Policy-based Self-Management of Wireless Ad Hoc Networks

Antonis M. Hadjiantonis

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Communication Systems Research
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey GU2 7XH, UK

June 2008

Summary

Wireless ad hoc networks pose major research challenges because of their increasing ubiquity and user-initiated formation. *The motivation* of this thesis emanates from the need for unrestricted wireless communication in a scalable and predictable manner. This need is accentuated by the increasing users' demand for spontaneous communication and the deficiencies of existing management frameworks. *The objective* is to propose a management framework able to leverage the potential of wireless ad hoc networks as an alternative communication method allowing them to coexist with other networks and to emerge as their flexible extension. In the context of this thesis, w*ireless ad hoc networks* consist of a majority of end-user devices, capable of multihop communication, and optionally supported from limited infrastructure. The policy-based management (PBM) paradigm is employed to facilitate their self-management, combining design and theory with testbed implementation and simulation studies.

The thesis contribution can be identified in three areas: (1) *Design of a context-aware policy hierarchy and a hybrid role-based organisation model*: The integration of policies with contextual feedback enables the creation of a closed control loop at different organisational levels, forming the basis for self-management. (2) *Deployment of distributed PBM functionality*: The management of wireless ad hoc networks is possible with the decentralisation of traditional PBM concepts, based on the design and implementation of a Distributed Policy Repository (DPR). The DPR assists in the coordination of dispersed policy decision points (PDPs) by facilitating the synchronisation of policies and offering a uniform view of management objectives to the PDPs. (3) V*alidation of PBM functionality for self-management*: A case study addresses the deployment of a wireless ad hoc network on a testbed, attempting to overcome the lack of central coordination and the occurrence of interference, by using policies to control its dynamic channel assignment. Finally, the framework is extended for service management, implementing adaptable service provisioning and offering service customisation to end-users. The elements of this thesis contribution are combined under a unified policy-based framework for the self-management of wireless ad hoc networks.

Key words: wireless ad hoc networks, policy-based management, self-management, management framework, autonomic computing, distributed policies.

Dedicated to my family

Αφιερώνεται στην οικογένειά μου

Acknowledgments

I would like to express my sincere gratitude to Professor George Pavlou for his supervision and guidance throughout the long journey towards this PhD thesis. His knowledge and experience were invaluable for my progression as a researcher, while his support and unreserved belief in my initiatives have motivated my academic work.

My deepest thanks and gratitude go to my wife Athanasia and my parents Michalis and Maria for always being supportive and caring. Without the wholehearted support from all of my family, I would not have been able to embark on a doctoral degree. Their love and companion have always encouraged every step towards the completion of my PhD and I am forever grateful.

My appreciation goes to my colleagues at CCSR who through their input and cooperation have made the time at University of Surrey productive and fruitful. In particular, I would like to thank Dr. Paris Flegkas for his advice during my first years at CCSR, as well as all colleagues from Networks Group with whom I have collaborated on joint research efforts. Many thanks go to Aimilios Chourmouziadis, Marinos Charalambides and Oscar Gonzalez-Duque for helping create a friendly and enjoyable environment at Surrey.

Finally I would like to acknowledge the excellent collaboration with colleagues that were involved in the EMANICS European Network of Excellence. I would like to thank them all and especially Professor Olivier Festor for their support and opportunities provided.

Contents

List of Figures

List of Tables

Glossary of Terms

3GPP	Third Generation Partnership Project
ACPL	Autonomic Computing Policy Language
ANMP	Ad hoc Network Management Protocol
AODV	Ad hoc On-Demand Distance Vector
AP	Access Point (wireless)
API	Application Programming Interface
ARM	Advanced RISC Machines
BSS	Basic Service Set
CCP	Context Collection Point
CDP	Context Decision Point
CDR	Conflict Detection and Resolution
CDS	Connect Dominating Set
CH	Cluster Head
CIM	Common Information Model
CLI	Command Line Interface
CMIP	Common Management Information Protocol
CMT	Context Management Tool
CN	Cluster Node
COPS-PR	Common Open Policy Service for Policy Provisioning
CORBA	Common Object Request Broker Architecture
CR	Context Repository
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
DIT	Directory Information Tree
DMTF	Distributed Management Task Force
DN	Distinguished Name
DPR	Distributed Policy Repository
DS	Directory Server
DSA	Directory Server Agent
ECA	Event Condition Action
ESS	Extended Service Set
ETSI	European Telecommunications Standards Institute
FCC	Federal Communications Commission
FSM	Finite State Machine

GPL	General Public License
GUI	Graphical User Interface
IBSS	Independent Basic Service Set
ICO	Information Commissioner's Office
IDL	Interface Definition Language
IEEE	The Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IRTF	Internet Research Task Force
ISM	Industrial Scientific and Medical
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU	International Telecommunication Union
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LPDP	Local Policy Decision Point
MAC	Medium Access Control
MANET	Mobile Ad hoc NETwork
MAPE	Monitor-Analyse-Plan-Execute
MbD	Management by Delegation
MCDS	Minimal Connected Dominating Set
MIB	Management Information Base
MN	Manager Node
MO	Managed Object
MPR	Multi-Point Relay
NO	Network Operator
NP	Nondeterministic-Polynomial
OLSR	Optimized Link State Routing
OSI-SM	Open Systems Interconnection-Systems Management
P2P	Peer to Peer
PBM	Policy-based Management
PBNM	Policy-based Network Management
PCELS	Policy Core Extensions LDAP Schema
PCIM	Policy Core Information Model
PCIMe	Policy Core Information Model Extensions
PCLS	Policy Core LDAP Schema
PCO	Policy Conforming Objects

PDA	Personal Digital Assistant
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PFO	Policy Free Object
PHY	Physical Layer
PMAC	Policy Management for Autonomic Computing
PMT	Policy Management Tool
PO	Policy Object
PR	Policy Repository
QoS	Quality of Service
QPIM	QoS Policy Information Model
RAP	Resource Allocation Protocol
RDN	Relative Distinguished Name
RFC	Request For Comments
RPC	Remote Procedure Call
SDK	Software Development Kit
SEAL	Service Adaptation Logic
SMI	Structure of Management Information
SNMP	Simple Network Management Protocol
SOA	Service-Oriented Architecture
SOAP	Simple Object Access Protocol (deprecated)
SP	Service Provider
SSH	Secure Shell / Secure Socket Shell
SSID	Service Set Identifier
TC	Topology Control
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TMN	Telecommunications Management Network
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
VANET	Vehicular Ad Hoc Networks
W3C	World Wide Web Consortium
WG	Working Group
WLAN	Wireless Local Area Network
WSDL	Web Service Description Language
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

Publications

Journals

[1] A.M. Hadjiantonis, M. Charalambides, G. Pavlou, "An Adaptive Service Management Framework for Wireless Networks", *IEEE Vehicular Technology Magazine*, Special Issue on Policy-based Management for Wireless Multimedia Services, Vol.2,Iss.3, pp.6-13, Sep.2007.

[2] A. Malatras, A.M. Hadjiantonis, G. Pavlou, "Exploiting Context-awareness for the Autonomic Management of Mobile Ad Hoc Networks", *Springer Journal of Network and System Management* (JNSM), Special Issue on Autonomic Pervasive and Context-aware Systems, Vol. 15, No.1, pp.29-55, Mar.2007.

Conferences

[3] A.M. Hadjiantonis, G. Pavlou, "Policy-based Self-Management of Hybrid Ad hoc Networks for Dynamic Channel Configuration", *Proc. 11th IEEE/IFIP Network Operations and Management Symposium* (NOMS), Salvador, Brazil, pp.433-440, Apr.2008.

[4] A.M. Hadjiantonis, M. Charalambides, G. Pavlou , "A Policy-based Approach for Managing Ubiquitous Networks in Urban Spaces", *Proc. IEEE International Conference on Communications* (ICC), Glasgow, UK, pp.2089-2096, Jun.2007.

[5] A.M. Hadjiantonis, A. Malatras, G. Pavlou, "A Context-aware Policy-based Framework for the Management of MANETs", Proc. 7th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY), Ontario, Canada, Jun.2006.

[6] F. Liu, A.M. Hadjiantonis, H.M. Tran, M. Amin, "An Architecture for Supporting Network Fault Recovery Management", *Proc. 2nd Int. Conf. on Autonomous Infrastructure, Management and Security* (AIMS), Bremen, Germany, Jul.2008, to be published

Book Chapters

[7] A.M. Hadjiantonis, G. Pavlou, "Self-management of wireless ad hoc networks using the policy-based paradigm and context-awareness", in *Context-Aware Computing and Self-Managing Systems*, W. Dargie, Ed, CRC Studies in Informatics Series, to be published.

Chapter 1

Introduction

Wireless networks have become a ubiquitous reality and evermore surround our everyday activities. They form and disappear spontaneously around us and have become new means for productivity and social interaction. Access to corporate networks, e-mail or simply entertainment are new necessities posed on an increasingly networked wireless world. In the era of mobility and connectivity, a multitude of wireless devices interact with us in our everyday life. Wireless digital assistants such as mobile phones, laptops or personal organisers must be able to cope and offer the desired services at any place and at anytime. An increasingly *ad hoc* element facilitates the need for on demand connectivity and wireless communication. At the same time, increased complexity and heterogeneity have become barriers to wider adoption and ease of use. *Wireless ad hoc networks* have the potential to enable truly ubiquitous computing and pervasive networking. However, their diverse characteristics and special requirements pose the need for novel management paradigms.

Self-management is receiving intense interest from academia and industry, aiming to simplify and automate network management operations. Self-management capabilities aim to vanish inside devices, relieving both managers and users from tedious configuration and troubleshooting procedures. Ideally, self-managed devices integrate self-configuration, self-optimisation, self-protection and self-healing capabilities. When combined, these capabilities can lead to adaptive and ultimately self-maintained autonomic systems. In reality though, the deployment of self-managed networks is withheld from several obstacles that need to be overcome in order to realise such a vision. The use of *policies* for network and systems management is viewed as a promising paradigm to facilitate self-management. Policies can capture the high-level management objectives and can be automatically enforced to devices, simplifying and automating compound and time-consuming management tasks.

1.1 Research Motivation

The motivation for the research efforts of this thesis emanates from the need to facilitate unrestricted wireless ad hoc communication in a scalable and predictable manner. The increased penetration of wireless technologies and devices, combined with the user-oriented formation of wireless networks, create this need and motivate these research efforts towards an appropriate management framework. ***The objective*** of this thesis is to propose a new management framework, able to leverage the potential of *wireless ad hoc networks* as an alternative communication paradigm, allowing them to coexist with other networks and to emerge as their flexible wireless extension. For the purpose of this thesis, w*ireless ad hoc networks* consist of a majority of end-user devices, capable of ad hoc multihop communication, and optionally supported from limited infrastructure.

Nowadays, services targeting home and business users, such as Internet access, digital television or online entertainment, are taken for granted. However, modern lifestyle creates the need for extending the reach of such services but also creates the need for new ones, targeted to people on the move. The convergence of fixed and wireless technologies is inevitable and a rapidly evolving market brings new challenges. In recent years, we have experienced an unprecedented penetration of mobile phones, while the mobile industry growth and evolution continues steadily. Developed countries are planning their transition to fully converged networks and services, while wireless access capability (Wi-Fi) is becoming a common feature of mobile phones. At the same time, newly industrialised countries and emerging markets are just discovering wireless technologies and offer an impressive drive for low cost infrastructure development. Their massive potential customer base is contrasted to low population density, making the cost of wired technologies prohibitive and deeming current management paradigms as inapplicable. At the same time, new wireless networking paradigms are investigated, offering a promising and challenging ground for research and innovation.

To fully appreciate *wireless ad hoc networks*, we need to understand how they fit into the big picture of the next generation wireless landscape. The legacy of Mobile Ad Hoc Networks (MANET) has restricted the popularity of future ad hoc networks. As a result, the hidden potential of multihop ad hoc communications is often underestimated. Admittedly, in spite of extensive research efforts in MANETs, their market penetration has been negligible. Their industrial exploitation and adoption remains limited to specialised military or emergency response scenarios. Researchers note that the major reason for the negligible market impact of the generic MANET paradigm is the lack of realism in the research approach. On the contrary, field measurements of urban Wi-Fi® network deployments have identified that 10% of connections worldwide are in ad hoc mode (IBSS), while the rest are conventional connections to

infrastructure wireless access points (BSS/ESS) [details in §2.3.5,pp.33]. Moreover, in cases of wireless access at large IT events, the percentage of ad hoc connections is further increased. These surprisingly high percentages verify the popularity of spontaneous ad hoc communication of wireless devices in a peer to peer manner. This popularity is attributed to the convenience and self-directed deployment offered by *ad hoc* mode. In addition, new technologies for multihop wireless communication are under development, with *mesh networks* being the most mature one. Opportunistic networking and vehicular ad hoc networks (VANET) are also emerging as novel manifestations of the *wireless ad hoc networking* paradigm. These facts reveal the prospects of *wireless ad hoc networks* as a paradigm and as a technology, able to coexist with other networks and ideally can emerge as their flexible and self-managed wireless extension.

The apparent potential of *wireless ad hoc networks* has motivated a realistic research approach towards their management. Efforts were centred on pragmatic assumptions and coupled design with implementation and deployment on a wireless testbed. By considering the deployment of *wireless ad hoc networks* in real life scenarios, research efforts can disengage from MANETs isolation, opening new possibilities for innovation. Having in mind the vast numbers of user-owned wireless devices, a management framework for emerging wireless ad hoc networks would facilitate the deployment of new services and would encourage the wider use of spontaneous communication in ad hoc manner. Unfortunately, large-scale deployment and management of such networks is a daunting task, hindered by the intermittence of wireless links, the resource constraints of participating devices and their highly distributed nature. The capacity and throughput are limited and severely degrade as the user population and number of hops grow. Intermittence and interference amplify the problem, since enabling wireless technologies need to share the same spectrum. These factors deem conventional management frameworks for fixed networks inapplicable. Even the frameworks for today's wireless networks are unsuitable, because of their centralised organisation and strict management objectives. A novel management framework is required, taking into account the diverse conditions and requirements of *wireless ad hoc networks*. Specialised solutions are needed that can autonomously adapt to changing network conditions in a fast and reliable manner.

Improved network organisation can increase scalability and decentralise management responsibilities, but one has to consider that the majority of wireless networked devices are not under the strict control of a network operator as in traditional infrastructure-based networks. Therefore, a critical management requirement is to respect the owner relationship between end-users and managed devices. Individual users are reluctant to entrust the command of their devices to an operator and demand more control. The lack of a single administrative authority complicates management tasks, but at the same time motivates research on collaborative management schemes. Open standards and contractual agreements can facilitate the interests of different

managing entities, e.g. network operators or service providers. The goal is to provide an adaptive framework for network and service management, where users' privacy and preferences are respected, while multiple managing entities can offer services tailored to the users' needs.

The *policy-based management (PBM) paradigm* can provide the means to integrate self-management capabilities and *policies* can capture the high-level management objectives to be autonomously enforced to devices. Although the PBM paradigm has been traditionally employed in large-scale IP networks, its *controlled programmability* can significantly benefit the highly dynamic environment of wireless networks. PBM can offer a balanced solution between the strict hard-wired management logic of current management frameworks and the unrestricted migration of mobile code offered from *mobile agents*. This has motivated the adoption of PBM for the self-management of wireless ad hoc networks, aiming to simplify and automate compound time-consuming management tasks. The centralised orientation of policy-based operations requires significant research efforts to accommodate the needs of wireless ad hoc networks. In addition, in a rapidly evolving multi-player environment, policies can express the interests of different players and facilitate their cooperation. PBM can be a future-proof solution and can provide the flexibility to adapt to change. At the same time, the users' requirements for control and privacy can be encapsulated in policies and with minimum intervention their devices can operate autonomously.

1.2 Thesis Contribution

The contribution of this thesis focuses on the proposal of a new management framework for wireless ad hoc networks based on distributed policy operations and integrated self-management capabilities. The proposed framework is a composition of concepts and implementation efforts, aiming to contribute towards leveraging the potential of wireless ad hoc networks as an emerging communication method. This subsection outlines the different aspects of this thesis contribution and introduces the partial elements that constitute the overall framework:

a. Policy-based organisational model

The proposed organisational model adopts a hybrid architecture by combining the benefits of both hierarchical and distributed models for policy-based management (PBM). By integrating organisational roles and policies, the tiered model distributes management responsibilities. The role-based node classification adopts a distributed algorithm to select the most capable nodes to participate in PBM operations. This dynamic distribution of management tasks increases scalability and robustness. The flexibility of the proposed model allows a customisable degree of distribution which has enabled its adoption in various scenarios. In addition, the integrated multi-manager capability can facilitate the interests of different managing entities and can enhance the potential of wireless ad hoc networks through collaborative management.

b. Policy hierarchy and enforcement scope for layered self- management

The organisational roles were mapped to a custom policy hierarchy allowing the majority of devices to participate in a PBM network and contribute towards its management. The concept of *policy enforcement scope* was integrated with the role-based policy hierarchy, to form three control layers with a closed loop. This was made possible by using context-aware components that sense the environment and provide feedback to policies at different hierarchy levels. At the top hierarchy level, an automated conflict detection and resolution mechanism was integrated to ensure conflict-free operation of multiple managing entities.

c. Policy design and implementation methodology

A step-by-step methodology was introduced to guide the design and implementation of policies, from requirements' gathering to the deployment of policy instances in a policy repository. The essential benefit of using the proposed methodology is the ability to create lightweight technology-independent policy specifications that can interoperate with full-fledged PBM systems. This work fills the gap between existing policy-based fixed networks with adequate power and emerging wireless ad hoc networks based on portable wireless devices.

d. Distributed Policy Repository design and implementation

The designed and implemented *Distributed Policy Repository (DPR)* is a physically distributed set of components, consisted of interconnected repository replicas hosted on selected capable nodes. The DPR is deployed and maintained using special *DPR management policies* and is based on LDAP (Lightweight Directory Access Protocol). The policy repository encapsulates the management logic of the network therefore it is one of the most critical elements for every policy-based system. To avoid a fatal failure point, the centralised PR philosophy is adapted through the DPR for the management of wireless ad hoc networks, as detailed in §5.3. The proposed DPR glues together the distributed nodes that are responsible for collaborative management. In addition, it offers a logically uniform view of management objectives through policies, it distributes traffic load, and it provides alternative access options for policy access. The DPR also implements the ability to deploy and maintain special purpose *partial* replicas, offering a partial view of network policies that can relate to a specific service or location. This feature can be employed when there is a need for localised control or bottlenecks to increase scalability and availability of wireless networks.

The DPR component was implemented for portable wireless nodes in order to validate the design's feasibility. Based on testbed deployment, measurements of traffic and latency were taken for different topologies, which provide valuable performance indicators for large-scale deployment. Extracted evaluation results of the proposed distributed policy replication methods were compared to those of centralised methods with excellent results.

e. Lightweight policy provisioning and selective policy enforcement

A technology-independent *policy provisioning protocol* was implemented to transfer policy decisions for enforcement on distributed wireless nodes. The use of standardised communication protocols has significantly preserved the system's extensibility and wider applicability on heterogeneous devices. Policy provisioning and enforcement on end-users' devices adds the requirement to respect their preferences and safeguard the unfair use of their personal data. This was addressed by proposing a twofold protection scheme that prevented managing entities to acquire information against the users' will and offered more control to the device owner. *User-centric control* allowed individuals to set their privacy preferences to their controlled networked devices and explicitly restrict access to their personal data, regardless of network policies. In addition, a p*olicy-based regulation* scheme integrated policies from data protection authorities to ensure users' personal data are not collected or exploited.

f. Implementation of self-management capabilities

The contribution of this case study is the realisation of self-management capabilities for policy-based wireless ad hoc networks. Based on a realistic scenario, the proposed policies and implementation facilitated predictable and controlled deployment of wireless ad hoc networks. The performance of wireless ad hoc networks was significantly improved by integrating self-configuration and self-optimisation capabilities for dynamic channel assignment. A wireless channel selection algorithm was integrated with policies to identify channel occupation by competing wireless networks, managing to avoid the most busy channels. Testbed experiments investigated the dynamic adaptation of wireless ad hoc networks, managing to anticipate throughput degradation by reconfiguring their transmission channel and avoiding interference in real-time. This deployment can be considered as a first step towards the implementation of fully self-managed systems.

g. Adaptive service management framework

The contribution of this case study is the extension of the proposed PBM framework to accommodate adaptive service management for wireless networks. The extended framework accommodates a level of control from end-users through preferences. While these preferences can guide the provider to offer a fully customised service, they can also be influenced to achieve optimised service utilisation. Another important feature of the framework is the support for service adaptation based on statistical and contextual information. The simulated deployment of a media service illustrated the concepts and demonstrated the tangible benefits of such an approach.

1.3 Thesis Structure

This chapter has introduced the research motivation and objective, highlighting the novelty and contributions of this work. The thesis is organised as follows:

Chapter 2 provides a background of the investigated area and presents related work found in the literature. The principles of network management and wireless networking are introduced, with emphasis on special issues for wireless ad hoc networks. The policy-based management paradigm is also presented in the context of self-management and autonomic computing.

Chapter 3 introduces a policy-based organisational model for wireless ad hoc networks and elaborates on its novel features. The role-based hybrid organisation with cluster formation and multiple manager capability is examined. The model's scalability is also investigated

Chapter 4 analyses policy design aspects and introduces a custom policy notation and hierarchy. Through the definition of the policy enforcement scope and feedback from context-aware components, closed control loops are formed at three layers. Self-management for ad hoc networks is illustrated using policy examples for different layers. Examples of automated policy conflict detection and resolution are also provided to facilitate a multi-manager environment.

Chapter 5 delves into policy implementation issues and elaborates on the implementation and management of a Distributed Policy Repository (DPR). A step by step design and implementation methodology is provided to realise policies for wireless ad hoc networks. Most of this chapter is dedicated to DPR design and implementation. Implementation details for full and partial policy replication are presented and compared to centralised policy access methods. Finally measurements and evaluation results from testbed deployment are provided.

Chapter 6 presents the implementation of a policy provisioning protocol and introduces selective enforcement for end-user privacy protection. In addition, a twofold protection scheme is proposed to prevent managing entities from acquiring information against the users' will and to offer more control to the device's owner.

Chapter 7 elaborates on two validation case studies. The first one deals with the design and implementation of self-management capabilities for dynamic deployment of wireless ad hoc networks. A channel selection algorithm is integrated in policies to achieve the self-configuration and self-optimisation. Implementation efforts and testbed measurements verify the effectiveness of proposed policies for self-management. The second case study extends the policy-based framework for adaptive service management. The concepts are demonstrated through simulation, to provide customised media delivery according to the preferences of wireless network users.

Chapter 8 provides the summary and final conclusion. A high-level description of the proposed framework summarises the contributions and identifies open research issues for future work.

Chapter 2

Background and Related Work

2.1 Introduction

There exists an enormous amount of literature and research efforts regarding wireless networks and ad hoc networking in particularly. After more than 20 years of research on network management and mobile ad hoc networks, their successes and failures need to be evaluated. This is a necessary process before constructively combining the wealth of knowledge and attempting to contribute towards the management of wireless ad hoc networks. The increasing academic and industrial interest towards wireless technologies, combined with the abundance of portable devices, has motivated research efforts on their management. These efforts need to be examined in the light of the established policy-based management (PBM) paradigm and the emerging self-management or autonomic paradigms.

2.2 Network, System and Service Management

2.2.1 Paradigms, approaches and organisational models

Since the early days of networking, the need for management has remained undeterred. The evolution of networks was accompanied with an evolution of management approaches. However, after more than 20 years of intense scientific research, a consensus has not been reached [23]. According to [23], "research is expected to continue *ad infinitum* as different networking environments emerge with new management needs, providing fertile soil for applying new problem solving techniques".

Among first efforts to standardise management, Open Systems Interconnection (OSI) Systems Management (OSI-SM) [187] from ITU-T* defined five generic functional areas of management, according to the type of operations and information handled. These areas are widely referred to as FCAPS operations, according to their initials:

- *F*ault Management
- *C*onfiguration Management
- *A*ccounting Management
- *P*erformance Management
- *S*ecurity Management

In [23], a thorough analysis of management approaches, frameworks and protocols was provided, offering an insightful historical perspective on Network and Systems Management. A detailed taxonomy was presented, based on the high-level distinction of management approaches in Remote Invocation (RI) and Management by Delegation (MbD) [28]. The former category (RI) remains increasingly popular, with two model subcategories identified as *Manager-Agent* (e.g. SNMP [196], COPS [201],[205], NETCONF [216]) and *Distributed Object/Service Interfaces* (e.g. CORBA [175] ,Web Services [176]). In the latter (MbD) category, notable approaches were based on code mobility [29], including ScriptMIB [30] and Mobile Agents [31],[32]. An introduction to aforesaid referenced technologies and models can be found in [23].

An earlier survey on paradigms for distributed enterprise network and systems management [25] adopts a different approach to provide two taxonomies. An enhanced taxonomy classifies management paradigms based on four criteria, i.e. delegation granularity, semantic richness of the information model, degree of task specification and degree of management automation. Another simpler taxonomy is also quite useful and remains relevant today. Based on a single criterion, the organisational model, this taxonomy identifies four paradigms. A relevant presentation [33] elaborates on that classification, by defining m as the total number of managers, a as the total number of agents, and $n = m + a$ as the total number of elements in the management system. The four paradigm classes of organisational models were identified as follows:

a. centralised management ($1 = m$)
b. weakly distributed management ($1 < m << n$)
c. strongly distributed management ($1 << m < n$)
d. cooperative management ($m \approx n$)

* ITU-T: International Telecommunication Union, Telecomm. Standardization Sector, formerly CCITT

The terminology used in [25],[29] has been adopted to clarify the distinction between management *paradigms* and management *technologies*. Software Engineering considers that *technologies implement paradigms*, using object-oriented analysis and design. According to [25]: "At the *analysis phase*, network and systems administrators select a management *paradigm* (e.g., distributed objects). At the *design phase*, they select a management *technology* (e.g. CORBA). At the *implementation phase*, they use that technology to program the network and systems management *application* [29]".

2.2.2 Evolution of Protocols and Technologies

The authors of [23],[25] note that in the early 1990's proprietary solutions were phased out due to their critical deficiency of supporting interoperability between multiple vendors. The standardisation of two open protocols was an important milestone for network and systems management, namely the Simple Network Management Protocol (SNMP [196]) and the Common Management Information Protocol (CMIP [189]). In the 1980's, the International Organisation for Standardisation (ISO) standardised the OSI System Management (OSI-SM) framework, using CMIP for the first object-oriented management approach targeted on OSI intermediate and end systems. Its adoption by ITU-T as the basis for its Telecommunications Management Network (TMN) model established CMIP in the Telecommunications world [188],[189].

In parallel, work on SNMP was completed around 1990 and was eagerly adopted by the Internet (IP, Internet Protocol) community to manage local area networks, wide area networks and intranets. Its "variable-based" information model and limited set of operations made it efficient and simple, leading to its storming adoption and deployment on the majority of IP-capable devices [23]. Over time, important revisions were made by IETF to keep it up-to-date with the increasing complexity of IP networks. The decision of IETF to stop the SNMP evolution in 2002 [23], solidified SNMPv3 as the final version and shifted IETF's interest to new Internet management technologies [24]. According to [25], network and systems management had thrived on either centralised or weakly distributed paradigms for many years and during the last few years a paradigm shift has been witnessed. In [24], the authors elaborate on the future of Internet management technologies and identify the significant deficiencies and challenges of existing technologies. They categorise the Internet community's management approaches as *evolutionary* and *revolutionary*.

Evolutionary approaches aimed at solving identified problems by gradually improving the existing Internet management framework. Main problems of SNMP were targeted, including the elementary information model, the use of unreliable UDP for transport and the lack of transaction support [23][26]. IRTF's efforts to develop a next generation data definition language (SMIng)

and IETF's work on the Evolution Of SNMP (EOS) could not reach consensus on proposals and both ceased activities in April 2003 [24]. A third approach to solve problems was driven by IETF Resource Allocation Protocol (RAP) WG [193] and resulted in the definition of the Common Open Policy Services Protocol for Policy Provisioning (COPS-PR) [205] and its associated data definition language, the Structure of Policy Provisioning Information (SPPI). COPS-PR was designed to provision complex and continuously changing device configurations generated from emerging policy-based management (PBM) systems.

As detailed later (§2.4), COPS efforts were closely related with the advent of the policy-based paradigm for network management (PBM or PBNM) and parallel work from IETF's Policy Framework WG (POLICY) [194]. IETF had defined a policy-based management framework with a series of new object-oriented information models [204],[207],[210], aiming on one hand to establish a common understanding about PBM [206] and on the other to alleviate SNMP deficiencies using the COPS-PR protocol. For example, it used TCP as its transport and supported transport layer security mechanisms [24]. In spite of initial expectations [121], COPS-PR failed to gain significant market acceptance because it failed to fully address SNMP deficiencies and introduced complexity. Maintenance costs and lack of backward compatibility further restricted its adoption. While researchers were looking into emerging technologies to substitute COPS-PR completely [34], the protocol had been adopted in 2002 by the telecommunications industry for policy control in 3GPP's specifications [179] for "3rd Generation Mobile System based on evolved GSM core networks", i.e. Mobile/Cell Networks based on 3GPP Release 5 and later.

In 2003, the Internet management community admitted that "evolutionary" approaches had failed or had limited market acceptance and focused its interest in "revolutionary" approaches. Revolutionary approaches try to replace existing management-specific technologies with standard distributed systems technologies [23],[24]. Since 2001, vendors had been shipping products that offered XML-based interfaces for configuration management [164]. While industry was already centred towards XML-based approaches, activities in Internet management community (IETF, IAB*) had a slow start. As mentioned in [24], conclusions from the IAB Network Management Workshop in June 2002 stated Internet community's support to investigate XML-based network management and in May 2003 a new IETF Network Configuration (NETCONF) working group was chartered [194] to standardise XML-based interfaces for configuration management [216]. Although initial products (e.g. routers) used either proprietary XML transport mechanisms [164] or CORBA/IIOP [165], the trend towards standardised Web Services and XML/HTTP-based management has continued to evolve and is currently embraced by both the network management

* IAB: Internet Architecture Board (www.iab.org)

community and industry [26],[27],[34],[166]. Web Services (WS) is an emerging Internet-oriented technology, developed and standardised by the WWW Consortium (W3C) [176]. WS were seen as the successor of distributed object technologies due to their strong analogies to CORBA.

Before the investigation of Web Services, a brief introduction to distributed object technologies is provided. According to [26],[23], one of the outcomes of intense research in the mid to late 1990s on the use of distributed object technologies was the Common Object Request Broker Architecture (CORBA) for network and systems management. CORBA uses a fully object-oriented information model supporting inheritance, where objects are defined through their interfaces, which are specified in the Interface Definition Language (IDL). CORBA specifies a general remote call protocol and its most common mapping over TCP/IP in known as Internet Inter-Operability Protocol (IIOP). In telecommunication environments, CORBA gradually phased out OSI-SM, with ITU-T translating original specifications to CORBA's IDL. The key problems of CORBA technology was its relatively heavyweight nature and expensive deployment [23]. In addition, critical requirements of network management were not satisfied, namely lack of support for elaborate information retrieval and scalability problems arising from excessive amounts of dynamic entities to represent the required managed objects [26]. These deficiencies led to CORBA's confinement for service management in telecommunication industry.

Web Services were seen as a promising technology for network management, in spite of XML's verbosity leading to increased overheads compared to SNMP and CORBA [26]. Conversely, the use of XML was also the main advantage of Web Services, due to its universal adoption as an interoperable data interchange format. Approaches such as DMTF* Web-Based Enterprise Management (WBEM), OASIS† Web Services Distributed Management (WSDM), SWSI‡ Web Services Management Framework (WSMF) are currently looking at the use of Web Services as a management framework.

The main building blocks of Web Services are WSDL, UDDI and SOAP [176]. Web Services Description Language (WSDL) is an XML-based language that provides a model for describing Web Services as collections of network endpoints, ports and messages. WSDL documents provide an abstract definition of available services, thus separating them from their implementation. The WSDL specification provides an XML format to compose documents for this purpose. Universal Description, Discovery and Integration (UDDI) is a platform-independent, XML-based registry for Web Services, which service consumers can query to discover services' location of interest.

* DMTF: Distributed Management Task Force (www.dmtf.org)

† OASIS: Organization for the Advancement of Structured Information Standards, (www.oasis-open.org)

‡ SWSI: Semantic Web Services Initiative, (www.swsi.org)

Accessing these services is mainly done using SOAP*[177]. SOAP is an application protocol for message exchange between service providers and consumers, mostly used in RPC (Remote Procedure Call) mode of operation. The default mapping of SOAP over HTTP/TCP/IP is dominant [26].

XML-RPC [157] is considered as the precursor of SOAP and has found acceptance both in industry and academia. It is a pure Remote Procedure Call (RPC) protocol which uses XML to encode its calls and HTTP as their transport mechanism. The reason for its sustained popularity is its apparent simplicity compared to SOAP. In academia XML-RPC has been used for communication of resource-contained portable devices, because it is lightweight, interoperable, easy to extend, easy to deploy and widely supported by devices [104],[105]. Its minimum requirements for XML processing and HTTP1.0 were satisfied even by the most limited Java platforms, i.e. MIDP1.0/CLDC1.0† for Java 2 Micro Edition [160],[161]. In the Internet industry, its simplicity and XML interoperability were its major aspects that opened the programmatic interfaces of popular websites to a massive audience of application developers, e.g. for video (YouTube,www.youtube.com/dev) and photo sharing (Flickr,www.flickr.com/services/api). In parallel, the authors of [170] introduced the Representational State Transfer (REST) architectural style, as an abstract model of the Web architecture. Technologies based on REST paradigm leverage HTTP standard operations to manipulate online resources.

Focusing on service management, it has been accepted that distributed objects are naturally suited to service and application management [23]. Service management involves mostly business process reengineering and automation, for which technologies like CORBA were well suited. The trend towards Web Services and XML/HTTP-based management is also evident in approaches for service management. Indicative of the industry momentum is an announcement of a major equipment vendor in 2007, mentioning that "XML- and SOAP-based web services are becoming the de facto communications and information exchange standard" [166]. However, as mentioned in [23], CORBA and existing technologies will continue being used for service management, given the prior investment in this area. The changing technological landscape blurs the differences between traditional roles like ISP (Internet Service Provider) and MNO (Mobile Network Operator), significantly affecting perception of service provisioning and management. Major market players are strengthening their position by increasingly offering bundled services. For example, a subscription to a single service provider can offer fixed telephony, mobile telephony, broadband access and digital television ("quad-play"). At the same time, the manifestation for

* SOAP: Simple Object Access Protocol. Acronym abandoned by W3C with SOAP v.1.2 [177]

† MIDP: Mobile Information Device Profile, CLDC: Connected Limited Device Configuration

"any-play" [167] appeals to providers, in a much discussed evolution of "enterprise software architectures from a client-server paradigm to a Service-Oriented Architecture (SOA)" [166].

Researchers had identified that service management could benefit from a policy-based approach [168] and the adoption of a policy provisioning protocol COPS-PR by the telecommunications industry points to that direction. In addition, after years of research on programmable routers, where their operating system can execute on demand plugins [158], recently "fully integrated and programmable routers" [159] appeared in the market, promising to revolutionise service management and provisioning. The impact of such technologies remains to be seen in the context of conflicting industry interests and ongoing standardisation efforts.

