exchnage

Figure 9-13 Mobile-IP-IPSec tunnel through NAT traversals

The implementation of the proposed architecture in a Windows environment is shown in Figure 9-14, where the mobile node has currently three interfaces (i.e. WLAN, Ethernet and a 3G network card). The WLAN interface has detected an available WLAN network together with its associated security and status information. The cost attribute is derived from an external profile while the priority attribute is derived from a user's personal preference i.e. internal profile.



Available Devices

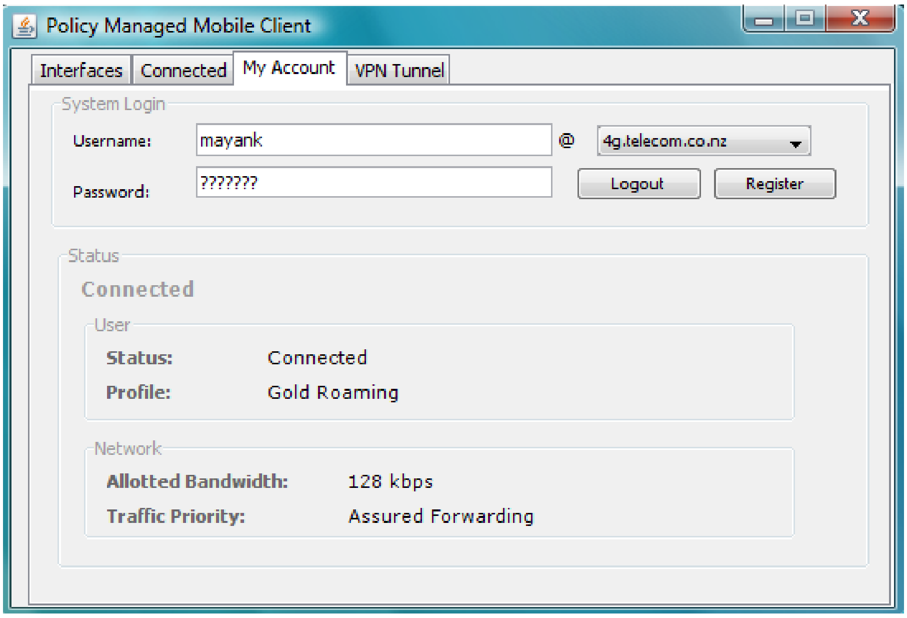
Broadcom 802.11a/b/g WLAN
Broadcom NetLink(TM) Gigabit Ethernet
Sierra Wireless AirCard 580

Available Networks

Name	Type	Status	Security	Cost	Priority
UC Secure	Wireless	Not Connected	Certificate	0.25c/Mb	3
UC Wireless	Wireless	Not Connected	Not Secure	0.18c/Mb	4
Costa A	Wireless	Connected	Secure (WEP)	Free	2
Telecom 3G	CDMA	Not Connected		0.67c/Mb	5
Ethernet	Ethernet	Not Connected		Free	1

Figure 9-14 Implementation of Policy Managed Mobile Client – 1

Figure 9-15 illustrates the profile download support integrated to the mobile node. When a user is authenticated to the 3G network (using its NAI e.g. mayank@4g.telecom.co.nz) then a subscription profile is downloaded into the mobile client (discussed in section 9.5.2.3).



System Login

Username: mayank @ 4g.telecom.co.nz

Password: ???????

Status

Connected

User

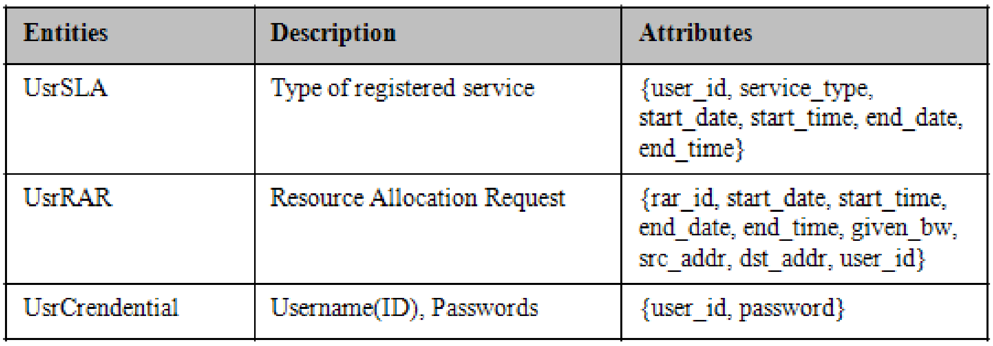
Status:	Connected
Profile:	Gold Roaming

Network

Allotted Bandwidth:	128 kbps
Traffic Priority:	Assured Forwarding

Figure 9-15 Implementation of Policy Managed Mobile Client - 2

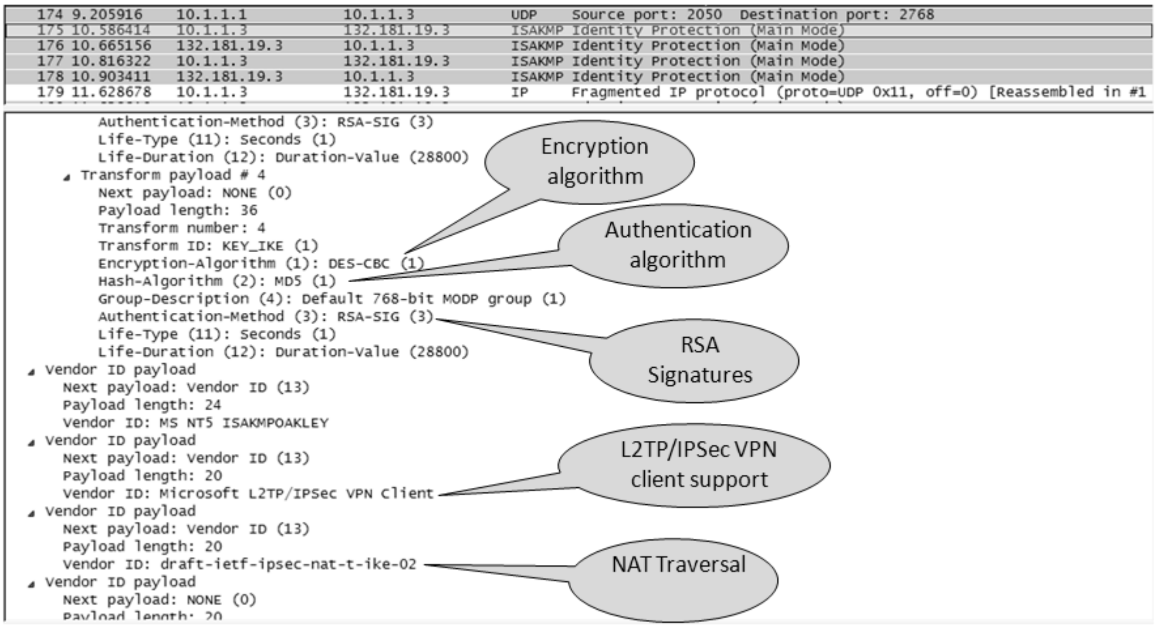
Any runtime request from the user is forwarded to the 3G network (service providing network) by using the Resource Allocation Request (UsrRAR) together with the user credentials (username-password in our case) and the user's current subscription with the service provider i.e. usrSLA. Table 9-5 presents the entities required to make a runtime request of resources. The UsrSLA maintains an initial subscription value (such as *gold user*) based on which a runtime request for UsrRAR is generated. In our implementation, a RAR is used to request a given bandwidth for a period of time. For example, request from a *gold* profile user for a *platinum* profile *bandwidth* for the *next hour* for a *video conference.* The request-response is based on the WSDL model and the message exchanges using SOAP.



Entities	Description	Attributes
UsrSLA	Type of registered service	{user_id, service_type, start_date, start_time, end_date, end_time}
UsrRAR	Resource Allocation Request	{rar_id, start_date, start_time, end_date, end_time, given_bw, src_addr, dst_addr, user_id}
UsrCrendential	Username(ID), Passwords	{user_id, password}

Table 9-5 Entities for SOAP for runtime requests by mobile user

A VPN tunnel is also supported where the mobile client can create a L2TP/IPSec VPN tunnel with its home agent or an external gateway of its organisation. This IPSec tunnel also facilitates NAT traversals and allows a mobile node to be connected with the home network via any hotspot. Figure 9-16 shows such a L2TP/IPSec VPN tunnel between a mobile node inside a hotspot (10.1.1.3) and its home agent (132.181.19.3).



```
174 9.205916   10.1.1.1       10.1.1.3       UDP    Source port: 2050  Destination port: 2768
175 10.586414  10.1.1.3       132.181.19.3   ISAKMP Identity Protection (Main Mode)
176 10.665156  132.181.19.3   10.1.1.3       ISAKMP Identity Protection (Main Mode)
177 10.816322  10.1.1.3       132.181.19.3   ISAKMP Identity Protection (Main Mode)
178 10.903411  132.181.19.3   10.1.1.3       ISAKMP Identity Protection (Main Mode)
179 11.628678  10.1.1.3       132.181.19.3   IP     Fragmented IP protocol (proto=UDP 0x11, off=0) [Reassembled in #1
```

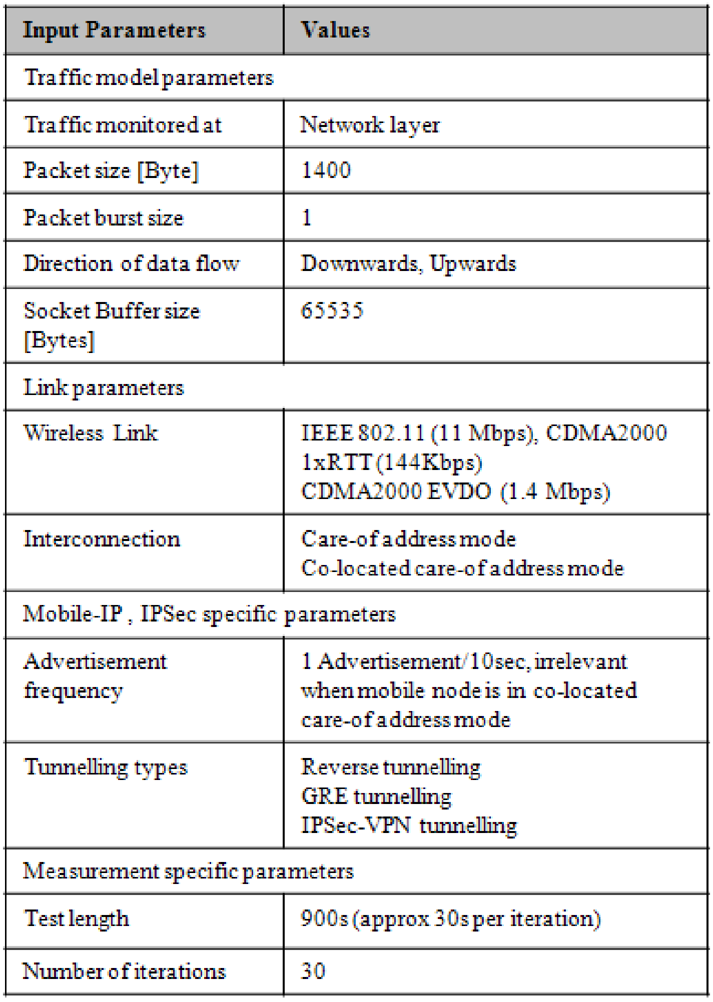
```
      Authentication-Method (3): RSA-SIG (3)
      Life-Type (11): Seconds (1)
      Life-Duration (12): Duration-Value (28800)
    ◢ Transform payload # 4
      Next payload: NONE (0)
      Payload length: 36
      Transform number: 4
      Transform ID: KEY_IKE (1)
      Encryption-Algorithm (1): DES-CBC (1)
      Hash-Algorithm (2): MD5 (1)
      Group-Description (4): Default 768-bit MODP group (1)
      Authentication-Method (3): RSA-SIG (3)
      Life-Type (11): Seconds (1)
      Life-Duration (12): Duration-Value (28800)
◢ Vendor ID payload
    Next payload: Vendor ID (13)
    Payload length: 24
    Vendor ID: MS NT5 ISAKMPOAKLEY
◢ Vendor ID payload
    Next payload: Vendor ID (13)
    Payload length: 20
    Vendor ID: Microsoft L2TP/IPSec VPN Client
◢ Vendor ID payload
    Next payload: Vendor ID (13)
    Payload length: 20
    Vendor ID: draft-ietf-ipsec-nat-t-ike-02
◢ Vendor ID payload
    Next payload: NONE (0)
    Payload length: 20
```

Figure 9-16 L2TP/IPSec tunnel support for VPN

9.7.4 Test Results

We will now present the testbed results to analyse the impact of hard and vertical handoffs on the network layer and analyse the overall performance degradation. For comparison we will then present the improvements in handoff latency resulting from our proposed solution.

To demonstrate the handoff latency, cases of LAN-WLAN-CDMA and WLAN-CDMA-LAN and CDMA-LAN-WLAN are included since these are the most common scenarios. The parameters of our experiments are summarised in Table 9-6. Throughout the Mobile-IP handoff experiments, the mobile node was moved between WLAN, CDMA and LAN networks and spends the same amount of time in each network. The following section presents the test results.



Input Parameters	Values
Traffic model parameters	
Traffic monitored at	Network layer
Packet size [Byte]	1400
Packet burst size	1
Direction of data flow	Downwards, Upwards
Socket Buffer size [Bytes]	65535
Link parameters	
Wireless Link	IEEE 802.11 (11 Mbps), CDMA2000 1xRTT (144Kbps) CDMA2000 EVDO (1.4 Mbps)
Interconnection	Care-of address mode Co-located care-of address mode
Mobile-IP , IPSec specific parameters	
Advertisement frequency	1 Advertisement/10sec, irrelevant when mobile node is in co-located care-of address mode
Tunnelling types	Reverse tunnelling GRE tunnelling IPSec-VPN tunnelling
Measurement specific parameters	
Test length	900s (approx 30s per iteration)
Number of iterations	30

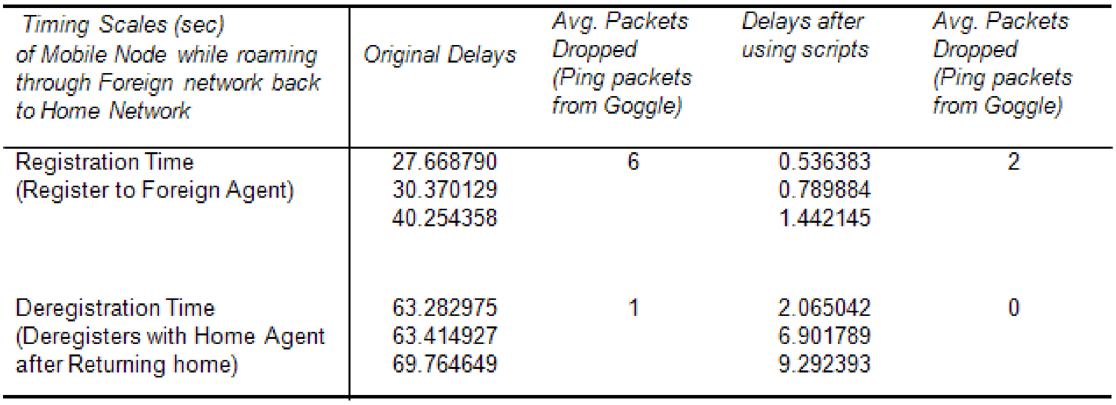
Table 9-6 Standard parameters used in the experiments

9.7.4.1 **Handoff latency**

When a mobile node changes the interface being used for communication, a *Registration Request* is sent from the mobile node to the home agent. While this *Registration* is being sent, any packets that were en route for the mobile node will be dropped, as its IP address has changed. This causes a noticeable delay while handoff occurs.

Table 9-7 shows the average values over 30 iterations of the handoff components for different handover scenarios. These iterations were performed for 3 times a day (i.e. at 10 am, 3 pm and 7 pm) for a period of 7 continuous days. All the experiments were performed in a real environment where the data packets were injected to a live 3G network while the WLAN connections were available from the University's Internet Security lab.

The handoff latency values included in the default Dynamics Mobile-IP implementation are too high. For example, the original handoff latency in Dynamics Mobile-IP implementation when a mobile node registers to a foreign network takes around 30 seconds to complete.



Timing Scales (sec) of Mobile Node while roaming through Foreign network back to Home Network	*Original Delays*	*Avg. Packets Dropped (Ping packets from Goggle)*	*Delays after using scripts*	*Avg. Packets Dropped (Ping packets from Goggle)*
Registration Time (Register to Foreign Agent)	27.668790 30.370129 40.254358	6	0.536383 0.789884 1.442145	2
Deregistration Time (Deregisters with Home Agent after Returning home)	63.282975 63.414927 69.764649	1	2.065042 6.901789 9.292393	0

Table 9-7 Improvement in handoff latency

We have written some scripts (discussed in section 9.7.2) which considerably decreases the handoff latencies with reduced number of dropped packets. Table 9-7 compares the improvement in handoff performance after using the scripts.

When the mobile node returns to the home network, it needs to deregister from the home agent so that any packets destined for the mobile node are not required to be forwarded by the home agent. In Table 9-7, the *Deregistration* time seems to be higher than the *Registration* time, since the mobile node waits for the current *binding* to expire (usually based on an exponential back-off algorithm) and then deregisters from the home agent. However, this wait does not have any considerable side effects as the mobile node accepts packets as soon as returning home.

Our optimised scripts based on the proposed handoff selection algorithm considerably decrease the handoff latencies with a reduced number of handoffs, thus reducing the total number of dropped packets (Figure 9-17).

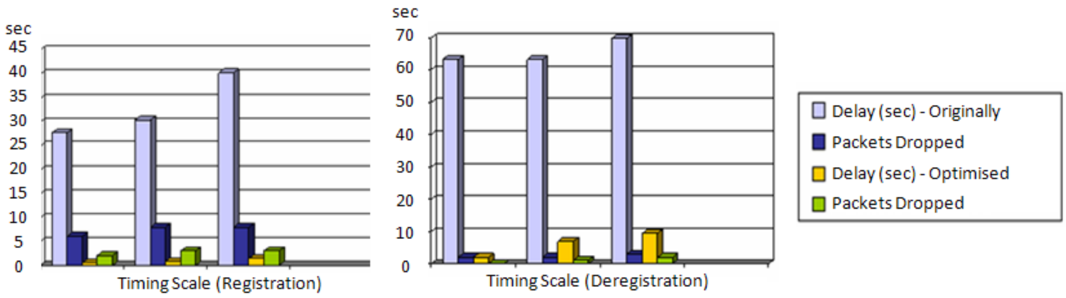
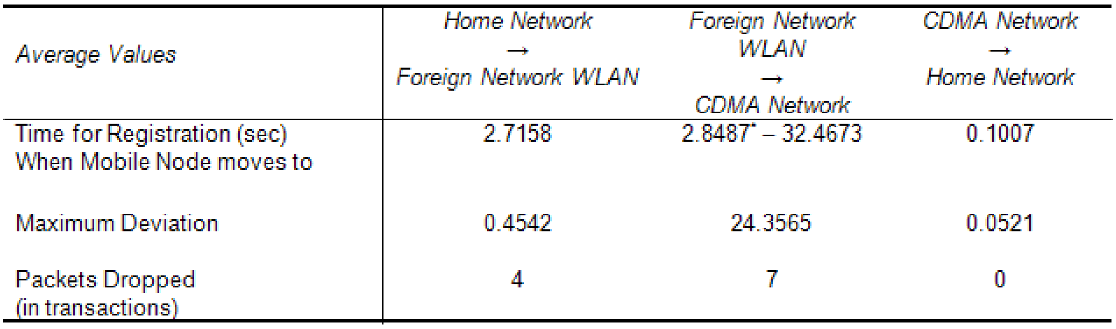


Figure 9-17 Improvement in handoff latency

The scripts which were developed for the foreign WLAN network handoff were extended to incorporate the CDMA network. Table 9-8 shows the observed handoff latencies. Not surprisingly, the delay when connecting to a foreign WLAN network is higher in comparison with the previous experiments since IPSec tunnels are being used, and the home agent and the mobile node needs to negotiate, authenticate and then create IPSec tunnels.



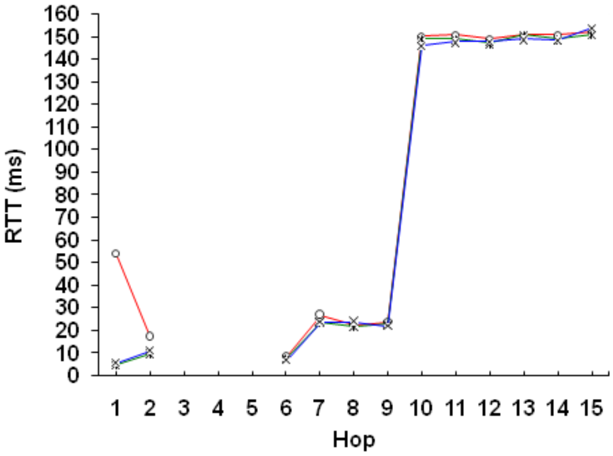
Average Values	*Home Network* → *Foreign Network WLAN*	*Foreign Network WLAN* → *CDMA Network*	*CDMA Network* → *Home Network*
Time for Registration (sec) When Mobile Node moves to	2.7158	2.8487* – 32.4673	0.1007
Maximum Deviation	0.4542	24.3565	0.0521
Packets Dropped (in transactions)	4	7	0

Table 9-8 Handoff latencies

However, when the handoff latency is small, while moving towards the CDMA network, the IPSec tunnels are not created smoothly i.e. only one way (from mobile node to home agent and not vice versa). The probable reason may be because the Dynamics Mobile-IP implementation cannot update the Home agent's tunnels rapid enough.

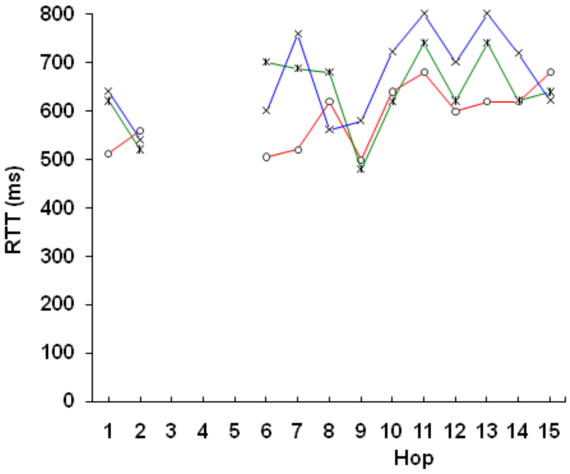
9.7.4.2 **Round Trip Time (RTT)**

Graph 9-1 shows a trace route from the mobile node to *www.google.co.nz* (Google), when the mobile node is at a foreign network using the 802.11b interface. A decrease in the round trip time at the beginning is as expected resulting from nodes caching addressing information, as well as unreliability in the wireless communications. The gaps at hops 3, 4 and 5 may be caused by university firewalls which do not respond to *trace route*. The high leap at hop 9 shows the slow speed of the Internet in comparison with the high speed at the local connection.



Graph 9-1 Round Trip Time when mobile node is at foreign WLAN network

Graph 9-2 shows trace route from the mobile node to *Google,* when the mobile node is at the foreign network in a co-located care-of address mode using the CDMA interface. The round trip time is much higher compared with that experienced in the WLAN, as the *ping request* packets from the CDMA interface go firstly to the home agent (via slow CDMA network) and then home agent relays the packets to *Google* again via the same slow CDMA network. Any *ping reply* from Google is sent back through the same slow speed CDMA network to the home agent, which then forwards it to the mobile node.