2.3 Wireless ad hoc networking paradigm

2.3.1 Definitions, characteristics and challenges

Wireless Ad Hoc Networks consist of a majority of end-user devices capable of multihop communication and optionally supported from limited infrastructure. The given definition is in line with literature efforts approaching wireless ad hoc networking as a paradigm rather than as a specific technology [10][13][47]. As previously mentioned, *technologies implement paradigms* [25],[29]. Research on the *wireless ad hoc networking paradigm* has been intense and dates back in military research from the 70's and 80's. With the establishment of IETF's Mobile Ad hoc Networks Working Group (MANET WG [190]) in 1997, the Internet community became actively engaged in an effort to coordinate research and standardisation on the emerging paradigm. The main concern that had monopolised IETF's interest was research on routing protocols, an effort that is still ongoing today. With the exception of a few standardised routing protocols that were adopted in practise, the research impact of MANET on industry and mass market penetration have been minimal. The view of MANETs in isolation from an increasingly networked world has been the main reason for their lack of impact. However, IETF has recently revisited the MANET paradigm and has chartered a new Working Group for ad hoc networking in 2007: the *Ad-Hoc Network Autoconfiguration* (AUTOCONF) WG [191]. The main purpose of AUTOCONF WG is "*to standardise mechanisms to be used by ad hoc nodes for configuring unique local and/or globally routable IPv6 addresses*". It should be stressed that AUTOCONF WG output is currently work in progress [195] and WG's charter has set November 2008 as a review date with the task to "close or recharter the WG". However, market momentum is eminent [41],[47] and the renewed view of *ad hoc networking* is indicative of the abandonment of MANETs' isolation and their definite need to coexist and gracefully integrate with today's networks.

The revitalised approach towards MANET is evident in their definition from IETF. In Jan.1999, the MANET WG defined them *as an autonomous system of mobile nodes* (RFC2501)[198]. Work in progress in the newly formed AUTOCONF WG defines a MANET as *a routing domain containing MANET routers*, or simply put a *loosely connected domain of routers*. The clear view of a technology interacting with other networks, combined with the salient qualities of wireless *ad hoc networking* paradigm, set a pragmatic perspective for renewed research efforts.

The salient characteristics of this paradigm according to the MANET WG ([198], 1999) include:

- *Dynamic topologies*: nodes are free to move arbitrarily in typically multihop network topologies where both bidirectional and unidirectional links may exist.
- *Bandwidth-constrained, variable capacity links*: wireless links continue to have lower capacity than hardwired ones and their realised throughput remains much less than their maximum transmission rate, due to the effects of multiple access, fading, noise and interference conditions.
- *Limited physical security*: mobile wireless networks are generally more prone to physical security threats than are fixed networks, with an increased possibility of eavesdropping, spoofing, and denial-of-service attacks
- *Energy-constrained operation*: nodes may rely on batteries or other limited energy resources.

A slight interest deviation can be identified in AUTOCONF WG, were three fundamental qualities of this paradigm are acknowledged ([195], 2007):

- *Wireless interface characteristics* like variable link quality, interference issues and environmental factors result in a very dynamic temporal performance in terms of packet loss and data rates. In addition, wireless links may exhibit *asymmetric reachability*, causing performance issues with many protocols.
- *Mobility* naturally aggravates communication issues and drastically hinders the attainment, establishment and maintenance of network relationships between nodes.
- *Ad hoc interaction* further complicates the above issues, by allowing nodes to join and leave the network at any time or even form new networks autonomously.

These salient features raise a series of hard challenges for researchers and practitioners. For the purpose of routing and management of wireless ad hoc networks, critical challenges are related to the distinct neighbourhood views among nodes, the maintenance of an accurate neighbourhood view, as well as the very participation of a node in the ad hoc network.

The set of neighbours of a node's neighbours have an extended neighbour relationship, referred to as the *node's neighbourhood* [209]. The possibility of *asymmetric reachability* between nodes results in different view of their neighbourhood. According to [195], *asymmetric reachability* describes two communication properties between wireless interfaces: (a) *non-transitive communication* means packets from X can reach Y, and packets from Y can reach Z, but packets from X may not reach Z, and (b) *non-bidirectional communication* means that packets from X can reach Y but packets from Y may not reach X. These properties can be related to the hidden/ exposed terminal problems that have beset the design of wireless networks' MAC protocols [69].

The local creation of a node's neighbourhood is a complicated procedure which a node must carry out on its own. Therefore, defining the process of determining neighbouring node's existence, presence and loss of existence is a fundamental challenge in MANET. As previously said these relationships are hard to define and even harder to maintain. Historically, two nodes are either neighbours or not neighbours and several simple mechanisms have been used to determine neighbour relationships, e.g. single packet reception, acceptable loss rates, and simple handshakes [195]. This model is not suitable for MANETs, where wireless interfaces may exhibit *asymmetric reachability.* These dynamic neighbourhood relationships affect the performance of IP-based protocols, most of which were designed for bidirectional, transitive and stable communication links, assuming a model like fixed Ethernet [15]

Given the MANET characteristics, determining membership in a MANET can be quite difficult. As nodes can move arbitrary and initiate communications in an ad hoc manner, spatiotemporal participation of nodes in a particular MANET is volatile. Such volatility significantly affects routing and management of MANETs, especially if gateway nodes or MANET border routers exist. The existence of multiple wireless interfaces as well as multiple routing protocol instances on the same node, complicate issues further. In addition, gateway nodes are required to have a consistent view of the nodes reachable through them, while nodes need to determine routes and reachable destinations possibly in a fully distributed manner.

In wireless networks, and especially in ad hoc ones like MANETs, information exchange among protocol layers can be very useful. For example, link layer feedback of failure to sent or failure to receive a frame could be used by the network layer to indicate that a neighbouring node is no longer reachable. Such *cross-layer interactions* can reduce overheads of upper layers and significantly reduce the latency in decision making [44],[45]. In some cases though, such an approach overrides the crucial property of layers' independence and modularity, which had made interoperability possible over the years. In [46], four cross-layer approaches and their effect on modularity were defined: (a) *interlayer communication* preserves layers modularity, (b) *interlayer design* breaks modularity (c) *interlayer tuning* is seen as an intermediate solution, while (d) *non-layered design* adopts no layering at all. Care should be taken when using any cross-layer

approach, in order to extract meaningful conclusions and indications regarding lower layer conditions. Conclusions should be extracted from statistical processing or extrapolation of collected information and not from isolated reports [48]. For instance, failure to deliver a single frame by itself may not be a good indicator that a node is or is not reachable. The task of processing collected information is also a challenge itself for lightweight nodes.

The challenge of scalability obviously is not unique to MANET. Aforementioned challenges and characteristics strongly affect scalability both for routing and management purposes. In 1999, a *large* MANET was considered as constituted from tens or hundreds of nodes (RFC2501). In 2007 the enormous proliferation of wireless technologies and devices is evident, where a *large* MANET is constituted from 100-1000 nodes, while a *very large* one is larger than 1000 nodes [195]. IETF observed an apparent maturity of small (2-30 nodes) to moderate (30-100 nodes) MANET, as admitted in [195] from reasonable test and deployment experiences of routing scenarios. Research in MANET WG is ongoing (e.g. OLSRv2, [190]) and several methods of topology control can be found in the literature [11]. However, reminded of the lack of impact of over 20 years of MANET research, an overview of research efforts and pitfalls can assist in appreciating the reasons and avoiding the same mistakes.

2.3.2 MANET research and experiences

Among various wireless ad hoc technologies, Mobile Ad hoc Networks (MANETs) [190] are probably the most well-known, having received intense interest, especially from the research community. According to RFC2501, published in 1999 [198], synonymous paradigms to MANET included *Mobile Packet Radio Networking*, a term coined during early military research in the 70's and 80's; while the *Mobile Mesh Networking* terminology had appeared in an article in "The Economist" in 1997, referring to the structure of future military networks. In [198], *Mobile Multihop Wireless Networking* was also mentioned as the most accurate synonym of MANET. The novel feature that differentiated this paradigm from existing ones was the dynamic multihop traversal of wireless links (wireless multihop routing), in the absence of infrastructure or external coordination. Multihop routing protocols have been studied extensively for MANETs and this remains among most active research topics of IETF's MANET WG.

Admittedly, in spite of vibrant MANET research for many years, results have not led to significant industrial exploitation or widespread adoption. According to [46], the major reason for the negligible market impact of the "pure general-purpose MANET" paradigm is the lack of realism in the research approach. *Pure MANET* [46] refers to the complete absence of any infrastructure or management authority, as opposed to "*hybrid general-purpose MANET*". *Hybrid MANET* relaxes the constraint of no infrastructure support. Traditional *pure* MANETs have been

mainly investigated through analytical modelling and simulation studies. Important work on analytical modelling established the theoretical foundations of multihop ad hoc communications (e.g. the capacity of wireless networks [35]) and provided technology specific guidelines to assist MANET designers (e.g. improvements of distributed coordination function of 802.11 [36]).

Simulations have been widely used in academia, mainly to overcome the lack of experimental testbeds and to enable large-scale deployment of networks. Unfortunately, extensive simulation studies and lack of real life deployment have been the Achilles' heel for the applicability of MANET research. For example, topology control (TC) has been a well studied area. According to [11] though, there is no experimental evidence that the considerable theoretical and simulation-based research on topology control (TC) can actually benefit the network, e.g. by reducing node energy consumption or radio interference. The author states that "*the lack of experimental demonstrations of the usefulness of TC mechanisms is probably the most important open issue in this research field*" [11](§15.5,pp.199).

According to a survey of MANET simulation studies [37] , the majority of research efforts published in the MobiHoc conference [38] had been supported by simulation, particularly 114 out of the 151 papers published (75.5%). Simulations were typically based on open-source network simulators for academic use (53.8%), e.g. ns-2*, Glomosim†, and occasionally on commercial simulators (12.6%), e.g. Opnet (www.opnet.com), Qualnet (www.scalable-networks.com). Unreservedly, simulations can provide insightful results and indications of problems and bottlenecks in protocol and network design. They need however to be used with caution and with careful parameter setup. The authors of [37] expressed their concern that over 90% of the MobiHoc published simulation results may include bias, since very few addressed initialisation bias and random number generator issues. In [39], the authors question the credibility of MANET simulation by citing comparative studies of different simulators and highlighting the incoherence of their results under the same conditions. For instance, they have observed that "*differences in comparative analysis between routing protocols can be due to underlying (and possibly undocumented) parameter settings and not the protocols being compared*" [39]. Another important issue highlighted is the combined treatment of Physical Layer (PHY) and Medium Access Control sub-layer (MAC) by most simulators, and the lack of customisation of MAC/PHY simulation parameters according to each scenario.

Contrary to the minimal MANET impact, research interest for other multihop ad hoc networking technologies has been renewed recently. Technologies like Mesh networks [41],[42] and

* http://www.isi.edu/nsnam/ns/

† http://pcl.cs.ucla.edu/projects/glomosim/

Vehicular Ad Hoc Networks (VANET) [43] are gaining popularity, by adopting realistic assumptions and coupling theoretical studies with implementation and testbed deployment [47]. Novel variations of wireless ad hoc networking like opportunistic networking [49] have also emerged, while the related paradigm of wireless sensor networks has already been applied in practise [11],[50] with significant industrial support.

2.3.3 Research on the management of ad hoc networks

In spite of the MANETs' limited market impact, the research community has identified the need for their management and early attempts date back to 1999 [51]. The value of managing a MANET would stem from their scalable coordination and the ability to deploy services with sufficient QoS. The apparent contradiction of effectively managing autonomous ad hoc deployments of individual wireless nodes has been addressed by researchers under different assumptions [9], [51], [52], [53], [54], [55], [56], [57], [58], [59], [62], [64], [65], [67]. The main assumption was the uniform deployment of homogeneous wireless devices, under the direct or indirect control of a central authority. Overall, related literature on MANET management has been limited and proposed solutions attempt to partly solve relevant problems. Existing approaches vary regarding the adopted organisational model. Recently, there has been a shift towards the policy-based paradigm and hierarchical PBM systems for MANETs have been considered. Table 2-1 summarises related work on management of MANETs. These efforts are particularly related to the wireless ad hoc networking paradigm and the management of relevant technologies, because they differentiate this paradigm from traditional ones for wireless networks. Therefore a critical evaluation of their contributions is attempted below.

Table 2-1. Taxonomy of related work on MANET management

	Tiers	Hierarchical	Distributed	Policy-based	Agent-based	Modules	Storage	Managers
W.Chen et al [51]	3	+	-	-	-	1	anmpMIB	1
C.Shen et al [52]	2	+	+	-	+	4	MIB	0-1
R.Chadha et al [53][54]	3	+	-	+	+	1	mySQL	1
K.Phanse et al [55][56]	3	+	-	+	-	2	PIB	1
R.Badonnel et al [57][58]	3	+	+	-	-	1	anmpMIB	1
A.Hadjiantonis et al [5]	**2-3**	**+**	**+**	**+**	**-**	**1-3**	**LDAP**	**≥1**

The first efforts to tackle MANET management were presented in [51]. The suggested Ad hoc Network Management Protocol (ANMP) was based on hierarchical clustering of nodes in a three level architecture. At the top level there is a manager, who manages various cluster heads which

in turn manage the agents at the lower level (nodes). The protocol is compatible with SNMPv3 and uses the same PDUs (Packet Data Units). It introduces an extension to MIB II (Management Information Base), called anmpMIB, to facilitate mobile specific properties. Furthermore, it exploits the triggering capabilities of SNMP in order to dynamically reconfigure the agents by setting alarms. This is done by associating different and downloadable functions to alarms. In this proposal, clustering for management purposes is used and can be combined with routing clustering. A group of agents form a cluster which is managed by a cluster head. The creation of management clusters is performed by using one of two proposed clustering algorithms. The first algorithm is graph-based clustering and the second one is geographical clustering. The latter algorithm depends on the availability of GPS data on each node in order to perform a spatial cluster formation. ANMP also introduced a "guest protocol" which allowed some isolated nodes to be served by another cluster, instead of creating a new one. The performance of the described algorithm was evaluated through simulations and the results showed that unmanaged nodes and overheads increase as the periodic clustering interval and the ping interval are increased. In spite of some pioneering concepts introduced at that time, e.g. GPS-based clustering, ANMP was severely restricted by its centralised philosophy. It was based on the strict hierarchy of SNMP manager-agent model, which is not well suited for ad hoc networks. In addition, both clustering algorithms proposed were also limited by their centralised conception. An interesting idea introduced was the "guest protocol", which can effectively refrain the creation of small clusters and excessive cluster partitioning.

Another management model examined is the "Guerrilla" architecture, described in [52]. This model adopts a peer-to-peer paradigm and facilitates a supervisor/agency model. Its architecture is based on a two-tier distributed infrastructure where at the higher level "nomadic managers" possess most management intelligence, make decisions and launch active probes to fulfil management objectives. "Nomadic managers" execute the "Nomadic Management Module" which facilitates decision theoretic and mobile code techniques. This enables the Nomadic Managers to make decisions and to spawn when needed. At the lower level, actives probes (lightweight programmable scripts) perform localised and remote management tasks using SNMP agents. The role of the active probes can be either monitoring or task specific. Monitoring probes are used to explore, discover and maintain the ad hoc network's topology. An interesting approach was presented to enable decision making, i.e. the use of a utility function. The utility function indicates mathematically the current network conditions and can trigger the execution of an action if its value drops below a threshold. The major advantage of this model was its high degree of decentralisation. Though high-level policies were mentioned, the model cannot be considered as policy-based. The reason is that policies were implemented in the form of a utility function which estimated the current network status based on predefined static parameters.

The next model examined is the DRAMA architecture [53][54],which proposed a policy-based network management system using intelligent agents. Policy agents were deployed to manage the network through a tiered hierarchical architecture. This model was developed under the US Army CERDEC Dynamic Re-Addressing and Management for the Army (DRAMA) program to address the special needs of tactical mobile ad hoc networks. The use of several proprietary military protocols (YAP [60], DRCP/ DCDP [61], AMPS [62]) restricted the wider adoption of this system. This system adopts a policy-based approach and automatically enforces high-level policies using intelligent agents. Three types of policy agents were defined to manage the network through a two level architecture. A Local Policy Agent (LPA) manages a single node, Domain Policy Agents (DPA) manage DPAs or LPAs in a Policy domain and Global Policy Agents (GPA) manage DPAs. The system initially used a Data Distribution Service (DDS) based on proprietary military protocols, i.e. DRCP/DCDP for unicast and enhanced YAP for event reported. A variety of management agents (configuration, reporting etc) was used to accomplish the task of policy enforcement. Policy definition followed the principles of the IETF/DMTF by adopting the ECA notation through the PECAN (Policies Using Event–Condition–Action Notation) specification language [59]. The taxonomy of policies included general-purpose, monitoring, configuration, reporting, filtering and aggregation policies. This work mentioned some experimental measurements and concluded that aggregation and filtering reduced overhead while automatically triggered reconfiguration improved management performance. The use of proprietary, non standardised protocols (YAP, DRCP/ DCDP, AMPS) prevented interoperability and wider use of the architecture. Its military orientation was obvious in the architectural design of the system, since it encapsulates a hierarchical military scheme. In other words, although filtering and aggregating enabled some local management control, the management hierarchy was rather strict and the overall control remained centralised.

Another policy-based approach was presented in [55],[56], aiming to provide QoS in MANET. This suite consisted of four schemes that cooperatively managed a MANET. The proposed schemes were k-hop cluster management, dynamic service redundancy (DynaSeR), service discovery and inter-domain policy negotiation. According to the k-hop cluster management scheme for clustering, the number of hops between a policy server and its clients is limited to k hops. Two ways to implement clustering were proposed, either by taking advantage of the topology information gathered by the underlying proactive ad hoc routing protocol or through interaction between the Common Open Policy Service (COPS) protocol-based application layer and the IP layer. The dynamic service redundancy (DynaSeR) solution implemented redirection and delegation that allowed the PBM system to improve its service coverage. Redirection was a server-centric way of helping a client leaving its current cluster to discover a new server. Delegation allowed dynamic invocation of policy server instances on-demand to cover as many

clients in the network as possible, covering those that lie outside all existing clusters. Extensions to COPS-PR protocol, added delegation capabilities. The suggested service discovery was a mechanism to facilitate automated discovery of policy servers in the network and extended COPS-PR with two messages (SA, CSRQ). A policy server periodically advertises itself via a limited k-hop broadcast of SA messages. A client that does not receive an SA message within a certain time interval broadcasts a CSRQ message. The server, which may have moved within k hops of the client, responds with a unicast SA message. Alternatively, a client node that is currently being serviced, upon hearing a CSRQ message, may volunteer to act as a delegated server. The inter-domain policy negotiation scheme extended the COPS-PR protocol to facilitate inter-policy server communication and to support policy negotiation between different network domains. The proposed schemes and protocols were implemented both as a prototype in a Linux-based ad hoc network testbed and as simulation models in QualNet.

The main drawback of the aforementioned policy-based work was its explicit dependence on the COPS protocol. As already discussed, COPS-PR has found little acceptance in network management. Its relatively heavyweight nature limits its applicability to resource constrained MANETs. Therefore the proposed solutions in [55],[56] may not be future-proof. On the other hand, the concept of DynaSer scheme using delegation and redirection is quite interesting and can provide a solution to scalability issues. A similar idea had been introduced in ANMP [51] with its "guest protocol".

One of the most recent approaches towards MANET management is *probabilistic management*, as introduced in [57],[58]. The authors proposed a management approach for ad hoc networks based on probabilistic guarantees, where instead of addressing the management of the whole network, the network is partitioned and only a subset is effectively managed. By introducing a "spatio-temporal connectivity measure", mathematical calculations extract "spatio-temporal connected components" as the subset of nodes with the highest management interest. The first or second connected components are only managed and among each component a manager node is elected. The election algorithm uses the "K-means classification methods" to select nodes based on their network behaviour. Probabilistic guarantees on the percentage of managed nodes were derived based on extensive ns-2 simulations. The proposed approach was integrated into the aforementioned ANMP architecture [51], by replacing its centralised clustering algorithms. Required protocol operations where piggybacked on OLSR, by defining new fields to carry the required information for probabilistic calculations among neighbours. Overall, the probabilistic management approach is quite interesting and its main innovation is focused on the recognition that the total number of MANET nodes cannot be guaranteed as being managed. Based on that, distributed algorithmic clustering was performed to partition the network and elect the most capable managers. The main drawback of this approach is the lack of a case study to support the

admittedly novel concepts, combined with its simulation-based evaluation. Both observations limit the scope of these efforts and restrict its adoption in real life wireless ad hoc scenarios. The suggested implementation is explicitly based on a proactive MANET routing protocol (OLSR), which is expected to incur limited management overheads for clustering and manager election. At the same time though, its general use in wireless networks is limited to OLSR-based MANETs.

The considerable rise in nature and bio-inspired computing research has also been suggested for MANET management. In [64],[65] the authors introduce stigmergic learning inspired by insect pheromones, i.e. the chemical substances that trigger a natural behavioural response in another member of the same species. Pheromones enable individual agents to adjust their level of activity as the system operates and extend this mechanism to the self-organisation of autonomous wireless nodes. These relatively new concepts of bio-inspired computing are mostly agent-based and have been developed through military funded projects. This is depicted in their initial MANET applicability scenarios, like the management of unmanned vehicles and foot soldiers operations.

Organisational models for ad hoc networking

Scalability has always been one of the main challenges of MANET and wireless ad hoc networks in general. Research and practise have shown that scalability can be enhanced with appropriate network organisation, for example the hierarchical IP addressing scheme for the Internet. It is therefore important to review approaches for wireless ad hoc networks' organisational models. The aforementioned literature for MANET management provides a useful starting point, since these approaches introduced the basic requirements of wireless ad hoc network management and the need for differentiation from traditional organisational models (§2.2.1,pp.9).

For wireless ad hoc networks, it is obvious that a centralised organisational model is not suitable. Assigning a single central entity to manage the whole network may be impossible, because nodes are intermittently connected. Nodes may appear and disappear at any time, for example due to radio environment variations or due to battery exhaustion. In the case that the manager node disappears or is disconnected, then inevitably the network remains unmanaged. The major problem of a single point of failure introduces the need for a distributed organisational model. Spreading management responsibility among nodes makes the network fault tolerant.

Beyond the aforementioned problems specific to ad hoc networks, problems that apply to fixed ones are magnified in ad hoc ones. In large scale networks the task of centralised management requires a considerable message overhead which may cause congestion problems. Overprovisioning network resources is a common remedy in these cases. Conversely, this solution is not applicable to ad hoc networks since they have very limited bandwidth and the high message overhead involved in management would consume the scarce nodes resources. In a few words, the special properties of ad hoc networks, like intermittent links, sparse bandwidth and limited

resources, make the centralised model for management inapplicable. To anticipate some of these issues, the *weakly distributed management* (or *hierarchical*) model has been proposed in the literature. As an alternative to centralised and hierarchical models, *strongly distributed* (or simply *distributed*) organisational models have emerged. Their distributed nature is more suitable for the management of an ad hoc network, because there is no single point of failure and nodes can connect and disconnect from the network without major disruption. The distributed and hierarchical approaches both rely on more than one entity to collectively manage the network by maintaining a loose hierarchy among nodes. In this way, management is fault tolerant and reliable. The extreme case of the *collaborative management* (or *peer-to-peer*) model has received interest in fixed network management, for the moment though its resource-demanding nature is restricting its applicability on lightweight wireless networks.

Focusing on the organisational models adopted by the aforementioned literature, it is noted that the centralised model is almost absent. In [63], the authors presented a mathematical evaluation of scalability in logically ad hoc networks. Their work reviewed current and future directions of ad hoc networks' organisation and provided useful directions and alternatives. From the analysis in [63] it is clear that a star topology, i.e. a centralised model, is not efficient for ad hoc networks and does not scale well. However "mesh" topologies, which include a combination of distributed and hierarchical models, scale efficiently. The important feature which is used in mesh models is the ability of exchanging management information (and management policies) between managed nodes. In this way, the overall manager, if it exists in the model, has less congestion probability. One of the models described in [63], referred to as "mesh with partial autonomy and hierarchical coalitions", is quite interesting since it combines the advantages of distributed and hierarchical organisational models. It uses the idea of clustering for management purposes and introduces "sub-controllers" which cooperate and exchange management information. These "sub-controllers" are loosely managed by a "controller" (manager) i.e. they receive directives but also have some freedom of choice.

The idea of a distributed and hierarchical model is realised in the aforementioned "Guerilla" [52] architecture with the use of "nomadic managers" that form an "agency". Among the agency's nodes, peer-to-peer communication takes place and an overall "supervisor" (manager) monitors the network. This "supervisor/agency" model is interesting because management responsibility and intelligence is distributed among the nomadic managers which form the higher tier of this two tier model. The hybrid distributed/hierarchical model is also adopted in [55],[56] using cluster heads to manage clusters and allowing cluster heads to communicate and redirect nodes from one cluster to another. The redirection technique combined with delegation enriches the model. The introduced delegation creates a control sub-tier where "delegated" nodes act as servers (similarly to proxy functions).

A different approach is adopted in ANMP [51] and DRAMA [53],[54]. Both follow a hierarchical organisational model with some deviations. The same technique as previously was used, which uses cluster heads to manage clusters (in [51]) or policy domains (in [53]). As a result a tiered management architecture is realised. The essential difference between these models and the previous ones is that cluster heads in [51] (or DPAs in [53]) do not communicate with each other in order to exchange management information. Instead information is gathered and reported to the overall manager (or GPA) who possesses a significant part of management responsibility and takes most decisions. This hierarchical approach is more suitable for tactical ad hoc networks and its applicability to general purpose ones may be limited. The reasons supporting this opinion are first the high dependence on a central manager and secondly the reduced flexibility for cooperation between clusters heads. Hence, the survivability of the network depends primarily on the survivability of the manager node. This is not an obvious disadvantage for military ad hoc networks, where the manager node is presumably well protected in friendly ground and has adequate resources available. However, in the general case, the possibility of network failure is high, since the manager can be abruptly disconnected from all other nodes. Considering cluster heads, their restriction from communicating with each other makes them highly dependent on the manager and possible disconnection from it can lead to unmanaged clusters. These drawbacks can be anticipated by allowing communication between cluster heads in order to have cooperation and updates for management tasks.

Algorithms for wireless ad hoc network organisation

The significance of network organisation has been introduced earlier and its main motivation is certainly scalability. Wireless ad hoc networks pose many more challenges, making their efficient organisation harder. However, beyond scalability, expected benefits can include the increase of following properties: coverage, capacity, bandwidth, battery drain time and more. These can be achieved through the reduction of traffic and signalling overheads, adjustment of transmission range and better methods for wireless channel access [11].

Regarding scalability, clustering has been widely used in ad hoc networks to form a hierarchy of nodes for routing and message dissemination. As seen already, clustering has been used at the application layer for management purposes to associate roles to devices, e.g. cluster heads and cluster nodes. A range of algorithms [81],[82],[83],[84],[85],[88] can be used for cluster formation and maintenance, depending on the requirements of the applicability scenario and network composition. For example, ad hoc deployments for tactical operations have quite different requirements than user-initiated wireless networks. Beyond the traditional research domain of algorithms for routing performance optimisation, the renewed view of multihop ad hoc networks as extensions to the Internet gave rise to a series of different problems. For example, the

cache placement problem relates to efficient temporary placement of information on nodes and the selection of those nodes in volatile wireless ad hoc networks [88]. Similar issues are related to the replica placement problem, where a set of data needs to be efficiently replicated in the network to minimise access overheads among nodes [89],[90].

Solutions based on node domination have been used extensively to optimise routing paths and reduce protocol signalling overheads in MANETs. The main idea is to select a set of nodes in a multihop network which can route or disseminate forwarded traffic efficiently, minimising a predefined metric, like path length, number of messages or consumed energy. [81],[82],[83],[84], [85],[88]. Algorithms based on Dominating Set creation are a popular solution used for virtual backbone formation and gateway selection. Virtual backbones create a connected sub-graph of a network which is used for traffic forwarding. The selection of a dominating set of nodes is used in proactive MANET routing protocols ([9]:§20,pp.425). The majority of these solutions are based on adapted distributed solutions of the Dominating Set (DS) problem. A special case of the DS problem is predominantly used, the Minimum Connect Dominating Set (MCDS) problem. Useful definitions are provided below ([11]:Elements Of Graph Theory, pp.225-8):

Dominating set: Given a graph $G = (N,E)$, a dominating set for G is a set D of nodes such that for any $u \in N - D$ there exists $v \in D$ such that $(u, v) \in E$; that is, any node in the graph is either in D or adjacent to at least one node in D.

Connected dominating set: Given a graph $G = (N,E)$ and a dominating set D for G, D is said to be a connected dominating set if G_D is connected; that is, if the subgraph of G induced by node set D is connected.

Connected graph: A graph $G = (N,E)$ is connected if for any two nodes $u, v \in E$ there exists a path from u to v in G.

The optimal solution of these problems, i.e. minimum sets calculation (MDS,MCDS), is NP-hard [81], hence is rarely addressed in practical networks. Non-optimal solutions to the defined problems have been proven to be NP-complete, therefore various optimisation heuristics have been used in literature. Departing from the mathematical strictness of Graph Theory, it is common practise in Computer Science and Engineering to use *heuristics* to reduce the computation time for a problem, yielding a near-optimal solution under certain conditions much faster. The use of heuristics is especially necessary for efficient distributed execution of algorithms, suitable for portable wireless devices with limited resources. A ρ-approximation algorithm is defined as a polynomial time algorithm that always finds a feasible solution with an objective function value within a factor of ρ of the optimal cost [90]. This metric is used to compare and evaluate the performance of proposed heuristics, compared to optimal solutions calculated by *brute force*, i.e. exhaustively testing all possible solutions to find the best.

An efficient distributed execution of the connected dominating set calculation was proposed by Wu [78] and has been widely used to create virtual backbones in MANET [79],[80]. The Connected Dominating Set (CDS) creation algorithm by Wu is briefly described below, while additional information is provided in Appendix A.. All nodes (devices) execute the distributed algorithm to independently decide whether they should mark themselves as gateway nodes. Marked nodes create the connected dominating set of the graph, thus ensuring one-hop accessibility for the remaining nodes. A final set is created in two rounds. The first round involves neighbourhood information exchange among nodes and a marking process. Every node marks itself as a CDS node if it has two unconnected neighbours, i.e. two nodes without a direct link. The second round involves an optimisation process: after CDS nodes advertise their selection, they locally apply two rules that result in the elimination of redundant CDS nodes and a smaller Connected Dominating Set. Two optimisation rule heuristics make use of an arbitrary unique node identifier (*node id*) during the pruning process to assist in the unambiguous elimination of CDS nodes.

An important aspect of this algorithm is that it only requires two rounds to conclude, leading to a relatively fast distributed selection. In addition, the complexity and message overhead cost is quite small compared to other proposed algorithms, e.g. from Das et al [83],[84]. Particularly, the selection cost at each node is $O(\Delta^2)$, where Δ is the maximum node degree (the maximum number of node's neighbours). The total amount of message exchanges is $O(\Delta v)$, where v is the total number of nodes in the network. The algorithm also defines efficient update and reconstruction procedures for the maintenance of the CDS under node movement and failure. An adapted version of Wu's algorithm has been used for management of clustered MANET based on context-aware heuristics [67], while power-aware heuristics were used by [80]. By replacing the *node id* with a context-aware [2],[67] or power-aware [80] function, the optimisation rules ensure that not only the most connected nodes remain in the CDS, but also the most capable. Additional solutions based on dominating sets can be found in [81], [82], [88].

A notable solution for the distributed creation of a connected dominating set is also provided by OLSR [209], a standardised proactive MANET routing protocol. OLSR uses a fully distributed algorithm to select Multi-Point Relay (MPR) nodes that form a connected dominating set for efficient flooding and reduction of protocol overheads. The MPR algorithm provides highly distributed solutions and aims to minimise the MPR set through the use of heuristics. An MPR set is similar to a *virtual backbone*, as previously examined.

For the purpose of literature examination, a data provider DP is the information host, whereas Master DPs hold original data and Slave DPs host replicated data. The *replica placement problem* is defined here:

Given an arbitrary network G and a number M of Master DP, select a number of N network nodes to place a Slave DPR, such as to minimise the total cost of replicating the data of M to N plus the cost of data access for the rest of the G-(M+N) nodes.

Replica/cache placement and management remain active research topics for both fixed and wireless networks. Multihop ad hoc networks have differentiated from established solutions, to anticipate their inherent dynamic nature and link instability. A range of algorithms has been proposed [88],[89],[90],[91], combined with the need to efficiently organise the ad hoc network and reduce management overheads. While the aforementioned optimal *replica placement problem* has not been formally proven as a computationally infeasible task, the majority of the algorithms adopted for its solution are considered not "feasibly computable" [22],[76], formally proven to be at least NP-complete, if not NP-hard.

Node domination based solutions have already been examined. Solutions in this family have been used extensively to optimise routing paths and reduce protocol signalling overheads in MANET. In addition, their excellent distributed performance has motivated their use for management purposes as well. For example, OLSR has been used by [55],[56],[58], piggybaggging its messages and exploiting its distributed MPR creation. Wu's algorithm has also been used for the management of clustered MANETs [2],[67]. Similarly, the issues of *cache or replica placement* in MANETs have been addressed through various algorithmic methods that use a variety of heuristics to yield distributed solutions [88], [89], [90], [91]. Recent work in [89] addressed the cache location problem for wireless ad hoc networks and suggested *benefit-based* data caching algorithms to solve the problem. These solutions are particularly attractive since they take into consideration multiple data items, where each data item has a server and a set of clients that wish to access the data item at a given frequency. An algorithm selects data items to cache at each node under memory restrictions. The authors claim their centralised approximation algorithm delivers a 4-approximation solution and a localised distributed version of the algorithm performs very close to that approximation, handling mobility of nodes and dynamic traffic conditions.

Solutions based on *facility location problems* adopt concepts of Location Analysis and Operational Research (an interdisciplinary branch of applied mathematics) [87]. For example the connected facility location problem, has been used to address the replica placement problem and has been proven to be NP-hard [90], [91]. In general, facility location problems involve a given number of facilities that needs to be optimally located in an existing area and fulfil given requirements. Facility location problems are particularly attractive as solutions to the replica placement problem because they follow similar requirements, e.g. cost minimisation or minimisation of the facilities number. Facility location problems were traditionally encountered in urban design and applications [87], e.g. provisioning of public services, like determining the locations of post offices, transportation terminals or fire fighting units. The general requirements

case can be narrowed to *Median* and *Centre* problems. In median problems a pre-specified number of facilities must be located so as to minimise the average distance. In *Centre* (aka *minimax*) problems, a pre-specified number of facilities must be located so as to minimise the maximum distance (or time or cost) to or from the facilities that any user will have to travel [86],[87]. During the last few years, these solutions have been investigated in the context of wireless networks design and have been adapted to approximate problems like the optimal cache placement [90],[91]. An approach based on code mobility was undertaken in [86], where mobile agents would autonomously decide on their optimal location, in order to partition the network and minimise total hop distance between cluster heads and their cluster nodes.

In [90], the authors elaborate on the "efficient cache placement in multihop wireless networks" and attempt to find the optimal cache placement which minimises the total cost, i.e. the incurred overheads from cache updates and requests to caches. They prove that the problem is equivalent to a special case of the NP-hard *connected facility location problem*, called the *rent-or-buy problem* [91]. The problem formulation as explained in [90],[91] is provided here: *An existing facility is given, along with a set of locations at which further facilities can be deployed. Every location is associated with a service demand, which must be served by one facility.* The rent-or-buy problem is also NP-hard [91], therefore several approximation algorithms (heuristics) have been developed in [90]. The described polynomial-time algorithm approximates the optimal (brute force) solution for arbitrary graphs within a factor of 6, in a distributed implementation.

Unfortunately, with the exception of a few cases that have been deployed in practise, the same issue exist regarding the usefulness of extensive simulation and the lack of experimental evidence [37],[39]. As with the case of topology control (TC), this issue is a critical open issue of this research field ([11]:§15.5)

2.3.4 Network Layer and Multihop Routing Issues

Routing is one of the most investigated areas of wireless ad hoc networks and MANET in particular. Several routing protocols have been investigated for ad hoc networks and different classifications have been defined [40]. For example, based on their route establishment strategy, two main categories can be identified according to IETF [190]. *Proactive* protocols maintain routes to known destinations and use periodically updated routing tables for traffic forwarding. The Optimised Link State Routing Protocol (OLSR) is an example in RFC status (RFC3626 [209]). Other examples of proactive protocols are TBRPF, DSDV [190]. *Reactive* protocols establish routes on-demand once communication with a destination is needed. The Ad hoc On-Demand Distance Vector (AODV) protocol is an example in RFC status (RFC3561 [208]). Other reactive protocols are DSR in RFC status and recently DYMO in Internet Draft status [190].

Hybrid routing approaches (e.g. ZPR, Zone Routing Protocol) and hierarchical ones attempt to combine the benefits of proactive and reactive routing [40]. It should be noted that the applicability of defined multihop routing protocols extents beyond the strict MANET paradigm and has been adopted by different multihop paradigms and technologies, e.g. mesh networks.

As already mentioned, after 20 years of routing protocol research, IETF has revisited the MANET paradigm and has chartered a new Working Group for ad hoc networking in 2007: the *Ad-Hoc Network Autoconfiguration* (autoconf) WG [191]. The main purpose of AUTOCONF WG is "*to standardise mechanisms to be used by ad hoc nodes for configuring unique local and/or globally routable IPv6 addresses*". Work in progress defines a MANET as *a routing domain containing MANET routers*, or simply put a *loosely connected domain of routers*. Accordingly, a MANET router is distinguished by having one or more MANET interfaces and in addition it may also have zero or more non-MANET interfaces. A MANET router is responsible for hiding MANET's characteristics from non-MANET nodes. This approach is indicative of the abandonment of MANET isolation and their definite need to coexist and integrate with today's networks. The reference to *one or more MANET interfaces* has already been investigated in the context of *mesh* networks [41], one of the most prominent applications of multihop ad hoc networking paradigm.

In fact in 1999, the first RFC of MANET WG (RFC2501,[198]) mentioned that mesh-based mobile networks can be operated as robust, inexpensive alternatives or enhancements to cell-based mobile network infrastructures. Although the telecommunications industry has showed some interest in the context of *multihop cellular* or *hybrid ad hoc networks*, in practise these approaches have not found much response. However, the Internet community has come to gradually adopt Mesh networks [41], which were boosted among other reasons from the maturity of routing protocols and the establishment of sufficient performance at the network layer of multihop networks. Economic reasons also supported this interest shift, since the advent of wireless technologies in emerging markets. These markets required a technology with low infrastructure cost and the ability to cover large geographic areas with low population densities [41],[42],[46]. Mesh networks combine the benefits and convenience of ad hoc networking with the support of wireless infrastructure. As such, they use MANET routing protocols and extend those in proprietary ones depending on deployment needs and applicability. Commercial mesh networks have been successfully deployed, using proprietary equipment and protocols, while companies like Tropos (www.tropos.com) or Bel-air (www.belairnetworks.com) advertise large-scale mesh networks deployments. Rooftop mesh networks for free public wireless access have also been developed, e.g. the Meraki (meraki.com) project. According to [42],[47], the increased interest in multihop ad hoc networks, both from industry and academia, has been fuelled by their pragmatic approach towards realistic deployment and real life experiences.

Another critical issue that needs to be addressed in wireless ad hoc networks is the assured forwarding of packets among participating nodes [70],[71]. This is one of the basic requirements for any networked application to be deployed over multihop ad hoc networks. The duties of fixed routers are carried out by the participating wireless nodes and network operation relies on their good intentions to forward the received traffic. This not always the case and often selfish or malicious nodes refuse to forward packets, leading to congestion or even worse network downtime. Incentives mechanisms have received a lot of research interest [92], their deployment though is limited. Detection mechanisms are also investigated, aiming to determine which nodes are misbehaving and take appropriate measures against them [70],[71].

The strengthened interest in multihop ad hoc networking has assisted divergence from traditional MANET research and helped overcome the aforementioned pitfalls of simulation studies. As suggested in [46], the lack of realism is considered as the main reason for the negligible market impact and deployment of MANETs. Coupled with limited attention to users' requirements and non-existent deployment of network prototypes, this approach gradually restricted MANET to a few special purpose scenarios. Renewed interest on an *evolved multihop ad hoc paradigm* has flourished and poses increasing requirements for efficient and scalable management.

2.3.5 Physical and Data Link Layer Issues

The deployment of wireless ad hoc networks suffers from limitations in wireless link connectivity and capacity, due to the design of Physical (PHY) and Data Link layers (MAC sub-layer). The capacity and throughput are limited and severely degrade as the user population and number of hops grows [35]. Intermittence and interference amplify the problem, since enabling wireless technologies need to share the same spectrum, while used ISM (industrial, scientific and medical) frequency bands are by definition subject to interference. In real life, in order to deploy wireless network testbeds, the family of IEEE 802.11 standards [183] is usually considered, since it is the most widely deployed technology. Devices based on 802.11(a,b,g,n) are operating in unlicensed ISM radio bands and can arbitrarily use any of the defined channels for deployment.

The design of appropriate MAC layer algorithms makes these technologies fairly tolerant against interference and noise, but this comes at a price. Speed and performance are sacrificed in order to allow multiple stations to share the same wireless medium, i.e. the available spectrum. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) protocols attempt to reduce the collision probability by sensing the wireless channel and backing off if it is sensed busy. The classic problems of hidden and exposed terminals are quite common [69]. An additional measure to prevent collisions can be used, the RTS/CTS handshake (Request To Send / Clear To Send) [183]. The use of Spread Spectrum modulation techniques can cause increased collisions due to

interference between channels with inadequate frequency separation (inter-channel interference). This happens because channel spacing is overlapping for maximum frequency reuse. Depending on the enabling technology and modulation, different channels are likely to interfere with each other and interference increases the nearer the central frequencies of channels are. Recommended deployments normally assume use of non-overlapping channels for collocated deployments, while spatial reuse is also possible. An example *channelisation* for IEEE 802.11 in 2.4GHz band is shown Figure 2-2, with three non-overlapping channels.

One of the crucial problems of wireless ad hoc networks is the establishment of lower layer (MAC/PHY) connectivity without central administration. Most of MANETs research takes this connectivity for granted, assuming a single channel for the communication of all MANET nodes. The basic connectivity settings for devices joining existing WLANs, e.g. public hotspots or home networks, are automatically provisioned by the controlling wireless access point (AP). Lower layers (MAC/PHY) are automatically configured by the wireless hardware drivers, based on the AP management frames (beacons). For ad hoc wireless networks, the apparent obstacle is how to establish communication in the absence of an AP.

IEEE Std 802.11 and IBSS (ad hoc) mode

The IEEE 802.11 protocol family [183] is a remarkable standardisation achievement, having been established as the dominant Wireless Local Area Network (WLAN) protocol to date. IEEE 802.11 defines the MAC/PHY protocols for WLAN technologies. Over the years it has been revised several times with continually increasing data rates, security and functionality, maintaining and solidifying its market presence. A number of factors have played their role towards its dominance, including among others the coordinated standardisation activities, continuous specification updates and maintenance of backward compatibility through IEEE, as well as strong industrial support and consensus through the Wi-Fi® Alliance. Another important aspect was the use of unlicensed ISM spectrum, which expedited its worldwide adoption by avoiding regional regulatory delays. As a result, myriads of devices support at least one of IEEE 802.11 specifications and in spite of its identified deficiencies and competition, it is expected to remain the dominant protocol for WLANs. Predictions in [15] on the adoption of the standard mentioned that "it is likely that 802.11 will do to the Internet what notebook computers did to computing: make it mobile".

IEEE 802.11 is a member of the IEEE 802 family (Figure 2-1), which is a series of specifications for local area network (LAN) technologies. Like IEEE 802.3 (Ethernet) and IEEE 802.5 (Token Ring), the 802.11 standard focuses on the two lower layers (L1 and L2) of the OSI/Internet reference model. This is the Data Link Control Layer (i.e. DLC or DLL layer 2), further divided into Logical Link Control (LLC) and Medium Access Control (MAC) sub-layers. 802.11 defines

Physical layer (PHY) transmission schemes (layer 1), and the MAC protocol, but no LLC functionality. For LLC, the 802.11 system may rely on general protocols that are usable with all 802 standards. In a few words, IEEE 802.11 defines the MAC/PHY for WLAN technologies.

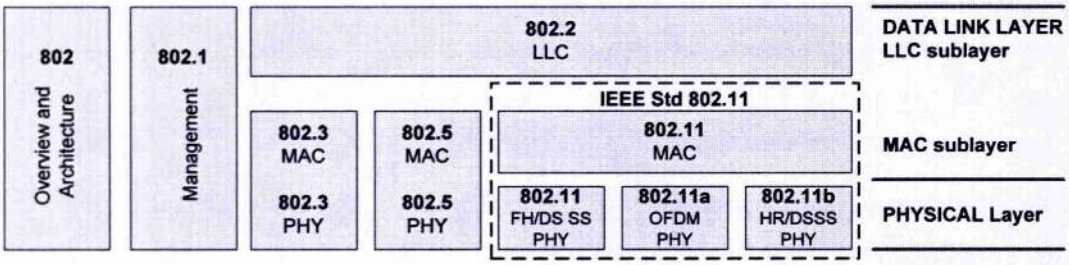

Figure 2-1. 802 Protocol Family, Standards and Layers

Since 1997 when the standard became public, several revisions and incremental updates were added. Most important milestones of the standard include its first version of "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ratified as IEEE *Std 802.11-1997*. This version was superseded in Jun.1999 with *IEEE Std 802.11-1999*, which was reaffirmed and endorsed by ISO/IEC in Sep.2005. The *802.11-1999* version is the core standard which remains in force, subject to a number of Amendments. The core standard is also referred to as *ANSI/IEEE Std 802.11-1999(R2003)* or *ISO/IEC 8802-11: 1999*. The latest standardised version is *IEEE Std 802.11-2007*, which is a "Standard Maintenance & Revision" of the 1999 version.

The various sub-standards or Amendments of 802.11 define improvements of MAC/PHY methods as well as different management extensions. Versions 802.11b (1999, Amend.2) and 802.11g (2003, Amend.4) operate in 2.4GHz ISM frequency band and support maximum data rates up to 11Mbps and 54Mbps respectively. Newer version 802.11g maintained backward compatibility with previous version 802.11b and to date is widely used in Europe. Version 802.11a (1999, Amend.1) was standardised and operates in 5GHz frequency band supporting maximum data rates up to 54Mbps respectively. Version 802.11a has been more popular in Americas, since initial Spectrum Regulation issues prevented the use of 5GHz band in Europe. A newer amendment (802.11h-2003, Amend.5) addressed "*Spectrum and Transmit Power Management Extensions in the 5 GHz Band in Europe*". Another anticipated amendment is 802.11n (currently in pre-standard status) and adds support for multiple-input multiple-output (MIMO) antennas for higher throughput. In spite of not being finalised, products compliant with draft v.2.0 became available in June 2007, under the certification and endorsement of the Wi-Fi® Alliance.

The tremendous popularity of 802.11 has been linked mainly with the convenience of deploying wireless access points (AP) in public spaces, airports or homes, and through them easily providing

wireless Internet for connected hosts. This is the primary operation mode of 802.11, named Basic Service Set (BSS) and typically used in wireless home networks and small WLAN deployments. It implies a single AP that advertises a specified Service Set Identifier (SSID), i.e. a name for the wireless network, which wireless hosts can recognise and use to connect. The name and other MAC/PHY information necessary to synchronise hosts are included in special 802.11 management frames (*beacons*), periodically emitted by the AP. An Extended Service Set (ESS) creation is also possible, where interconnected BSS, use the same network name (SSID) to provide transparent BSS handoffs to connected users, i.e. allow users to roam transparently between different APs. BSS mode and extensions are sometimes referred to as Infrastructure BSS, but they should not be confused with Independent BSS (IBSS) mode.

Independent Basic Service Set (IBSS) mode is the second mode of operation for 802.11 hosts, defined in 802.11 standards as *ad hoc* mode [183]. It is also referred as *peer-to-peer* mode by the management software for user devices. In this least known mode, no infrastructure AP exists and all nodes execute the same operations in a distributed manner. The first of the ad hoc devices to initiate communication assumes the role of a limited AP, advertising in beacon frames the properties of the new ad hoc network, like its name (SSID) and connectivity parameters, the beaconing interval and any encryption methods used. Nearby wireless devices that can hear the beacons can connect to the ad hoc network in a peer-to-peer manner, i.e. establish single hop wireless links with their neighbours that use the same SSID. Participating ad hoc nodes also use a built-in distributed algorithm to periodically rotate the AP role and emit beacons [183]. Obviously, IBSS mode of operation does not imply or assume any multihop behaviour, but as said ensures per hop wireless connectivity at MAC/PHY layers. The widespread availability of 802.11 devices has made IBSS/ad-hoc mode quite popular. In most cases, initial MAC/PHY configuration is arbitrarily set at the initiating device, by adopting default software driver and/or hardware dependent parameters. The use of "default" settings can lead to interference and performance degradation in the cases of simultaneous collocated network deployments, due to channelisation issues explained below.

An interesting series of real life field measurements is available in [156], based on the assessment of urban Wi-Fi® network deployments (wardriving) in various cities. In spite of mentioned drawbacks of ad hoc, field measurements have identified that 10% of connections worldwide are in ad hoc mode (IBSS), while the rest are conventional connections to infrastructure Access Points (BSS/ESS). Moreover, in cases of wireless access at large IT events, the percentage of ad hoc connections rose to 50% at Infosecurity'06 exhibition in London and to 42% at CeBIT'06 trade show in Hannover. These surprisingly high percentages verify the popularity of spontaneous and temporary ad hoc communication of wireless devices in a peer to peer manner. Popularity is

attributed to the convenience and self-directed deployment offered by ad hoc mode. These facts indicate the significant scope for *wireless ad hoc networks*.

Another important observation from aforementioned field measurements is that the vast majority of measured WLANs were using the "default" vendor settings regarding the deployment channel, leading to overcrowding of those channels. Although the geographic proximity of deployments is not mentioned in [156], the practise of using default settings is obvious. This practise may work for geographically isolated networks, but in cases of collocated network deployments it can lead to interference and performance degradation [172]. The selection of those default channels by equipment vendors is related to the *channelisation* of 802.11 technologies and vendor recommendations use of "non-overlapping" channels to avoid interference [93].

Interference is a major issue in WLAN, especially when deployed in unlicensed bands. For example, 802.11b/g technology defines 13 channels in the 2.4GHz ISM band, with centre frequency separation of only 5 MHz and overall channel frequency occupation of 22 MHz , as shown in Figure 2-2 (channel 14 has also been defined for use in Japan only). Due to frequency overlap between consecutive and nearby channels, interference may occur between channels because of the small frequency spacing (inter-channel interference). Recommended deployments in the Federal Communications Commission (FCC) region use three non-overlapping channels (1,6,11) [93] as shown in Figure 2-2.

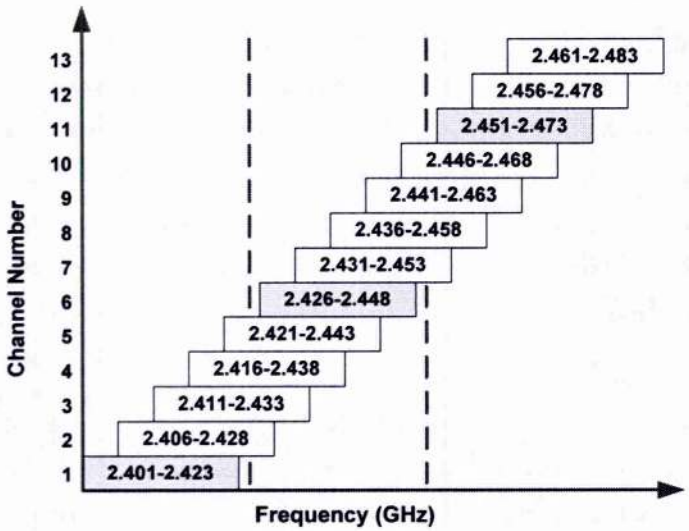

Figure 2-2. Defined Channels and Spacing for 802.11 in 2.4GHz ISM band

This explains the results presented in [156], where more than 50% of recorded WLANs were deployed on Channel 6 and about 25% on Channel 1. However, researchers have shown that interference is still noticeable even when "non-overlapping" channels are used in dense WLANs [74]. This can be explained because of the proximity of most devices which results in the near-far

effect. The near-far effect is encountered due to 802.11 MAC/PHY operations that aim to achieve fairness in channel throughput and utilisation based on channel sensing measurements (CSMA/CA) [75]. In addition, it should be noted, that the central frequency of Channel 11 (2.462GHz) is very near the operating frequency of microwave ovens (2.45GHz). This explains why Channel 11, although it is considered non-overlapping, is not used as the default one to avoid interference. Some vendors offer high-end AP which automatically switch channel with proprietary algorithms if they identify intense interference. Due to the proneness of 802.11 to inter-channel interference, researchers have also suggested dynamic channel assignment and selection algorithms [72],[73].

2.4 Policy-Based Management (PBM)

Policy-Based Management (PBM) [17][18] and policies have been envisioned as encapsulating business objectives which in turn are autonomously applied to managed systems, requiring minimal human intervention. However, practise has shown that what was initially conceived as the instant panacea of network management is in fact a long journey towards self-managing networks, hampered by severe obstacles. The views published in [16] by a major infrastructure vendor are illustrative of initially overestimated expectations from policies: "to many people, it suggests that, by some magic, you get something for nothing, or at least without needing to think through what needs to be precisely done to achieve those objectives. Of course, there is no magic, and anyone expecting magic is bound to be disappointed". Beyond initially high expectations, research on PBM has gradually verified its enormous potential and showed that it can simplify complex management tasks of large-scale systems. The concept of high-level policies monitoring the network and automatically enforcing appropriate actions has received intense interest and has been fuelled by the renewed interest in Self-Management and Autonomic Networking [17],[18],[94],[96],[97],[98],[99], [173] ,[192].

In general, policies can be defined as Event-Condition-Action (ECA) clauses, where on event(s) E, if condition(s) C is true, then action(s) A is executed. Different definitions and classification of policies can also be found in the literature and are presented later. The main advantage which makes a policy-based system attractive is the functionality to add *controlled programmability* to the managed system, without compromising its overall security and integrity [96],[97]. Real time adaptability of the system can be mostly automated and simplified by the introduction of the PBM paradigm. According to [96],[97], *policies can be viewed as the means to extend the functionality of a system dynamically and in real time in combination with its pre-existing hard-wired management logic*. Policies offer the unique functionality to the management system of being re-programmable and adaptable, based on the supported general policy types. Policies can be

introduced to the system and parameterised in real time, based on management goals and contextual information. Policy decisions prescribe appropriate actions on the fly, to realise and enforce those goals.

A block diagram of PBM functional elements is shown in Figure 2-3, using a simplified UML notation of their relationships. These four elements constitute IETF's policy-based framework, as proposed through the work of the Policy Framework WG (POLICY) [192],[206],[207] and the Resource Allocation Protocol WG (RAP) [193][202]:

- **Policy Management Tool (PMT):** the interface between the human manager (e.g. a consultant or network administrator) and the underlying PBM system..
- **Policy Repository (PR):** the blueprint of policies that a PBM system is complying with at any given moment. In essence, it encapsulates the operational parameters of the network and therefore it is one of the most critical elements.
- **Policy Decision Point (PDP):** a logical entity that makes policy decisions for itself or for other network elements that request such decisions. These decisions involve on one hand evaluation of policy rule's conditions and on the other hand dealing with the actions' enforcement when conditions are met.
- **Policy Enforcement Point (PEP):** a logical entity that enforces policy decisions. Traditionally, the sole task of PEP is to execute policy decisions, as instructed by the controlling PDP.

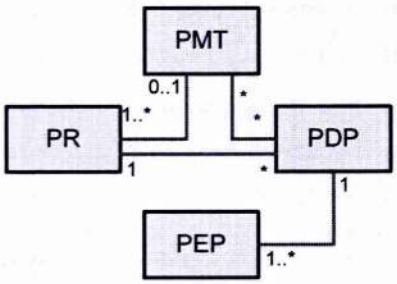

Figure 2-3. PBM functional elements

The IETF framework is widely used and accepted in research and industry and has served as a reference model for PBM systems [17],[18],[98],[99]. The operation of a Policy-Based Management (PBNM) system is outlined here: Managing Entities using a Policy Management Tool (PMT) to introduce and store policies in the Policy Repository (PR). The PR is a vital part for every policy-based system because it encapsulates the management logic to be enforced on all networked entities. Stored policies can be subsequently retrieved, either by Policy Decision Points (PDP) or by another PMT. Once relevant policies have been retrieved by a PDP, they are

interpreted and the PDP in turn provisions any decisions or actions to the controlled Policy Enforcement Points (PEP).

2.4.1 Policy representation and specification languages

The system representation of policies and the policy specification language used are two important issues for PBM design. Both can be affected by the definition of an appropriate information model, which can provide the common ground for identifying managed objects as well as representing policies. Joint standardisation efforts from IETF and DMTF had focused on the development of an Information Model rather than a formal language for policy specification. This has allowed the establishment of a technology-independent common ground for policy specification, which has also provided the established functional policy-based architecture. IETF's PCIM (Policy Core Information Model) and DMTF's CIM (Common Information Model) remain widely used because of these attributes [17],[18],[102].

From the historical perspective of [102], Clark's *policy term* (1989) is considered one of the first attempts for network policy specification. Since then, many policy specification languages have been introduced from academia and industry with varying support and impact. The Ponder policy language and toolkit from Imperial College has been among most prominent academic efforts. Notable research efforts from industry include the "Policy Definition Language" (PDL) from Bell Laboratories (1999) [103], the "eXtensible Access Control Markup Language" (XACML) from OASIS (2003)[178] and the "Autonomic Computing Policy Language" (ACPL) from IBM Research (2005) [174]. Among the very few efforts dedicated to MANETs, the PECAN framework and policy specification language were introduced in [59]. PECAN initially stood for *Policy-Enabled Configuration Across Networks* and was a CORBA-based military-oriented framework to support hierarchical management structures and to administer policy operations for MPLS traffic management in tactical networks. In the context of the DRAMA project, PECAN meant *Policies using Event Condition Action Notation* [9][53][54] and maintained a simplified ECA policy specification inspired from PDL. A comparison of PECAN with Ponder can be found in [9]:§3.3, pp.96.

The Ponder toolkit from Imperial College has been a popular PBM suite and has been developed over a period of 10 years [107]. It was among the first general purpose software tools supporting policy-based concepts and offered an open source implementation. Ponder is a declarative, object-oriented language that can be used to specify security and management policies. It does not rely on an information model to define policies; instead a formal grammar is introduced and policies must comply with it [107][108]. Ponder has four basic policy types: authorisations, obligations, refrains and delegations; as well as three composite policy types: roles, relationships and

management structures that are used to compose policies [108]. The Ponder framework integrated a centralised Domain Service based on LDAP technology to store groups of managed components, as well as software implementing the actual policies. In addition, a generic asynchronous notification service called Elvin [109] was integrated to support event-based policies. Elvin was primary designed as a middleware for distributed systems. Concepts of CIM as an extensible information model were used in combination with Ponder policy language to manage DiffServ domains [110] and to demonstrate a mapping of ECA policies from CIM to Ponder [111]. Recently, Ponder2 was also released, integrating a new high-level language named PonderTalk, used to control and interact with managed components and self-managing entities [112],[113].

Returning to standardisation efforts, the output of IETF's Policy Work Group [192] was a series of RFC documents, defining the *Policy Core Information Model* (PCIM) [204] and its extended version, PCIMe [207]. These efforts were driven by the combined work of IETF and DMTF. In addition, the model has been further extended and standardised in *Policy QoS Information Model* (PQIM) [210]. The defined information models are conceptual vendor-independent models for representing and organising policies across a spectrum of technical domains. Their purpose is to provide a consistent definition and structure of data (including policies), using object-oriented techniques. These models define policy classes and associations sufficiently generic to allow them to represent different policies [148]. Although IETF models (PCIMe, PQIM) were technology independent regarding their system representation, IETF has standardised their mapping guidelines to the LDAP [213] Data Model, describing their schema definitions in [211],[212]. After the conclusion of IETF's Policy Framework WG (2004) and Resource Allocation Protocol WG (2005), DMTF continued the development of newer versions of the information model, referred to as the Common Information Model (CIM) [180]. CIM is composed of a Specification (v.2.4, Nov.2008) that details integration with other management models and a Schema (v2.18 Apr. 2008) that provides the actual model descriptions. New concepts introduced have further extended the initial PCIMe model (based on CIM v2.2) into CIM Policy Model (v.2.13,Sep.2006).

It should be outlined that IETF did not define a policy specification language but implicitly provided a generic specification of policy rules through PCIM. This specification is in the form of: *if<condition>then <action>*, and defines a *policy as a set of rules to administer, manage and control access to network resources* [204]. IETF's policies can have some additional functionality like policy roles, grouping and prioritisation, which are defined in the PCIMe version [207]. Models for application-specific areas may extend PCIMe or CIM Policy Models in several ways. Recent work of DMTF has produced the CIM Simplified Policy Language (CIM-SPL) [181] aiming to provide a means for specifying *if<condition>then <action>* policy rules to manage computing resources using constructs defined by the CIM Policy Model and Schema. The design

of CIM-SPL was inspired by existing policy languages and models including PDL [103], Ponder [107] and ACPL [174]. A missing element from IETF's PCIMe solution is an explicit triggering mechanism which would allow event representation and would make the system event-driven. This is important in a policy-based system, since the generic policy rule *event-condition-action* is more powerful and widely accepted [17],[18],[102]. Work in DMTF's CIM Event Model [182] (since CIM Schema v2.9) suggested a triggering mechanism that could be integrated with the CIM Policy Model. "Rule triggering" events and a special query language, named WBEM Query Language (WQL) are under research within CIM Event Model [182]. Using CIM models and CIM-SPL, event-based or unsolicited policy evaluation can be provided implicitly by the instances of CIM Indication classes [181],[182]. DMTF's CIM and WBEM framework are provided as open source software through the OpenPegasus development project (www.openpegasus.org).

Importance of Information and Data Models

PCIMe [204],[207] provides a vendor and language independent way to represent policies. Use of such standards allows flexible and extensible policy modelling, regardless of the implementing technology. A part of PCIMe class hierarchy is shown in Figure 2-4.

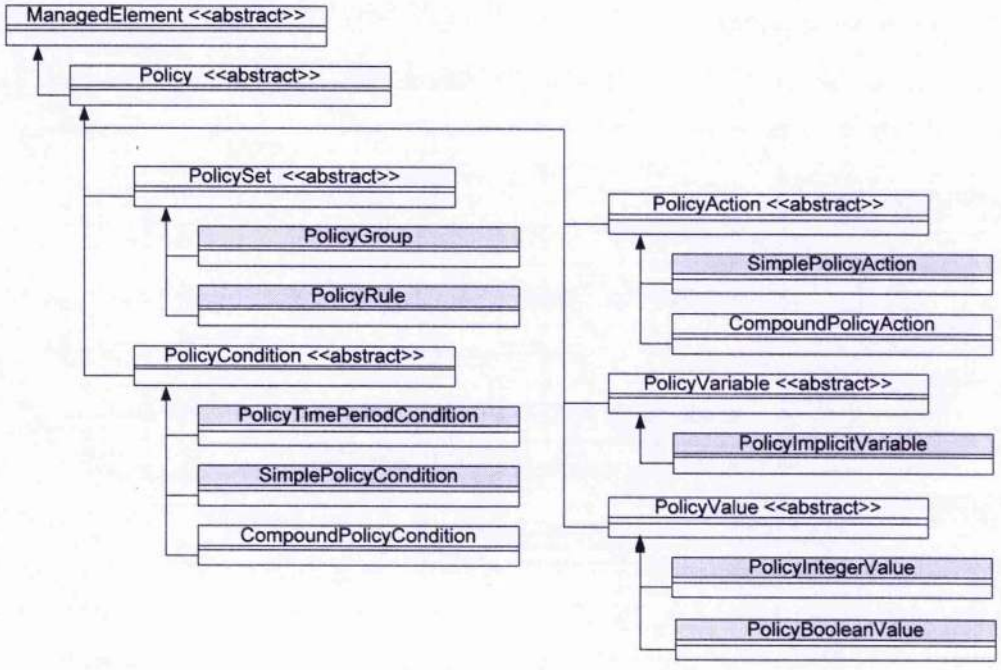

Figure 2-4. Partial hierarchy of PCIMe classes

IETF recommends the use of LDAP as the implementation technology for policy system representation and storage. A brief overview of LDAP technology is provided in Appendix B. The mapping between the PCIMe Information Model to the LDAP Data Model is guided by two IETF

RFC Standards Track documents: Policy Core LDAP Schema (PCLS) [211] and Policy Core Extensions LDAP Schema (PCELS) [212]. The collection of all "objectclass" and "attribute" LDAP definitions constitute the LDAP schema that a Directory Server uses to verify directory entries. These RFC also provide guidelines on extending these schemas, in order to include new custom classes. The LDAP Schema is an interoperable format for the required Data Model that is widely supported by LDAP Directory Servers (e.g. OpenLDAP, Fedora Directory).

The PCIMe [204],[207] model defines two hierarchies of object classes. *Structural* classes encapsulate information for representing and controlling policy data, while *relationship* classes indicate how instances of the structural classes are related to each other. Therefore two types of mappings can be performed:

- For the structural classes in the information model, a one-to-one mapping is defined and information model classes map to LDAP classes, while information model properties map to LDAP attributes.
- For the relationship classes in the information model, different mappings are possible. Classes and their properties are mapped in three ways: to LDAP auxiliary classes, to attributes representing distinguished name (DN) references, and to superior-subordinate relationships in the Directory Information Tree (DIT)

The mapping of specific PCIMe classes (e.g. pcimGroup and pcimRule) is designed to be as flexible as possible. For this reason, three LDAP classes are defined by IETF for each of these classes:

- An abstract superclass is defined that contains all required properties of each PCIMe class (pcimGroup, pcimRule)
- The abstract class is subclassed as a structural class that can be instantiated independently (pcimGroupInstance, pcimRuleInstance)
- In addition, an auxiliary class is also subclassed for use as an attachment to structural entries (pcimGroupAuxClass, pcimRuleAuxClass)

According to object-oriented design (OOD) principles, an abstract class cannot be instantiated, but includes the properties to be inherited to its subclasses. The structural subclass is the main class for instantiating the required class, as it can be deployed in a stand-alone manner within the Directory Information Tree (DIT). On the contrary, auxiliary classes cannot be instantiated independently, but instead are attached to existing structural classes of any type. This provides maximum flexibility for an LDAP designer and implementer. A part of the defined PCIMe LDAP hierarchy (schema) is shown in Figure 2-5 The parent class of all LDAP entries is top defined in X500 [186], while dlm1ManagedElement is defined in CIM LDAP schema [180].

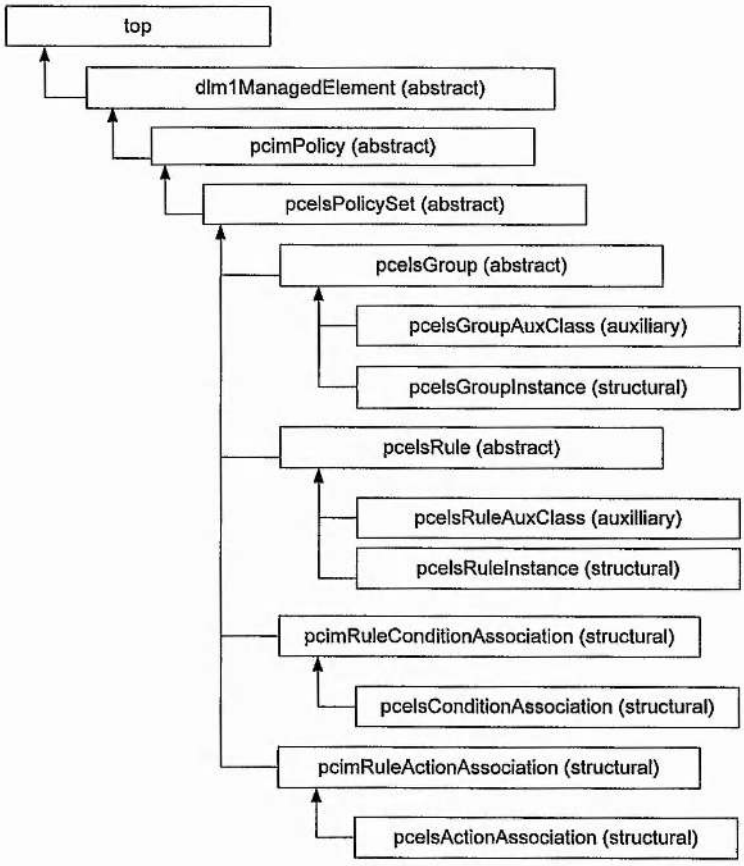

Figure 2-5. Data Model (PCELS) Class Inheritance Tree

Roles and conflict analysis

Roles and policies have a close relationship, stemming from Role-Based Access Control (RBAC) security mechanisms. The adoption of roles for policies is extended beyond security and access control issues associated with the definition of RBAC policies. Roles have been adopted as a method to collectively group policies according to the responsibilities pertaining to the role of a physical person (e.g. network manager), a process (e.g. user login) or a network component (e.g. border router). The extensive use of *roles* with *domains* [107][108] was among the main innovative aspects of the Ponder policy specification language, i.e. using domains as hierarchical collections of objects explicitly grouped together for management purposes and assigning those collections to respective roles.

Roles and high-level goals are particularly useful in complex management systems and dramatically assist in simplifying and abstracting management operations. However, the translation of those goals to low-level policy specification is an important open research topic. Policy refinement is the process of deriving a concrete policy specification from higher-level objectives or goals [150]. It is an important process that leverages the potential of PBM frameworks, therefore it has received significant research interest, aiming to provide automated

solutions [114][115][116]. The process is further hampered by the risk of producing inconsistent policy specifications, giving rise to concerns about policy conflicts and the need for policy analysis.

The need for policy analysis and the lack of tested solutions is one of the main drawbacks of policy-based systems. Policy analysis [117] refers to the examination of policies and the verification of their current and future consistency. In complex environments where a number of policies need to coexist, there is always the likelihood that policies may conflict, either because of a specification error or because of application-specific constraints. It is therefore important to provide the means of detecting conflicts in the policy specification [117],[120]. Generally speaking, conflicts can be detected as inconsistent policy parameters or actions. A classification of policies can be found in [117]. Conflicts can be generally classified as dynamic and static. A number of static conflicts may arise during policy specification, like modality conflicts, conflicts of duties and multiple manager conflicts. As an example, specifying the execution of mutually exclusive policy actions at the same time is apparently a conflict (modality conflicts, [117]). Another conflicting situation may arise when the sets of managed objects affected by the actions of different policies overlap. When these policies are provided from multiple managers with semantically incompatible goals, then there is a potential conflict for overlapping objects (multiple manager conflicts) [117]. Conflict Detection and Resolution (CDR) is an active research area of PBM and different approaches have been proposed to address aforementioned issues. Special rules can be used to recognise conflicts in the policy specification. These rules usually come in the form of logic predicates and encapsulate application-specific data and/or policy information as constraints. Examples on how these rules can be used as part of a detection process can be found in [118]. Dynamic policy analysis and conflict resolution is proposed in [119], showing how event calculus can be used to detect run-time conflicts and providing an approach for rule specification to automate conflict resolution. The authors of [119] implemented their approach in a case study for QoS management.