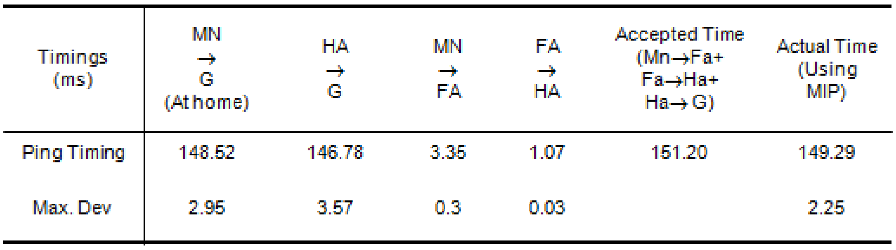


Graph 9-2 Round Trip Time when mobile node is in CDMA network

9.7.4.3 **Response Time**

The response time is determined by the time packets take to travel between the mobile node and the correspondent node (the node with which the mobile node is communicating with such as a *Google* web server or organisation’s file server). Given the low load in the home agent (HA), the response time is close to the round trip time (RTT).

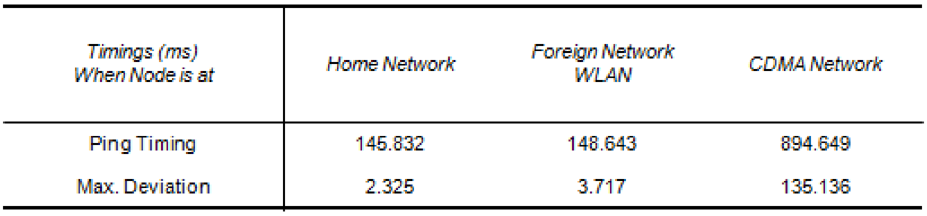
Using RTT measures to the *Google* gave response-times observed in Table 9-9. These results display the expected characteristics i.e. mobile node at home requires less time for data communication compared with when it resides at a foreign network. The packets move from mobile node (MN)→foreign agent (FA)→home agent (HA)→*Google* (G) and backtrack along the same path. Additional measurements were made to predict *accepted time* compared to the *actual time* taken.



Timings (ms)	MN → G (At home)	HA → G	MN → FA	FA → HA	Accepted Time (Mn→Fa+ Fa→Ha+ Ha→G)	Actual Time (Using MIP)
Ping Timing	148.52	146.78	3.35	1.07	151.20	149.29
Max. Dev	2.95	3.57	0.3	0.03		2.25

Table 9-9 Response time - Mobile IP

Table 9-10 presents the response times observed in the respective foreign WLAN and CDMA networks while using Mobile-IP in co-located care-of address mode with reverse tunnelling and IPSec enabled.

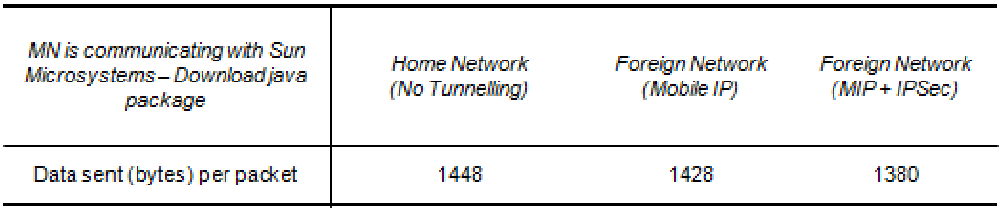


Timings (ms) *When Node is at*	*Home Network*	*Foreign Network* *WLAN*	*CDMA Network*
Ping Timing	145.832	148.643	894.649
Max. Deviation	2.325	3.717	135.136

Table 9-10 Response time - Mobile IP and IPSec tunnels

9.7.4.4 **Throughput**

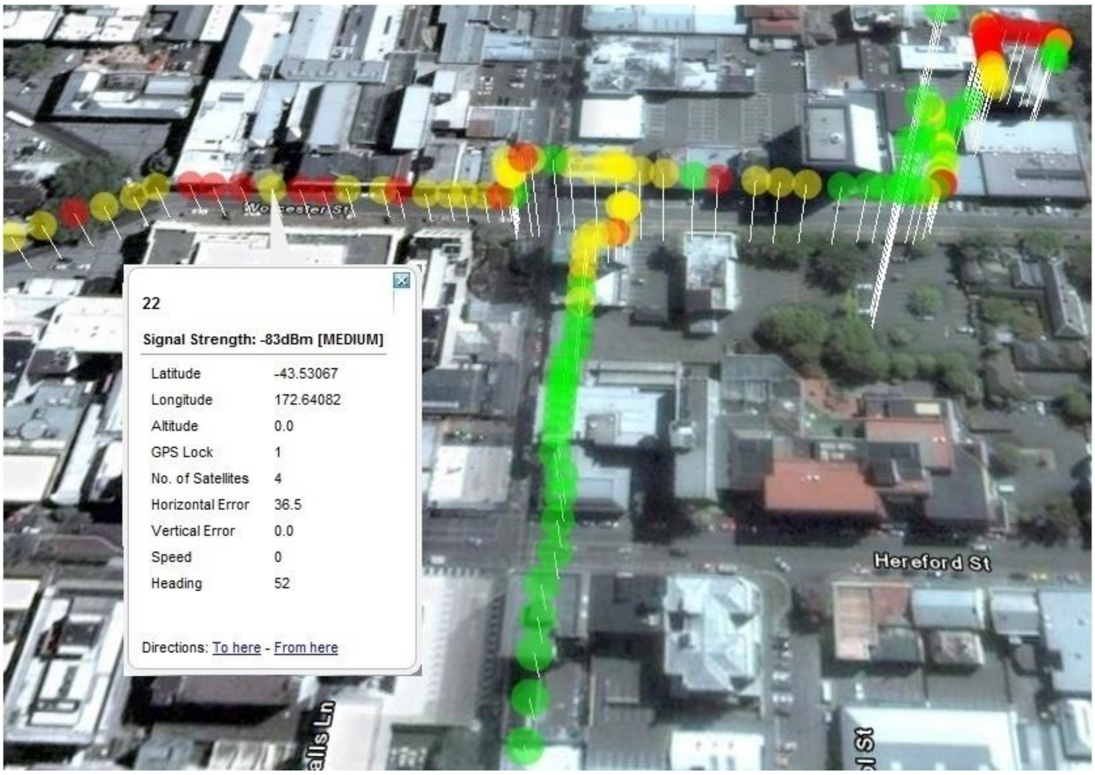
Mobile-IP introduces a second IP header (in addition to the original IP header, i.e. IP-IP tunneling) which decreases the payload of data transmission. Additionally IPSec headers create additional tunnel overhead on top of MIP transmission. Table 9-11 details the effects of various tunneling options on payload data.



MN is communicating with Sun Microsystems – Download java package	Home Network (No Tunnelling)	Foreign Network (Mobile IP)	Foreign Network (MIP + IPSec)
Data sent (bytes) per packet	1448	1428	1380

Table 9-11 Effect of tunnelling on payload data

Figure 9-18 shows the monitoring of signal strength of available hotspot areas based on the proposed network selection algorithm in a Windows environment. A typical case scenario is described as when a mobile user is moving in an area with hotspots and a ubiquitous wide area network. The policy-managed mobile client continually monitors and updates the internal database with the available network services and, in accordance with the internal profile, performs a handoff if a better access network is available.



22

Signal Strength: -83dBm [MEDIUM]

Latitude	-43.53067
Longitude	172.64082
Altitude	0.0
GPS Lock	1
No. of Satellites	4
Horizontal Error	36.5
Vertical Error	0.0
Speed	0
Heading	52

Directions: To here - From here

Figure 9-18 Handoff decisions based on internal profile

9.8 Summary

One of the main challenges in offering ABC services to mobile users from the client side is heterogeneity. The diversity in the environments augments the complexity in every stage of the handover process i.e. selecting the best available network, selecting an optimum time to handoff and continuous analysis of dynamic network conditions to provide the mobile user a best available service.

In addition, as the mobile user prefers to connect to the best network available, and wants to be connected at all times with acceptable QoS and security, a number of decisions are required to fully support and to some extent automate this seamless and transparent handoff. A multitude of security, authentication and other services protocols need to be managed together with the runtime requests of the mobile user and network's ability to provide those services. User's preferences are still important in this (semi)transparent handoff scenario, where a user may not want to handoff to a better service when other factors such as cost are considered.

The mobile device needs to have a management framework which can continuously analyse current options of available access networks, select the best time for the handoff, and then perform the handoff. The handoff must be fast and transparent with minimal effect on a user's current sessions and occur *only* when necessary. Similarly, a handoff algorithm must, for example, be able to evaluate all available networks and select the best performing network as fast as possible in order to avoid interruptions in communications. This is particularly difficult as wireless performance can fluctuate rapidly due to radio interference, especially if the coverage is bad.

Throughout this chapter, we have proposed an architecture of a policy-managed mobile client to support seamless handoff across multiple access networks. The proposed mobile client architecture supports multi-domain authentication and security for different scenarios such as a user in a hotspot or as a road warrior. The network selection algorithm and the introduced Infrastructure parameter help in selecting an optimum time and the best available access network to handoff. The runtime requirements and preferences of mobile users are managed by their internal profile with an ability to demand a service in a real-time environment and, external user profile is downloaded from the access network. We have also presented the implementation of the proposed architecture in Linux and Windows environments. The features offered by our proposed policy-managed mobile client architecture are summarised in Figure 9-19.

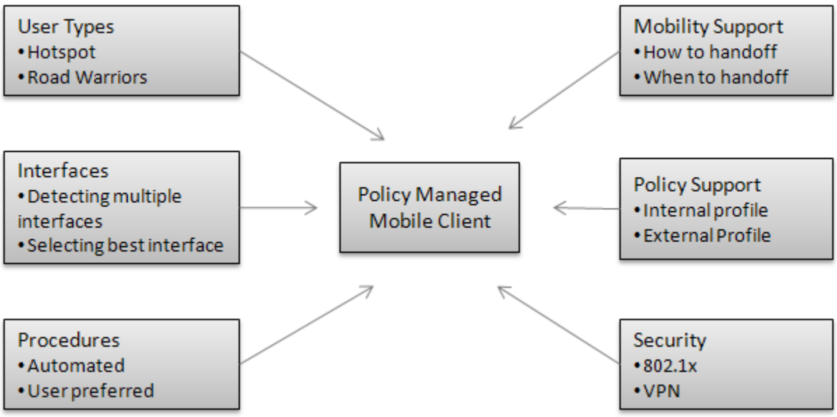


Figure 9-19 Features of proposed policy-managed mobile client

By analysing our test results, we understand that it is inevitable to completely eliminate the side effects of any handoff and there will be effects of reduced payload when additional headers of protocols (such as Mobile-IP, GRE and IPSec) are added to the data packet. However, we believe that this side effect is well balanced by the offered services of ABC where mobility, authentication and security are provided. The performance analysis of the proposed architecture also showed the improved overall performance resulting from reducing the number of handoffs and minimising the number of dropped packets, which validates our architectural framework.

Chapter 10
Conclusions

10.1 Summary

It is becoming evident that future network environments are unlikely to consist of simply one access technology but will integrate multiple access technologies, adding complexities to the mobility management systems. Users will want to be connected at all the times, and preferably with the best access network available. The seamless inter-networking to provide ABC services will be a basic feature in mobile terminals to allow connectivity in heterogeneous environments.

To achieve this ABC state, mobile devices need more intelligent solutions to offer seamless connectivity to mobile users. Additionally, service providing networks need to support a management framework to dynamically manage their resources to offer requested services in real-time, and satisfy service-level agreements with their home and roaming users.

The overall objective of this research was to provide managed ABC services over underlying heterogeneous wireless and mobile platforms, while maintaining negotiated security and QoS in a modular and scalable environment. In this research work, we have proposed, designed and implemented a model and architecture for the management of heterogeneous networks.

This work extends the existing IETF PBNM model by introducing a new *layered-approach*. This layered approach provides a unique flexibility to the organisation so that they can choose favourable semantic and syntax approaches, which facilitates the separation of management policies from their implementation in an open and heterogeneous environment.

Furthermore, we have proposed, designed and implemented a model of a policy-managed mobile client and its architecture, to support seamless handoff across multiple access networks. The proposed mobile client supports multi-domain authentication and security with a downloadable user profile.

In the following section, we will present a summary of the work proposed in this thesis. In the last section we will discuss future work which can be carried out to enhance both the design and the implementation proposed in this work.

10.2 Ideas presented in this research

Before starting the design of the proposed models, we set a number of initial requirements that the framework had to handle. Many of these requirements were due to the fact of managing heterogeneous and mobile (mainly wireless) environments, although there were other requirements that had their origin in the initial goals of the framework, and were considered as basic functionalities of any management system.

The proposed management model presents a number of novel features that either fill gaps not yet covered in the current state-of-the-art research projects, or suggest new modular solutions to specific problems which may be utilised as desired by the organisations implementing this model. The proposed model for managing a network is independent of the management level. Furthermore, the architecture design described in this document contains all the management functionality needed to work as a complete solution.

1. A new *layered-approach* is proposed to extend the IETF PBNM model to allow separate policy representations for semantic and syntactic analysis.

   Semantic analysis defines a formal methodology to understand and analyse policies for detecting inherent conflicts and for providing conflict resolution. *Syntax analysis* supports the distribution of policies in a consistent format for deployment in network elements involved. The primary focus of such semantic and syntax representations is to define a formal policy structure and allow policy negotiations based on current network conditions for intra- and inter-network management.

This separation of different abstract layers allow a stepwise refinement from an informal high-level business policy to network specific operational commands, and provides a clear distinction between understanding and implementing policies. The concept of *mapping translators* is introduced which provides a formal procedure to translate the policies from one representation to another (i.e. from a semantic analysis approach to a syntax representation approach).

2. After analysing various semantic analysis approaches (e.g. based on modern algebra such as predicate logic, event calculus or deontic-logic and based on a structure state such as tree, graph or finite state automata), we have proposed a new method for policy conflict detection and resolution.

   The proposed method extends the Role Based Access Control (RBAC) approach to define a Policy Schema to represent an overall structure of the organisation by grouping entities (i.e. Roles, Operations, Objects) into different hierarchies. A *composite mapping* technique is later defined to reflect the relationships between them.

   Assuming that the Policy Schema is in a consistent state (i.e. there are no inherent conflicts within policies), whenever a new policy is entered by the administrator, the resultant Policy Schema is analysed by the proposed semantic analysis methodology. The proposed semantic analysis is based on first-order logic and composite mapping to detect any conflicts and maintain an overall consistent state of the Policy Schema. To avoid any policy conflicts, we have used Separation of Duty and Chinese Wall security constraints. In case of any conflict(s), other factors such as time period, validity period and On-event conditions have been introduced to resolve these conflicts.

3. As new classes have been introduced in the NIST RBAC model (i.e. hierarchies of action and objects have been introduced), we have also extended the RBAC Policy Information Model (RBPIM) to support new RBAC policies. RBPIM is an

extension of IETF Policy Core Information Model (PCIM) and is specifically designed as a policy Information Model for storing and enforcing RBAC policies in a distributed heterogeneous system. We have extended RBPIM to introduce the *PolicyOnEventCondition* and *RBACObject* classes to provide time-based validity and support Chinese-Wall security model.

4. Several extensions to traditional approaches have been proposed by various working groups (e.g. SNMPConf, COPS) which could provide necessary capabilities for syntax analysis of policies for configuration, deployment and monitoring of network devices. On the other hand, Web services are emerging as a de-facto for access control in a request/response format which allows policies to be published by the service providers and offer support for policy negotiations (e.g. XACML).

   We have encapsulated the extended RBPIM by the COPS protocol for intra-domain policy deployment and XACML encapsulation for inter-domain policy negotiations. This provides a unique opportunity to utilise traditionally employed SNMP/COPS type protocols for intra-domain management with the monitoring elements, while using XACML type languages to support SLA negotiators to resolve any run-time policy conflicts.

5. A layered model for a policy-managed mobile client is designed and implemented to support seamless handoff, authentication and security to mobile clients across multiple access networks. This layered approach allows for extension of the IETF proposed Policy Enforcement Point (PEP) into the mobile device and manages it to behave in accordance with its profile. We have also introduced a new *Infrastructure parameter* to ensure whether the offered infrastructure support from the service provider is acceptable to the mobile client. The profile management component uses web services to identify its privileges with an ability to demand services in a run-time environment.

6. We have also proposed and implemented the architectural framework of the generic layered models for extended PBM framework and policy-managed mobile client. The introduced concepts of a *Policy Semantic Engine* and a *Policy*

Syntax Engine in a PBM framework allow dynamic configuration changes and enable policy deployment in a run-time environment. The *Roaming Manager* in policy-managed mobile clients provides mobility to the roaming mobile users acting as a home agent in Mobile-IPv4 environment. A handoff selection algorithm at the mobile node is also proposed while introducing various parameters to optimise the overall handoff timings.

Overall, the proposed models and architectures for the management of access networks and mobile clients provide ABC services over heterogeneous mobile environments allowing flexibility in policy negotiations and cater for the demands of a mobile user in run-time environments.

10.3 **Future Scope**

The work proposed in this thesis can be used as a starting point for other lines of research related to the management of heterogeneous networks. Some of them could be related to the enhancement of proposals in this thesis, and others might identify new fields where the developed concepts might be applied. Also, new methodologies might be explored to define ways to achieve the initial objectives.

Hereafter, we enlist possible future lines of work that might be followed to either extend or enhance the proposed solution.

- The performance of the proposed management architecture can be enhanced to achieve an optimum implementation in terms of performance.
- The security of the management system can be enhanced by encryption of the policies while they are stored in the policy repository, and while they are deployed to the network elements, as they contain credentials of the users whose resources are being modified by the policy.
- Another field where some of the proposed concepts can be applied and evaluated is inter-domain management. New techniques may be introduced to implement accounting models together with service negotiations, where mobile users using the services at other networks

can be charged with different pricing schemes, which provide for commercial adaptations.

- For large distributed networks, inter-operation requires a large number of policies to be defined, stored in the repository, and implemented in as-and-when required basis. An optimum approach to search the policies from the repository is a further interesting challenge.
- With the proposed model for management of a mobile entity, the optimisation in handover techniques, minimising delay times and supporting a greater number of real-time services is an open and challenging area of further research.

References

1. M3010, “International Telecommunication Union - Telecommunication Standardisation Sector (ITU-T), Principles for a Telecommunications Management Network,” *Recommendation M3010*, 2000.
2. J.D. Case, M.S. Fedor, M.L. Schoffstall, and C. Davin, “RFC 1157: Simple Network Management Protocol (SNMP),” *DDN Network Information Center, SRI International*, 1990.
3. ISDN, “ITU-T Recommendation I. 321 - B-ISDN Protocol Reference Model and its Application,” *International Telecommunication Union*, 1991.
4. Corba, “The Common Object Request Broker: Architecture and Specification,” Cited Sep 2008, 1991; http://www.corba.org/.
5. T.B. Downing, *Java RMI: Remote Method Invocation*, IDG Books Worldwide, Inc. Foster City, CA, USA, 1998.
6. L. Dimopoulou, E. Nikolouzou, P. Sampatakos, and L.S. Venieris, “QMTool: an XML-based management platform for QoS-aware IP networks,” *Network, IEEE*, vol. 17, no. 3, 2003, pp. 8-14.
7. S. Wright, R. Chadha, and G. Lapiotis, “Special Issue on Policy Based Networking,” *IEEE Network*, vol. 16, no. 2, 2002, pp. 8-56.
8. IPHighway, “Policy Standards and IETF Terminology,” *White Paper, Jan*, 2001.
9. DMTF, “Distributed Management Task Force,” Cited Sep 2008; http://www.dmtf.org/home.
10. IETF; http://www.ietf.org/html.charters/policy-charter.html.
11. M. Frodigh, S. Parkvall, C. Roobol, P. Johansson, and P. Larsson, “Future-generation wireless networks,” *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 8, no. 5, 2001, pp. 10-17.
12. W.W. Lu, “Fourth-generation mobile initiatives and technologies [Guest Editorial],” *Communications Magazine, IEEE*, vol. 40, no. 3, 2002, pp. 104-105.
13. E. Gustafsson, A. Jonsson, E. Res, and S. Stockholm, “Always best connected,” *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 10, no. 1, 2003, pp. 49-55.
14. D. Kosiur, *Understanding Policy-based Networking*, John Willey & Sons, 2001.
15. O. Prnjat, I. Liabotis, T. Olukemi, L. Sacks, M. Fisher, P. McKee, K. Carlberg, and G. Martinez, “Policy-based Management for ALAN-Enabled Networks,” *IEEE 3rd International Workshop on Policies-Policy*, 2002.
16. Incits, “Role Based Access Control,” *359-2004, American National Standards Institute/INCITS*, 2004, pp. 1-56.
17. IETFPol, “IETF Policy Framework group,” Cited Sep 2008; http://www.ietf.org/html.charters/policy-charter.html.
18. A. Smith, and M. Stevens, “Resource Allocation Protocol Working Group,” November 1999; http://www.ietf.org/html.charters/rap-charter.html.

19. N.C. Damianou, A. Bandara, M. Sloman, and E. Lupu, "A Survey of Policy Specification Approaches," *Department of Computing, Imperial College of Science Technology and Medicine, London*, 2002.
20. S. Duflos, G. Diaz, V. Gay, and E. Horlait, "A Comparative Study of Policy Specification Languages for Secure Distributed Applications," *Management Technologies for E-commerce and E-business Applications: 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2002*
21. I. Aib, N. Agoulmine, M.S. Fonseca, and G. Pujolle, "Analysis of Policy Management Models and Specification Languages," *Network control and Engineering for QoS, security and mobility II, Kluwer Academic Publishers, Norwell, MA*, 2003.
22. G. Tonti, J.M. Bradshaw, R. Jeffers, R. Montanari, N. Suri, and A. Uszok, "Semantic Web Languages for Policy Representation and Reasoning: A Comparison of KAoS, Rei, and Ponder," *Lecutre notes in Computer Science*, 2003, pp. 419-437.
23. T. Phan, J. Han, J.G. Schneider, T. Ebringer, and T. Rogers, "A Survey of Policy-Based Management Approaches for Service Oriented Systems," *19th Australian Conference on Software Engineering, 2008. ASWEC*, 2008, pp. 392-401.
24. J. Vivero, "MANBoP (Proposal of a Model for the Management of Active Networks Based on Policies)," PhD Thesis, Universitat Politecnica de Catalunya, 2003.
25. A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, "RFC3198: Terminology for Policy-Based Management," *Network Working Group*, 2001.
26. DAIDALOS, "Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services (DAIDALOS)," Cited Aug 2008; http://www.ist-daidalos.org.
27. D.F. Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, 2001, pp. 224-274.
28. S.G. Sohn, J. Kim, and J.C. Na, "Design of network security policy information model for policy-based network management," *The 7th International Conference on Advanced Communication Technology, 2005, ICACT 2005*, pp. 701-705.
29. DMTF, "Common Information Model Standards," Cited Aug 2008; http://www.dmtf.org/standards/standard_cim.php.
30. DMTF, "Directory Enabled Network," Cited Aug 2008; http://www.dmtf.org/standards/standard_den.php.
31. B. Moore, E. Ellesson, J. Strassner, and A. Westerinen, "RFC 3060: Policy Core Information Model–Version 1 Specification," *Internet RFCs, Internet Engineering Task Force*, 2001.
32. B. Moore, "RFC3460: Policy Core Information Model (PCIM) Extensions," *Internet RFCs, Internet Engineering Task Force*, 2003.
33. IETFRAP, "Resource Allocation Protocol group," http://www.ietf.org/html.charters/rapcharter.html.

34. Y. Snir, Y. Ramberg, J. Strassner, R. Cohen, and B. Moore, “RFC 3644: Policy Quality of Service Information Model,” *Network Working Group,* 2003.

35. J. Jason, L. Rafalow, and E. Vyncke, “RFC3585: IPsec configuration policy information model,” *Internet Engineering Task Force,* 2003.

36. D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, “RFC 2748: The COPS (Common Open Policy Service) Protocol,” *Internet Engineering Task Force,* 2000.

37. J.P. Thompson, “Web-based enterprise management architecture,” *Communications Magazine, IEEE,* vol. 36, no. 3, 1998, pp. 80-86.

38. SNMPConf, “Configuration Management with SNMP workgroup (snmpconf) IETF,” Cited Sep 2008; http://www.ietf.org/html.charters/OLD/snmpconf-charter.html.