2.4.2 Distributed policy storage, provisioning and enforcement

The architecture of PBM systems is predominantly based on a centralised or hierarchical paradigm, following the organisation of the managed networks. As a result, the majority of PBM functionality and protocols follow these paradigms, e.g. the manager-agent model for policy provisioning and the centralised policy repository storage. To enable distributed PBM, the coordination of multiple policy decision points (PDP) needs to be addressed in combination with decentralised policy storage and provisioning.

Recently, the emergence of highly distributed computing systems (grids) has motivated the decentralisation of policies and their distributed management. Departing from centralised PDPs deployment, the distributed control of multiple PDPs has been investigated in [100], providing a conceptual model for their coordination. Issues of distributed or centralised decision making were examined, defining policy elements that can control coordination, and rules for the refinement of coordination policies. Distributed management of policies for *Grid* networks has been investigated in the context of the *Globus Toolkit* [101]. An authorisation framework was investigated to provide support for multiple security policies from different autonomous domains. This work has been targeted on Grid systems, aiming to coordinate distributed access control lists and distributed provisioning of defined policies.

Policy provisioning is the process of communicating policy decisions and directives between a Policy Decision Point (PDP) and a Policy Execution Point (PEP) using a suitable protocol [202][206]. A PDP is also known as a *policy server*, reflecting its responsibility to serve a number of PEP with policy decisions and relevant PBM information. On the other hand, PEP are also known as *policy clients* since their operation depends on these decisions, as provided by their parent PDP. The protocol involved in this communication is the *policy provisioning protocol.* Efforts from IETF's Resource Allocation Protocol Working Group (RAP WG) have produced the COPS (Common Open Policy Service) Protocol [201] and COPS protocol for Policy Provisioning (COPS-PR) [205]. COPS is a simple query and response protocol that can be used to exchange policy information between a policy server (PDP) and its clients (PEP). The basic model of interaction between a policy server and its clients is compatible with IETF's policy-based framework. The focus of IETF's efforts has been mainly to provide a protocol to carry out the task of policy provisioning mostly related to QoS parameters and setup. In academia the efforts described in [55],[56] utilise COPS-PR solely for the purpose of QoS configuration for MANETs. Furthermore, different architectures have introduced dual node functionality [97], where each managed device acts both as a PDP and as a PEP, thus making the usage of COPS unnecessary.

In spite of protocol drawbacks, the concepts behind COPS have found general acceptance as policy provisioning principles. Specifically, the interaction between PEP and PDP can be done based on two models, stemming from the definition of IETF's COPS protocol and relevant IETF terminology [201],[205]:

1. Outsourcing model: "an execution model where a policy enforcement device issues a query to delegate a decision for a specific policy event to another component". This external component is the parent PDP of the requesting PEP.
2. Provisioning model: "an execution model where network elements are pre-configured, based on policy, prior to processing events. Configuration is pushed to the network

device, e.g. based on time of day or at initial booting of the device. The focus of this model is on the distribution of configuration information".

These two models are often contrasted, based on their traditional applicability to different QoS paradigms. The *outsourcing* model has been proposed for use with RSVP and Integrated Service (IntServ, RFC2210). For example, the arrival of a new RSVP message to a PEP requires a fast policy decision (to avoid delaying the end-to-end setup). The PEP may use COPS to send a query to the PDP, asking for a policy decision. On the other hand, the *provisioning* model is used with Differentiated Services (DiffServ, RFC2475) where based on the events received, devices (PEP) use downloaded (pre-provisioned) mechanisms to implement QoS policies. However, the two models are not mutually exclusive and PBM systems may combine both. In RFC2753 [197], the concept of a Local PDP (LPDP) is introduced, where a provisioned PEP is able to make local decisions. The requirement was that partial decisions and the original policy request needed to be sent to the PDP which would render a final decision, possibly overriding LPDP. Hence, the PDP acts as the final authority for decisions applying to PEP and PEP must enforce the decision [197].

Beyond COPS, no other dedicated policy provisioning protocol has been standardised by the IETF and policy provisioning has been viewed under the general umbrella of configuration management protocols. Traditional management protocols (SNMP) and interfaces (command line interface) are in use to carry out policy provisioning in an application-dependent manner. PBM frameworks based on Java (e.g. Ponder) have used Java RMI (Remote Method Invocation) to carry out provisioning. However, having in mind their deficiencies [23],[24],[25] and the need for interoperable standards, both the research community and industry have been moving towards XML-based management protocols. The trend towards Web Services and XML/HTTP-based management has also affected PBM [129],[130],[178].

The concept of Remote Procedure Calls (RPC) has been integrated to middleware and distributed management approaches (e.g. CORBA [175], XML-RPC [157]). It is also being used for interoperable management operations based on Web Services [176] using SOAP [177]. XML-RPC [157], as the lightweight precursor of SOAP, has found acceptance in resource-contained portable devices. Its main requirement is HTTP/XML processing capability, which is available on the majority of networked devices. Its compact specification and minimum device requirements have supported its wide use on portable devices as an interoperable, easy to extend and easy to deploy middleware platform [104],[105].

Management middleware and policy provisioning are tightly related to Policy Object Management. Based on RFC2753 [202], definitions of *Policy Object* and *Policy Element* are provided below. Object Oriented terminology is used in parallel with policy-based terminology, clarifying terms *Object*, *Class* and *Instance*:

- *Object*: the general representation of data and methods.
- *Class*: the static representation of a collection of objects.
- *Instance*: the runtime representation of a class, initialised during execution.
- *Policy Object (PO)*: represents policy-related information, such as policy elements, and is carried in a request or response related to a decision.
- *Policy Element*: subdivision of policy objects, containing single units of information necessary for the evaluation of policy rules.

In the Ponder framework, PO are organised and managed in *domains*, following a hierarchical organisation [108]. The drawbacks of a strict network hierarchy are inherited by POs, mainly with the creation of policy decision bottlenecks and a single point of failure. On the other hand, hierarchical domains have been useful for grouping PO related to particular roles or device types and have assisted in delegating responsibility. Research has shown that PDPs are common bottlenecks of traditional PBM systems, since normally they have to provision and control large numbers of PEPs [100]. Finite state machines and automata have been employed for managing PO in state-full PBM systems, controlling their state transitions (e.g. DEN-ng [136], FAIN [169]).

Policy Enforcement Issues

Important PBM issues are related to the decision making process and the enforcement of policies in the network. One has to consider whether the enforcement of policies needs to be uniform or choice will be given to nodes. According to the IETF's architecture, final policy decisions are made at a PDP and policy enforcement is expected to be uniform [202],[206], i.e. all nodes conforming to same policies. However in a user-created wireless ad hoc network this is not necessary [63], since the purpose and formation of such networks is different from fixed ones. An important issue emerges, regarding whether the policies should apply to all users and how their preferences are respected.

Recent concepts on policy enforcement were introduced in [63] to allow network nodes to partly conform to a global policy set. In [63], cases are examined where no absolute control from an authority is accepted, discussing whether all policies should apply to all users and how their preferences should be respected. In [122] a "promise theory" attempts to provide "political autonomy" to entities and decentralise policy management. Such requirements significantly increase the system's complexity. On the other hand, these concepts can be used to address the users' demand to control owned devices and the need to respect their privacy. In the European Union for example, strict legislation by the European Data Protection Supervisor (EDPS) mandates the processing and acquisition of personal data (Directive 95/46/EC, edps.europa.eu). National authorities have been established to monitor their enforcement, for example the

Information Commissioner's Office (ICO,ico.gov.uk). In spite of regulatory directives, consumers remain increasingly concerned with the acquisition and exploitation of their personal data.

Policy Storage and Distribution Issues

The existence of a policy repository (PR) in PBM architectures requires an efficient policy storage implementation. Typical implementations of a PR are based on Lightweight Directory Access Protocol (LDAP) Servers (RFC 4511, LDAPv3, [213]), also known as Directory Servers (DS). A Directory Server including its directory content (i.e. policies) is simply referred to as a *Directory*. As already mentioned, IETF standardisation efforts have specified an LDAP schema to represent PCIMe policies and vendor-specific extensions.

LDAP [123],[213] was designed to provide access to the X.500 Directory [186] without incurring the resource requirements of Directory Access Protocol (DAP) [186]. LDAP is specifically targeted at management applications that provide simple read/write interactive access to Directories. The reasons for the dominance of LDAP as a policy repository are some of the useful features it has to offer. The object-oriented design and implementation of a Directory using LDAP, makes storage of policy objects very convenient and easy to access [211],[212]. The offered operations/services (e.g. search, modify, add etc.), combined with filtering and authentication capabilities, can be used in a natural way for policy retrievals, modifications and look-ups. Furthermore, the capabilities to distribute and/or replicate the directory among network nodes make it very versatile. The LDAP directory can be distributed on several physical nodes by utilising its inherent replication capabilities. Finally, LDAP has sophisticated built-in security mechanisms that can provide various levels of access control for contents retrieval and for directory management purposes. On the other hand, it should be noted that LDAP technology is optimised towards frequent search and look-up operation rather than updates and modifications. These limitations should be considered in combination with the frequency of policy modifications [123],[124],[125],[126],[127]. On another perspective, XML-based solutions have also been considered as an alternative to LDAP for storing policies, in spite of XML's verbosity [128],[129],[130]. The reasons are the significant penetration of XML in several devices and systems and its wide support as a uniform and interoperable format for sharing and representing data. Relational databases have also been used for a Policy Repository, as mentioned in [54]. A database server based on MySQL stores policies and configuration data on every MANET node, using a proprietary format.

Although a PR is a centralised concept, various techniques exist to physically distribute its contents. The reasons for distribution are obviously resilience and load balancing [99],[100], [173]. A single point of failure would make policy-based systems vulnerable; therefore features of DS are often exploited, e.g. content synchronisation operations [215], multi-master replication

[131]. A commercial solution for a distributed LDAP Directory has been proposed [132] to support large-scale PKI (Public Key Infrastructure) deployments. Large-scale distributed repositories of digital content have also been deployed (www.dspace.org), based on relational databases. Distributed database management has been extensively studied [21] and distribution of directories follows similar concepts [19][132]. For deployments over wireless networks, there is a significant differentiation of requirements. This has led to different approaches and algorithmic solutions for replica management, as already described in §2.3.3, pp.26.

2.5 Self-management and the Autonomic Paradigm

Self-management refers to the ability of independently achieving seamless operation and maintenance by being aware of the surrounding environment [7]. It has been closely related with autonomic computing and self-maintained systems [133],[134]. This ability is widely embedded in the natural world, allowing living organisms to effortlessly adapt to diverse habitats. For example, the ability of warm-blooded species to regulate their body temperature. Without planning or consciousness, body's mechanisms work in the background to maintain a constant temperature. To imitate nature's self-managing abilities and apply them to the management of network and systems, the latter should be provided with the logic and directives for their operation and in addition the means to sense their operating environment. Self-management has been closely related with control systems and particularly to closed-loop controllers (Figure 2-6). By using a system's output as feedback, a feedback loop allows the system to become more stable and adapt its actions to achieve desired output. From the definitions above, it is evident that two main functions are required to support *self-management*. These two functions are interrelated and interdependent, thus forming a closed control loop with feedback as shown in Figure 2-6:

A. Provide the logic and directives to achieve seamless operation and maintenance.

B. Provide the means to sense and evaluate their operating surrounding environment.

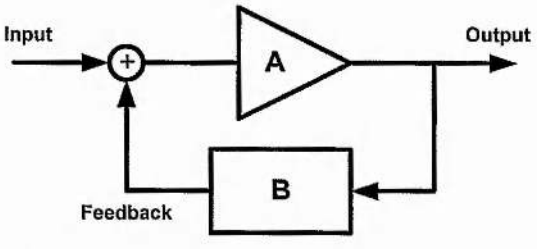

Figure 2-6. Closed control loop with feedback

In 2001, an influential research declaration from IBM had introduced the concept of *Autonomic Computing*, which encapsulated the aspects of self-management in an architectural blueprint [133]. The concept was inspired by the ability of the human nervous system to autonomously

adapt its operation without our intervention and has appealed to researchers worldwide. IBM's vision [134] has fuelled intense research efforts both in industry and academia. In essence, autonomic computing and self-management are considered synonymous. According to IBM, "*autonomic computing is a computing environment with the ability to manage itself and dynamically adapt to change in accordance with business policies and objectives.*" In addition, four quintessential properties of a self-management system were identified [134], frequently referred as self-* or self-CHOP properties:

- Self-Configuration
- Self-Healing
- Self-Optimisation
- Self-Protection

Self-management concepts are increasingly used in research [135] following the introduction of the *autonomic manager* (AM) component, as proposed by IBM [134]. Major IT and Telco players are showing their research interest in autonomic networking and self-management, e.g. Motorola in [136] and Microsoft in [137] . In addition, intense interest is shown in *autonomic network management* from Academia [138]. The *autonomic manager* [134] architectural component has become the reference model for autonomic and self-managing systems. It is a component that manages other software or hardware components using a control loop. The closed control loop is a repetitive sequence of tasks including Monitoring, Analysing, Planning and Executing functions. The orchestration of these functions is enabled by accessing a shared Knowledge base. The reference model is frequently referred as K-MAPE or simply MAPE, from the initials of the functions it performs. The use of a feedback loop raises concerns about a system's stability and according to control theory, a "valid operating region" of a feedback loop should be specified, indicating the range of control inputs where the feedback loop is known to work well [135],[137]. Based on the definition of *autonomic management*, *policies* are identified as the basis of self-managing systems, encapsulating high-level business objectives. This direction has been clearly advocated by IBM, with the introduction of Policy Management for Autonomic Computing (PMAC) [173]. PMAC is an infrastructure that uses policy-based management to simplify the management and automation of products and complex systems. It has been supported by its own policy specification language, namely IBM's Autonomic Computing Policy Language (ACPL) [174].

The sensor-monitor functionality of *self-management* has been linked with context and context-awareness [67],[68],[134]. Different definitions and meaning have been given to these terms. Context has been defined in [66] as any information that can be used to characterise the situation

of an entity, whereas an entity is defined as the person, place or object that is considered relevant to the interaction between a user and an application. According to [67], context-awareness refers to the ability of a system to dynamically and continuously adapt its status and operation according to context. In essence, context is synonymous to information and it is this information that needs to be collected, modelled and processed to become useful Knowledge. Self-management systems can exploit *knowledge*, combined with *policies* [67],[135]. Context modelling can help to achieve context-awareness, since various context sources produce different data that have to be structured and organised under a unified representation scheme. In other words, a context model acts as a communication protocol among context-aware entities, allowing interoperable and efficient processing. Context modelling includes approaches based on the entity-relationship model, Unified Modelling Language (UML) and Ontologies [67]. Collaborative context determination for MANETs is introduced in [139], where a mobile node collects context from its neighbouring peers.

Research on autonomic systems has been intense during the past years, aiming to embed the highly desirable self-managing properties to existing and future networks. The roadmap to autonomic management [133] is indicative of a gradual evolution and can be used to evaluate a system's progress [135],[138]. Accordingly, management frameworks can advance through different maturity phases before becoming autonomic:

- Basic: manually operated management operations
- Managed: management technologies used to collect and synthesise information
- Predictive: correlation among management technologies provides the ability to recognise patterns, predict optimal configuration and suggest solutions to administrators
- Adaptive: management framework can automatically take actions based on available knowledge, subject to the supervision of administrators
- Autonomic: business policies and objectives govern infrastructure operation. Users interact with the autonomic technology tools to monitor business processes and/or alter the objectives

Apparently the road to self-management is long and a series of issues will need to be resolved on the way. Until now, a complete self-management solution is not available. Instead, researchers and practitioners have attempted to partially tackle self-management by implementing some of the desired properties and adopting a gradual transition. Each of the four desired capabilities is contributing to the overall goal of enabling truly self-managed systems.

Chapter 3

Policy-based Organisational Model for Wireless Ad Hoc Networks

3.1 Introduction

One of the most critical requirements of network management is scalability. Research and practise have shown that scalability can be enhanced with appropriate network organisation. In other words, a proper organisational model can reduce management overheads and thus increase the efficiency and responsiveness of management operations. This increases the maximum number of effectively managed nodes without saturation or system failure. For the management of wireless networks these issues are magnified, mainly because of reduced bandwidth, variable link quality, limited device resources and predominantly uncontrolled large-scale deployments. Management overheads refer mostly to generated traffic and resource utilisation caused by the management components and protocols. In wireless networks, it important to keep overheads as low as possible and one method to achieve that is by designing an organisational model that takes in mind their special requirements.

By analysing the definition for Self-Management, policies are identified as the basis of self-managing systems, encapsulating high-level business objectives. Policy-Based Management (PBM) is the major building block of the organisational model presented in this chapter and effectively of the overall Self-Management framework presented in this thesis. In this chapter, emphasis is given to the organisational aspects of wireless ad hoc networks, introducing a series of original features based on a hybrid organisational model. Beyond policy-based functionality, the key features of the presented model are listed below and will be further analysed in forthcoming paragraphs:

- *Hybrid model*, to combine the benefits of both hierarchical and distributed models
- *Role-based*, to integrate roles and policies for allocation of management responsibilities
- *Multi-manager capability*, to facilitate the interests of different managing entities
- *Hypercluster formation*, to distribute management tasks and increase robustness
- *Context-awareness*, to sense the environment and provide feedback
- *Module differentiation*, to enable various implementation and deployment scenarios

3.2 Model Overview

The effective management of wireless ad hoc networks poses diverse requirements. In this chapter, an attempt is made to tackle them by designing a novel policy-based organisational model. The main advantage which makes a policy-based system attractive is the functionality to add controlled programmability in the management system without compromising its overall security and integrity. Real time adaptability of the system can be mostly automated and simplified with the introduction of the PBM paradigm. At the same time, the managed system reports contextual information and events, providing the necessary feedback to close the control loop. A high-level system view is depicted in Figure 3-1, where the closed control loop of Figure 2-6 has been integrated as the basis for the presented policy-based organisational model.

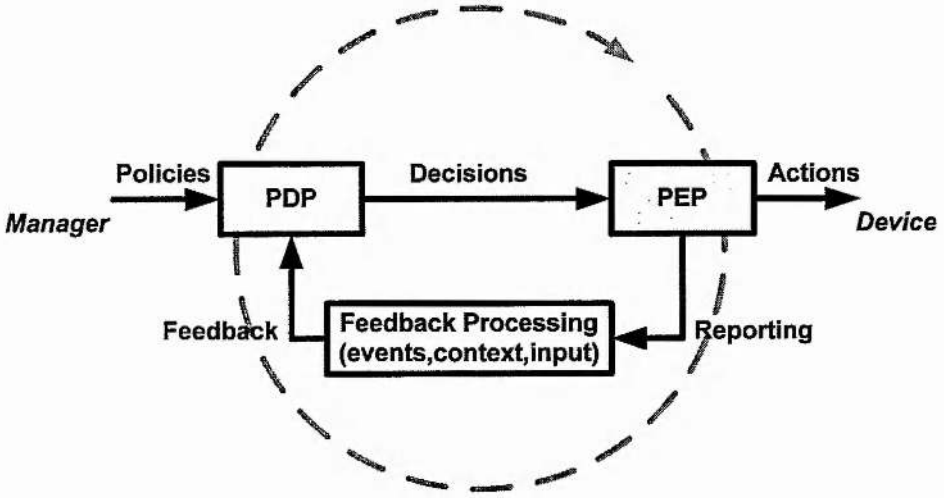

Figure 3-1. High-level view of policy-based closed control loop for self-management

Traditionally, hierarchical models are used for large-scale fixed IP networks. In such networks, over-provisioning of bandwidth and physical resources eliminates any single points of failure and traffic bottlenecks. Obviously, this solution can not be applied to wireless networks because resources are quite limited. Resources like battery power and bandwidth need to be optimally

utilised by employing a suitable model. While fully distributed organisational models are an attractive alternative for large-scale networks, e.g. P2P overlays for file sharing, such models are exceedingly resource-hungry and demand significant bandwidth for signalling and control messages to maintain such overlays in wide-area deployments. On the other hand, a combination of both paradigms in a hybrid model is promising. One of the innovative features of the proposed model is its deployment flexibility with a varying degree of distribution. Stemming from its hybrid design the model aims to offer a balance between the strictness of hierarchical models and the fully-fledged freedom of distributed ones. At the same time this model embraces both as it can be deployed as either of these.

Wireless networks have an amplified element of locality, which is evident in a wide range of applicability scenarios. For example, ad hoc networks can be formed for a corporate meeting or can be formed from an emergency response unit, responding to a confined disaster area. Bearing in mind the characteristics of wireless links, unpredictable delays and traffic flooding can be restricted if decision making is performed locally. To achieve that, a local control loop is needed, capable of provisioning the network with fast and reliable responses. By enabling clustering for management purposes, the element of locality is preserved and the requirements mentioned above are achieved. Hence, the motivation for a clustered organisational model is founded. Additional important requirements of wireless networks are the increased node heterogeneity and capabilities diversity. These issues further motivate the decision to employ a role-based organisational model, which allows natural integration with the overall policy-based system and clustering for management. By employing three different roles, distinct levels of increasing capability demands were created, able to suite nodes' heterogeneity. The defined roles are *Cluster Node* (**CN**), *Cluster Head* (**CH**) and *Manager Node* (**MN**).

A brief example explains the three different roles, while a detailed description is given in the following Section (§3.3). It is assumed that user-owned devices, like laptops or PDA, become Cluster Heads (*CH*) and form clusters that cover their nearby geographic area and include other user devices. Lightweight user devices with limited resources, like mobile phones or media players, are able to participate in the wireless network, assuming the least demanding role of a Cluster Node (*CN*). On the other hand, powerful devices can be inserted in the wireless network by the network operator and host fully-fledged management software that introduces business objectives and policies, i.e. assume the role of a Manager Node (*MN*). Multiple managers may coexist in a deployed network, based on a "*multi-manager*" paradigm. On top of clusters, a distributed management coalition forms the "*hypercluster*", including one or more privileged nodes (MN) and the local cluster managers (CH). The multi-manager paradigm and the hypercluster formation are two of the distinctive elements of the introduced organisational model.

3.3 Hybrid organisational model and role entities

Before examining the novel aspects of the organisational model, the distinction between *network formation* and *network organisation* is clarified. *Formation* refers to the purpose of the network and its deployment attributes, normally defined beforehand if a business model exists and coded in the preinstalled software of participating devices. As detailed later (§3.5,pp.69), the *formation* purpose of a wireless ad hoc network affects the algorithm to be used for its partitioning and the assignment of each participating device to a role, i.e. its *organisation*. Furthermore, *organisation* directives may allow for customisation and integration with policies and different clustering algorithms, aiming to improve network's scalability and survivability. In brief, *network formation* deals with the business model's requirements while *network organisation* deals with functional and operational requirements. Based on the above, the following paragraphs deal with *network organisation* issues, providing a configurable platform for *network formation* to suit various business models.

The basic concept behind the proposed model is the combination of a hierarchical model with a distributed one in a novel hybrid organisational model. Looking at the two extreme cases of organisational models, we have on one hand strictly hierarchical ones and on the other fully distributed ones. Each is better suited to different networks, but for the needs of wireless ad hoc networks, a hybrid approach was deemed necessary. The aim is to offer a balance between the strictness of hierarchical models and the fully-fledged freedom of distributed ones. This creates a flexible model with a variable distribution degree, able to accommodate different case studies. Beyond hybrid deployment, the proposed model embraces both paradigms and can also be deployed as either distributed or hierarchical. This is shown in Figure 3-2, using the aforementioned role types in all three models for clarity. A qualitative assessment of benefits and drawbacks of each model is provided below, justifying the decision for adopting a hybrid one. The figure depicts the implications and overheads implied by each model. Solid lines represent direct communication of management information while dashed lines represent auxiliary management information exchange.

The proposed hybrid model is based on a loose tiered hierarchy by employing distributed node clustering to achieve locality and restrict dissemination of traffic overhead. Static and dynamic cluster creation is discussed in §3.5, while further details of the algorithmic role selection process are provided in Appendix A. For the case of a hierarchical model, the elegant and strict cascading organisation is based on the centralised model of manager-agent. At first sight the model appeals as quite efficient in terms of required management information exchange. What is not shown though is the required backup infrastructure and messages for the avoidance of the single point of failure, i.e. the single Manager Node. On the other hand, the fully distributed model has no single

point of failure, but the flat node organisation requires all nodes to be involved in management and exchange significantly more control messages. The customised hybrid model for the management of wireless ad hoc networks attempts to lessen these drawbacks and combine the advantages of both. The strictness of a hierarchical model is relaxed by allowing nodes in CH and CN roles to communicate with neighbouring nodes in the same role. The apparent drawback of additional management traffic is outweighed by increased network robustness and survivability, having in mind that node disconnection and disappearance is frequent in wireless networks. Contrary to traditional management schemes for fixed networks, in wireless networks a link failure is not considered as a fault and the hybrid model aims to counterbalance the transient nature of links, by using indirect communication between nodes. For example, if a CH is temporarily disconnected from the controlling MN, it may be possible to retrieve management information from a neighbouring CH within the hypercluster. In this way, management overheads and delays imposed by a hierarchical model are avoided. In a hierarchical model, if the link between a CH and a MN was interrupted, node isolation would occur and a repair procedure to restore the link would be initiated.

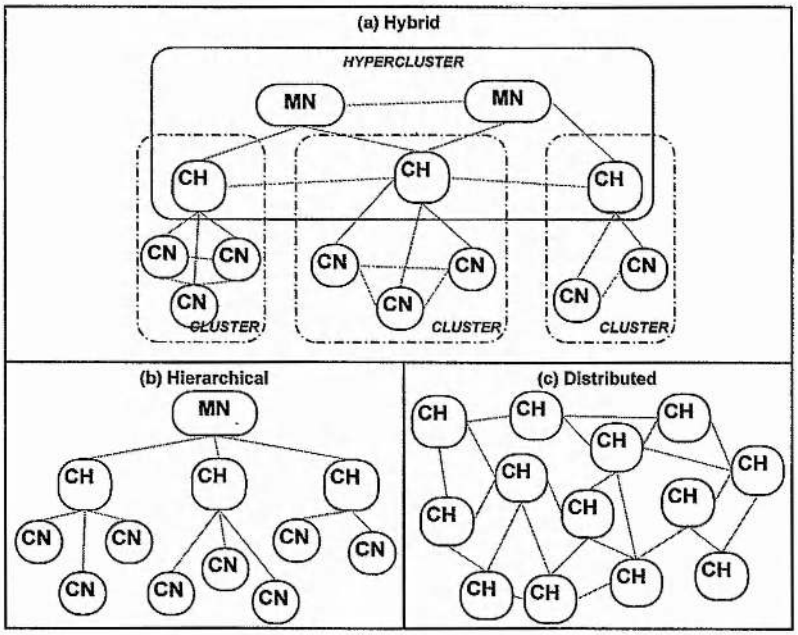

Figure 3-2. Organisation Models: (a) Hybrid (b) Hierarchical and (c) Distributed

If a similar case is assumed in a fully distributed model, node failure or disconnection are not considered a problem, since such models achieve robustness by flooding the network with management and signalling information. The relatively uncontrolled nature of this model has attracted interest in unstructured P2P networks used by different applications. The participating end-user equipment is typically a computer with broadband Internet access and continuous power supply. As a consequence, fully distributed models involve a significant traffic overhead and

resource utilisation, deeming such models inappropriate for the majority of wireless scenarios where lightweight battery-powered user devices participate and the bandwidth is an expensive, limited resource.

Wireless ad hoc networks are consisted of highly heterogeneous devices, thus motivating the decision to employ different node roles. In addition, the natural integration of a role-based organisational model with the overall policy-based design further supports this decision. Three different roles were employed; a decision that aims to create distinct levels of progressively increasing capability and responsibility demands. As mentioned, the three roles are *Cluster Node* (CN), *Cluster Head* (CH) and *Manager Node* (MN). These roles imply a clustered wireless network, borrowing some concepts from research in MANETs. Similarly, a Cluster Head is in charge of a number of Cluster Nodes, thus forming a *cluster*. Contrary to hierarchical models, the proposed model allows communication between Cluster Heads for exchanging management information and collaborative management. A number of privileged Manager Nodes can also exist, responsible for introducing the policies that express the overall management objectives, realising a multi-manager paradigm. Manager Nodes and Cluster Heads create a cluster with higher hierarchy level, referred to as the "*hypercluster*".

3.3.1 Policy and context interaction

The concepts of policy and context interaction are detailed here. To close the feedback loop, the main policy-based components require an appropriate reporting mechanism. Context-awareness and context can provide an elaborate reporting mechanism and policies can exploit context information as policy events and conditions parameters. With the introduction of context-aware counterparts to the standard functional elements of a policy-based system, their interaction in a closed control loop is investigated to achieve self-management.

By separating the core PBM functionality in four layers according to IETF's framework, the policy-based operations have been integrated to the organisational model by distributing the four basic PBM components among nodes according to their role. The defined functional elements of PBM are used, i.e. the Policy Management Tool (PMT), the Policy Decision Point (PDP), the Policy Enforcement Point (PEP) as well as a special version of the Policy Repository (PR), the Distributed PR (DPR). To form a closed feedback loop, the above elements are complemented with their context-aware counterparts. Hence the introduced components were added respectively: Context Management Tool (CMT), Context Decision Point (CDP), Context Collection Point (CCP) and Context Repository (CR) [2][5].

The extended set of components is shown in Figure 3-3, regardless of organisational roles. With the exception of Context and Policy Repositories, each pair of components is collocated. As

shown in Figure 3-3, component pairs exchange management information, with solid lines representing policy-related information and dashed ones context information. The major design difference of context-aware functions is that the flow of context information is reverse to the one in PBM systems. Context is collected at the lower layers of the architecture (CCP) then is processed and forwarded (CDP) to the higher layers for management decisions to be taken (CMT). On the other hand, policies are initially defined at the top layer (PMT) and then are propagated to decision points (PDP), which in turn provision their actions to end-devices (PEP). At each level, respective components interact, with the policy-based ones configuring the operation of the context-aware ones. In turn, context is provided to policy-based decision points, in order to evaluate policy conditions. As it will be explained in §4.2 (pp.82), this interaction takes places in three different layers leading to an adaptive closed control loop at each layer.

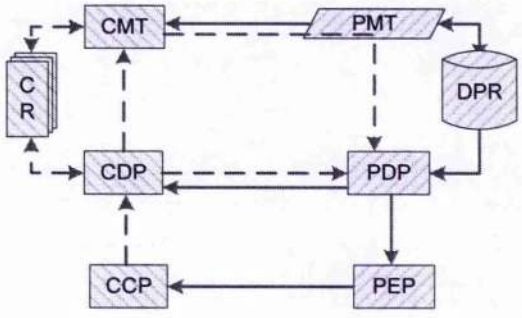

Figure 3-3. Interaction of policy and context components

The context-aware components were first introduced in [5] to assist in the management of Mobile Ad Hoc Networks (MANET). Their detailed architecture was presented in [2], elaborating on internal structure and functionality to achieve autonomic management of MANETs. In the scope of this thesis, the design of context-aware components is decoupled from the limitations imposed by MANET paradigm and the original concepts presented in [2][5] were extended to suit a wider range of wireless ad hoc networks. Context-awareness remains an open research area covering several aspects of Autonomic Computing and Computer Science. Formal techniques for context modelling and representation, as well as algorithms for context inference and aggregation, have been successfully integrated in deployments of the presented model for MANETs [2]. However, these issues remain out of the scope of this thesis, since they require a thorough investigation of a different research area and they are orthogonal to the investigated one.

3.3.2 Roles and Components for Self-Management

In this subsection, the relation between *roles* and *components* is clarified. The self-management framework is built from the composition of communicating basic ***components***. A defined set of

components is required for acquiring one of three ***roles***. In addition, each subordinate *role* is a *component* subset of its superior *role*. These concepts are visually represented in Figure 3-4.

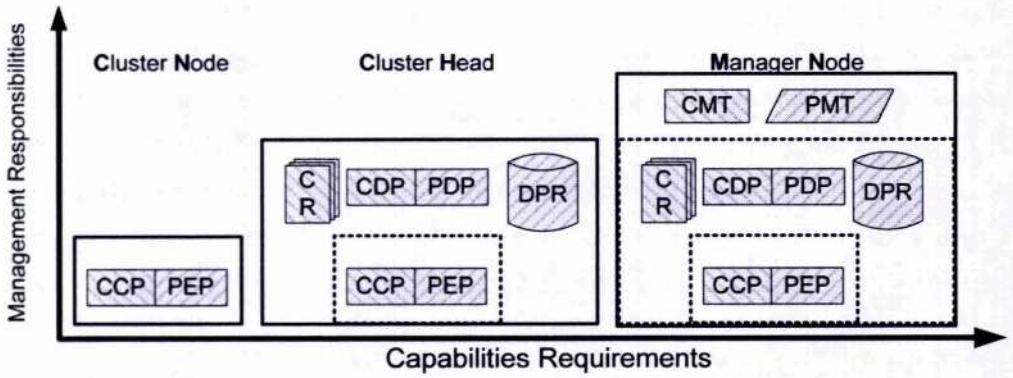

Figure 3-4. Node roles and required components

Based on the three roles and their respective components, Figure 3-5 shows a sample small-scale topology of 12 nodes, using the presented organisational model with node roles. This example employs two MN and an additional CH to control three clusters of CNs. In addition, the three leader nodes form the *hypercluster* thus collaborating to exchange management information and to share management tasks, e.g. the distribution and synchronisation of policies.

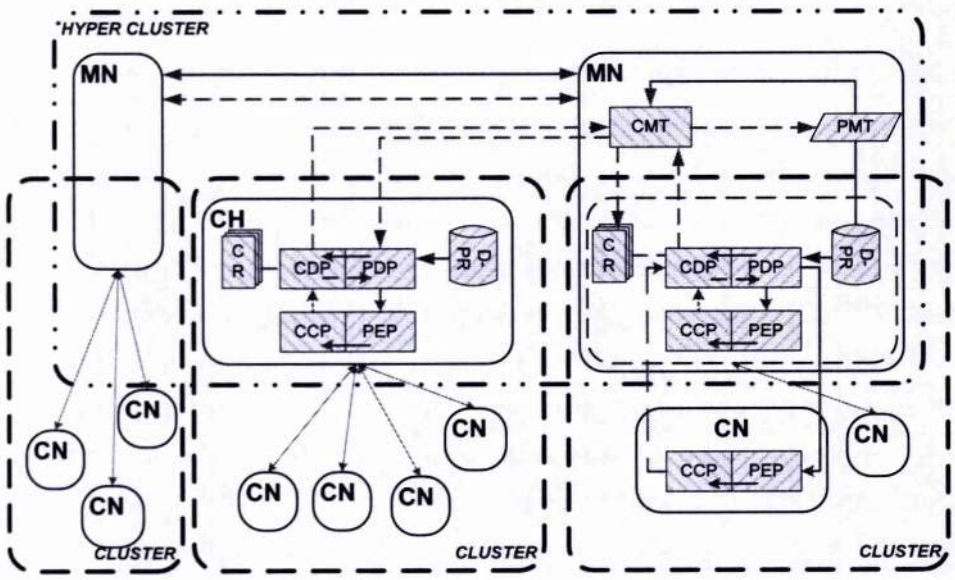

Figure 3-5. Example of Organisational model with node roles

In the next subsection (§3.3.3) an implementation perspective is taken by differentiating between node "*roles*" and "*modules*". *Appendix A* also provides additional details. A "module" is the preinstalled management software of a node, needed to realise the management functionality and operations of the framework. By elaborating on the deployment issues of the model, Figure A-5

mirrors the small-scale topology of Figure 3-5, showing an example with modules realising the organisational roles.