39. S. Waldbusser, J. Saperia, and T. Hongal, “Policy Based Management MIB,< draft-ietf-snmpconf-pm-10. txt>,” *Internet Engineering Task Force (IETF), Feb,* 2002.

40. R. Story, C. Wang, S. Inc, M. Baer, R. Charlet, and W. Hardaker, “IPSP Working Group, Network Associates Inc, Internet Draft <draft-ietf-ipsp-ipsec-conf-mib-06. txt> (work in progress),” 2003.

41. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “The Ponder Specification Language,” *Workshop on Policies for Distributed Systems and Networks, Lecture Notes in Computer Science,* pp. 18-38.

42. N.C. Damianou, “A Policy Framework for Management of Distributed Systems,” Department of Computing, University of London, Imperial College of Science, Technology and Medicine, London, UK, 2002.

43. J.D. Moffett, and M.S. Sloman, “Policy hierarchies for distributed systems management,” *IEEE Journal on Selected Areas in Communications,* vol. 11, no. 9, 1993, pp. 1404-1414.

44. Jasmin, “The Jasmin Project,” Cited Aug 2008; http://www.ibr.cs.tu-bs.de/projects/jasmin/policy.html.

45. ANDRIOD, “Active Network Distributed Open Infrastructure Development (ANDRIOD),” Cited Aug 2008; http://www.cs.ucl.ac.uk/research/andriod/.

46. ALAN, “Application Layer Active Networking,” Cited Aug 2008; http://dmir.it.uts.edu.au/projects/alan/.

47. M. Fry, and A. Ghosh, “Application Level Active Networking,” *Computer Networks,* vol. 31, no. 7, 1999, pp. 655-667.

48. Y. Kanada, I. Center, and H. Ltd, “Dynamically extensible policy server and agent,” *Policies for Distributed Systems and Networks, 2002. Proceedings. Third International Workshop on,* 2002, pp. 236-239.

49. A. Banchs, “Analysis of the Distribution of the Backoff Delay in 802.11 DCF: A Step Towards End-to-End Delay Guarantees in WLANs,” *LECTURE NOTES IN COMPUTER SCIENCE,* 2004, pp. 54-63.

50. R. Prior, S. Sargento, D. Gomes, and R.L. Aguiar, “Heterogeneous Signaling Framework for End-to-End QoS Support in Next Generation Networks,” *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), 2005.*

51. I. Draft, “draft—ietf-mobile-fast—mipv6—05. txt,” *Fast Handover for Mobile IPv6.*

52. D.B.L. Technologies, and S.G.L. Technologies, “The Daidalos project and standardizing NGN in ETSI TISPAN-An overview.”

53. S. Cantor, J. Kemp, R. Philpott, and E. Maler, “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2. 0,” 2005.

54. Hewlett-Packard, “HP OpenView PolicyXpert 2.0 - Users Guide, Edition 1,” October 2000.

55. AllotCommunications, “NetPolicy Policy Based Management System, product documentation,” 2000.

56. Cisco, “CiscoAssure Policy Networking: End-to-End Quality of Service,” *Cisco Systems*, 1998.

57. J. Conover, “Policy-Based Network Management,” Cited Sep 2008; http://www.networkcomputing.com/1024/1024f1.html.

58. M. Galic, A. Halliday, A. Hatzikyriacos, M. Munaro, S. Parepalli, and D. Yang, “A Secure Portal Using WebSphere Portal V5 and Tivoli Access Manager,” *ISBN: 073849853X*, 2003.

59. D. Ferraiolo, J. Cugini, and D.R. Kuhn, “Role-Based Access Control (RBAC): Features and Motivations,” *Proceedings of the Eleventh Annual Computer Security Applications Conference*, 1995.

60. M.D.F. Ferraiolo, M.D.M. Gilbert, and M.N. Lynch, “An Examination of Federal and Commercial Access Control Policy Needs,” *National Computer Security Conference, 1993 (16th) Proceedings: Information Systems Security: User Choices*, 1995.

61. H.L. Feinstein, “Final report: NIST Small Business Innovative Research (SBIR) grant: Role Based Access Control: phase 1,” *SETA Corporation. SETA Corporation*, 1995.

62. D.F. Ferraiolo, J.F. Barkley, and D.R. Kuhn, “A Role-Based Access Control Model and Reference Implementation Within a Corporate Intranet,” *ACM Transactions on Information and System Security*, vol. 2, no. 1, 1999, pp. 34-64.

63. R. Sandhu, D. Ferraiolo, and R. Kuhn, “The NIST model for role-based access control: towards a unified standard,” *Symposium on Access Control Models and Technologies: Proceedings of the fifth ACM workshop on Role-based access control*, vol. 26, no. 28, 2000, pp. 47-63.

64. V.D. Gligor, S.I. Gavrila, and D. Ferraiolo, “On the formal definition of separation-of-duty policies and theircomposition,” *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, 1998, pp. 172-183.

65. C. Perkins, “IP Mobility Support for IPv4,” *RFC 3344*, Aug 2002.

66. P. Vidales, J. Baliosian, J. Serrat, G. Mapp, F. Stajano, and A. Hopper, “Autonomic system for mobility support in 4G networks,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 12, 2005, pp. 2288-2304.

67. IETF, “IP Routing for Wireless/Mobile Hosts (mobileip),” 2003; http://www.ietf.org/html.charters/OLD/mobileip-charter.html.

68. J.D. Solomon, “Mobile IP: The Internet Unplugged,” *PTR Prentice Hall, Upper Saddle River, NJ*, 1998.

69. M. Ylianttila, M. Pande, J. Makela, and P. Mahonen, “Optimization scheme for mobile users performing vertical handoffsbetween IEEE 802.11 and GPRS/EDGE networks,” *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, vol. 6, 2001.

70. Dynamics, “Dynamics Mobile IP,” *Helsinki University of Technology, Finland, http://dynamics.sourceforge.net/*, 2001, Cited Sep 2008.

71. R.H. Katz, “Adaptation and mobility in wireless information systems,” *Communications Magazine, IEEE*, vol. 40, no. 5, 2002, pp. 102-114.

72. K. Lai, M. Roussopoulos, D. Tang, X. Zhao, and M. Baker, “Experiences with a Mobile Testbed,” *LECTURE NOTES IN COMPUTER SCIENCE*, 1998, pp. 222-237.

73. P. Philippopoulos, P. Fournogerakis, I. Fikouras, N. Fikouras, and C. Gorg, “NOMAD: Integrated Networks for Seamless and Transparent Service Discovery,” *Proceedings of the IST Mobile Summit 2002*, 2002.

74. N.A. Fikouras, “Filters for Mobile IP Bindings (NOMAD),” *IETF Draft, draftnomad-mobileip-filters-05. txt (work in progress)*, 2003.

75. K. Kuladinithi, N.A. Fikouras, C. Goerg, K. Georgios, and F.N. Pavlidou, “Filters for Mobile IPv6 Bindings (NOMADv6),” *draft-nomadv6-mobileip-filters-03 (work in progress)*, 2005.

76. Mind, “IST Project: Mobile IP based Network Developments,” 2000.

77. A. Cuevas, P. Serrano, C.J. Bernardos, J.I. Moreno, J. Jaehnert, K. Hyung-Woo, J. Zhou, D. Gomes, P. Gonçalves, and R. Aguiar, “Field Evaluation of a 4G True-IP network,” *IST Mobile Summit 2004*, 2004.

78. A. Misra, S. Das, A. Dutta, A. McAuley, S.K. Das, I. Center, and N.Y. Hawthorne, “IDMP-based fast handoffs and paging in IP-based 4G mobile networks,” *Communications Magazine, IEEE*, vol. 40, no. 3, 2002, pp. 138-145.

79. R. Ramjee, K. Varadhan, L. Salgarelli, S.R. Thuel, S.Y. Wang, and T. La Porta, “HAWAII: A Domain-Based Approach for Supporting Mobility in Wide-Area Wireless Networks,” *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 10, no. 3, 2002.

80. A.T. Campbell, J. Gomez, S. Kim, A.G. Valko, C.Y. Wan, and Z.R. Turanyi, “Design, Implementation, and Evaluation of Cellular IP,” *IEEE PERSONAL COMMUNICATIONS*, vol. 7, no. 4, 2000, pp. 42-49.

81. S. Aidarous, and T. Pevyak, “Telecommunication Network Management: Technologies and Implementations,” *IEEE Press*, 1997.

82. M3400, “International Telecommunication Union - Telecommunication Standardisation Sector (ITU-T), TMN management functions,” *Recommendation M3400*, 2000.

83. D.C. Verma *Policy-based Networking - Architecture and Algorithms*, New Riders, November 2000, pp. 9-12.

84. P. Horn, “Autonomic Computing: IBM’s Perspective on the State of Information Technology,” *IBM TJ Watson Labs, NY, 15th October*, 2001.

85. J.C. Strassner, *Policy-based Network Management - Solution for the Next Generation*, Morgan Kaufmann Publishers, pp 3-65, 2004.

86. K. Chan, R. Sahita, S. Hahn, and K. McCloghrie, “RFC3317: Differentiated Services Quality of Service Policy Information Base,” *Internet Engineering Task Force*, 2003.

87. R. Boutaba, and A. Polyrakis, “Extending COPS-PR with Meta-Policies for Scalable Management of IP Networks,” *Journal of Network and Systems Management*, vol. 10, no. 1, 2002, pp. 91-106.

88. R. Wies, “Policy Definition and Classification: Aspects, Criteria and Examples,” *Proceedings of the IFIP/IEEE International Workshop on Distributed Systems: Operation and Management*, 1994.

89. C.M. Schwarz, and P.K. Frost, *Chambers English Dictionary*, Larousse Kingfisher Chambers, 1988.

90. F.J. Garcia, G. Martinez, J.A. Botia, and A.F.G. Skarmeta, "Representing Security Policies in Web Informaion Systems," *Policy Management for the Web (PM4W), 14th International World Wide Web Conference*.

91. H. Kamoda, M. Yamaoka, S. Matsuda, K. Broda, and M. Sloman, "Policy conflict analysis using free variable tableaux for access control in web services environments," *Policy Management for the Web Workshop, Chiba, Japan, May*, 2005.

92. W3C, "Web Services Activity," 2002; http://www.w3.org/2002/ws/.

93. R. Sahita, S. Hahn, K. Chan, and K. McCloghrie, "RFC 3318: Framework Policy Information Base," *Network Working Group*, 2003.

94. T. Moses, "eXtensible Access Control Markup Language (XACML) Version 2.0," *OASIS Standard*, vol. 200502, 2005.

95. P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "RFC 3588: Diameter Base Protocol," *Network Working Group*, 2003.

96. DMTF, "Specification for the Representation of CIM in XML - Preliminary," *DSP0201, Version 2.2*, 2004.

97. M. Nakhjiri, *AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility*, Wiley, 2005.

98. M. Cappiello, A. Floris, and L. Veltri, "Mobility amongst heterogeneous networks with AAA support," *IEEE International Conference on Communications, ICC 2002*, vol. 4, 2002.

99. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," 2002.

100. J. Strassner, "Policy Based Network Management - Solutions for the Next Generation," R. Adams ed., Morgan Kaufman Publishers, 2004, pp. 3-16.

101. L. Howard, "RFC2307: An Approach for Using LDAP as a Network Information Service," *Network Working Group*, 1998.

102. Schema, "IBM," http://www.serc.iisc.ernet.in/ComputingFacilities/systems/cluster/vac-7.0/html/glossary/czgs.htm.

103. Y.T. Lee, "Information modeling: From design to implementation," *Proceedings of the Second World Manufacturing Congress*, pp. 315-321.

104. E.C. Lupu, and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, 1999, pp. 852-869.

105. M. Sloman, and K. Twidle, "Domains: a framework for structuring management policy," *Network and Distributed Systems Management*, 1994.

106. B.P. Dinsmore, M. Heyman, P. Kruus, and C. Scace, "Dynamic Cryptographic Context Management (DCCM) Report# 4: Final Report:[NAI Report# 0776]," 2000.

107. R. Sandhu, and Q. Munawer, "How to do discretionary access control using roles," *Proceedings of the third ACM workshop on Role-based access control*, 1998, pp. 47-54.