Before explaining the concepts of a multi-manager paradigm and the hypercluster, the architecture and responsibilities of the three different roles are outlined. Roles are progressively more demanding and complex in terms of management responsibilities and each simpler role is a subset of a more complex one. A highly modular design takes into account the heterogeneity of wireless networks and their wide applicability range. Each role has an increasingly more complex structure and added functionalities as shown in Figure 3-4. In other words, the most demanding role of a MN, encapsulates the CH role, which in turn encapsulates the CN role.

Cluster Node (CN)

The Cluster Node (CN) is the simplest role a node can assume and is designed to be simple and lightweight. A device in this role is participating in the network as a member of a single cluster and is under the control of its "parent" Cluster Head (CH). As a managed device it does not employ any additional management functionality beyond the functionality needed to configure itself. Lightweight implementations of CN functionality can be hosted on mobile phones, media players, routers or legacy (programmable) equipment.

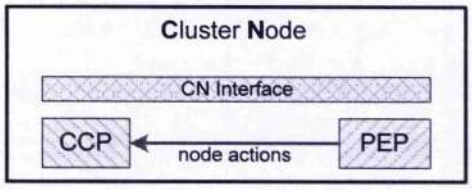

Figure 3-6. Cluster Node (CN)

Its main component is the Policy Enforcement Point (PEP), responsible for receiving and executing the provisioned policy actions from its parent CH's Policy Decision Point (PDP). The PEP acts as a middleware between the PBM system and the actual device hardware. The PEP should support at least one policy provisioning protocol and depending on the protocol it may be required to translate provisioned policy actions into low-level device-dependent commands. From an implementation perspective, it can be separated in two parts, i.e. device-dependent and device-independent.

Collocated with PEP, is its context-aware counterpart, the Context Collection Point or CCP. The CCP also communicates with device hardware to extract contextual information needed for network management. The actual context collection by CCP is locally configured by the PEP, based on received policy actions. E.g. policies define the required context polling and reporting frequencies to avoid excessive traffic and resources consumption. Based on policies the CCP reports collected node context to its parent CH.

Finally, an auxiliary component (*CN Interface*) aggregates the communication functionality between the CN and its parent CH as well as its neighbouring CN when needed. An optional extension to the policy-based system has been designed in [4] and affects CN Interface. Through a graphical user interface, it enables a user to locally control its owned device by setting preferences and privacy settings (see §6.3 for more details).

Cluster Head (CH)

The *Cluster Head* (**CH**) role is designed to enable distributed management operations by maintaining an overall control through policies. These contradicting requirements complicate CH design and increase the demand for device capabilities to assume this role. As a result, relatively capable devices can become CH. The minimum requirements depend on actual implementation. PDAs, smartphones and internet tablets are a sample of capable devices, while more powerful one (e.g. laptops) can also assume this role depending on network density and population.

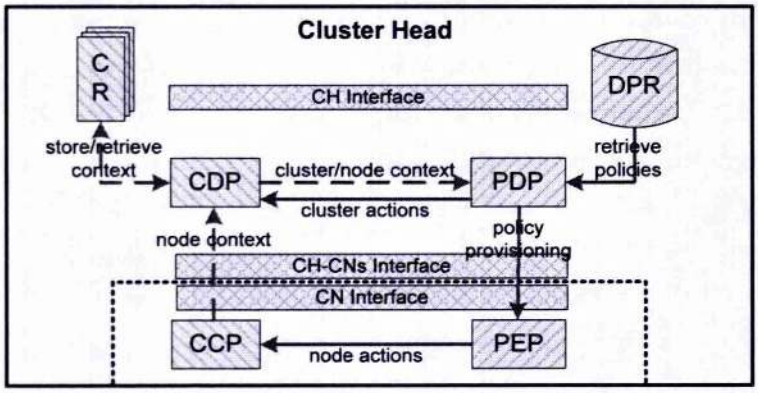

Figure 3-7. Cluster Head (CH)

A device in CH role controls a number of devices in CN role. As depicted in Figure 3-7, it also controls its encapsulated CN components. The "CH-CNs Interface" coordinates communication between all CN (by contacting their "CN Interface"), including the encapsulated local one. This interface has a twofold mission:

(1) to distribute policy actions from its *Policy Decision Point* (**PDP**) towards all nodes belonging to its cluster, i.e. its controlled PEPs

(2) to receive the reported context information from cluster nodes and forward it to its *Context Decision Point* (**CDP**), the context-aware counterpart of PDP

The Policy Decision Point (PDP) is the most important component of a CH as it is responsible for hosting all active Policy Objects (PO) and actively ensures their applicability and enforcement within its own cluster, formed by a number of CN devices. This involves the monitoring and evaluation of policy conditions. Based on conditions evaluation, it caters for the provisioning of

policy actions to the set of controlled PEP within its cluster, either proactively (provisioned) or reactively (outsourced).

The PDP interacts with a collocated Context Decision Point (CDP). This is the second context-aware counterpart of policy-based functional elements, as introduced within the presented framework. As in the case of PEP-CCP interaction, the PDP locally enforces special policies that guide the operation of its CDP. Thus the CDP is configured to report only relevant context by actively aggregating and processing cluster context. Reported context is the input for the evaluation of policy conditions and can be locally stored at the *Context Repository* (**CR**) component. In §4.2.1 (pp.81), further details are provided on policy and context interaction and the creation of adaptive control loop for cluster autonomy.

Another important task of a CH and specifically of its PDP, is the continuous acquirement of updated policies. Normally a Policy Repository (PR) is contacted to retrieve appropriate policies. For the purpose of the presented framework for wireless networks, the traditional PR has been replaced with **DPR** (*Distributed Policy Repository*), as a set of distributed and interconnected directories hosted on selected hypercluster nodes. Hence a CH contacts its nearest active DPR instance, either locally or remotely. The presence of an activated and up-to-date directory at a CH is dictated by the enforcement of DPR management policies. These policies aim to balance the resource utilisation at CH, reduce traffic overhead for synchronising directories and adapt policy distribution according to network dynamics. Full details on physical and logical repository distribution as well as mentioned DPR management policies are given in §5.3.

Finally, another auxiliary component is present within CH, to interface the device with fellow hypercluster nodes. The "*CH Interface*" also has a twofold mission:

(1) Communication and coordination with other hypercluster nodes, including Manager Nodes (MN), to enable the distribution of management responsibilities in the clustered network.

(2) Special functionality related to the network formation and its purpose. Depending on the network type and purpose, clusters creation and maintenance can be either dynamic/algorithmic (e.g. MANETs) or static/preconfigured (e.g. home and personal area networks).

Manager Node (MN)

The *Manager Node* (**MN**) role is the top hierarchy level of the framework. The role is an extension of the encapsulated Cluster Head (CH) role, with added management responsibilities and privileges. Due to the increased management capabilities of this role, devices assuming it are expected to be controlled by eligible managing entities. For deployments of pure user-created

networks, i.e. where no managing entities exist, it is possible for privileged users to control a device in MN role, e.g. in a home/personal area network.

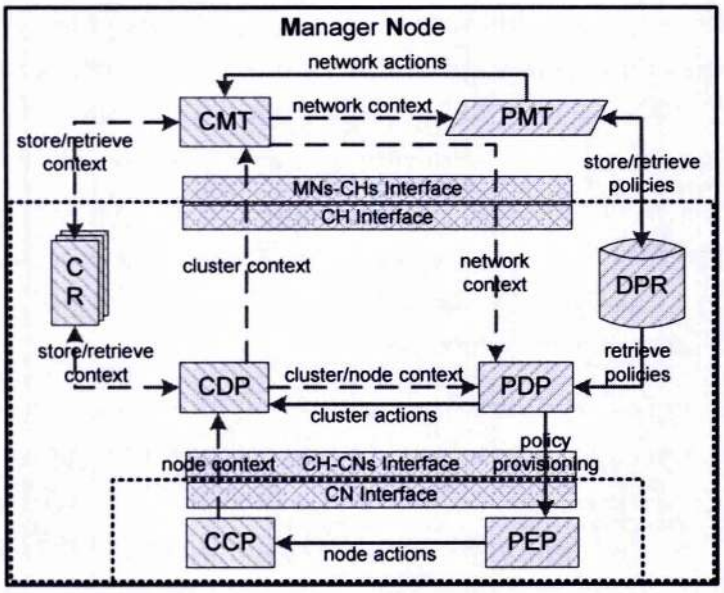

Figure 3-8. Manager Node (MN)

The main additional component of the MN role is the *Policy Management Tool* (**PMT**). The PMT is the interface between the human manager (e.g. a consultant or network administrator) and the underlying PBM system. In other words, a PMT offers a management interface between the policy makers and the deployed policy-based network. Contrary to traditional PBM frameworks, the designed MN role and PMT can offer a varying degree of overall network control depending on the purpose of its formation and the business objectives of managing entities. The MN role is the only role that has write access to the Policy Repository. It is allowed to introduce or change policies using its PMT to communicate with the local instance of the DPR. Contrary to the CH role, where the presence of an instance depends on DPR management policies, a MN always hosts an active and updated DPR instance.

An additional task for PMT is the interaction with its local **CMT** (Context Management Tool), the introduced context-aware counterpart. As in the cases of PEP-CCP and PDP-CDP, the PMT enforces special policies that guide the operation of the local CMT. This policy-based operation involves the aggregation of network-wide context, needed for the evaluation of specific policy conditions. The CMT also communicates locally with the *Context Repository* (**CR**) component to store or retrieve context information.

Finally, an auxiliary component is present within MN, mainly to provide an interface between managing entities (i.e. MN) in a multi-manager deployment. The "*MNs-CHs Interface*" also has a twofold mission:

(1) Communication and coordination with other Manager Nodes (MN), particularly for the purpose of policy conflict resolution and updates.

(2) Communication with Cluster Heads (CH) within the hypercluster, to provide context for the evaluation of policy conditions requiring network-wide knowledge.

Summary of Roles and respective components

Table 3-1 summarises the minimum set of components required to assume each role and realise its policy-based functionality. Auxiliary components and interfaces are omitted. It is implied that additional components may be present (installed) on a node and may be dormant if the assumed role does not require them. Further details on such deployment aspects are given in Appendix A.

Table 3-1. Summary Table for Roles and Components

Role:	Manager Node **MN**	Cluster Head **CH**	Cluster Node **CN**
Policy-Based Components Required			
PMT	Yes	No	No
DPR	Yes	Yes	No
PDP	Yes	Yes	No
PEP	Yes	Yes	Yes
Context-Aware Components Required			
CMT	Yes	No	No
CR	Yes	Yes	No
CDP	Yes	Yes	No
CCP	Yes	Yes	Yes

3.3.3 Motivation for Module Differentiation

Having introduced the concept of *"roles"*, the motivation for "module" differentiation is discussed, with some deployment issues. Further details of an example deployment of the model and high-level implementation guidelines are provided in Appendix A. Beforehand, the need to differentiate between node *"roles"* and *"modules"* is explained. A "module" is the preinstalled management software of a node, needed to realise the management functionality and operations of the presented framework. Effectively, this differentiation refers to the differentiation of the organisational role of an entity in the network as opposed to the actual software capabilities it carries. The self-management framework is built from the composition of communicating basic ***components***; while a preinstalled group of *components* constitutes a software ***module***. A defined set of *components* is required for acquiring one of three ***roles***; therefore the installed *module* on each device determines which *roles* it is capable of. As said, each *role* is a *component* subset of its superior *role*.

The same motivation for introducing different roles also applies to the concept of different modules and mainly stems from the increased heterogeneity of nodes in wireless ad hoc networks. As described earlier, each role requires certain functional *components* to carry out its specified management tasks. Each inferior role is deliberately encapsulated within its superior, in an effort to facilitate software reuse and a highly modular design. If the top hierarchy role is disintegrated, i.e. the Manager Node (MN), it results in the full set of components needed for the implementation of the proposed framework. A subset of these components is used by other node roles, i.e. the Cluster Head (CH) and Cluster Node (CN).

However, not all devices are capable of hosting every component, as device capabilities vary significantly, due to increased heterogeneity. For example, a mobile phone is not capable of carrying the DPR component and to host the required LDAP Directory Server. As a consequence, a mobile phone can only assume the most lightweight role (CN). This is the case of a large class of lightweight devices or terminals that it is desirable to participate in the wireless ad hoc network but will always retain the CN role due to their capabilities. It would be reasonable to implement the *components* subset for the CN role as a separate *module.* The motivation for this module separation is to extend the reach of the wireless ad hoc network to those lightweight devices that are very likely to initiate ad hoc communications. In addition, depending on the deployment scenario and the business model, node population increase may be translated in increase in customer numbers and consequently increase of revenue through consumption of services. Another issue that motivates module separation is the possible disclosure of management functionality and business policies to devices not controlled by the managing entities. Once again, this is tightly dependent on the deployment scenario and the business model. In such cases, it would make sense to exclude components like the Policy Management Tool (PMT) from a module that is publicly available.

3.4 Multi-manager paradigm

Management of large-scale networks is traditionally performed by a single logical managing entity, the *manager* of the network operator. As a logical entity, the *manager* may be a team of human administrators and network engineers, responsible for implementing management requirements in networks and systems. The described operations are performed in a logically centralised fashion, expressing the interests of the single managing entity. A simplification and acceleration of management operations can be achieved by policy-based management, as already described. In this case, one logical manager is employed for network management, strictly specifying through policies the behaviour of managed devices, e.g. routers, firewalls etc.

The idea behind the multi-manager paradigm lies in the nature of wireless networks and the purpose of their formation. Especially in the case of wireless ad hoc networks, it is desirable to allow more than one logical manager to coexist, thus catering for a variety of multi-manager scenarios. Having more than one manager gives the flexibility to form networks between distinct trusted administrative authorities. This composition is performed without any of the managers being forced to forfeit its management privileges. Instead, managers cooperatively introduce policies which guide the overall network's behaviour. Hence, the motivation behind the *multi-manager paradigm* for the designed model.

By employing the proposed multi-manager paradigm, the coexistence of more than one managing entities that control Manager Nodes (MNs) is possible. Most notably, this paradigm can become the basis for novel business models aimed at the exploitation of emerging wireless networks. For instance, a network operator (NO) can provide limited infrastructure support to assist the deployment of spontaneous wireless ad hoc networks. The multi-manager paradigm provides a controlled environment for additional managing entities, e.g. service providers. The NO can contractually lease management privileges to SP, allowing them to deploy additional services for wireless users.

In another example, a MANET can be setup for a corporate meeting between two companies' representatives. The multi-manager paradigm treats the companies' managers as equals and allows both to affect network behaviour by introducing policies. Another applicability example relates to the spontaneous formation of ad hoc networks from users or groups with no previous affiliation. The increasing popularity of social networking is expected to expand in wireless networks and establish wireless communities. Bringing such communities together can be catered by providing each community with management privileges and allowing them to define management policies. The managed devices in such scenarios are user-owned devices, like PDAs, media players etc. These devices are not strictly managed by an administrator and can benefit from a multi-manager paradigm.

From a functional point of view, an additional benefit from a multi-manager paradigm is scalability and resilience to a single point of failure. This is evident especially in the deployment of large-scale ad hoc networks where scalability issues demand more than one manager in order to control and administer effectively the numerous cluster heads and cluster nodes. The frequent disconnections and variable link quality may lead to network partitioning, leaving a portion of the network without a Manager. For example, a tactical MANET deployment may involve different platoons moving in adverse conditions and terrain. Having more than one manager increases the survivability of the network and reduces the risk of becoming unmanaged should a manager is disconnected or destroyed. Apparently these issues are minimised in fixed networks where link failures are less frequent and backup plans for link restoration exist. Departing from military

applications, infrastructure-less wireless networks created by users may increase resilience by deploying a number of physical managers based on reputation or capabilities. This role can be rotated among user devices to share out resource consumption and responsibilities.

Without depending on the classic manager-agent paradigm, the presented model facilitates a uniform management capability through multiple managers. Regarding the aforementioned roles in the framework, the Manager Node (MN) role is the equivalent of a *manager* as described above. Nodes assuming this role are controlled by eligible *managing entities* and define the network's behaviour through policies, i.e. policies are introduced in the system by using the PMT at each MN. Based on the above, four different cases are distinguished with regard to management authority:

1. *Single MN – Single Managing Entity*: This case is equivalent to traditional management, as described above. The presented model can be deployed for the hierarchical management of wireless networks and selectively utilise some of the advanced framework features, e.g. the Distributed Policy Repository.
2. *Multiple MN – Single Managing Entity*: This case is an extension of the previous one, by allowing more than one MN in a single administrative domain. In a simple scenario, MN are employed to increase network scalability and eliminate single point of failure. More complex scenarios are also possible, where different MN may represent different departments of the same organisation, or geographically distributed managers of a wide area network.
3. *Multiple MN – Multiple Managing Entities*: This is the most complicated case of the three, since different managing entities aim to manage the same network based on their own management goals. These entities may include network operators and service providers that use their controlled MN to introduce policies with different parameters and objectives.
4. *Multiple MN – No Managing Entities*: In the extreme case where no managing entities exist, the creation of user-managed networks is implied. Multiple MN may be controlled by privileged users (similarly to case 3) or may be algorithmically assigned to increase scalability (similarly to case 2).

Each MN can introduce policies in the system to express the high-level goals of each entity and these policies are interpreted in the management logic of the network. The distribution of policies among the hypercluster nodes helps on one hand to distribute management load and decision making and on the other hand gives localised control to Cluster Heads.

However, the coexistence of distinct administrative authorities raises issues of conflict detection and resolution. It can be assumed that the freedom to introduce policies in more than one physical node, i.e. different MN, could add significant complexity to the task of coherent network management and would increase the possibilities of policy conflicts. However this complexity is controlled, depending upon the purpose and formation of the wireless network. In the case of multiple managers under a single logical managing entity, complexity is almost the same as if a central manager introduced a policy, with a small overhead to control the serialised introduction of policies. However, in the case of multiple managers under different managing entities, there is a probability to have a special case of conflicts, namely inter-manager policy conflicts. This happens due to the different high-level objectives of each entity that are expressed in conflicting policies. These conflicts are examined in §4.3 where a solution for automated conflict resolution is provided.

3.5 Hypercluster formation and network clustering

The notion of the "*hypercluster*" has been introduced in the presented organisational model to cope with the management requirements of wireless ad hoc networks [5] [8]. The *hypercluster* is a special set of nodes that are assigned the collaborative management of the wireless network. It is consisted of devices having the Manager Node (MN) or the Cluster Head (CH) role, according to the role-based concepts described earlier. The *hypercluster* emerges as an overlay above individual clusters, whereas the remaining Cluster Nodes (CN) are effectively managed by hypercluster's nodes.

The formation of a hypercluster is valuable for efficient management of wireless networks since management intelligence and the policy repository are distributed among hypercluster's nodes. The model allows communication between Cluster Heads for exchanging management information, e.g. policies. This is also necessary for the efficient management of the Distributed Policy Repository, where bypassing a central manager for a policy retrieval or update can be faster and more efficient. Communication between CHs is extremely important in wireless ad hoc networks since links between nodes are intermittent and the bandwidth limited. Therefore where a CH can acquire information available at another CH, direct communication would save bandwidth and resources at the MN. Peer-to-peer communication between *hypercluster* nodes is allowed for exchanging management information and context.

Clustering is widely used in ad hoc networks for the reasons already explained (§2.3.3,pp.20). Roles were naturally introduced to cope with the complexities of cluster creation and maintenance and have been traditionally used in network layer clustering schemes for proactive MANET routing. However, in the presented work, clustering is used at the application layer for

management purposes and each role is associated and guided by special policies. In addition, the introduced *hypercluster* is formed to distribute and load-balance management tasks among resource-constraint wireless nodes.

The hypercluster can execute distributed algorithms for its own maintenance (e.g. reformation or reaction to node disconnection), eliminating the single point of failure of a strict hierarchy. A range of algorithms (§2.3.3,pp.26) can be used for cluster formation and maintenance, depending on the requirements of the applicability scenario and network composition. For example, ad hoc deployments for tactical operations have quite different requirements than user-initiated social networks. The flexibility of a policy-based design allows the integration of different clustering schemes and in addition the real-time parameterisation of their operation. Before network deployment and the actual formation of the hypercluster, a decision needs to be made regarding the selection method of participating nodes, i.e. Manager Nodes (MN) and Cluster Heads (CH). Selection can be either static or dynamic depending on the scenario and the wireless ad hoc network composition. A combination of both methods is possible and the use of default policies can assist further the initial setup and deployment.

Static assignment predefines which devices are selected as MN and CH to form the hypercluster. This selection method can be used in wireless scenarios where eligible managing entities (e.g. network operator or service provider) loosely control the overall network and services are deployed around a defined geographic area. Assignment of nodes to the MN role needs to be static, in order to ensure that the eligible managing entities always control privileged nodes. Fixed privileged nodes can be used as dedicated managers. Such case studies where examined for ubiquitous urban environments [4] and mobile wireless networks onboard trains [1]. Details are presented in later sections.

Dynamic assignment of MN and CH roles can be performed with the use of distributed algorithms, based for example on the selection of the most capable nodes. Scenarios with increased mobility in isolated deployments may also benefit from algorithmic hypercluster selection. Also, when there is a lack of fixed nodes or infrastructure, dynamic assignment increases the survivability of the ad hoc network. Case studies for the management of MANETs were examined in [2] and [5], using a distributed clustering algorithm.

The combination of both methods is also possible, e.g. using static assignment for MN and dynamic assignment of CH. For the rest of this section, the combined general case is examined, i.e. a *dynamic* hypercluster assignment with default MN, useful in cases of loosely managed ad hoc networks. MANETs have been traditionally used in military or emergency response scenarios. In such scenarios, a rapidly changing topology is assumed by a highly mobile network where a logical managing entity may not exist and all nodes are equal. As a consequence, static

assignment of hypercluster nodes is not advised as it would reduce network's survivability and increase management overheads. To tackle these issues, algorithmic clustering methods can be used to select the most appropriate set of nodes, based on capabilities, mobility and other metrics of interest. In some cases though, especially in military deployments, Manager Nodes (MN) need to be predefined depending on hierarchy and security issues, while Cluster Heads (CH) are dynamically assigned by distributed algorithms.

In [2] and [5] a dynamic cluster creation method has been introduced and evaluated, based on an adapted version of the Connected Dominating Set (CDS) creation algorithm by Wu [78]. In brief, all nodes (devices) execute the distributed algorithm to assign a role to themselves and to select the most capable ones to form the hypercluster. These nodes create the dominating set of the graph of capable nodes, thus ensuring one-hop accessibility for the remaining nodes. The novelty adopted in this work was the use of a context-aware capability function (CF) for DS algorithm's optimisation heuristics, replacing the arbitrary node ID selection criteria of the original one. Based on collected local information, the CF of a node indicates its current ability to assume resource-consuming roles (i.e. MN or CH). CF reflects two aspects of the nodes' capabilities, one referring to their computing attributes and another to their relative mobility. The algorithm by Wu was selected because of its fully distributed execution, low complexity and low message exchange overheads compared to other algorithms (§2.3.3). Finally, the algorithm caters for the dynamic environment of wireless ad hoc networks by defining efficient update and reconstruction procedures for the maintenance of the CDS under node movement and failure. For completeness, Appendix A provides further details about the Dominating Set creation algorithm [78] and its use and modification in this work [2],[5].

3.6 Scalability Investigation of Organisational Model

In this section, different aspects are investigated regarding the creation of the hypercluster and the overall proposed network organisation. Based on performed implementation measurements and literature on adopted algorithms, an evaluation of the model's scalability and traffic overheads for policy distribution is provided.

The traditional hierarchical policy-based network organisation is depicted in Figure 3-9. According to this design, a centralised manager and a fixed predefined number of PDP control the total number of network devices. Normally there is no provision for dynamic network reorganisation, since the expected number of devices is relatively stable. On the contrary, the proposed hybrid model combines both hierarchical and distributed approaches. Using a fully distributed algorithmic method ([2],[5]), it reorganises itself on the fly to anticipate fluctuation in node (device) population. Hence, dynamic network conditions and node reorganisation can be

catered. An example topology is shown in Figure 3-10, where on the left side a sample network deployment is shown and on the right the respective network graph. Hypercluster nodes are depicted using PDP/DPR components (left) and boxed circles (right). In this example, both MN nodes (in red) need to be included in the hypercluster, hence their inclusion in the graph.

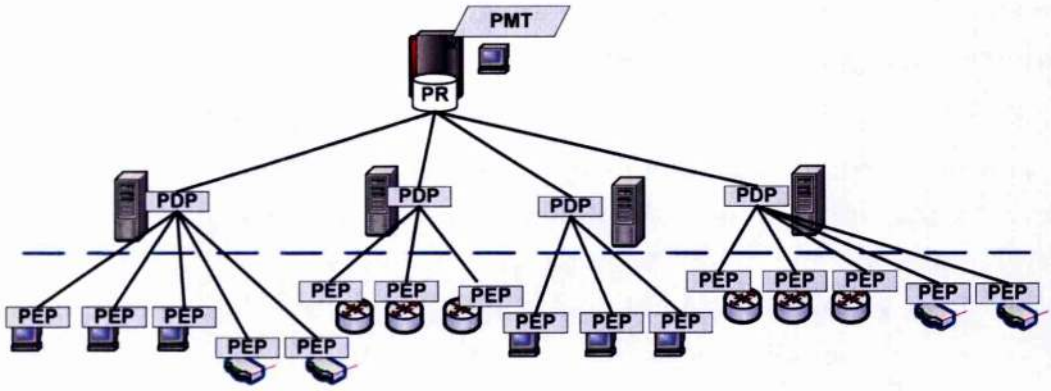

Figure 3-9. Centralised and hierarchical policy-based network organisation

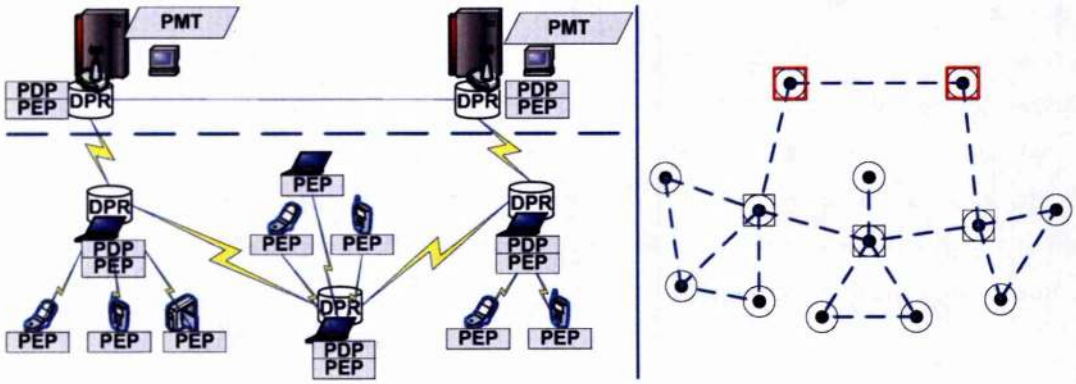

Figure 3-10. Hybrid organisational model and network graph for hypercluster creation

3.6.1 Evaluation of algorithmic hypercluster creation

The first aspect examined was the size (population) of the created hypercluster set for different network node populations and for different node densities. It is important to maintain a reasonable size for the hypercluster, able to adapt to change and to the variable conditions of wireless ad hoc networks. The presented results were calculated by converting the simulation parameters used in [78] and combining those with implementation measurements [2],[5]. The use and interpretation of the presented results justify the selection of Wu's algorithm for hypercluster creation and compare these results to traditional alternatives for PBM beyond the suggested hypercluster formation. To assess the behaviour of the algorithm in terms of hypercluster population, different cases of node density were examined against an increasing number of node populations. The

density ratio is defined as the ratio of the total network population over the simulation area (nodes/m^2).

Specifically, four cases were examined, two with a fixed density ratio and two with a variable density ratio. For simulations with fixed node density, as the node population increases, the deployment area is also increased to maintain a stable (fixed) node density (Fix.Dens(1:1600), Fix.Dens(1:27800)). For example, for a fixed density of 1:1600 nodes:m^2, the number of nodes is varied from 25 to 400 and the area size from 200x200m^2 to 800x800m^2 respectively. This case was particularly examined in [2], where beyond the evaluation of hypercluster size, additional evaluation aspects were presented like the hypercluster's construction time. For variable node density, the deployment area is fixed and the node population is increased, resulting in increased density ratio. For example in a fixed area of 1000x1000 m^2, node population is varied from 25 to 400 leading to a variable density ratio between 1:40000 to 1:2500 nodes:m^2 (Var.Dens(~1:2500)). For a fixed area of 500x500 m^2 and the same increase in node population, a variable density ratio between 1:10000 to 1: 625 nodes:m^2 was examined (Var.Dens(~1: 625)). By examining both fixed and variable node densities, a better understanding of algorithm's performance was acquired and useful guidelines for the deployment of the proposed model were extracted. Simulation parameters and results can be found in Table A-1 of Appendix A, where additional details about the employed algorithm are also provided.

Figure 3-11 demonstrates the behaviour of Wu's algorithm [78] under the aforementioned varying conditions for small to medium scale networks populations. By analysing these data, it was observed that for cases of fixed density ratio, the hypercluster population increases linearly, thus linear trendlines were calculated. The slope is inversely proportional to node density, i.e. more hypercluster nodes are needed for less dense networks. This is reasonable and linearity proves the scalability of the algorithm because it ensures efficiency in construction time and overheads [2],[5].

However, it is noted that for less dense deployments (e.g. 1:27800 nodes/m^2), the hypercluster population is quite increased, reaching almost half of the total population. As explained later, this behaviour can lead to scalability issues regarding policy distribution because all hypercluster nodes host a PDP that needs to be informed of current policies. For the examined cases of variable density ratios, the algorithm demonstrated logarithmic behaviour. This is an important property of the algorithm, since it guarantees adaptive hypercluster behaviour in a defined geographic area, while node population increases. A broader view of algorithm's behaviour is also shown in Figure 3-12, based on large-scale network deployments. This graph confirms the scalability of the algorithm, provided the managed network is not exceedingly sparse in terms of node density.

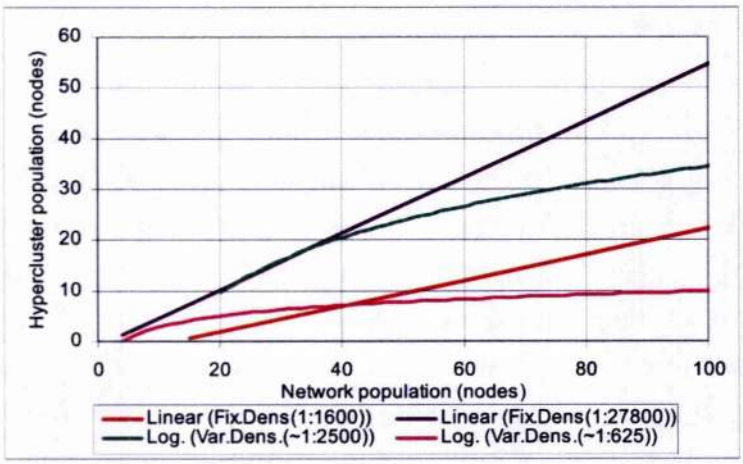

Figure 3-11. Distributed hypercluster calculation for medium-scale networks

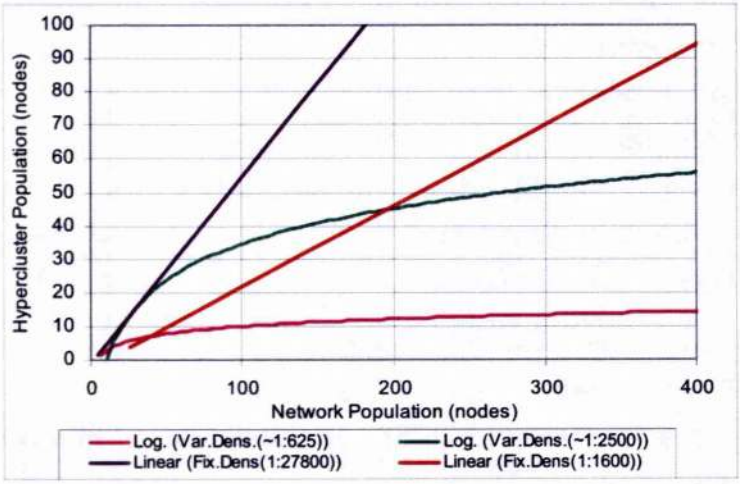

Figure 3-12. Distributed hypercluster calculation for large-scale networks

3.6.2 Evaluation of organisational model for policy distribution

Having verified the behaviour and scalability for the creation of the hypercluster, the policy-based aspects and duties of hypercluster nodes are examined in this section. As explained, hypercluster nodes (CH and MN) employ a PDP that is used to manage all PEP in their cluster (Figure 3-10). Therefore each PDP must be aware of network policies. This is the case for both centralised and hybrid models. For all examined cases, it was assumed that all PDP should become aware of the same policies, i.e. same policy retrieval queries were used.

For the proposed hybrid organisation, policies are distributed among hypercluster nodes (and their PDP) by creating a Distributed Policy Repository (DPR) overlay. As will be detailed in Chapter 5, policies are replicated to CH nodes from MN nodes using LDAP Content Synchronization Operation ([215], RFC4533), which is a special directory replication directive (*syncrepl*). Policy

replication and synchronisation are part of a policy-based DPR Management scheme discussed in §5.3 (pp.108). For the purpose of this section's analysis, PDP of centralised deployments acquire policies from the central PR, using standard LDAP Search Operation ([213], RFC4511). This is the normal practise for accessing a centralised repository, but it assumes PDP have been notified in advance about the search criteria and the PR location. It is also assumed that a predefined number of PDP has been deployed on prespecified nodes.

In 2004-5, initial policy implementation and measurements were presented [5], based on local deployment of OpenLDAP Directory Server Agent (DSA). OpenLDAP v.2.2 provided an early implementation of the Content Synchronisation Operation (RFC Draft, Sep.2004). Due to the standardisation of *syncrepl* operation and the availability of improved OpenLDAP implementations (v.2.3), new extended measurements were taken in 2007-8. A comprehensive analysis of replication methods and measurements is provided in Chapter 5. For the purpose of evaluating the proposed organisational model for policy distribution, an extract of policy measurements was used for this section. The traffic cost incurred for point to point retrieval of 200 policies is shown in Table 3-2. The implemented policy representation requires 4 LDAP entries per policy rule instance, while compound policies may require more entries. LDAP traffic is extracted from measurements, to calculate the exact traffic required by each operation. By combining these traffic measurements with previously presented algorithmic results (Figure 3-11, Figure 3-12), the implications imposed on management traffic overheads can be evaluated and better understood. The distributed hypercluster creation and the replication-based policy distribution, strongly influence the effectiveness of the proposed hybrid organisational model.

Table 3-2. Traffic measurements for policy retrieval

200 policies (816 entries)	Total Traffic (bytes) [inc. headers]	LDAP Traffic (bytes)
ldapsearch (RFC4511)	237318	233688
syncrepl (RFC4533)	365920	360310

As measured, the traffic cost of *ldapsearch* operation for initial policy retrieval would be significantly less than that of *syncrepl* operation. However, each PDP must be informed in advance about the existence and exact location of policies it needs to retrieve, i.e. the rules' distinguished names (DN) [213](Appendix B). Additional notification is also needed when new policies are added or existing ones change. The traffic cost of notifications was not included in measurements, as it is dependent on the actual implementation of each centralised system. Using distributed replication, *syncrepl* operation automatically disseminates all changes and new policies to Cluster Heads' repositories, making them available to collocated PDP.

Measurements of *syncrepl* LDAP traffic were used with hypercluster sizes to estimate the total LDAP traffic generated for distributed policy replication on all hypercluster nodes. Varying wireless network populations and densities were examined (§3.6.1). Generated LDAP traffic for centralised deployments with fixed PDP numbers (Figure 3-9) was also estimated. These values were calculated by multiplying the fixed number of PDP with the measurements taken using *ldapsearch* operation. For comparison, two static centralised cases were examined with a preconfigured number of 25 and 75 PDP. As network population increases, these values remain constant (fixed PDP number), hence are shown as horizontal dashed lines. As expected, graphs of LDAP traffic for distributed policy deployment remain similar in shape with the ones depicting hypercluster size. What is important though is how these graphs compare to respective centralised deployments. Figure 3-13 shows generated traffic in large-scale network deployments, providing an overview of all examined cases.

The first observation was that the generated traffic is significantly increased for large-scale sparse networks. This was expected since a larger hypercluster set was created to accommodate connectivity among widely dispersed nodes. Such deployments would be difficult to maintain, since generated traffic would need to traverse long multihop paths and can lead to congestion of wireless links. The same difficulties apply to a centralised deployment with a fixed predefined number of PDPs, since these PDPs will need to be placed away from the central repository in order to provide coverage to all network nodes. Compared to the centralised approach and a constant predefined number of 75 PDP, large-scale deployments of fixed density ratio (1:1600) generated less traffic for node populations less than 210 nodes. This dynamic behaviour can significantly benefit wireless networks, because the traffic needed to distribute policies adapts to the network population. Thus for large-scale networks, policy distribution to an adaptive set of hypercluster nodes was comparable to centralised deployment with 75 fixed PDP.

Figure 3-14 focuses on medium-scale wireless networks, depicting a selection of the examined cases. Due to network size, the inclusion of the centralised approach with a predefined number of 75 PDPs is omitted. For the depicted cases it was observed that small to medium networks with variable node densities in an area of 1000x1000 m^2 (Var.Dens(~1:2500)) required less traffic to distribute policies to PDPs of an adaptive hypercluster set. These traffic measurements were compared to the traffic cost of policy retrieval from 25 fixed PDPs for the centralised case. In addition, generated traffic for hybrid deployment in dense wireless networks was very limited and did not incur significant overheads even for large scale networks of 400 nodes in an area of 500x500 m^2. This was due to the small and stable size of the created hypercluster, since most devices in dense networks have direct wireless links with each other.

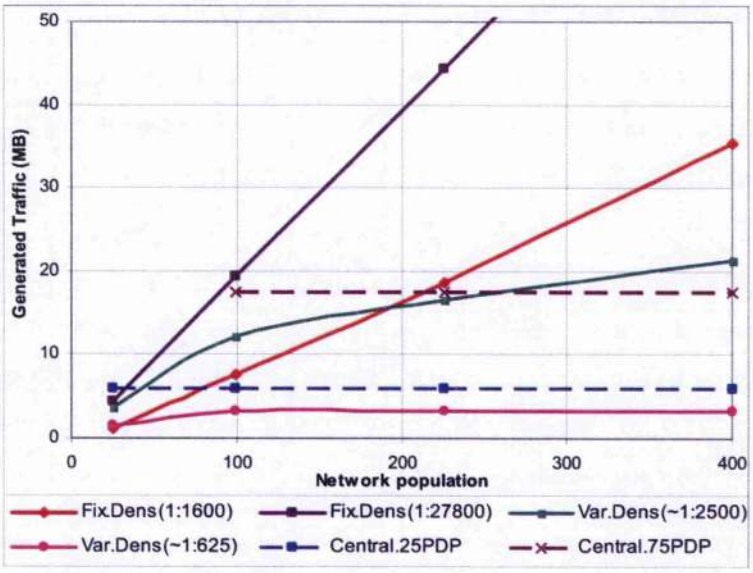

Figure 3-13. Policy Distribution Traffic to Hypercluster (large-scale networks)

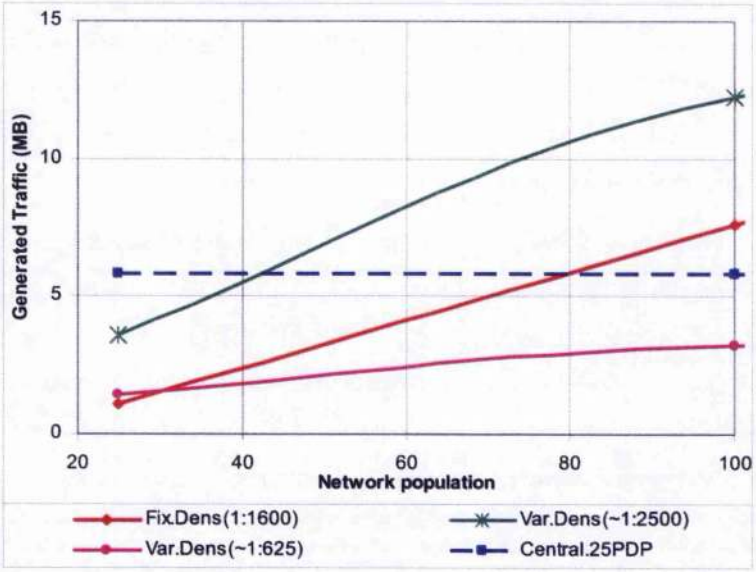

Figure 3-14. Policy Distribution Traffic to Hypercluster (medium-scale networks)

As observed, a non negligible traffic increase is incurred by *syncrepl* operation, when compared to the *search* operation for the same policy retrieval. However, the traffic overhead can be counterbalanced by improved network organisation and adaptive behaviour to dynamic variations in the population of wireless ad hoc networks. The aforementioned results verify the viability of policy distribution using *syncrepl* operation and multiple distributed policy repositories. These issues are further analysed in Chapter 5, by elaborating on the Distributed Policy Repository design and relevant DPR Management policies.

3.7 Summary and Conclusions

As networks become more and more complex, it is evident that frameworks with self-management capabilities can significantly expedite and simplify management tasks. Towards this direction the designed self-management framework for wireless networks is based on policies and context to realise an adaptive closed feedback loop.

The combination of these two concepts, namely policy-based management (PBM) and context-awareness, has made possible the implementation of realistic case studies on wireless testbeds. As it was explained, policies and context interact by exchanging information to proactively achieve management tasks. Policies express high-level objectives, guiding the self-management of wireless networks and provisioning which actions should be executed when certain conditions are met. At the same time, context monitoring achieves a real-time understanding of network conditions and surrounding environment and is used for policy conditions evaluation. In order to achieve self-management according to high-level objectives, the described process is repetitive, leading to an adaptive closed loop of control. The adaptation loop is initiated with the uniform deployment of policies which are dynamically translated into management logic and distributed to capable wireless nodes. Policies can also drive context collection, i.e. the monitored context may depend on the types of policies deployed, and in turn collected context drives policy activation and execution, leading to self-managed decision making.

As explained (§3.5), an algorithmic process organises the wireless network in clusters, where assigned Cluster Heads (CH) perform local management tasks. The rest of the nodes become Cluster Nodes (CN), register to their nearest CH and remain under its supervision. The correspondence of physical devices to roles depends on their capabilities and their ownership. Generally speaking, lightweight devices like mobile (cell) phones become CN, while more powerful devices like laptops or access points can become CH. Depending on the formation purpose of the wireless network and the business model of the network operator, one or more privileged nodes are assigned the Manager Node (MN) role. Together MN and CH constitute the *hypercluster* and perform management tasks in a distributed and cooperative manner. The integrated policy-based features add the desired self-management capabilities and controlled programmability to wireless ad hoc networks.

Chapter 4

Policy Design Aspects for Wireless Ad Hoc Networks

4.1 Introduction

Wireless ad hoc networks are the target applicability domain of the designed policy-based self-management framework. As emphasised already, these networks are significantly different from today's fixed conventional ones and even from cellular ones. Participating devices are increasingly heterogeneous and can be quite lightweight in terms of processing capabilities. Therefore, it is important to differentiate the design of the adopted Policy-Based Management (PBM) paradigm, in order to satisfy the requirements of wireless ad hoc networks.

First, an appropriate policy language and representation are needed to enable the majority of devices to participate in collaborative management. Policies should be represented and stored in a lightweight format that would be space efficient and easily processed on heterogeneous devices. Therefore, the Event-Condition-Action (ECA) notation of policies was employed, due to its simplicity and effectiveness. Based on the presented hybrid organisational model, policies were designed in a matching hierarchy according to introduced roles. The purpose of a policy hierarchy was the efficient accomplishment of distributed management tasks, based on the role that each device holds. Furthermore, the concept of *policy enforcement scope* was introduced to assist in the integration of a triple layered closed-control loop.

The potential commercial exploitation of wireless ad hoc networks can be enhanced with the introduction of a multi-manager environment. To support the operation of multiple Managing Entities, a Conflict Detection and Resolution (CDR) Tool was integrated, implementing a

manager communication protocol that ensures consistent introduction and editing of policies. The occurrence of inter-manager conflicts was investigated, since each manager has its own high-level objectives and inevitably their policies may contradict due to incompatible management interests. Building on extensive literature on policy analysis, a CDR solution was integrated that makes automated conflict detection and resolution possible. At the same time, the solution satisfied the interests of all managers involved, based on their contractual agreements.

4.2 Policy Notation and Hierarchy

The designed self-management framework targets increasingly heterogeneous devices that can be quite lightweight in terms of processing capabilities. It is therefore important to decide on an appropriate policy language and representation so that the majority of devices are able to participate in a PBM network and contribute to its collaborative management.

In particular, policies need to be represented and stored in a lightweight format that would be space efficient and easily processed. Therefore, the first step is the adoption of an appropriate information model and subsequently its translation to a data model. In addition, a formal representation needs to be interpreted between human-readable format and machine-processed language. Such computationally intensive tasks need to be uncomplicated and avoid operations that would drain the limited resources of devices participating in a wireless ad hoc network. The mentioned requirements were not addressed by existing policy languages. This has lead to the decision to employ a custom lightweight policy notation based on the established *Event-Condition-Action (ECA)* notation and on existing IETF/DMTF information and data model specifications [5]. However, the focus was placed on the definition of the necessary policies and structures for wireless network management, rather than the formal definition and implementation of a complete policy language or toolkit.

The ECA policy notation is widely used in literature due to its simplicity and effectiveness [9],[17],[18],[102]. This representation is both efficient and lightweight so as to cater for the policy needs in the resource-poor wireless ad hoc environment; therefore the presented policies follow this notation. In addition, this notation is generic enough to allow the translation of policies to other formal policy languages, e.g. PDL [103] or Ponder [107]. The selected format of policies follows this structure:

{Roles} [Event] *if* {Conditions} *then* {Actions}

Roles element defines which devices will need to apply the specific policy. It also helps grouping policies and easily retrieving them from a Policy Repository. The presented PBM design adopts the three roles defined for the organisation of the framework, i.e. Cluster Node (*CN*), Cluster Head (*CH*) and Manager Node (*MN*). Each policy can be assigned to a role or a role combination,

creating a hierarchy of policies that facilitate distribution of management tasks. As discussed in the following section, role combinations have a special meaning related to the concept of *policy enforcement scope.*

Event element triggers the evaluation of policy conditions. It can be a periodic, time-based or scheduled event, as well as dynamic real-time event or event correlation. Depending on system's capabilities and complexity, a sophisticated event bus and correlation engine can be implemented. For the presented design, events were emitted from sensing and monitoring processes. Such processes may gather information from local systems events or from reported context. An approach based on remote procedure calls (RPC) was adopted for external events and reporting.

Conditions element is a Boolean expression containing one or more conditions to be evaluated. If the condition is true, that would trigger the execution of specified actions. Composite policy conditions can be formed from simple Boolean variables (e.g. Device==ON) or mathematical expressions (e.g. Battery>50%) In addition, access control restrictions and time-based conditions can be added. The introduction of context-aware components to the PBM framework closes the feedback loop and can provide context-aware parameters for condition evaluation.

Actions element contains one or more actions needed to be enforced, once the specified event has occurred and policy conditions are true. Essentially, these elements encapsulate the appropriate parameters that need to be transferred to enforcement points through a policy provisioning protocol. Therefore their representation is closely related with policy provisioning and enforcement (§6,pp.137). The actual implementation of actions is independent of Action elements, providing an extensible and customisable solution to suit different devices and platforms.

4.2.1 Policy hierarchy and enforcement scope for self-management

Policies and roles were combined in the proposed framework and a "Roles" argument has been introduced to the proposed ECA policy clause. As discussed, three organisational roles have been defined to assist in the distributed management of the network. Based on the organisational hierarchy of the presented framework, policies were also designed in a matching hierarchy. Beyond the corresponding policies for each of the three roles (MN, CH, CN), the concept of *policy enforcement scope* was introduced to further assist in the layered closed-control loop. The ***"enforcement scope"*** *of a policy is defined as the set of nodes where actions need to be enforced, when the policy is triggered by the context collected within this set.* As expected, the three organisational roles defined earlier were respectively mapped to policy roles and can group policies accordingly i.e. {Roles}:= {CN} or {CH} or {MN}. Such policies have *local enforcement scope* in the sense that their conditions and actions are evaluated and enforced locally. This is the

common practise in most PBM systems, for example policies controlling router configuration would set configuration values individually, based on {CN} policies.

The purpose of a policy hierarchy is the efficient accomplishment of distributed management tasks, based on the role that each device holds. Policies assist in role and duty separation by associating specific functionality with appropriate devices. In addition, a policy hierarchy maintains information locality and reduces dissemination overheads. Policy conditions can be evaluated locally from available knowledge, without the need for decision outsourcing to higher hierarchy layers. The maintenance of locality and restricted information flooding is very important for the resource-constraint environment of wireless ad hoc networks.

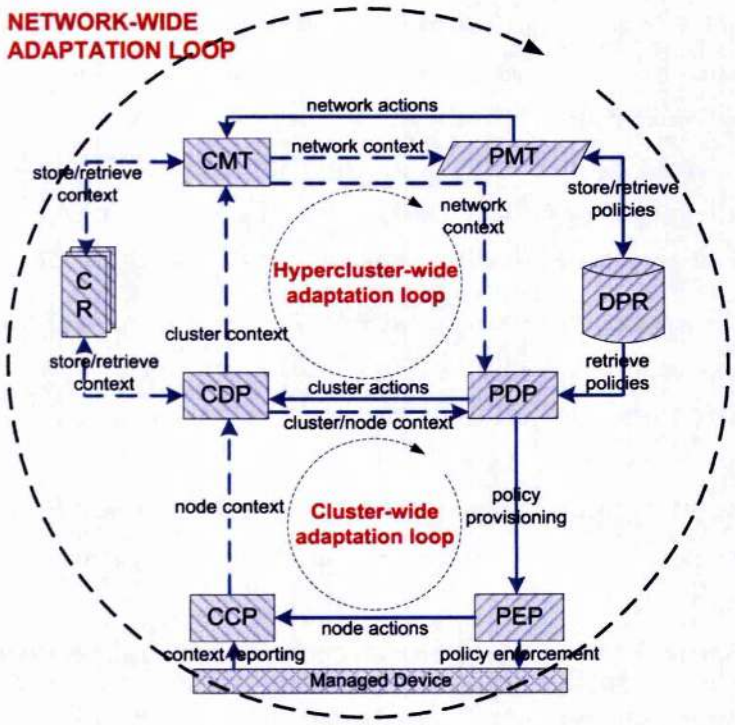

Figure 4-1. Diagram of layered closed-loop adaptation

One of the novel concepts of the proposed self-management framework is the triple layered closed-control loop. As introduced earlier, cooperation between policies and context creates a closed-control loop, with context providing feedback to policies and policies controlling context processing. Further details on these concepts are provided below, with reference to Figure 4-1. This figure depicts the main functional components and their interactions, regardless of role separation. By introducing policy hierarchy and enforcement scopes, three independent control layers were created based respectively on three different enforcement scope levels. In the next subsection, realistic examples of policy types are given, illustrating the introduced concepts:

1. Network-wide: Policies with *network-wide enforcement scope* are triggered at the MNs by the context collected and aggregated from all network nodes. The PDP of all MNs decide to enforce the actions network-wide and delegate those actions to CHs to be enforced to all PEP of network nodes. These policies are identified by their assignment to all three roles: {Roles}:={MN&&CH&&CN}. The control loop involves the hierarchical collection and aggregation of context, starting from cluster CCPs reporting to their CDP and in turn CDPs of all cluster heads reporting information to one of the managers' CMTs. The cluster-wide aggregated context (e.g. relative mobility) is reported to CMTs and is subsequently exchanged among them to ensure a network-wide context representation. More details on context management can be found in [2]. The bottom-up *context information flow* is followed by a top-down *policy decision flow*.

The benefit of the designed control loop and policy hierarchy stems from the achieved network-wide awareness and policy control of managers (MN). The network-wide context gathered from CMTs is withheld among managers and only returns to cluster heads (CH) if policy actions are required. This explains the second "context-flow" arrow (Figure 4-1), going from CMT, through PMT, to PDP. The context collection path is reversed, returning context to reporting cluster heads. The network-wide context value is used by all PDP for condition evaluation of policies with network-wide enforcement scope. Triggered actions are provisioned to all PEP in the network, ensuring uniform network-wide enforcement of policies. Finally, the loop is closed and repeated by collecting new context at CCP, in order to evaluate when needed the results of enforced actions. In addition, policy actions can affect CCP by fine-tuning parameters related to network-wide context collection and processing.

The aggregation of network-wide context and triggering of special (network-wide) policies at the top hierarchy layer implies an indirect policy (action) provisioning from CMT-PMT to PDP. This can be explained if one considers that CMT-PMT of MN, contact the local and remote PDPs only if conditions of network-wide policies are met. In turn PDP of CHs provision policy actions to the PEP of all nodes. The apparent hierarchical context exchange between Manager Nodes and Cluster Heads can create a bottleneck at MN, since the ratio of CH:MN can be potentially large. This issue can be alleviated by employing the hypercluster distribution network and exploiting the Distributed Policy Repository (DPR). Further details on DPR and policy provisioning are given in Chapters 5 and 6 respectively.

2. Hypercluster-wide: Policies with *hypercluster-wide enforcement scope* can be triggered at all hypercluster nodes by the context aggregated within the hypercluster. Decisions are enforced only at the hypercluster nodes. These policies are identified by their assignment to {Roles}:={MN&&CH}. A second layer of control loop is formed among hypercluster's nodes, adding an extra degree of automation to the PBM framework. The hypercluster-wide adaptation loop operates similarly to network-wide adaptation. It involves the CDPs of all cluster heads, where each CDP reports

information to one of the managers' CMTs. Such information may either be a cluster-wide aggregated context (e.g. average cluster fluidity) or context individually perceived by the local CCP of a cluster head (e.g. wireless channel quality). In the latter case, the participation of remote cluster's CCPs is not required. As previously for *network-wide scope*, CMTs process received context and return a hypercluster-wide context value to PDPs for condition evaluation. In this case though, only conditions of policies with hypercluster-wide enforcement scope are evaluated and actions are provisioned only to PEPs belonging to hypercluster nodes, i.e. CH and MN. The loop can be closed by enforcing actions on hypercluster CDPs, fine-tuning context-related parameters.

3. Cluster-wide: Policies with *cluster-wide enforcement scope* can be triggered at a hypercluster node by the context aggregated within its cluster. Decisions are enforced only at the cluster nodes belonging to the cluster where the policy was triggered. These policies are identified by their assignment to the roles: {Roles}:={CN&&CH}. The formation of a cluster adaptation loop is evident, since the PDP of a cluster head (CH), receives localised context information (e.g. remaining battery) from its collocated CDP, as aggregated from the CCPs of nodes it controls. Based on that cluster-wide context, a cluster PDP (CH) can autonomously decide by evaluating policies with cluster-wide enforcement scope and provision further actions to controlled PEPs. Again the loop is closed with the policy-based continuous context collection, as a result of provisioned actions. By enforcing node actions on CCPs, it is possible to fine-tune context collection and reporting parameters.

To better illustrate the use and applicability of aforementioned policy design and concepts, realistic examples of policy types are presented in the following paragraphs. These policies provide a first step towards an automated policy-based management framework, specifically designed for the needs of wireless ad hoc networks. Policies are intentionally simple to serve as proof of concept examples and a guideline to realising more complicated functionality. In the next Chapter, more details are given on actual policy representation, providing a step-by-step methodology for policy implementation.

4.2.2 Policy examples for resource-constrained devices

To illustrate the aforementioned concepts, three examples are presented below, aiming on one hand to demonstrate the effectiveness of simple policy rules and on the other the applicability of the defined policy enforcement scope. The chosen policies are not overly complex for clarity and serve as an introduction to policy design. Depending on management goals, compound conditions and actions can be introduced in all policy types, in order to take more parameters into account. All three examples were taken from a realistic case study of a wireless network, i.e. the management of MANETs [5].

1. Routing adaptation policy with network-wide enforcement scope

A plethora of protocols has been proposed to solve the multihop routing problem in MANETs. As detailed earlier in literature review, a generic classification can distinguish routing protocols into proactive and reactive, depending on the strategy used to establish routes between nodes. Therefore, a policy type is modelled that would enable dynamic on-the-fly adaptation of the routing protocol. Network conditions/context on one hand and manager defined goals on the other, can be both expressed by this type of policy, which alters routing strategy and increases network performance [2]:

{MN&&CH&&TN}[**E**] if {RM=(**n..m**)} then {RoutProt:=**k**}

The above policy type is used to adapt network behaviour by switching the routing protocol (RoutProt) according to the network's *relative mobility (*RM*)*. Bold fields can be customised during and after policy instantiation. RM is aggregated context information extracted from the network-wide knowledge of node movements, e.g. GPS positioning data, mobility patterns or other context. The simple condition monitors whether RM value lies within the range (n..m), in order to enforce an action that activates the appropriate routing protocol. For implementation purposes, the idea is to use a proactive routing protocol (OLSR [209], k=1) when relative mobility is low and a reactive (AODV [208], k=2) when high. Two policies can enforce the described management goals:

{ MN&&CH&&TN }[rm_event] if {RM=[0..35)} then {RoutProt:=1:OLSR }

{ MN&&CH&&TN }[rm_event] if {RM=[35..100]} then {RoutProt:=2:AODV}

The network-wide enforcement scope of this policy implies that the condition variables used (e.g. RM) should have an aggregated network-wide value. For example, the value of RM is extracted from the gradual aggregation and processing of simple low-level node context (e.g. speed) to cluster context and eventually network context. Cluster context is collected at CMT components of managers (MN) and this allows them to compose the network-wide context variables. This higher level context information drives the triggering of actions that should be enforced globally (network-wide). If conditions are met, each CMT forwards this value to the local PDP and to the PDP of all CH it controls. In turn, each PDP enforces the triggered action to all cluster node PEP, including its local one. This sequence of actions ensures the smooth and controlled execution of network-wide adaptation, in a self-managing and policy-based manner.

2. Repository replication policy with Hypercluster-wide enforcement scope

The need for Policy Repository (PR) distribution has already been explained and is mainly required to diminish the single point of failure in centralised PR (§2.4.2,pp.44). For this purpose a policy type is modelled to guide the replication degree of the Distributed Policy Repository (DPR). A manager node has the ability to dynamically define the behaviour and the replication

degree of DPR by introducing related policies on the fly and without disrupting its operation or system operation:

{MN&&CH}[E] if {FM=(n..m)} then {ReplDegState:=k}

The above policy type is used to guide the **replication degree** (ReplDegState) of DPR component. *Replication degree* is the level of DPR distribution in terms of replica numbers in the network and is expressed as different *replication states*. The *fluidity metric (FM)* is a hypercluster-wide aggregated context that represents how volatile the network is. Three states of replication are implemented, namely k=1:Single, k=2:Selective and k=3:Full. These states reflect the current need for repository replicas within the hypercluster nodes and adapt according to the volatility of the MANET as shown in Figure 4-2. As mentioned earlier, the idea is to increase the DPR replication degree when network fluidity increases, hence the three policies below:

{MN&&CH}[fm_event] if {FM=[0..25)} then {ReplDegState:= 1:Single}

{MN&&CH}[fm_event] if {FM=[25..70)} then {ReplDegState:= 2:Selective}

{MN&&CH}[fm_event] if {FM=[70..100)} then {ReplDegState:= 3:Full}

Based on the collected hypercluster-wide information (in this case the FM), the CDP of each CH informs the collocated PDP and policies of this type may be triggered for hypercluster-wide enforcement. Once triggered, their respective actions are enforced only to the PEP of the hypercluster nodes. The formed adaptation loop ensures the correct replication state is enforced depending on perceived network fluidity among hypercluster nodes.

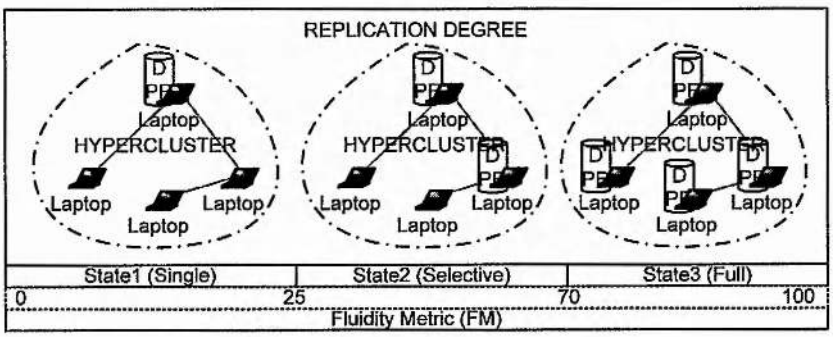

Figure 4-2. Replication States of the Policy Repository

The importance of a reliable and robust DPR has motivated the decision to choose this policy type for further analysis and implementation. In §5.2 a step-by-step methodology is provided to assist in the design and implementation of policies. Based on the described concepts, a detailed investigation of DPR Management is presented in §5.3. By exploiting LDAP synchronisation features, a highly customisable deployment of a DPR overlay can be formed. This is possible by extending DPR management policies to combine a-priori knowledge of localised events (e.g. scheduled sport event) with dynamic real-time context information (e.g. processing load or free memory of each PDP).

3. Energy conservation policy with Cluster-wide enforcement scope

A major issue in MANET is the conservation of device resources, hence a policy-based attempt to tackle the problem is presented. A policy type is introduced that adaptively configures node's energy consumption according to current state and environment as well as the overall management objectives:

{CN&&CH}[E] if {BP=(n..m)} then {TransPow:=k}

This policy type is used to efficiently manage devices' resources by influencing relevant configuration parameters. The Battery Power (BP) context is the average percentage of remaining battery power among cluster nodes. It is used here to affect the transmission power (TransPow) of cluster nodes. For implementation k ={1,2}, where 1=Normal Power and 2=Low Power, therefore two policies are implemented:

{ CN&&CH }[bp_event] if {BP=(0..33]} then {TransPow:= 2:Low Power)}

{ CN&&CH }[bp_event] if {BP=(33..100]} then {TransPow:= 1:Normal Power }

The idea is to use a threshold average battery level in order to reduce transmission power and conserve remaining battery power. Policies of this type only need cluster-wide context knowledge since their enforcement is independent among clusters. The PDP of every CH receives context information for the registered variables and enforces the actions to all PEP (CN) within its cluster. Periodic receipt of individual BP context subsequently generates periodic bp-event, causing the evaluation of the two conditions and triggering of respective actions. In these cases, context information is withheld within the cluster, thus reducing overall traffic load and processing resources.

The effect of this policy is battery power conservation, since one of the main energy consumers of mobile devices is actually their wireless transceiver. The cluster-wide, instead of per node, enforcement of power reduction is necessary to avoid asymmetric wireless links among cluster nodes. In practise, this policy is better suited for relatively dense network deployments, to avoid node disconnection with their CH. The reduction of transmission power causes a reduction of transmission range that may result in one way link breaks from CN to CH as well as two-way link breaks between CN. To anticipate potential disconnection and asymmetric links, additional conditions may be added to policies depending on network deployment parameters.

4.3 Multi-manager environment and policy analysis

One of the major concerns of IT community regarding the entrustment of management to policy-based solutions is the likelihood of policy conflicts and the risk of inconsistent configurations. Research community has investigated policy analysis issues including detection of policy conflicts and most importantly their resolution. Therefore, critical issues of policy analysis were

examined as they are closely related to the designed PBM framework. The work presented in this section attempts to integrate an automated conflict detection and resolution mechanism to the framework, focusing mainly at the higher policy hierarchy level, i.e. conflicts arising between multiple managers. While conflicts may occur at all policy levels, conflicts were investigated at the level of managing entities, because such solutions can significantly increase the applicability of the proposed policy-based framework.

The motivation behind a multi-manager paradigm has been explained in §3.4. This section further elaborates on its applicability, by focusing on issues of policy analysis. To support the operation of multiple managing entities, a conflict detection and resolution tool was integrated, implementing a manager communication protocol that ensures a conflict-free system operation. The presented research efforts were focused on a case study to better illustrate concepts and provide realistic examples. In order to assist the investigation of multiple manager environments, the case study of "urban spaces" was introduced. "*Urban spaces*" are a subset of the general case of *ubiquitous computing*, defined as the complex networked environments deployed in urban centres employing fixed and wireless devices owned by users and operators [4]. This definition matches the general definition of wireless ad hoc networks (§2.3.1) and closely resembles mesh network deployments [41],[42]. Urban spaces can be seen as the convergence of fixed and wireless networks, where the majority of devices are individually owned and controlled by users (Figure 4-3).

Let us consider a network formed by the infrastructure of a network operator (NO) and the devices of individual users, as illustrated in Figure 4-3. The excess of user-owned wireless devices differentiates such networks from the ones where established concepts have been applied, e.g. mobile/cellular networks or the Internet. The operator's infrastructure may include media servers, information kiosks, traffic cameras etc. User devices may include mobile phones, laptops, PDAs, as well as home network devices like TVs or media players. For this case study, it is assumed that the network operator has agreements with independent service providers, who may use the network infrastructure to offer different services to the users. Based on the aforementioned policy-based design and organisational model, a novel approach is proposed to manage the whole network and allow more than one entities to cooperatively perform management tasks.

With the adoption of a *multiple manager paradigm* both the network operator (NO) and the service providers (SP) can exploit the network by introducing their own policies, while a conflict detection and resolution mechanism is in place. As it will be explained in an example, users participating in the network are willing to share resources and consume available services. These shared resources can be utilised either by the NO or by the SP for different purposes, hence there is a high possibility of conflicting interests. To alleviate such issues and maintain smooth network

operation, each managing entity is allowed to specify its own policies to affect network operation, while the introduced conflict detection and resolution mechanism (CDRT) ensures consistency.

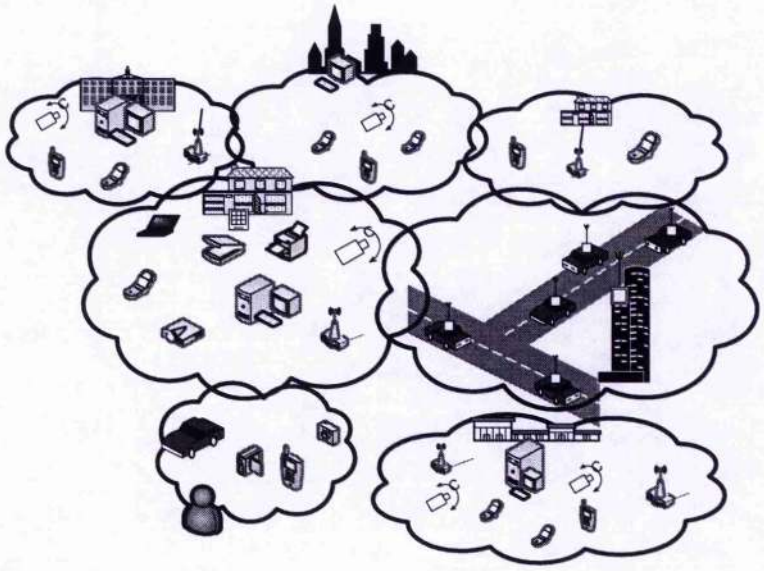

Figure 4-3. Urban Space as a subset of Ubiquitous Computing

4.3.1 Conflict detection and resolution

To complement the design of a policy-based self-management framework, a mechanism for the automated detection and resolution of policy conflicts was employed. Conflicts can be generally classified as dynamic and static (§2.4.1,pp.39). A number of static conflicts may arise in the policy specification, like modality and mutual exclusion conflicts, conflicts of duty and multiple manager conflicts. This work focused on the latter type and addressed the inconsistencies that may occur due to the adopted multi-manager paradigm. The novelty was focused on the integration and extension of existing policy analysis methods rather than the investigation of new formal techniques for policy analysis. Based on the introduced case study, a detailed conflict detection and resolution example is presented in the next subsection.

The presented case study aims on one hand to demonstrate application-specific methods for conflict detection and resolution and on the other to alleviate problems arising from the multi-manager paradigm. Since more than one manager may have different management objectives, it is essential to detect and resolve any conflicts among managers to avoid inconsistencies. The policy-based multi-manager paradigm is ideal for the case study. By applying the defined roles and organisational model, an automated conflict detection and resolution mechanism is integrated, aiming to make a first step towards a self-managed policy analysis solution.

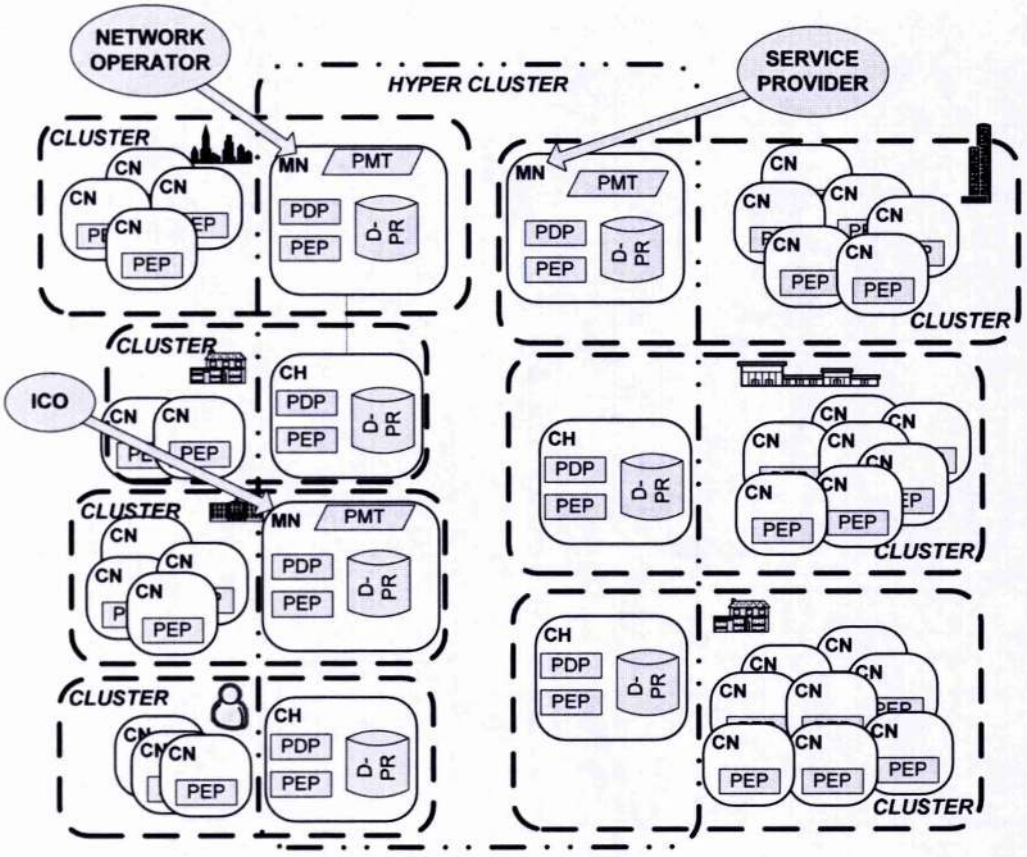

Figure 4-4. Organisational model in an urban space

For the examined case study of urban space networking, *eligible entities* refer to authorised management entities able to introduce their management objectives through policies. Such *eligible entities* include network operators (e.g. mobile networks operators), service providers (e.g. multimedia providers), local authorities (e.g. tourism office) and data protection agencies (e.g. ICO). In this scenario, the assignment of nodes to the role of a Manager Node (MN) needs to be static, in order to ensure that the authorised *eligible entities* always control MN. To demonstrate the concepts, three entities with competing interests in managing the network were chosen: a network operator (NO), a service provider (SP) and a data protection agency (ICO). The participation and role of a data protection agency is related to regulatory policies and user protection, as will be clarified in subsequent sections (§6.3,pp.146). Figure 4-4 displays the deployment of the proposed organisational model in the urban space depicted in Figure 4-3. Each cloud of devices from Figure 4-3 forms a cluster. The three eligible entities (MN) and the local cluster leaders (CH) form the *hypercluster*. The rest of the devices take the simplest role (CN). At the top hierarchy level, managers (MN) need only high-level information and do not need to know about the specifics within each cluster. Their interactions in this multi-manager scenario are explained in following subsection.