108. F. Dridi, G. Pernul, and T. Sabol, "The Webocracy Project: Overview and Security Aspects," *Schnurr et al., ProffessionellesWissensmanagement: Erfahrungen und Visionen. Shaker Verlag, Aachen*, 2001.

109. M. Nyanchama, and S.L. Osborn, "Access Rights Administration in Role-Based Security Systems," *Proceedings of the IFIP WG11. 3 Working Conference on Database Security VII*, 1994, pp. 37-56.

110. D.F.C. Brewer, and M.J. Nash, "The Chinese Wall security policy," *Proceedings of IEEE Symposium on Security and Privacy*, 1989, pp. 206-214.

111. Allot, “Policy Based Networking Architecture,” 2001; http://www.allot.com/media/EnternalLink/policymgmt.pdf.

112. R. Nabhen, E. Jamhour, and C. Maziero, “RBPIM: a PCIM-based framework for RBAC,” *Local Computer Networks, 2003. LCN'03. Proceedings. 28th Annual IEEE International Conference on*, 2003, pp. 52-61.

113. J. Strassner, B. Moore, R. Moats, and E. Ellesson, “RFC 3703: Policy Core Lightweight Directory Access Protocol (LDAP) Schema,” *Network Working Group*, 2004.

114. P. Marcu, “Reference Installation of the Ponder Policy Toolkit,” *Systementwicklungsprojekt, Ludwig–Maximilians–Universitat Munchen, April*, 2005.

115. F.J.G. Clemente, G.M. Perez, and A.F.G. Skarmeta, “An XML-Seamless Policy Based Management Framework,” *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3685, 2005, pp. 418.

116. H.S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, “XML Schema Part 1: Structures,” *W3C Recommendation*, vol. 2, 2001.

117. T.M.F. Gb, “Shared information/data (SID) model (Release 6.0),” *The TeleManagement Forum*, GB922, Dec 2005.

118. A. Tymofiejewicz, and A. Mickiewicza, “Policy-based management framework for an integrated heterogeneous network,” *Krajowe Sympozjum Telekomunikacji i Teleinformatyki, KSTiT'2006, Sep 2006*.

119. M.S. Sloman, “Policy Driven Management for Distributed Systems,” *Journal of Network and Systems Management*, vol. 2, no. 4, 1994, pp. 333-360.

120. J. Strassner, “Policy Based Network Management - Solutions for the Next Generation,” R. Adams ed., Morgan Kaufman Publishers, 2004, pp. 56-63.

121. R. Hilpinen, S. Kanger, K. Segerberg, and B. Hansson, *Deontic logic: Introductory and Systematic Readings*, Reidel, 1971.

122. F. Chen, and R.S. Sandhu, “Constraints for role-based access control,” *Proceedings of the first ACM Workshop on Role-based access control*, ACM New York, NY, USA, Nov 1995.

123. N. Dunlop, J. Indulska, and K. Raymond, “Dynamic conflict detection in policy-based management systems,” *Enterprise Distributed Object Computing Conference, 2002. EDOC'02. Proceedings. Sixth International*, 2002, pp. 15-26.

124. H. He, H. Haas, and D. Orchard, “Web services architecture usage scenarios,” *W3C Working Group Note*, 2004.

125. G.J. Ahn, and R. Sandhu, “Role-based authorization constraints specification,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, 2000, pp. 207-226.

126. E. Syukur, S.W. Loke, and P. Stanski, “Methods for Policy Conflict Detection and Resolution in Pervasive Computing Environments,” *Proceedings of Policy Management for Web Workshop, Chiba, Japan*.

127. A. Toninelli, J. Bradshaw, L. Kagal, and R. Montanari, “Rule-based and Ontology-based Policies: Toward a Hybrid Approach to Control Agents in Pervasive Environments,” *Proceedings of the Semantic Web and Policy Workshop*, 2005.

128. F. Baader, I. Horrocks, and U. Sattler, “Description logics as ontology languages for the semantic web,” *Festschrift in honor of Jorg Siekmann, Lecture Notes in Artificial Intelligence. Springer-Verlag*, 2003.

129. S. Barker, and P.J. Stuckey, “Flexible access control policy specification with constraint logic programming,” *ACM Transactions on Information and System Security (TISSEC),* vol. 6, no. 4, 2003, pp. 501-546.

130. P. Saint-Dizier, “A constraint logic programming treatment of syntactic choice in natural language generation,” *Aspects of Automated Natural Language Generation,* vol. 229, pp. 119–134.

131. L. Kagal, T. Finin, and A. Joshi, *A Policy Language for a Pervasive Computing Environment*, Defense Technical Information Center, Department of Computer Science, Maryland University, Baltimore, 2005.

132. J. Bradshaw, A. Uszok, R. Jeffers, N. Suri, P. Hayes, M. Burstein, A. Acquisti, B. Benyo, M. Breedy, and M. Carvalho, *Representation and reasoning for DAML-based policy and domain services in KAoS and nomads*, ACM New York, NY, USA, 2003.

133. K.L. Rei, “A policy language for the Me-Centric project,” *Technical Report HPL-2002*, HP Labs, pp. 1-57.

134. N. Damianou, N. Dulay, E. Lupu, and M. Sloman, “The Ponder Policy Specification Language. In proceedings of Workshop on Policies for Distributed Systems and Networks (POLICY 2001),” Springer-Verlag, LNCS.

135. A. Schaad, and J. Moffett, “Delegation of Obligations,” *3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002),* 2002.

136. J. Keeney, and V. Cahill, “Chisel: A Policy-Driven, Context-Aware, Dynamic Adaptation Framework,” *Proceedings of the Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003),* 2003, pp. 3–14.

137. OASIS, “eXtensible Access Control Markup Language (XACML) Version 2.0,” 2005, Cited Sep 2008; http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

138. G. Stone, “A Path-Based Network Policy Language,” Department of Computer Science, Naval Postgrad School, Monterey, California, United States, 2000.

139. P3P, “Platform for Privacy Preferences Project,” Cited June 2008; http://www.w3.org/P3P.

140. J.A. Hoagland, R. Pandey, and K.N. Levitt, “Security Policy Specification Using a Graphical Approach,” *Arxiv preprint cs/9809124*, 1998.

141. Y. Kanada, “Taxonomy and Description of Policy Combination Methods,” *LECTURE NOTES IN COMPUTER SCIENCE,* 2001, pp. 171-184.

142. L.Z. Granville, G. Coelho, M. Almeida, and L. Tarouco, “PoP-An Automated Policy Replacement Architecture for PBNM,” *Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)*, 2002, pp. 140.

143. H. Kreger, “Web Services Conceptual Architecture (WSCA 1.0),” *IBM Software Group,* vol. 5, 2001, pp. 6-7.

144. L.M. Correia, and European Cooperation in the Field of Scientific and Technical Research (Organization). COST 273., *Mobile broadband multimedia networks : techniques, models and tools for 4G,* Elsevier/Academic Press, 2006.

145. H. Kamoda, M. Yamaoka, S. Matsuda, K. Broda, and M. Sloman, “Access Control Policy Analysis Using Free Variable Tableaux,” *IPSJ Digital Courier,* vol. 2, no. 0, 2006, pp. 207-221.

146. H. Kamoda, A. Hayakawa, M. Yamaoka, S. Matsuda, K. Broda, and M. Sloman, "Policy Conflict Analysis Using Tableaux for On Demand VPN Framework," *Proceedings of the the First International Workshop on Trust, Security and Privacy for Ubiquitous Computing (TSPUC 2005), Taormina, Sicily, Italy, June*, 2005.

147. S. Jajodia, P. Samarati, and V.S. Subrahmanian, "A Logical Language for Expressing Authorizations," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 31-42.

148. R. Bhatti, J.B.D. Joshi, E. Bertino, and A. Ghafoor, "Access Control in Dynamic XML-based Web-Services with X-RBAC," *proceedings of The First International Conference on Web Services, Las Vegas, June*, 2003, pp. 23-26.

149. A. Anderson, "Core and hierarchical role based access control (RBAC) profile of XACML v2. 0," *OASIS Standard (Feb 2005).*

150. N. Dunlop, J. Indulska, and K. Raymond, "Methods for conflict resolution in policy-based management systems," *IEEE 7th International Enterprise Distributed Object Computing Conference (EDOC'03), September*, 2003, pp. 16-19.

151. J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "RFC 1905: Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)," *Network Working Group*, 1996.

152. S. Herzog, J. Boyle, R. Cohen, D. Durham, R. Rajan, and A. Sastry, "RFC 2749: COPS usage for RSVP, Standards Track RFC," *Internet Engineering Task Force*, 2000.

153. K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, and A. Smith, "RFC 3084: COPS Usage for Policy Provisioning (COPS-PR)," *Internet Engineering Task Force*, 2001.

154. D. Rawlins, A. Kulkarni, K.H. Chan, and D. Dutt, "Framework of COPS-PR Policy Information Base for Accounting Usage," *draft-ietf-rap-acct-fr-pib-00. txt (work in progress)*, 2000.

155. J. Ottensmeyer, M. Bokaemper, and K. Roeber, "A Filtering Policy Information Base (PIB) for edge router filtering services and provisioning via COPS-PR," IETF, Internet draft, draft-otty-cops-prfilter-pib-00. txt (work in progress), 2000.

156. M. Li, D. Arneson, A. Doria, J. Jason, C. Wang, and M. Stenberg, "IPsec Policy Information Base," *IETF Internet draft, draft-ietf-ipsp-ipsecpib-02.txt (work in progress).*

157. H. Hedge, and B. Stone, "Load balancing policy information base," *IETF, Internet draft, draft-hedge-load-balancing-pib-00.txt (work in progress)*, 2001.

158. S. Salsano, "COPS usage for Diffserv Resource Allocation (COPSDRA)," *draft-salsano-cops-dra-00. txt>, September*, 2001.

159. R. Fenger, H. Hegde, D. Larson, and R. Sahita, "Simplifying Support of New Network Services Using COPS-PR," *Intel Corporation white paper*, 2002.

160. K. McCloghrie, M. Fine, J. Seligson, K. Chan, S. Hahn, R. Sahita, A. Smith, and F. Reichmeyer, "RFC 3159: Structure of Policy Provisioning Information," *Network Working Group*, 2001.

161. R. Mameli, and S. Salsano, "Use of COPS for Intserv operations over Diffserv: Architectural issues, Protocol design and Test-bed implementation," *IEEE International Conference on Communications, ICC 2001,* vol. 10, 2001.

162. M. Li, N.R. Center, and M.A. Burlington, "Policy-based IPsec management," *Network, IEEE,* vol. 17, no. 6, 2003, pp. 36-43.

163. C. Király, Z. Pándi, and T. Van Do, "Analysis of SIP, RSVP, and COPS interoperability," *LECTURE NOTES IN COMPUTER SCIENCE*, 2003, pp. 717-728.

164. A. Tsalgatidou, and T. Pilioura, "An overview of standards and related technology in Web services," *Distributed and Parallel Databases*, vol. 12, no. 2, 2002, pp. 135-162.

165. D. Verma, M. Beigi, and R. Jennings, "Policy Based SLA Management in Enterprise Networks," *LECTURE NOTES IN COMPUTER SCIENCE*, 2001, pp. 137-152.

166. M. Brunner, and R. Stadler, "The Impact of Active Networking Technology on Service Management in a Telecom Environment," *IFIP/IEEE International Symposium on Integrated Network Management (IM '99), Boston, MA, May*, 1999, pp. 10-14.

167. M. Gudgin, M. Hadley, N. Mendelsohn, J.J. Moreau, and H.F. Nielsen, "SOAP Version 1.2," *W3C Working Draft*, vol. 9, 2001.

168. L. Clement, A. Hately, C. von Riegen, and T. Rogers, "UDDI Version 3.0. 2," *UDDI Spec Technical Committee Draft*, vol. 20041019, 2004.

169. E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," *W3C Working Draft*, 2001.

170. S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, and A. Malhotra, "Web Services Policy Framework (WS-Policy)," *Version*, vol. 1, no. 2, 2006, pp. 2006-2003.