4.3.2 Inter-manager conflicts

As emphasised, every policy-based system inevitably needs to deal with arising policy conflicts. Therefore the proposed PBM framework has been enhanced with a conflict detection and resolution (CDR) mechanism. Although several conflict types can be identified with regard to the examined application domain, interest focused on conflicts arising between policies originating from different managing entities (MN), as these are closely related to the adopted multi-manager paradigm. These conflicts are referred to as *inter-manager conflicts*.

The proposed CDR mechanism is part of a communication protocol between manager nodes (MN). The protocol defines the procedure for policy updates with conflict detection and resolution and ensures the consistency of the Distributed Policy Repository [4]. This is presented by the sequence diagram in Figure 4-5. In this case study, three *eligible entities* cooperatively manage the network: the service provider (MN1), the network operator (MN2) and a data protection agency (MN3). The procedure is the same for any number of manager nodes.

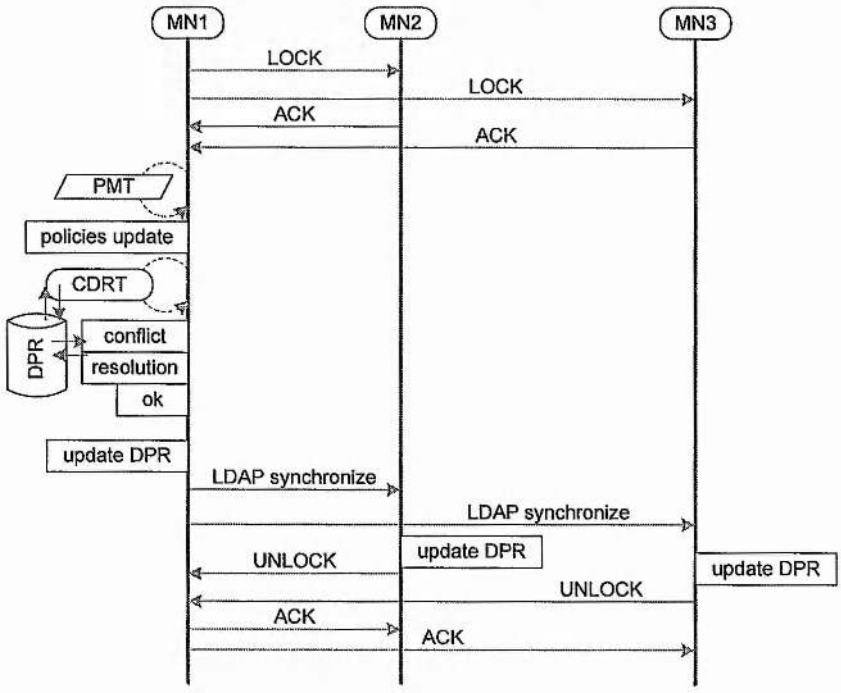

Figure 4-5. Sequence diagram for policy updates

For the introduction or editing of policies in the system, a MN must send a LOCK message to all other MNs to ensure that no concurrent policy changes occur and ensure the consistency of the Distributed Policy Repository (DPR). Once acknowledgements (ACK) are received the initiating manager can use its Policy Management Tool (PMT). Using the CDR Tool, all new or changed policies are analysed locally for conflicts, based on a set of global detection rules that eligible entities have agreed upon and specified a priori. In the event of a conflict, resolution can be

achieved in different ways depending on the conflict type, the entities involved and any prior agreements between managing entities. Once CDRT has verified the consistency of all policies, the initiating MN can update the DPR, which will automatically propagate changes to other MNs. Once the Manager Nodes have updated their DPR, they reply with an UNLOCK message to the first MN to confirm changes. The MN that initiated the changes sends ACK to all MNs which release all PMT for further policy updates.

The occurrence of inter-manager conflicts lies in the fact that each manager has its own high-level objectives which are expressed by different policies. Inevitably, these policies may contradict because of incompatible management interests. An illustrative example is provided below which describes such situations and serves as proof of concept for the proposed method of conflict detection and resolution.

Proof of concept example of conflict detection and resolution

Let us consider a scenario where two managers want to configure users' devices that are located in a specific area with low bandwidth availability and high user density, e.g. a stadium. In the examined case study, a service provider (SP or MN1) specialising in media delivery wants to maximise services utilisation by providing media to as many users as possible. The network operator (NO or MN2) on the other hand, monitors the network to discover bottlenecks and ensure its stable operation by configuring infrastructure devices and user-owned ones.

According to this scenario, user devices support packet forwarding and together with a limited number of NO nodes form a multihop network; a typical case of wireless ad hoc networks. In addition, users are willing to share some of their bandwidth in exchange for connectivity and services, by allowing policies to configure relevant managed object, e.g. the shared bandwidth (SBW). For simplicity, let us assume that managers are only interested in configuring how the shared bandwidth (SBW) of users is utilised. Being lightweight in capabilities, user devices do not implement any QoS traffic classification and prioritisation. Instead the have a simple scheduling mechanism that utilises a set percentage of shared bandwidth for management traffic and the remaining for forwarded user traffic. Hence, SBW value is divided in bandwidth for management (mngBW) and bandwidth for forwarded user traffic (p2pBW).

Both managers (NO and SP) want to achieve their objectives by configuring infrastructure devices (access points, information kiosks) as well as user devices (mobile phones, PDAs). The network operator's policy is to use most of the shared bandwidth for management purposes because a stable network is more important than providing services. Hence its policy should declare that management traffic receives more percentage of shared bandwidth. Using the PMT at MN2 the following policy (p1) is composed that sets SBW to 40% of which 30% will be used for management traffic and routing data and 10% for peer-to-peer and forwarded traffic:

{CN} [newUser] *if* { locateUser(Stadium) }
then {setBW((SBW:=40%),(mngBW:=30%),(p2pBW:=10%))} (p1)

The service provider on the other hand wants to utilise the users' shared bandwidth for distributing media and content (e.g. advertisements, video replays) among customers and through policies defines more shared bandwidth for forwarding traffic. To realise these goals, the PMT at MN1 is used to formulate the following policy (p2) that sets SBW to 60% of which only 20% will be used for management traffic and routing data and 40% for forwarded user traffic:

{CN} [newUser] if { locateUser(Stadium) }
then {setBW((SBW:=60%),(mngBW:=20%),(p2pBW:=40%))} (p2)

Since both policies are triggered by the same event (i.e. the entrance of a user to the stadium area), obviously they would be triggered simultaneously. Simultaneous policy triggering is acceptable and is not a conflict as such. The problem arises from their action part, because they attempt to modify the same managed objects. The two policies are conflicting since they both aim at configuring the same resource with inconsistent parameters. This is a specialisation of an inter-manager conflict that needs to be addressed.

For this example, it is fairly obvious when and why a conflict arises since both policies affect only three managed objects. Nevertheless, the PBM system needs a conflict detection mechanism as a first step of policy analysis. Conflict detection is an open research issue and is tightly related to the policy language used and its expressiveness. For this case study, static conflict detection is addressed, i.e. policies are analysed during their introduction to the PBM system. The defined policy update protocol in Figure 4-5 depicts a sequence diagram that serialises policy introduction, reducing the complexity of concurrent policy changes. As said, all managers have access to the Distributed Policy Repository that offers a uniform view of active policies. During policy editing, the CDRT uses conflict detection rules to analyse edited or new policies together with existing ones [119],[120]. These rules are special policies triggered by policy changes and their respective implementations reside within the CDRT. For the example policies p1 and p2 above, the conflict can be detected with a rule of the following form:

```
if {[p1.setBW(SBW1, mngBW1, p2pBW1) ^
   p2.setBW(SBW2, mngBW2, p2pBW2)] ^
  [(SBW1 != SBW2) v (mngBW1 != mngBW2) v
  (p2pBW1 != p2pBW2)] ^
   p1.locateUser(_) == p2.locateUser(_)}
then {signalConflict(BWAlloc(p1, p2))}
```

The condition part of this rule deals with the detection of a conflict and the action part triggers the resolution process. There are different possible solutions to address conflicts:

- The first solution would be to notify affected managers (MN1,MN2) about the conflicting policies and await for them to edit and reintroduce their policies. In this case, neither policy *p1* nor policy *p2* are enforced, until managers agree on common policies. This resolution process is manual, requiring human intervention and effort. The result would affect the management objectives of both the network operator and the service provider, inducing delays and overriding the benefits of using a PBM system.
- A second solution would be to prioritise policies depending on their origin and execute the one with the higher. E.g. assign higher priority to network operator policies compared to other managing entities. In this case, policy *p1* is enforced and policy *p2* is ignored. This solution can be considered automated since human intervention in not required when conflicts arise, although an agreement on policy priorities is needed beforehand. In case of a conflict, only the objectives of the manager with higher priority will be satisfied.

The presented solution uses and extends an automated resolution process [119],[120], aiming to better satisfy the management objectives of all involved entities. The basic concept is to replace the conflicting policies with a single policy that combines the interests of involved policy makers. Specifically, a resolution action set for each conflict type is agreed upon and pre-specified by the managing entities. This action is triggered when the conflict has been detected and acts as a mediator between the managers' objectives.

In this example scenario, the actions of the following rule, allocate a weighted average value for conflicting objects, based on the values provided by the two initial policies:

```
if {signalConflict(BWAlloc(p1, p2))}
then {setBW((SBW:= p1.getSBW * 0.6 + p2.getSBW * 0.4),
    (mngBW:= p1.getmngBW * 0.6 + p2.getmngBW * 0.4),
    (p2pBW:= p1.getp2pBW * 0.6 + p2.getp2pBW * 0.4))}
```

The value of the weights used in the averaging process depends on the contractual agreement between management entities and the business model of the managed network. In this example the network operator policy values have a weight of 0.6 while the weight for service provider policies is 0.4. This agreement reflects the importance of maintaining a stable network in an area with limited connectivity and gives more bandwidth to management traffic.

The conflict resolution rule above automatically constructs the following policy that is applied immediately and through the DPR is propagated to the other managers:

```
{CN} [newUser]      if {locateUser(Stadium) }
                    then {setBW((SBW:=48%),(mngBW:=26%),(p2pBW:=22%))}
```

In summary, a manager communication protocol ensures serialised introduction and editing of policies. At the same time, the CDRT takes automated conflict detection and resolution decisions

using the presented solution, provided the parties involved have previously agreed on the following:

1. Detection rules: specify system parameters and policy actions of interest to agree on when a conflict occurs and which resolution procedures should be followed.
2. Resolution rules: specify contractual agreements for each conflict type and translate them to weights or parameters to be included in resolution rules.

The use of different resolution methods is also possible, e.g. policy priorities etc. Compared to the aforementioned ones, the proposed resolution methods is considered superior because it provides an automated resolution mechanism that does not require human intervention and in addition satisfies the interests of all managers involved, based on their contractual agreements.

4.4 Summary and Conclusions

To anticipate the diverse needs of wireless ad hoc networks, a custom lightweight policy notation was employed, based on the established *Event-Condition-Action (ECA)* notation and existing IETF/DMTF specifications. Because of the increasingly heterogeneous and lightweight nature of target devices, a simplified policy language and representation allows to the majority of devices to participate in a PBM network and contribute to its collaborative management. Beyond the corresponding role-based policy hierarchy, the concept of *policy enforcement scope* has been introduced and examined, further assisting the layered closed-control loop. By using three examples taken from a realistic wireless ad hoc network case study, the formation of three closed control loops has been illustrated at the network-wide, hypercluster-wide and cluster-wide layers.

Issues of policy analysis were investigated, in order to support the cooperation of multiple managing entities. By designing a manager communication protocol that integrated an automated conflict detection and resolution tool (CDRT), the conflict-free operation of multiple managers can be ensured. CDRT warrants the consistency of the Distributed Policy Repository by using a protocol for the communication of manager nodes (MN). The presented methodology specifically addressed *inter-manager conflicts* and was demonstrated through a proof of concept example of automated policy conflict and resolution. The proposed resolution method provided an automated resolution mechanism that did not require human intervention and in addition satisfied the interests of all managers involved, based on their contractual agreements.

Chapter 5

Policy Implementation and Distributed Policy Repository Management

5.1 Introduction

Beyond policy notation, a suitable system representation should be used to allow policy processing and management. The standardised by IETF information and data models for policy representation were adopted and customised for the PBM of wireless ad hoc networks, aiming to build on existing concepts and maintain interoperability. The Policy Core Information Model (PCIM) and its extensions (PCIMe) are defined in RFC3060 [204] and RFC3460 [207] respectively. In addition, these information models are converted to concrete system implementations, based on their mapping to Lightweight Directory Access Protocol (LDAP) data model, as described in RFC3703 [211] and RFC4104 [212], also standardised by IETF. In the following subsection these decisions are further explained and more details are provided on the followed specification and mapping procedure. For clarity, a policy design example is also provided in §5.2.1.

The introduction and use of a five step methodology can represent and implement complex policy functionality in a straightforward and methodical manner. By building on existing standards the methodology results in future-proof, interoperable policies that encapsulate management logic and objectives in technology-independent representations. Technology-dependent implementation details, like storage and action enforcement, are also analysed exhaustively by providing concrete LDAP mapping guidelines and a working implementation based on examples. The mapping procedure from a generic Information Model representation to a solid implementation-ready Data

Model format is detailed below. The outcome of this methodology is compatible with all popular LDAP Directory Servers and it has been tested on OpenLDAP DS.

The ***Distributed Policy Repository*** (*DPR*) was designed as an extension of the traditional PR, to tackle identified policy distribution and storage problems. *The **DPR** is a physically distributed set of components consisted of interconnected directories hosted on selected hypercluster nodes.* Instead of simply replicating the PR among network nodes, a sophisticated policy-based replication scheme has been incorporated. In essence, DPR is responsible for the distribution of policies in the network and for logically connecting the devices that collaboratively participate in management. The DPR component was deemed necessary for the management of wireless ad hoc networks, such as user-owned networks, because of their spontaneous nature and the different ownership relation between networked devices and the network manager. The motivation for a DPR lies in the need for provisioning large-scale wireless ad hoc networks without the need for over-provisioning management resources, e.g. access points, bandwidth or human effort. Because the deployment of such networks varies significantly in terms of spatial and temporal parameters, accurate planning and pre-provisioning is extremely difficult. Hence the proposal for distribution of management tasks among PDPs, where PDPs are hosted on user devices and use policy guidelines stored in the DPR.

5.2 Policy Representation and Implementation

The presented design was based on standardised models developed within IETF, aiming for increased interoperability and standards compliance. As explained, IETF standardisation efforts had initially focused on the development of an Information Model and PBM Framework, something that has allowed the establishment of a technology-independent common ground for policy design and specification.

IETF/DMTF's Information Model is specified in two RFC Standards Track documents: RFC3060 for *Policy Core Information Model* (PCIM) [204] and RFC3460 for its extended version (PCIMe) [207]. For the rest of this work, the acronym *PCIMe* is used to indicate both models. As already mentioned, a missing element from IETF's PCIM model is an explicit triggering mechanism which would make the system event-driven. This is important in a policy-based system, since the generic policy rule *event-condition-action* (ECA) is widely accepted [17],[18],[102]. To overcome the lack of an event notation in PCIM, the abstract Policy element is extended as PolicyEvent, without loss of interoperability. By combining and grouping simple policy rules, complex policy structures can be formed (e.g. policy groups), leading to an increasingly complex policy-based design, as well as allowing the reuse of both policy Conditions and Actions.

As an example, a previous policy type from §4.2.2 was implemented for this Chapter. The policy type was used to model policies that drive the placement and replication degree of the Distributed Policy Repository (DPR). The defined policies were represented according to PCIM/PCIMe information model and complied with the standardised class hierarchy. When necessary, new classes were defined to accommodate the needs of wireless ad hoc design. As it will be further explained, after the Information Model (IM) representation of the defined policies is complete, the mapping to Data Model classes followed. For this purpose, the LDAP Data Model was used, as defined by IETF in two RFC Standards Track documents: RFC3703 for Policy Core LDAP Schema (PCLS) [211] and RFC4104 for Policy Core Extensions LDAP Schema (PCELS) [212]. These schemas were extended to cover the custom-made classes of the Information Model. In the following subsections, a step by step methodology for designing and implementing policies is presented. In spite of the uncomplicated nature of introduced policies, this example serves as a hands-on guide for policy designers and system developers.

5.2.1 Policy design and implementation methodology

For the actual deployment of a policy-based management framework, a step-by-step methodology is provided to assist in the design and then the implementation of policies. Most of the presented methodology is not limited to the case of wireless ad hoc networks but is generic enough to apply to various domains. *The essential benefit of using this methodology is the ability to create lightweight technology-independent policy specifications that can be fully interoperable with full-fledged PBM systems. This work fills the gap between existing specifications/implementations oriented towards fixed networks with adequate power and the need for specifications suitable for the emerging wireless ad hoc paradigm.* Towards this direction, models for application-specific areas may extend the Policy Model in several ways. The preferred way according to [204],[207] is to use PolicyGroup, PolicyRule and PolicyTimePeriodCondition classes directly, as a foundation for representing and communicating policy information. For this reason the generic definitions of IETF are followed, allowing the specification and customisation of new policies by creating subclasses of existing objects defined in PCIM.

The five steps of proposed methodology are listed below and are explained through an example case study:

Step 1: Requirements gathering and system description
Step 2: Policy type design and definition
Step 3: Policy Information Model Representation
Step 4: Mapping the Information Model to the Data Model
Step 5: Implementation, Deployment and Testing

The example case study deals with the management of the Distributed Policy Repository (DPR) in an ad hoc environment and modelled policies drive its replication degree and replica placement. As described later in detail, the *DPR* is a physically distributed Policy Repository, which consists of a number of repository replicas, placed on selected network nodes that form the hypercluster. One or more replicas may exist depending on network purpose and node mobility. For management purposes, different replication states of DPR are allowed and a manager has the ability to dynamically define the behaviour and the replication degree of the DPR by introducing related policies on the fly and without shutting down the system or the DPR component. Through the five steps described below, the whole implementation procedure is followed from requirements gathering to implementation and system deployment.

Step 1: Requirements gathering and system description

The first step is to gather the requirements of the managed system and express the management goals to be achieved. According to the example case study, it was desirable to design policies that would allow efficient and robust deployment of the DPR component in a wireless ad hoc environment. The high-level management goal was defined:

> *Depending on network's volatility, the system should automatically decide on appropriate Distributed Policy Repository deployment to maintain efficient policy distribution and provisioning*

The issues stemming from this goal and the target environment are listed here and need to be addressed:

(1) PDP may be intermittently connected to the ad hoc network but should maintain contact with the PR

(2) The nearest PR instance may be several hops away from PDP, thus introducing significant traffic and latency overhead to the propagation of new or updated policies.

(3) Multihop networks suffer from severe bandwidth degradation as the number of hops per route increases.

(4) Wireless ad hoc networks exhibit spatiotemporal density fluctuation in PEP population.

(5) Wireless networks are increasingly consisted of heterogeneous end-user devices.

Having in mind the above issues, the system was modelled to be in one of three possible states, with respect to the replication degree:

(i) Single replication: At this state the ad hoc network is considered as relatively static, i.e. node mobility is low and the link quality is fairly good. Therefore all PDP of hypercluster nodes can efficiently retrieve policies from a single PR master copy.

(ii) Selective replication: At this state the ad hoc network volatility is increased, i.e. node mobility causes frequent link breakage and the link quality is fair. Additional DPR replicas are instantiated in critical points within the hypercluster to reduce bandwidth utilisation and increase efficiency of policy retrieval.

(iii) Full replication: At this state the ad hoc network is considered as extremely volatile, i.e. node mobility is high and the link quality is very poor. Therefore all hypercluster's nodes need to keep a local DPR replica in order to efficiently retrieve policies and provision their cluster with them.

A graphical representation of these states was shown in Figure 4-2 (pp.86). It should be noted that these policies are applied only within the hypercluster nodes, as indicated by their hypercluster-wide enforcement scope and their assignment to {MN&&CH} roles (§4.2.1,pp.81). To facilitate the selection of the replication state, an appropriate metric was required to facilitate management goals, i.e. express the volatility of the network. Therefore, a new scalar metric was also defined: the *Fluidity Meter* (FM), which characterises how fluid and volatile the ad hoc network is. It ranges from *minFM* to *maxFM*, with bigger values representing higher fluidity. This metric can be extracted from collected network and context information.

Based on the above, the system representation was described by three replication states (Single, Selective, Full) and a scalar contextual metric (FM). Using this information, management goals can be expressed in policies.

Step 2: Policy type design and definition

Having gathered the required information, management goals can be expressed in ECA policies. Intuitively, policies will define in which state the network should be, by defining the limiting values for each state. PDP will enforce the defined actions by monitoring the Fluidity Meter (FM) of the network and checking the conditions. Therefore, the generic ECA policy type can be parameterised in a specific policy type that would control DPR replication:

{ MN&&CH}**[E]** if (FM = {**x** .. **y**}) then (Repl_Deg := **n**)

To instantiate the above policy type, the bold parameters (E,x,y,n) need to be specified. Since the system specification had defined three states, three different policies were needed to control the system. Two limiting values (*LowLim* and *HighLim*) form three mutually exclusive conditions. Hence, the three policies defined were the following:

{MN&&CH}[checkFM] if (FM = {MinFM .. LowLim}) then (Repl_Deg := State1)
{MN&&CH}[checkFM] if (FM = {LowLim .. HighLim}) then (Repl_Deg := State2)
{MN&&CH}[checkFM] if (FM = { HighLim to MaxFM }) then (Repl_Deg := State3)

The change of the replication state was guided by the above policies. The roles assigned for these policies were those of MN and CH, i.e. the devices belonging to the *hypercluster*. The periodic

triggering event *checkFM,* caused the evaluation of the three mutually exclusive conditions, leading to the appropriate state of replication degree. These policies would be introduced in the system by a managing entity that controls a MN, using the interface of PMT.

Step 3: Policy Information Model Representation

The aforementioned three policies, which guide the DPR replication state, were represented using IETF's information model. PCIMe provides a vendor and language independent way to represent policies. Use of these standards allows flexible and extensible policy modelling, regardless of the implementing technology. This model defines two hierarchies of object classes: *structural* classes encapsulate information for representing and controlling policy data, while *relationship* classes indicate how instances of the structural classes are related to each other. A part of PCIMe's class hierarchy was used for the presented case study (Figure 2-4, pp.41).

As pointed out earlier, PCIMe does not incorporate the notion of *events* and is based on the limited notation of if {conditions} then {actions} for policies. To alleviate this deficiency and apply the ECA notation and its benefits, PCIMe was extended by creating new classes to represent events. An abstract **PolicyEvent** class was created first and in addition a structural subclass: **SimplePolicyEvent** was extended (Figure 5-1). The new classes can be used to represent events and can be attached to PolicyRule or PolicyGroup instances as required. To allow this, a relationship class was also modelled: **PolicyEventInPolicy**. These concepts and their realisation are shown in Figure 5-2 and also in an example instantiation in Figure 5-3. As with previous definitions, this event specification is implementation-independent and the realisation of an event bus or event notification mechanism is a separate topic. The important issue here is the maintained interoperability since IETF directives were followed for the extension of PCIMe.

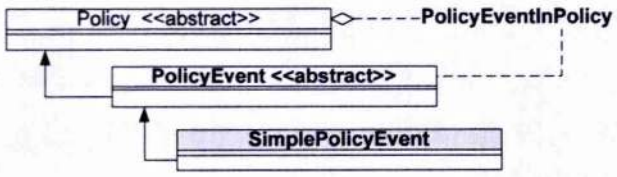

Figure 5-1. Extended PCIMe classes to support event representation

For the representation of domain-specific variables, PCIMe suggests the extension of PolicyImplicitVariable and the use of class inheritance mechanism. The reasons are thoroughly explained in PCIM ([204]: §5.8.9) and the main argument given is the ease of extension and better clarity. Based on that, two new classes were defined: PolicyFMVariable and PolicyReplDegStateVariable. Finally, SimplePolicyEvent class was extended to cater for the required checkFM event as EventCheckFM. The used classes and their relationships are shown in Figure 5-2. Based on this figure, the methodology of representing the defined policies to PCIMe classes is detailed.

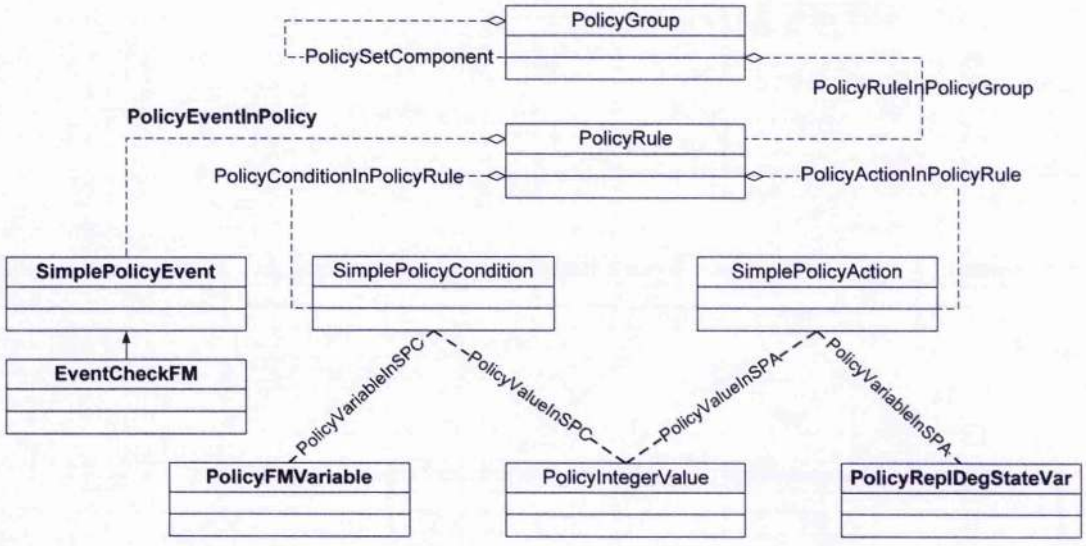

Figure 5-2. Class Hierarchy and Relationships for example policies

Figure 5-2 shows the new implemented classes (bold typeset) needed to accommodate events and two new variables. Both variable classes are subclasses of PolicyImplicitValue. The PolicyFMVariable class represents the Fluidity Meter (FM) variable that appears in the conditional part of the three defined rules. The PolicyReplDegStateVar represents the replication degree state of the DPR component which appears in the action part of the rules. Both variables accept integer values therefore PolicyIntegerValue class was reused. Using the SimplePolicyCondition and SimplePolicyAction classes the condition and action part of a rule were realised respectively. Finally, by adding SimplePolicyEvent class and its extension EventCheckFM, the missing event element was introduced to complete the definition of ECA policies. According to PCIMe, four self-explanatory associations (relationships) are used to glue the above classes together:

- PolicyValueInSimplePolicyCondition
- PolicyVariableInSimplePolicyCondition
- PolicyValueInSimplePolicyAction
- PolicyVariableInSimplePolicyAction

Three more relationships are needed to link an event, a condition and an action to an actual policy rule. Therefore PolicyEventInPolicy, PolicyConditionInPolicyRule and PolicyActionInPolicyRule were used to aggregate classes in common PolicyRule class. Obviously a PolicyRule class is the representation of a policy rule in PCIMe Information Model, accompanied with the relevant events, conditions, actions and their values. To clarify and apply the above, an example policy rule instantiation is presented in Figure 5-3:

{MN&&CH}[checkFM] if (FM = {LowLim .. HighLim}) then (Repl_Deg := State2)

The rule instance above is named "SelectRep" and is shown with all its accompanied class instances. This rule, together with "SingleRep" and "FullRep" (not fully shown in Figure) are members of the "DPR Management" PolicyGroup that logically groups policies related to the management and replication of the DPR component. In addition, PolicySet specification includes a

"PolicyRoles" property, which was used to define the {MN&&CH} roles for this group of hypercluster-wide policies. Similarly the other two policies defined in the previous step (SingleRep and FullRep) can be represented as PCIMe objects.

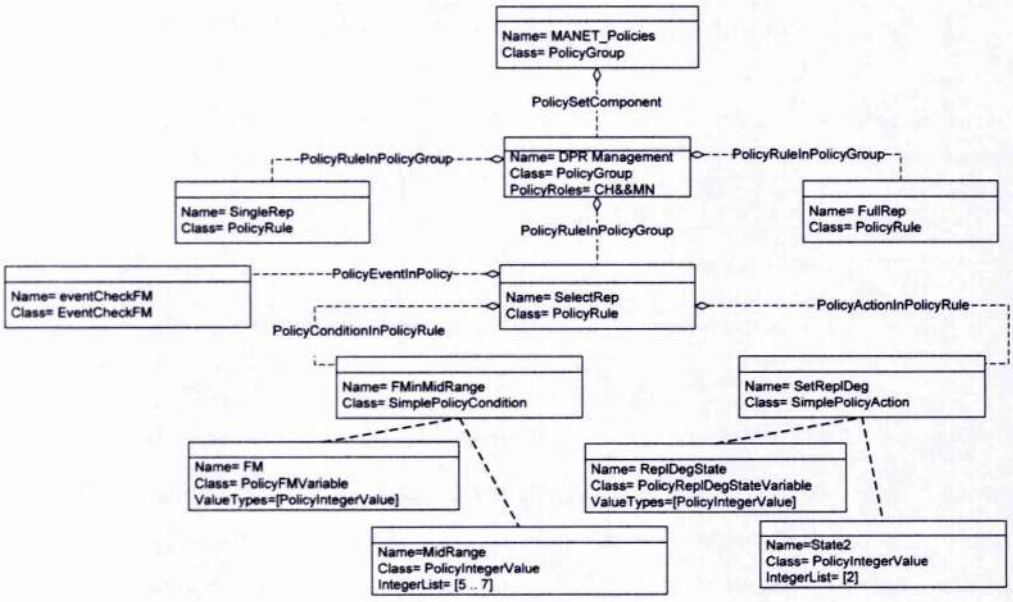

Figure 5-3. Example Policy Rule Instantiation

Step 4: Mapping the Information Model to the Data Model

The 4th step involves the mapping of designed classes from the Information Model to a concrete Data Model. This step is more related to policy storage and requires the selection of an appropriate technology to implement the actual Policy Repository. The main task of this step is to define new Data Model classes to map any new PCIMe classes defined previously.

Following IETF recommendation and based on the analysis of LDAP capabilities and features, the decision was made to use the LDAP Directory Server for the implementation of the policy repository for this framework. A brief overview of LDAP protocol is provided in Appendix B. The mapping between the PCIMe Information Model to LDAP Data Model is guided by two Standards Track RFC: Policy Core LDAP Schema (PCLS) [211] and Policy Core Extensions LDAP Schema (PCELS) [212]. These RFC define a collection of all "objectclass" and "attribute" LDAP definitions, constituting the LDAP schema that a Directory Server uses to verify directory entries. Surprisingly, a usable format of these two schemas was not available, therefore they were gathered in two new schema files (PCELS.schema, PCLS.schema) and where made available publicly [154]. These files are compatible with the majority of existing LDAPv.3 DS. In addition, IETF guidelines were followed to extend these schemas, in order to include the new classes needed for the customised PBM framework design.

To maintain interoperability and global uniqueness, all new LDAP classes and attributes should use a unique Object Identifier (OID). "Base OIDs" are assigned by naming authorities (e.g. Internet Assigned Numbers Authority, www.iana.org) and beyond their typical use in SNMP MIB configurations they are also used to extend LDAP Schemas. After an application to IANA, a Private Enterprise Number had been assigned (#30895,[155]) with OID base 1.3.6.1.4.1.30895 and was used for experimental implementation of the proposed schema extensions. To avoid "*OID hijack*" ([19]:pp.338), policy designers following the proposed methodology should use their organisation's OID or apply for a new one to a naming authority (Appendix B).

For clarity and readability, the special prefix wah- was defined and used with all new definitions. Prefix **wah-** stands for *wireless ad hoc*. All new definitions were included in a new schema file (`EXT.schema` [154]) and can be used by existing LDAPv3 DS. This provides an easy and straightforward extension mechanism for new policy types. In addition, a convenient LDAP configuration directive was used to give a symbolic name to the assigned long OID base, thus facilitating easy reuse and extension of examples.