171. A.H. Anderson, "An Introduction to the Web Services Policy Language (WSPL)," *POLICY*, vol. 4, 2004, pp. 189–192.

172. E. Newcomer, *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Addison-Wesley, 2002.

173. S. Hada, and M. Kudo, "XML access control language (XACL): Provisional authorization for XML documents," *Standard, OASIS*, 2001.

174. T. Yu, N. Li, and A.I. Antón, "A formal semantics for P3P," *Proceedings of the 2004 workshop on Secure web service*, 2004, pp. 1-8.

175. WS-Policy, "Web Services Policy Framework (WS-Policy)," http://www-106.ibm.com/developerworks/library/specification/ws-polfram/.

176. B. Parsia, V. Kolovski, and J. Hendler, "Expressing WS Policies in OWL," *14th International World Wide Web Conference, Chiba, Japan*, 2005.

177. WSPL, "Web Services Policy Assertions Language," Cited Aug 2008; http://www-106.ibm.com/developerworks/library/ws-polas.

178. A.H. Anderson, "An Introduction to the Web Services Policy Language (WSPL)," *POLICY*, vol. 4, 2004, pp. 189-192.

179. E. Toktar, E. Jamhour, and C. Maziero, "A XML policy-based approach for RSVP," *LECTURE NOTES IN COMPUTER SCIENCE*, 2004, pp. 1204-1209.

180. SLS, "Service Level Specification, The QoS Forum," Cited Aug 2008; http://www.qosforum.com/.

181. T.E. Squair, E. Jamhour, and R.C. Nabhen, "An RBAC-based Policy Information Base," *Sixth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, IEEE.

182. S. Vegesna, *IP quality of service, Modular Quality of Service Command-Line Interface (Appendix A)*, Cisco Press, 2001.

183. D. Evans, and A. Twyman, "Flexible policy-directed code safety," *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, 1999, pp. 32-45.

184. J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin, "RFC1157: Simple network management protocol (SNMP)," *DDN Network Information Center, SRI International*, 1990.

185. K. McCloghrie, and M.T. Rose, "RFC1213: Management Information Base for Network Management of TCP/IP-based Internets: Mib-II," *RFC Editor United States*, 1991.

186. L. Sanches, K. McCloghrie, and J. Saperia, "Evaluation of COPS/PIB and SNMP/MIB approaches for configuration management of IP-based networks," *internet draft, draft-ops-mumble-confmanagement-00. txt (work in progress)*, 1999.

187. W. Stallings, *SNMP, SNMPv2, and CMIP: the practical guide to network management*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1993.

188. H. Kvarnström, "A survey of commercial tools for intrusion detection," *Technical Report TR99-8, Chalmers University of Technology*, pp. 1-47.

189. H. Hazewinkel, and D. Partain, "RFC 3747: The Differentiated Services Configuration MIB," *Network Working Group*, 2004.

190. DiffServ, "Differentiated Services (diffserv)," Cited Aug 2008; http://www.ietf.org/html.charters/OLD/diffserv-charter.html.

191. T. Howes, M.C. Smith, and S. Gordon, *Understanding and deploying LDAP directory services*, Addison-Wesley Boston, 2003.

192. R. Chaudhury, E. Ellesson, S. Kamat, J.C. Martin, G. Powers, R. Rajan, D. Verma, and R. Yavatkar, "Directory Schema for Service Level Administration of Differentiated Services and Integrated Services in Networks" draft submitted to Directory Enabled Networks Ad Hoc Working Group," *Jun*, vol. 28, 1998, pp. 1-17.

193. L. Bartz, "LDAP schema for role based access control," *Internet Draft, Internet Engineering Task Force, (work in progress)*, Oct. 1997

194. T. Delot, P. Déchamboux, B. Finance, Y. Lepetit, and G. LeBrun, "LDAP, Databases and Distributed Objects: Towards a Better Integration," *Proceedings of Databases in Telecommunications II: VLDB 2001 International Workshop, 2001*.

195. G. Good, "RFC2849: The LDAP Data Interchange Format (LDIF)-Technical Specification," *Network Working Group*, 2000.

196. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475: An Architecture for Differentiated Services," *Network Working Group*, 1998.

197. R. Braden, D. Clark, and S. Shenker, "RFC 1633: Integrated Services in the Internet Architecture: an Overview," *Internet Engineering Task Force*, 1994.

198. R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "RFC 2205: Resource ReSerVation Protocol (RSVP)," *Internet Engineering Task Force*, 1997.

199. J. Schönwälder, "Overview of the 2002 IAB Network Management Workshop," *Book Overview of the 2002 IAB Network Management Workshop*, Series Overview of the 2002 IAB Network Management Workshop, ed., Editor ed.^eds., RFC 3535, Internet Engineering Task Force May 2003, pp.

200. D.C. Verma *Policy-based Networking - Architecture and Algorithms*, New Riders, 2000, pp. 183-217.

201. RBAC, "RBAC in the Solaris Operating Environment," Cited April 2008; http://www.sun.com/software/whitepapers/wp-rbac/wp-rbac.pdf.

202. M. Kim, and P. Compton, “Formal concept analysis for domain-specific document retrieval systems,” *LECTURE NOTES IN COMPUTER SCIENCE*, 2001, pp. 237-248.

203. T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, and F. Yergeau, “Extensible Markup Language (XML) 1.0 (Fifth Edition),” *http://www.w3.org/TR/REC-xml/#dt-doctype, W3C Recommendation,* Cited Sep 2008, 2008.

204. B. Aboba, and M. Beadles, “RFC 2486: The Network Access Identifier,” *Network Working Group*, 1999.

205. V. Jacobson, K. Nichols, and K. Poduri, “RFC 2598: An expedited forwarding PHB,” *Network Working Group,* 1999.

206. T. Ahmed, A. Mehaoua, and R. Boutaba, “Dynamic QoS adaptation using COPS and network monitoring feedback,” *LECTURE NOTES IN COMPUTER SCIENCE,* 2002, pp. 250-262.

207. I. Fodil, and G. Pujolle, “Roaming and Service Management in Public Wireless Networks Using an Innovative Policy Management Architecture,” *Int. J. Network Mgmt,* vol. 15, 2005, pp. 103-121.

208. ISO/IEC, “Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 4 Management framework,” *7498-4, International Standard IS0,* 1984.

209. A. Le Hors, P. Le Hegaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne, “Document object model (DOM) level 2 Core Specification,” *W3C Recommendation,* 2004.

210. D. Megginson, “Sax 2.0: The Simple Api for XML,” Cited April 2008, 2000; http://www.megginson.com/SAX/index.html.

211. JAXP, “Java API for XML Processing, JAXP 1.4 Reference Implementation,” Cited Aug 2008, May 2007; https://jaxp.dev.java.net/1.4/.

212. JDOM, “Simplify XML Programming with JDOM,” Cited Sep 2008, May 2001; http://www.ibm.com/developerworks/java/library/j-jdom/.

213. S. Jha, and M. Hassan, “Implementing Bandwidth Broker Using COPS-PR in Java,” *CONFERENCE ON LOCAL COMPUTER NETWORKS,* vol. 26, 2001, pp. 178-181.

214. P. Vidales, “Seamless Mobility in 4G systems,” Computer Laboratory, University of Cambridge, Cambridge, 2005.

215. M. Stemm, and R.H. Katz, “Vertical handoffs in wireless overlay networks,” *Mobile Networks and Applications,* vol. 3, no. 4, 1998, pp. 335-350.

216. 3GPP, “Technical Specification Group Services and System Aspects,” *3rd Generation Partnership Project, http://www.3gpp.org/TB/home.htm,* 2003, Cited Sep 2008.

217. M. Buddhikot, G. Chandranmenon, S.J. Han, Y.W. Lee, S. Miller, and L. Salgarelli, “Integration of 802.11 and Third-Generation Wireless Data Networks,” *IEEE INFOCOM,* vol. 1, 2003, pp. 503-512.

218. H. Zimmermann, “OSI reference model--The ISO model of architecture for open systems interconnection,” *Communications, IEEE Transactions on [legacy, pre-1988],* vol. 28, no. 4, 1980, pp. 425-432.

219. C. Perkins, “RFC 2002: IP Mobility Support,” *Network Working Group,* 1996.

220. C. Perkins, “RFC 3344: IP Mobility Support for IPv4,” *Network Working Group,* 2002.

221. D. Johnson, C. Perkins, and J. Arkko, “RFC 3775: Mobility support in IPv6,” *Internet Engineering Task Force,* 2004.

222. C. Perkins, “RFC 2003: Encapsulation within IP,” *Network Working Group,* 1996.

223. C. Perkins, “RFC 2004: Minimal Encapsulation within IP,” *Network Working Group,* 1996.

224. S. Hanks, T. Li, D. Farinacci, and P. Traina, “RFC 1701: Generic Routing Encapsulation (GRE),” *Network Working Group,* 1994.

225. J. Solomon, “RFC 2005: Applicability Statement for IP Mobility Support,” *Network Working Group,* 1996.

226. D. Cong, M. Hamlen, and C. Perkins, “RFC 2006: The Definitions of Managed Objects for IP Mobility Support using SMIv2,” *Network Working Group,* 1996.

227. P. Srisuresh, “RFC 2709:Security Model with Tunnel-mode IPsec for NAT Domains,” *Network Working Group,* 1999.

228. S. Wong, “The evolution of wireless security in 802.11 networks: WEP, WPA and 802.11 standards,” *URL: http://www.sans.org/rr/whitepapers/wireless/1109.php Cited Jun 2008,* vol. 28, no. 7, 2003, pp. 1-10.

229. J.D. Moffett, and M.S. Sloman, “Policy Conflict Analysis in Distributed System Management,” *JOURNAL OF ORGANIZATIONAL COMPUTING,* vol. 4, 1994, pp. 1-22.

230. F. Feng, and D.S. Reeves, “Explicit proactive handoff with motion prediction for mobile IP,” *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNCt'04),* pp. 855–860.

231. F. Erbas, J. Steuer, D. Eggesieker, K. Kyamakya, and K. Jobinann, “A regular path recognition method and prediction of user movementsin wireless networks,” *IEEE 54th Vehicular Technology Conference, 2001. VTC 2001 Fall.*

232. G. Liodakis, and P. Stavroulakis, “A novel approach in handover initiation for microcellular systems,” *Vehicular Technology Conference, 1994 IEEE 44th,* 1994, pp. 1820-1823.

233. R. Rezaiifar, A.M. Makowski, and S. Kumar, “Optimal control of handoffs in wireless networks,” *Vehicular Technology Conference, 1995 IEEE 45th,* vol. 2, 1995.

234. K.D. Wong, D.C. Cox, T. Technol, and R. Bank, “A pattern recognition system for handoff algorithms,” *Selected Areas in Communications, IEEE Journal on,* vol. 18, no. 7, 2000, pp. 1301-1312.

235. N.D. Tripathi, J.H. Reed, and H.F. VanLandingham, “Adaptive Handoff Algorithms for Cellular Overlay Systems using Fuzzy Logic,” *IEEE 49th Vehicular Technology Conference.*

236. P.M.L. Chan, R.E. Sheriff, and Y.F. Hu, “An intelligent handover strategy for a multi-segment broadbandnetwork,” *Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on,* vol. 1, 2001.

237. R. Chellapa, A. Jennings, and N. Shenoy, “A comparative study of mobility prediction in fixed wireless networks and mobile ad hoc networks,” *IEEE International Conference on Communications, 2003. ICC'03,* vol. 26, no. 1, 2003, pp. 891-895, 2003.

238. C. Komar, and C. Ersoy, “Location tracking and location based service using IEEE 802.11 WLAN infrastructure,” *European Wireless.*

239. J. Kristiansson, and P. Parnes, “Application-layer mobility support for streaming real-time media,” *IEEE Wireless Communications and Networking Conference, WCNC, 2004*.

240. Q. Zhang, C. Guo, Z. Guo, and W. Zhu, “Efficient mobility management for vertical handoff between WWAN and WLAN,” *IEEE Communications Magazine*, vol. 41, no. 11, 2003, pp. 102-108.

241. M. Ylianttila, M. Pande, J. Makela, and P. Mahonen, “Optimization scheme for mobile users performing vertical handoffs between IEEE 802.11 and GPRS/EDGE networks,” *Proceedings of IEEE Global Telecommunications Conference, 2001. GLOBECOM'01, San Antonio, USA,* vol. 6, Nov. 2001, pp. 3439-3443.

242. M. Ylianttila, J. Makela, and P. Mahonen, “Supporting resource allocation with vertical handoffs in multiple radio network environment,” *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, pp. 64-68.

243. W.T. Chen, and Y.Y. Shu, “Active application oriented vertical handoff in next-generation wireless networks,” *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'05),* pp. 1382-1388, New Orleans, USA, Mar 2005.

244. H.J. Wang, R.H. Katz, and J. Giese, “Policy-enabled handoffs across heterogeneous wireless networks,” *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99),* pp. 51-60, New Orleans, USA, Feb 1999.

245. Y. Min-Hua, L. Yu, and Z. Hui-min, “The mobile IP handoff between hybrid networks,” *IEEE 13th International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC’02), 2002*.

246. S. Aust, D. Proetel, N.A. Fikouras, C. Pampu, and C. Gorg, “Policy based Mobile IP handoff decision (POLIMAND) using generic link layer information,” *5th IEEE International Conference on Mobile and Wireless Communication Networks (MWCN 2003), 2003*.

247. H.J. Wang, R.H. Katz, and J. Giese, “Policy-enabled handoffs across heterogeneous wireless networks,” *Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99),* pp. 51-60, 1999.

248. N.D. Tripathi, J.H. Reed, and H.F. VanLandinoham, “Handoff in cellular systems,” *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 5, no. 6, 1998, pp. 26-37.

249. K. Pahlavan, P. Krishnamurthy, A. Hatami, M. Ylianttila, J.P. Makela, R. Pichna, and J. Vallstron, “Handoff in hybrid mobile data networks,” *Personal Communications, IEEE,* vol. 7, no. 2, 2000, pp. 34-47.

250. D. Harkins, and D. Carrel, “RFC2409: The Internet key exchange (IKE),” *RFC Editor United States*, 1998.

251. Racoon, “IPsec-Tools,” *IPsec utilities, http://ipsec-tools.sourceforge.net/*, 2008, Cited Aug 2008.

252. IANA, “Assigned Internet Protocol Numbers,” Internet Assigned Numbers Authority, Cited Aug 2008; http://www.iana.org/assignments/protocol-numbers/.

253. J. Prosise, *Programming Windows with MFC (with CD-ROM),* Microsoft Press Redmond, WA, USA, 1999.

254. E. Lear, Y. Rekhter, B. Moskowitz, D. Karrenberg, and G. de Groot, “RFC 1918: Address Allocation for Private Internets,” *RFC Editor United States, BCP 5*, 1996.

255. R. Hinden, and B. Haberman, “RFC 4193: Unique Local IPv6 Unicast Addresses,” *Network Working Group of the Internet Engineering Task Force*, 2005.

256. G. Tsirtsis, and P. Srisuresh, “RFC 2766: Network address translation-protocol translation (NAT-PT),” *Network Working Group*, 2000.

257. H. Levkowetz, and S. Vaarala, “RFC 3519: Mobile IP traversal of network address translation (NAT) devices,” *Network Working Group*, 2003.

Appendix A

Abbreviations

3G	Third Generation wireless communication systems
3GPP	3rd. Generation Partnership Project
4G	Fourth Generation wireless communication systems

A

AAA	Authentication, Authorisation and Accounting
ABC	Always Best Connected
AP	Access Point
AR	Access Router

B

Bluetooth	A standard for short-range wireless communication between computing devices and associated peripherals, including laptop and mobile computers, personal digital assistance, and mobile phones

C

CDMA	Code Division Multiple Access
CIM	Common Information Model
CLI	Command Line Interface
CMIP	Common Management Information Protocol
CoA	Care-of Address
COPS	Common Open Policy Service protocol
CORBA	Common Object Request Broker Architecture
CoS	Class of Service

D

DAML	DARPA Agent Markup Language
DiffServ	Differentiated Services
DL	Description Logic
DMTF	Distributed Management Task Force
DSD	Domain Specific Document
DSL	Digital Subscriber Line
DTD	Document Type Definition

E

EDGE	Enhanced Data GSM Environment

F

FTP	File Transfer Protocol

G

GPRS	General Packet Radio Service
GPS	Global Positioning System
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile Communications

H

HMIP	Hierarchical Mobile IP
HTTP	HyperText Transfer Protocol

I

IEEE 802.16a	IEEE 802.16 is working group number 16 of IEEE 802, specialising in point-to-multipoint broadband wireless access (also known as Wi-Max)
IEEE 802.11x (family)	Refers to any type of wireless local area network technology
IEEE 80211b	802.11b (also referred to as WiFi) – an extension to 802.11 that applies to wireless LANs and provides 11Mb/s transmission (with a fall-back to 5.5, 2 and 1 Mb/s) in the 2.4GHz band. It was a 1999 ratification to the original 802.11 standard, allowing wireless functionality comparable to Ethernet
IETF	Internet Engineering Task Force

IP	Internet Protocol
IPSec	IP Security, One of two protocols (with PPTP) used for virtual private networks.
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6

L

LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LDIF	LDAP Data Interchange Format
LP	Logic Programming

M

ME	Mobile Entity
MIB	Management Information Base
MIP	Mobile-IP protocol
MIPv4	Mobile-IP version 4
MIPv6	Mobile-IP version 6

N

NAS	Network Access Server
NAT	Network Address Translator

O

OSI	Open System Interconnection
OWL	Web Ontology Language

P

P3P	Platform for privacy Preference Project
PBMS	Policy Based Management System
PBNM	Policy Based Network Management
PCIM	Policy Core Information Model
PCIMe	Policy Core Information Model Extensions
PDA	Personal Digital Assistance
PDL	Policy Definition Language
PDP	Policy Decision Point
PEP	Policy Enforcement Point

PIB	Policy Information Base
PPP	Point-to-Point Protocol
PPTP	Point-to-Point Tunnelling Protocol
PRC	Provisioning Classes
PRI	Provisioning Instances

Q

QoS	Quality of Service

R

RADIUS	AAA server within a GPRS network
RAP	Resource Allocation Protocol Working Group
RBAC	Role Based Access Control
RBAC-PIB	Role Based Access Control – Policy Information Base
RBPIM	Role Based Policy Information Model
RDF	Resource Description Framework
RFC	Request for Comments
RMI	Remote Method Invocation
RSVP	Resource ReSerVation Protocol
RTT	Round Trip Time

S

SAML	Security Assertion Markup Language
SIP	Session Initiation Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management protocol
SOAP	Simple Object Access Protocol
SOD	Separation of Duty
SSD	Static Separation of Duty

T

TCP	Transmission Control Protocol
TCPDUMP	Networking tool to collect TCP-traces
TCPTRACE	Networking tool to obtain graphs from TCP-traces
TINA-C	Telecommunications Information Networking Architecture Consortium

TMN	Telecommunications Management Network

U

UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System

V

VoD	Video on Demand
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network

W

WiFi	WiFi is short for wireless fidelity and is another name for IEEE 802.11b. It is a trade term used by Wireless Ethernet Compatibility Alliance (WECA).
Wi-Max	Wi-Max is an acronym that stands for Worldwide Interoperability for Micorwave Access.
WLAN	Wireless LAN
WSDL	Web Services Description Language
WSPL	Web Services Policy Languages

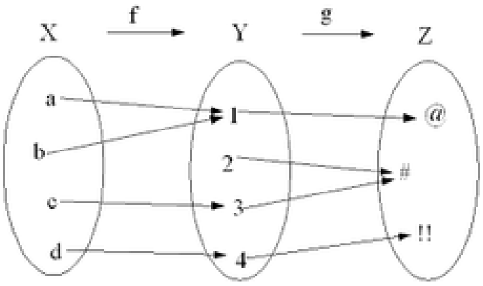
X

XACL	XML Access Control Language
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language

Appendix B

Composite Mapping

In mathematics, a composite function, formed by the composition of one function on another, represents the application of the former to the result of the application of the latter to the argument of the composite. The functions $f: X \to Y$ and $g: Y \to Z$ can be *composed* by first applying *f* to an argument *x* and then applying *g* to the result. Thus one obtains a function *g* o *f*: $X \to Z$ defined by (*g* o *f*)(*x*) = *g*(*f*(*x*)) for all *x* in *X*. The notation *g* o *f* is read as "*g* circle *f*", or "*g* composed with *f*", "*g* following *f*", or just "*g* of *f*". *g* o *f*, the composition of *f* and *g*. For example, (*g* o *f*)(c) = #.



The composition of functions is always associative. That is, if *f*, *g*, and *h* are three functions with suitably chosen domains and *codomains*, then *f* o (*g* o *h*) = (*f* o *g*) o *h*. Since there is no distinction between the choices of placement of parentheses, they may be safely left off.

The functions *g* and *f* are said to commute with each other if *g* o *f* = *f* o *g*. In general, composition of functions will not be commutative. Commutativity is a special property, attained only by particular functions, and often in special circumstances. For example, $|x| + 3 = |x + 3|$ only when $x \geq 0$. But inverse functions always commute to produce the identity mapping.

Considering functions as special cases of relations (namely functional relations), one can analogously define composition of relations, which gives the formula for $g \circ f \subset X \times Z$ in terms of $f \subset X \times Y$ and $g \subset Y \times Z$.

Derivatives of compositions involving differentiable functions can be found using the chain rule. Higher Derivatives of such functions are given by Faa di Bruno's formula.

Example

As an example, suppose that an airplane's elevation at time *t* is given by the function *h*(*t*) and that the oxygen concentration at elevation *x* is given by the function *c*(*x*). Then (*c* o *h*)(*t*) describes the oxygen concentration around the plane at time *t*.

Functional Powers

If $Y \subseteq X$ then $f : X \to Y$ may compose with itself; this is sometimes denoted by f^2. Thus:

$(f \circ f)(x) = f(f(x)) = f^2(x)$

$(f \circ f \circ f)(x) = f(f(f(x))) = f^3(x)$

Repeated composition of a function with itself is called *function iteration*.

The *functional powers* $f \circ f^n = f^n \circ f = f^{n+1}$ for natural n follow immediately.

By convention, $f^0 = id_{D(f)}$ (the identity map on the domain of f).

If $f : X \to X$ admits an inverse function, negative functional powers $f^{-k}(k > 0)$ are defined as the opposite power of the inverse function, $(f^{-1})^k$.

Note: If *f* takes its values in a ring (in particular for real or complex-valued *f*), there is a risk of confusion, as f^n could also stand for the *n*-fold product of *f*, e.g. $f^2(x) = f(x) \cdot f(x)$.

(For trigonometric functions, usually the latter is meant, at least for positive exponents. For example, in trigonometry, this superscript notation represents standard exponentiation when used with trigonometric functions: $\sin^2(x) = \sin(x) \cdot \sin(x)$. However, for negative exponents (especially −1), it nevertheless usually refers to the inverse function, e.g., $\tan^{-1} = \arctan$ (≠ 1/tan).

In some cases, an expression for *f* in $g(x) = f^r(x)$ can be derived from the rule for *g* given non-integer values of *r*. This is called fractional iteration. A simple example would be that where *f* is the successor function, $f^r(x) = x + r$.

Iterated functions occur naturally in the study of fractals and dynamical systems.

Alternative Notation

In the mid-20th century, some mathematicians decided that writing "*g* o *f*" to mean "first apply *f*, then apply *g*" was too confusing and decided to change notations. They wrote "*xf*" for "*f*(*x*)" and "*xfg*" for "*g*(*f*(*x*))". This can be more natural and seem simpler than writing functions on the left in some areas, and is called postfix notation.

For instance, in linear algebra, where *x* is a row vector and *f* and *g* denote matrices and the composition is by matrix multiplication. The order is important because this multiplication is non-commutative. Successive transformations applying and composing to the right agrees with the left-to-right reading sequence.

Category Theory uses *f;g* interchangeably with $g \circ f$. To distinguish the left composition operator from a text semicolon, in the Z notation a fat semicolon is used for left relation composition. Since all functions are binary relations, it is correct to use the fat semicolon for function composition as well.

Composition Operator

Given a function *g*, the *composition operator* C_g is defined as that operator which maps functions to functions as $C_g f = f \circ g.$