The lack of events in PCIMe is also reflected in PCELS. Therefore, in order to represent events, the customised PCIMe extensions were also mapped to LDAP data model. Following IETF's methodology, class SimplePolicyEvent was mapped to three LDAP classes for increased flexibility: **wahEvent** (abstract) **wahEventAuxClass** (auxilliary) **wahEventInstance** (structural). Their definitions follow below, while Figure 5-4 clarifies their relationship with existing classes. Note the use of symbolic OID base wahSchema, to facilitate new OID definitions:

```
objectidentifier wahSchema 1.3.6.1.4.1.30895

objectclass  ( wahSchema:1
    NAME 'wahEvent'
    DESC 'Base class for representing a policy event'
    SUP pcelsPolicySet
    ABSTRACT
    MAY ( pcimGroupName )
  )

objectclass  (wahSchema:2
    NAME 'wahEventAuxClass'
    DESC 'Auxiliary class for representing a policy event'
    SUP wahEvent
    AUXILIARY
  )

objectclass  (wahSchema:3
    NAME 'wahEventInstance'
    DESC 'Structural class for representing a policy event'
    SUP wahEvent
    STRUCTURAL
  )
```

To model the specified EventCheckFM class, a respective structural class **wahEventCheckFM** was introduced to allow its stand-alone instantiation. Its relationship with a specific rule (PolicyEventInPolicy) was modelled by superior-subordinate relationship in the DIT. Variables PolicyFMVariable and PolicyReplDegStateVar were mapped respectively to **wahFMVarAuxClass** and **wahReplDegVarAuxClass** LDAP auxiliary classes. Both classes have pcelsImplicitVariableAuxClass as their superclass. The new definitions follow below:

```
objectclass  (wahSchema:3.1
    NAME ' wahEventCheckFM'
    DESC ' A policy event to represent a check of FM'
    SUP wahEventInstance'
    STRUCTURAL
  )

objectclass  (wahSchema:4.1
    NAME 'wahFMVarAuxClass'
    DESC 'A policy variable representing the Fluidity Meter (FM)'
    SUP pcelsImplicitVariableAuxClass
    AUXILIARY
  )

objectclass  (wahSchema:4.2
    NAME 'wahReplDegVarAuxClass'
    DESC 'A policy variable representing the Replication State: 1=Single, 2=Selective, 3=Full'
    SUP pcelsImplicitVariableAuxClass
    AUXILIARY
  )
```

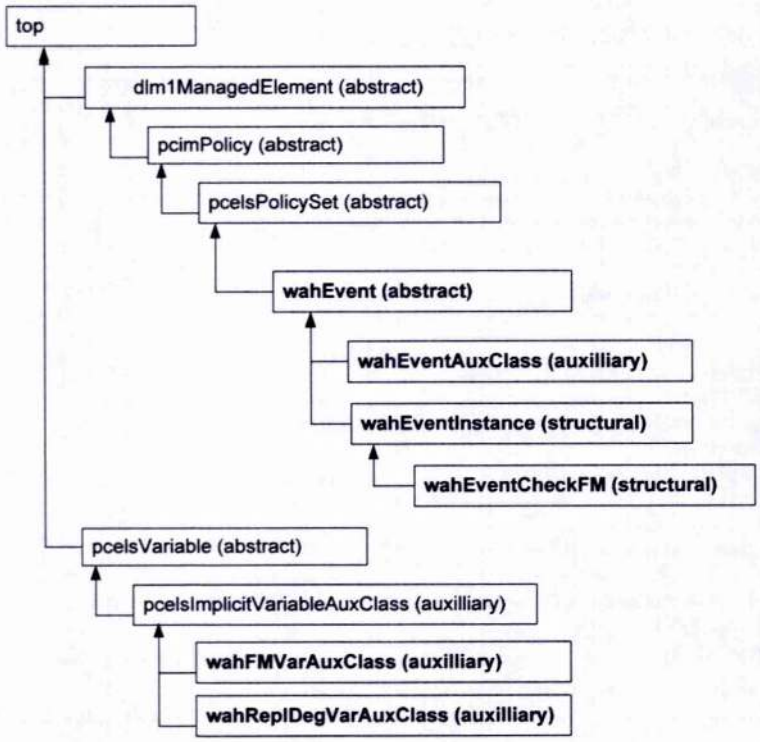

Figure 5-4. Extended PCELS Class Inheritance Tree

Step 5: Implementation, Deployment and Testing

The last step is technology-dependent because it relates to the technology implementing the Policy Repository and the actual storage of policies. Once all LDAP schemas are ready, they can be easily loaded to deployed Directory Servers. This requires their addition to DS configuration files (`slapd.conf`). Based on PCELS and introduced custom extensions, the defined policies can be implemented and mapped to a concrete machine representation that can be stored in an LDAPv3 DS.

An appropriate LDAP Client within the PMT can add the policy entries to a live *directory*, or alternatively they can be preloaded before start-up. Following up the example case study, a sample LDAP Data Interchange Format (LDIF) representation for one of the policies is provided. LDIF text is readable by most LDAPv3 DS as input.

```
dn: cn=DPRmanagement,cn=active policies,dc=ccsr,dc=com
objectclass: pcelsGroupInstance
cn: DPRmanagement
pcimGroupName: Policy group for DPR management

dn: cn= SelectRep,cn=DPRmanagement,cn=active policies,dc=ccsr,dc=com
objectclass: pcelsRuleInstance
cn: SelectRep
pcimRuleName: selective replication rule

dn: cn=CheckFM,cn= SelectRep, cn=DPRmanagement,cn=active policies,dc=ccsr,dc=com
objectclass: wahEventCheckFM
cn: CheckFM
pcimGroupName: Event to initiate FM check

dn: cn=FMinRange,cn= SelectRep, cn=DPRmanagement, cn=active policies,dc=ccsr,dc=com
objectclass: pcelsConditionAssociation
objectclass: pcelsSimpleConditionAuxClass
objectclass: wahFMVarAuxClass
objectclass: pcelsIntegerValueAuxClass
cn: FMinRange
pcimConditionGroupNumber: 0
pcimConditionNegated: FALSE
pcelsIntegerList: 25..70
pcimConditionName: Checks if FM is in the given Range

dn: cn=SetReplDeg,cn= SelectRep, cn=DPRmanagement, cn=active policies,dc=ccsr,dc=com
objectclass: pcelsActionAssociation
objectclass: pcelsSimpleActionAuxClass
objectclass: wahReplDegVarAuxClass
objectclass: pcelsIntegerValueAuxClass
cn: SetReplDeg
pcimActionOrder: 0
pcimActionName: Sets the Replication Degree of the DPR to the appropriate value
pcelsIntegerList: 2
```

The provided LDIF would insert new objects in an LDAP directory, instantiated from structural classes and accompanied by necessary auxiliary classes. First, a containing DPRmanagement policy group was created, for better directory organisation. Then a pcelsRuleInstance object named "SelectRep" was created to represent the defined Selective replication rule. Relationships PolicyConditionInPolicyRule and PolicyActionInPolicyRule were mapped as superior-subordinate

relationships in the Directory Information Tree (DIT), as shown from the DNs (distinguished name) of the last two objects in the LDIF text. Structural classes wahEventCheckFM, pcelsConditionAssociation and pcelsActionAssociation were placed under "SelectRep" DN to form the event, condition and action clause of the given rule. Additionally, condition and action classes have the relevant variables and values attached, in the form of auxiliary LDAP classes. The parsing of the above LDIF from an LDAP DS would produce the DIT in Figure 5-5, given that the "dn: dc=ccsr,dc=com" and "dn:cn=active policies, dc=ccsr,dc=com" objects already existed.

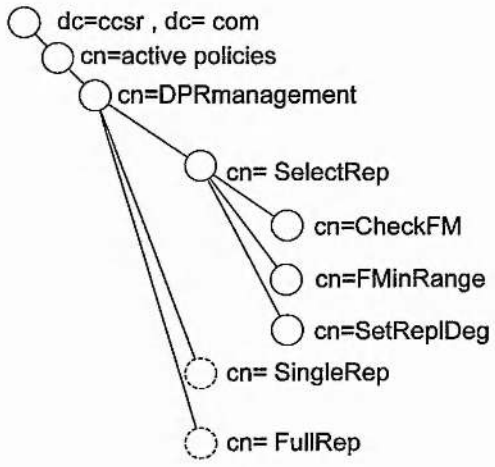

Figure 5-5. Example Directory Information Tree (DIT)

Once policies are stored in the Policy Repository, LDAP clients can query and retrieve them. Test queries can be sent to the PR to check for example *active* policies or *DPR Management* policies. It should be noted that the procedure of policy storage is completed here for centralised PBM systems. However, with the introduction of a Distributed Policy Repository, there are additional management operations and tasks that need to be performed which are detailed in §5.3 below.

5.3 Distributed Policy Repository

5.3.1 Motivation for DPR

The policy repository (PR) is a critical component for every policy-based system. Though traditionally the PR is centralised, PBM systems cannot rely on a single node to store it and use replication to increase its availability. The concept of replication is widely used for backup in case of failures or for load balancing in distributed database systems and commercial directories for fixed networks [19],[21],[132]. However, due to the intermittent nature of wireless links in ad hoc networks, it is expected that nodes will become disconnected frequently and multihop routes will be unstable. Thus access to a central repository cannot be guaranteed depending on the network's

volatility and mobility. On the other hand, the designed policy-based framework provides a highly distributed management environment that can anticipate variable link quality and counterbalance volatility with improved network organisation. Contrary to traditional management systems, the designed system can be deployed for "loose management" of wireless ad hoc networks, in the sense that it does not require the mandatory enforcement of policies and tight control of managed devices. Instead, the system physically and logically distributes the policies among devices, making them available to vast numbers of users that voluntarily choose to enforce the relevant policies that would eventually relieve them from manual configuration. This feature makes possible the configuration and optimisation of user devices with minimum or no intervention. Network operators and service providers can use the policy-based system to introduce the appropriate policies, aiming to set guidelines for the management of numerous user devices. As a result, management logic is encapsulated in policies that are transparently enforced to devices. To achieve the above and tackle identified problems, the Distributed Policy Repository (DPR) is designed, as an extension of the traditional PR:

> ***Distributed Policy Repository (DPR)**: a physically distributed set of components, consisted of interconnected directories hosted on selected hypercluster nodes.*

The introduced ***DPR component*** is different from other framework components, in the sense that its activation is variable and depends on ***DPR management policies*** and device role. The term ***DPR overlay*** is used to refer to the set of active instances of *DPR components* at any particular time. Instead of simply replicating the PR among network nodes, a sophisticated policy-based replication scheme has been incorporated. Basic DPR management policies have been introduced in §4.2.2. By utilising context information and based on such policies, the system automatically enforces the appropriate replication state among hypercluster nodes, depending on how volatile the network is. This policy distribution method provides alternative access options in case a repository is corrupted or disconnected, and distributes traffic load and processing overhead among nodes. However, the replica placement problem is a computationally hard problem and the proposed solutions (§5.4.3) attempt to tackle the problem with emphasis on practical engineering aspects of replica placement. Research on a formal algorithmic solution is not addressed here and lies in future work plans.

The proposed policy-based framework integrates a self-maintained DPR overlay, aiming to balance on one hand the traffic cost of policy transfers from a logical PR to numerous distributed PDP and on the other the traffic cost of synchronising distributed PR instances. In effect DPR management policies create a closed control-loop that guides the DPR behaviour and replicas' distribution; ensuring on one hand maximum repository availability (distributed copies) and on the other hand a single logical view of the stored policies (replicated content). Thus, scalable and

efficient management of wireless networks can be achieved even when partial and temporary disconnections from a network manager occur.

5.3.2 Designing a Distributed Policy Repository

The diverse nature of wireless networks prevents the unmodified adoption and deployment of a Policy Repository (PR) using the various techniques targeting fixed networks. This motivates research efforts for an enhanced PR, the *DPR* (Distributed PR). The policy-controlled DPR concept was introduced in 2004 and based on a MANET case study it was published in [5]. These concepts were presented in Section 4.2.2 (example 2) as an introductory example for policy design and implementation, enforcing different replication states depending on network mobility. Further work has extended and enhanced those concepts with sophisticated policies and applied them to the wider domain of wireless ad hoc networks. Using DPR management policies as an example, the high-level management goal for policy storage and distribution has been defined in §5.2.1:

> *Depending on network's volatility, the system should automatically decide on appropriate DPR deployment to maintain efficient policy distribution and provisioning*

As mentioned, additional requirements need to be taken into account when designing a Policy Repository for wireless ad hoc networks. These issues are discussed here, explaining how they have been tackled through the proposed Distributed Policy Repository solution:

(1) *PDP may be intermittently connected to the ad hoc network but should maintain contact with the PR:* Occasionally a PDP may not have a route to a central Policy Repository, due to the variable quality of wireless links. Wireless link disconnection is quite common and contrary to fixed networks is not considered a fault. However, each PDP should be aware of at least one instance of DPR and one route to reach it. This is necessary so that each distributed PDP can retrieve policies and updates to instantiate and maintain relevant policy objects. To anticipate the above, DPR instances are distributed among network nodes, making more replicas available either collocated with or nearer to PDP.

(2) *Multihop wireless networks suffer from severe bandwidth degradation as the number of hops per route increases:* If the nearest DPR instance is several hops away from a PDP, significant traffic and latency overheads are introduced to the dissemination of new or updated policies. Such overheads will have a detriment effect on the ability to manage the ad hoc network in a timely and consistent manner. To alleviate these effects, a proper network organisation is used and suitable algorithms are proposed for improved replica placement (§5.4.3,pp.130).

(3) *Wireless ad hoc networks exhibit spatiotemporal density fluctuation in PEP population:* Contrary to traditional fixed networks, the number of managed devices in wireless ad hoc networks can fluctuate unpredictably and special conditions may lead to temporary increase of PEP. For example mobile phone users attending a sports event or concert for a short time period. Even for managed wireless networks, accurate planning in such scenarios is extremely difficult. The proposed solution is to integrate self-management capabilities to user devices, enabling the ad hoc network to dynamically assign additional PDP and DPR instances.

(4) *Wireless ad hoc networks are increasingly consisted of heterogeneous end-user devices:* Due to increased market fragmentation, it is difficult to have a universal management solution and providers normally restrict device model availability for their customers. This issue becomes even harder for ad hoc networks, where any personal wireless device can participate and devices cannot be fully controlled by a network manager. This issue affects policy distribution and storage and is taken in mind by employing role-based DPR management policies and using capability information from devices.

The self-management framework is built from the composition of communicating basic *components* and a defined set of *components* are required for acquiring one of three *roles*. In addition, each *role* is a *component* subset of its superior *role*, as shown in Figure 3-4, pp.60. The *DPR component* is different from other framework components, in the sense that its activation is variable and depends on *DPR management policies* and device role. Devices in CH and MN roles have the capability and required software to host the DPR component. However, not all CH are required to activate their DPR component, according to policies and current network fluidity. As will be explained, DPR design is based on the advanced replication and distribution features of modern LDAP servers. The innovation lies in the adoption and customisation of such features for the implementation and deployment of a sophisticated DPR overlay to facilitate policy-based management in a wireless environment. To illustrate the above, Figure 5-6 graphically presents the contrast between traditional centralised policy repository design with the proposed multi-manager decentralised DPR design. A dashed horizontal line separates end-devices from core operators' network, while thin dashed lines depict backup components.

An additional important feature of the designed DPR overlay is the ability to deploy and maintain special purpose partial replicas of the repository. These replicas provide a partial view of network policies that may relate to a specific service or location. They can be employed when there is a need for localised control or bottlenecks, e.g. in areas with dense user population such as a conference site or a stadium. As an application scenario, dense WLAN deployments will be considered, where users manually initiate ad hoc networks without relying on infrastructure support that may or may not exist (§7.2, pp.158). This normally results in poor performance and

interference problems among WLAN, even regulatory violations. By making available appropriate policies in the DPR, user devices can be assisted by receiving guidelines that transparently configure the ad hoc network, choosing the best available wireless channel to avoid interference and dynamically switching channels if performance degrades. The scenario above has been used for demonstrating the applicability of management policies for DPR (§5.3.3) and was also used to elaborate on self-management policies for wireless devices (§7.2,pp.158).

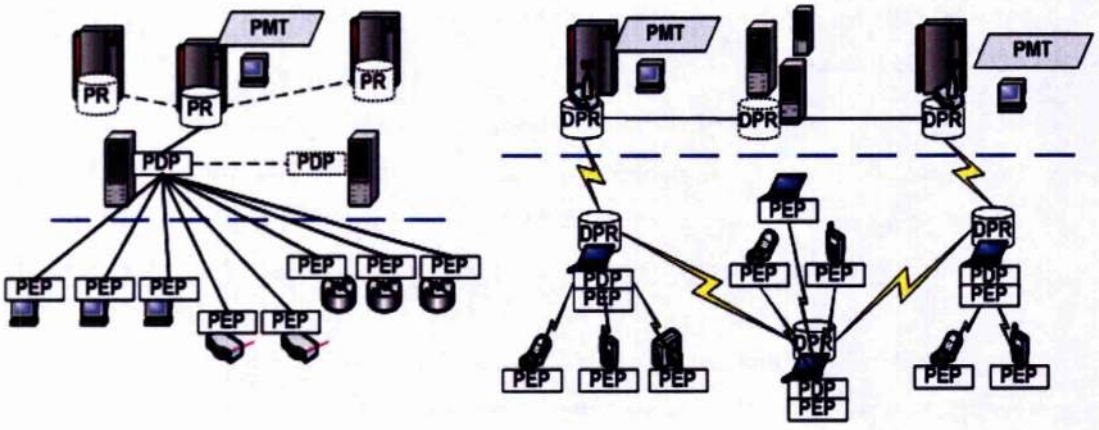

Figure 5-6. Traditional (left) Vs Proposed (right) PBM deployment

5.3.3 DPR Management Policies

In this subsection *DPR management policies* are detailed, explaining how they control the deployment of the DPR overlay among nodes. To realise the high-level management goal for policy storage and distribution, a special policy type has been already introduced aiming to control the *replication degree* of the DPR. By defining three *replication states* and the Fluidity Metric context, a set of three policy instances monitors and controls DPR among hypercluster's nodes. In a previous section (§5.2.1), the steps to design policy definitions and implement their LDAP storage representation were presented. The current section goes a step further, looking into implementation details of their actions, i.e. the implementation, deployment and management of the Distributed Policy Repository. Before proceeding, these policies are repeated here for continuity:

Policy type:

{MN&&CH}[**E**] if {FM=(**n**..**m**)} then {ReplDegState:=**k**}

Policies (policy type instances)

{MN&&CH}[fm_event] if {FM=[0..25)} then {ReplDegState:= 1:Single}

{MN&&CH}[fm_event] if {FM=[25..70)} then {ReplDegState:= 2:Selective}

{MN&&CH}[fm_event] if {FM=[70..100)} then {ReplDegState:= 3:Full}

The concept behind these policies is to select which of hypercluster's nodes activate their DPR component and carry a replica of network policies, in order to balance resource utilisation and policy accessibility across the network. The DPR state of each node is imposed by these policies that define the overall policy replication state. To realise the triggered policy action {ReplDegState:=k }, the actual enforcement needs to be carefully planned and implemented. For maximum flexibility, additional DPR management policies are employed to control deployment and maintenance of DPR components. In other words, the first high-level policy triggers another set of policies that through their actions will enforce the original one (e.g. Table 5-1 for *Selective*).

DPR overlay refers to the set of active instances of *DPR components* at any particular time. Additional *DPR components* may be present among network nodes and may remain inactive based on DPR management policies. Network volatility influences the DPR replication degree through the aggregation of special context information, e.g. the Fluidity Meter (FM). In brief, when network mobility is high and links are exceedingly intermittent, reliable access over many hops to a remote DPR may be inefficient, if not impossible. In this case, policy objects (PO) monitoring network fluidity detect the high volatility and proactively report that, aiming to increase the replication degree of DPR. Effectively the network will respond with increased decentralisation of the policy repository, pushing the storage points (DPR) closer to the decision points (PDP). Each MN or CH with an active DPR accommodates a full or partial replica of the repository and serves as an access point for repository requests within the neighbourhood. This balances processing load and traffic in the network and reduces latency. A CH with a dormant DPR can access policies from a list of neighbouring CH or MN with an active DPR.

Among the three replication states, *Single replication* is naturally the most simple and easy to implement. Since this state is employed when the Fluidity Metric is low, that implies a relatively static network with little disruption and infrequent changes to hypercluster's participant nodes. One of the manager nodes (MN) hosts the Master DPR instance and performs all policy updates. The rest of the hypercluster nodes that host a PDP, contact the DPR over LDAP to retrieve policies using an LDAP Search operation (RFC4515) [214]. It should be noted though, that Single replication refers to the single active DPR instance serving all LDAP requests from the network. However, this does not rule out the use of backup directory instances on capable neighbouring or remote nodes. In fact this is recommended to eliminate the single point of failure of a single activated Directory Server.

Full replication state is a straightforward technique used for highly volatile networks where the hypercluster is frequently reconstructed and the frequency of nodes connecting and abandoning the network is increased. In such scenarios, it is advisable to bind the process of node selection for DPR placement with the CH selection algorithm used for network clustering. While this method

avoids operation duplication and expedites DPR node selection, unavoidably it links two separate functions with potentially different objectives.

Finally, *Selective replication* state attempts to combine the benefits of both previous states and ameliorate their drawbacks. As a result it is the most complex replication state and therefore increased research efforts were dedicated. In order to decide where to place the DPR replicas, all Cluster Heads (PDPs) execute a special set of policies that combines a-priori knowledge of localised events (e.g. scheduled sport event) with dynamic real-time context information (e.g. processing load or free memory of each PDP).

Table 5-1 shows an example of three DPR management policies for selective replication [7]. Elaborating on these policies, a periodic chkDPR event causes the evaluation of conditions to determine if the current ratio of existing PDPs per DPRs or Users per PDPs in specified areas ($area_n, venue_n$) has exceeded the defined thresholds (thr_n). Additional time period constraints ensure triggering of policies when needed, e.g. on weekends or two hours before a sport's event kick-off ($t_{Weekend}$, $t_{Kickoff}$ - $2h$). Methods *locatePDPs()* and *selectDPRhost()* employ distributed algorithms (§5.4.3), for locating additional candidate PDP and for the best possible placement of replicated directories among hypercluster nodes. Different replica placement algorithms can be integrated in the implementation of policy actions, resulting in a customisable deployment of a DPR overlay.

Table 5-1. DPR Management Policies –Selective Replication

P	*Role*	*Event*	*if {Conditions} then {Actions}*
a	{CH}	chkDPR	if{$t_{Weekday}$}^{countPDPs(area$_1$)/countDPRs(area$_1$)>thr$_1$} then {locatePDPs(area$_1$)}, {selectDPRhost(algorithm $_a$, context$_1$)}, {deployDPR(all)}
b	{CH}	chkDPR	if{$t_{Weekend}$}^{countPDPs(area$_1$)/countDPRs(area$_1$)>thr$_2$} then {locatePDPs(area$_1$)}, {selectDPRhost(algorithm $_a$, context$_1$)}, {deployDPR(all)}
c	{CH}	chkDPR	if{$t_{Kickoff}$ - 2h}^{countPDPs(venue$_1$)/countDPRs(venue$_1$)>thr$_3$} ^ {countUsers(venue$_1$)/countPDPs(venue$_1$)>thr$_4$} then {locatePDPs(venue$_1$)}, {selectDPRhost(algorithm $_b$,context$_2$)}, {deployDPR(service1,service2)}

5.4 Distributed Policy Repository Implementation

To realise the aforementioned concepts, *OpenLDAP Directory Server* has been selected for DPR implementation. The main reasons for this decision were its lightweight open source distribution and its highly customisable replication features. Additional important reasons for OpenLDAP selection are listed here:

- A high performance LDAPv3 Directory Server. It offers production quality features including among other speed, robustness, reliability and replication.
- Lightweight and undemanding in terms of resources. The minimum required specifications for running OpenLDAP DS allow an extensive range of devices to efficiently host a directory replica, including low-spec laptops.
- Offers leading-edge replication and caching capabilities. The advanced replication options, including multi-master replication, were a critical factor for its selection.
- Open source code freely available under "OpenLDAP Public Licence", equivalent to "General Public License" (GPL) as defined by Free Software Foundation (FSF). This allows source code modification according to implementation needs.
- Provided in platform-independent source files that can be customised and compiled to a variety of Unix/Linux based platforms (also available in Microsoft Windows). In addition it supports different database backends to suite each platform.
- Its maturity and development support from the open source community were additional reasons to support this selection.

OpenLDAP's advanced replication capabilities were exploited to deploy and maintain the DPR overlay. The used features for implementing and deploying the DPR are explained here, while an additional description of these capabilities is available in Appendix B. OpenLDAP DS integrates a robust replication engine that is used to enable the policy-based DPR overlay. The overlay includes replicated read-only slave directories (shadow copies) on hypercluster devices, as well as partial copies for specific purposes (e.g. policies for services). OpenLDAP implements a "*synchronization replication engine*" (*syncrepl* for short), based on the "Content Synchronization Operation" (RFC4533 [215]). The functionality employed for DPR deployment and management is shown in Figure 5-7 and is explained below. An implementation of *DPR* consists of at least one read-write Master directory at a MN (provider, primary DSA) and a number of read-only Slave directories (consumer, replica, shadow copy, secondary DSA).

Master directories are normally hosted and controlled by the managing network entities, i.e. network operators and/or service providers. Regarding MN and the multi-manager case, the

activation of a DPR component depends firstly on the network deployment scenario (*network formation*) and secondly on policies. Therefore, in the case of a single Managing Entity for the network, one MN device is explicitly selected to host the master DPR and the remaining MN participate to DPR hosting according to policies, aiming to increase DPR and network scalability and survivability. For scenarios with multiple Managing Entities, each entity explicitly selects one MN device under its control, to host one of the master DPR. In this case, a special feature of LDAP DS is employed, known as Multi-Master Replication (MMR). OpenLDAP DS supports MMR since version 2.4 (Oct.2007). In the extreme case of no Managing Entity, e.g. for ad hoc social networks or user-owned networks, then active DPR components are algorithmically selected based on connectivity and scalability criteria.

Syncrepl engine offers client-side (consumer) initiation for replication of all policies or a customised selection, relieving the serving directory (provider) from tracking and updating replicas. This pull-based replication functionality (OpenLDAP directive: *refreshOnly*) is very useful since the operation of a directory provider (master) is not disrupted by the presence of consumers (slaves). In this mode, consumers are responsible to periodically poll their provider, in order to check if there are any updates available. Both can operate uninterrupted even when they are temporary disconnected due to wireless link intermittence. Upon link reestablishment, the directory consumers compare their current content with their provider's and retrieve any missed updates.

In addition, OpenLDAP's *Syncrepl* engine offers push-based replication (directive: refreshAndPersist), which allows a directory provider (master) to continuously update registered consumers (slaves) by sending them any updates in real time. In this mode, it is mainly the provider's responsibility to contact consumers once an update has been made and this is done through an open connection they maintain. This connection is initiated by consumers on their first attempt to contact their replication provider and retrieve initial directory contents. The positive feature that makes this push-based mode attractive to wireless networks is the ability to maintain stable operation even when temporary disconnections occur. The provider (master) marks the connection to that consumer (replica) as lost and periodically retries to establish contact. Upon link reestablishment, master synchronises the consumer with outstanding updates and normal push-based operation resumes.

One of the innovative features implemented and tested for the proposed DPR overlay is the ability to deploy and maintain special purpose *partial* replicas of the Policy Repository. These DPR replicas provide a partial view of network policies and can relate to a specific service or location. Partial replication is possible by defining appropriate scoping and filtering expressions that would replicate and keep synchronised a specific subset of total policies. It can be applied to all replication methods mentioned above, i.e. pull-based, push-based, even multi-master replication.

This functionality can significantly reduce policy retrieval traffic and synchronisation cost for wireless networks, while at the same time it increases policy availability. Accordingly, special PDP attached to partial replicas are responsible only for the enforcement of a policy subset and can be dynamically deployed to provision time-based events or local conditions. This feature can be employed when there is a need for localised control or bottlenecks, e.g. in areas with dense user population such as conference sites or stadiums. In such cases, while node population (i.e. users) increases, the management system can deploy special-purpose DPR replicas and accordingly more PDP that will be responsible for the distributed enforcement of specific management tasks, e.g. wireless parameters configuration or service provisioning.

Finally, an offline replication capability is implemented for DPR management, allowing Master directories to store their directory content to hypercluster nodes with dormant DPR components. In fact, this method is not strictly a replication process, but a method to keep extra repository copies in the network to anticipate disconnected or corrupted directories. This functionality further increases repository's survivability, by proactively creating offline replicas, that are able to return online when needed. Replicas can either be full or partial depending on demand and can be either scattered around the network or targeting a specific geographic area.

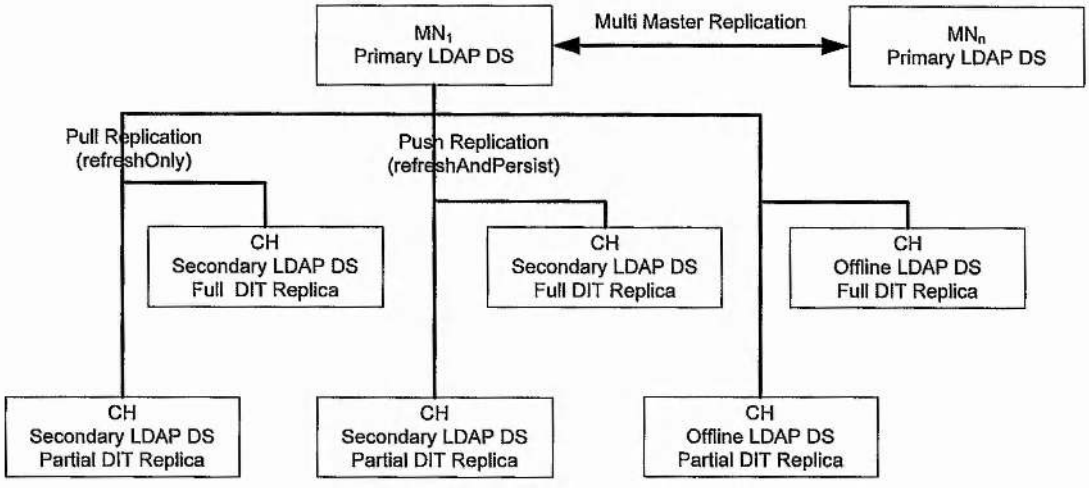

Figure 5-7. DPR Overlay Replication Functionality

5.4.1 Implementation Details and Evaluation Results

In Chapter 3, first attempts were presented to evaluate policy distribution traffic and the overall cost to the wireless network, in relation to the organisational model. In this section, an updated implementation has been used to deploy the DPR on testbed devices and measure traffic over wireless links. For the presented set of experiments a laptop was used to host the Master DPR and a portable wireless device (Internet Tablet/PDA) for the Slave DPR. Details of testbed equipment and software used for DPR implementation and measurements are given in Table 5-2.

Table 5-2. Hardware and Software for DPR Implementation and Measurements

Device	Processor (MHz -family)	Storage (GB)	Memory-RAM (MB)	Wi-Fi support
Sony Z1XMP (laptop dev.)	1500 - Intel	80 (Hard Disk Drive)	512	802.11b 802.11g
Nokia N800 (portable dev.)	330 - ARM	2 (External Flash)	128	802.11b 802.11g
Name,Version	**Details**	**Website**	**Category**	**License**
LAPTOP device software				
Debian R4.0	Debian GNU/Linux "etch" (Linux Kernel: 2.6.18)	www.debian.org	Operating System	GPL+ other
OpenLDAP v.2.3.32	OpenLDAP Software, [open source suite of directory software](slapd, ldapsearch)	www.openldap.org	Directory Server Agent and Client	GPL
Berkeley DB 4.2	Oracle Berkeley DB, transactional storage engine	www.oracle.com/technology/products/berkeley-db	LDAP Backend Database	Oracle Open Source
Wireshark 1.0	Wireshark network protocol analyser (formerly Ethereal)	www.wireshark.org	Packet capture and analysis	GPL
wireless-tools v.28	Tools for Linux Wireless Extensions manipulation (iwconfig)	www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/	CLI configuration tools	GPL
OpenSSH	OpenSSH Connectivity Tools(scp)	www.openssh.org	CLI security tools	GPL
phpLDAPadmin	Web-based LDAP client and browser	phpldapadmin.sourceforge.net	Directory client	GPL
Apache2 v.2.2.3	Apache2 HTTP Server	httpd.apache.org	HTTP server	ASL
PORTABLE device software				
IT OS2007	Internet Tablet OS "maemo bora" (Linux Kernel: 2.6.18)	www.maemo.org	Operating System	Nokia Open Source +GPL
OpenLDAP v.2.3.32	OpenLDAP Software, open source suite of directory software (slapd , ldapsearch)	www.openldap.org	Directory Server Agent and Client	GPL
Berkeley DB 4.0	Oracle Berkeley DB, transactional storage engine	www.oracle.com/technology/products/berkeley-db/	LDAP Backend Database	Oracle Open Source
Maemo 3.2 Bora SDK	Maemo 3.2 SDK Bora, Development tools for Nokia N800 Internet Tablet (ITOS2007)	maemo.org/development/sdks	Software Development Kit	Nokia Open Source
Apophis Scratchbox R4	Apophis Scratchbox cross-compilation tools	www.scratchbox.org	cross-compilation toolchain	GPL

Both *LDAP Directory Server Agents* (DSA) were compiled from the source of OpenLDAP ver.2.3.32 and an appropriate "database backend" was used for each device. Although a different backend was used, the underlying database was the same for both DSA, namely Berkeley DB [Table 5-2]. Unlike relational databases that mostly store tabular data, Berkeley DB is a hierarchical database developed specifically for LDAP storage. It is an open source, highly modular and embeddable database, distributed by Oracle under GPL-equivalent licence (following the acquisition of Sleepycat Software from Oracle). The Master DPR was hosted on a fully functional OpenLDAP DSA using BDB backend, the fully functional and high-performance transactional database backend of Berkeley DBv.4.2. To run OpenLDAP on a limited portable device, the source code of OpenLDAP was cross-compiled and LDBM backend was used. LDBM is a lightweight non-transactional DB management backend that uses Berkeley DB v.4.0. Cross compilation was needed due to different processor architectures, namely Intel for laptop and ARM for portable device. Open source development tools were available for *IT OS2007* (*Maemo3.2*), the Linux-based operating system of the portable device. Cross-compilation was done using *Maemo 3.2 Bora* SDK and *Apophis Scratchbox R4* [Table 5-2].

Based on the introduced ECA policy notation, a number of policies were implemented to test the behaviour of DPR. Following the methodology described in the previous Section, the LDAP representation of policies was defined, as described at Step 5 (pp.107) and was initially stored in relevant LDIF files (plain text representation). The size of LDIF files was the staring point for measurements and these files were used to populate policy entries in both Master DSA (laptop computer) and Slave DSA (portable device). Storage space and memory utilisation were measured for different directory sizes, i.e. for 100, 200 and 800 policies. These were equivalent to approximately 400, 800 and 3200 LDAP entries, with an average of 4 entries per policy. After the parsing of LDIF by each DSA, the disk allocated by the database backend was measured. Table 5-3 verifies that hosting a fully functional LDAP server on a portable device is possible, requiring reasonable storage space. Depending on resources, cache space can be reserved for quicker data access. In addition, database directives can define which LDAP entries (objectClass, entryCSN, entryUUID) and which operations (eq) to index, in order to accelerate search and replication access

Table 5-3. LDIF file size and database backend storage space

	Policies (approximate entries) [KB]			**Notes**
	100(~400)	**200**(~800)	**800**(~3200)	
LDIF file size	108	218	864	plain text
Slave DSA (ldbm)	704	1331	4301	default cache
Slave DSA (ldbm)	704	1331	4301	64MB cache
Master DSA (bdb)	1536	6451	21402	default cache
Master DSA (bdb)	167629	171315	185037	128MB cache

Comparison of Topologies for Policy Access

These series of experiments aimed to compare the traffic and time overheads incurred for policy access, using different topologies. The equipment and software used is shown in Table 5-2. Two devices where interchanged in the roles of DSA Host (*h*) and DSA Client (*c*), in order to evaluate their performance for different wireless topologies. Symbols ***h:*** for Host and ***c:*** for Client were used in graphs below, followed by the equipment type. Letter *M* indicated a laptop computer and letter *S* a portable device (PDA). For example, ***c:S, h:M*** indicates an LDAP client on a portable device remotely accessing an LDAP host on a laptop computer. A special case of local LDAP access on a single laptop computer was indicated by ***c:L, h:L*** and can be used for reference. Different *ldapsearch* queries were sent from the client to the host, aiming to retrieve different numbers of policies and LDAP entries by using different policy search base and scope. Each command was executed for three combinations of equipment topology (c:M,h:M , c:S,h:M , c:M,h:S), plus for local access for reference (c:L,h:L). The commands used are listed below:

$./ldapsearch -h host -b "cn=active policies,dc=ccsr,dc=com" -s sub	(4016 entr.)
$./ldapsearch -h host -b " CHANNELManagement, cn=active policies,dc=ccsr,dc=com" -s sub	(405 entr.)
$./ldapsearch -h host -b "cn= CHANNELManagement, active policies,dc=ccsr,dc=com" -s one	(101 entr.)
$./ldapsearch -h host -b "cn=active policies,dc=ccsr,dc=com" -s one	(3 entr.)

Figure 5-8 shows the total generated traffic for the first *ldapsearch* query, using "active policies" for search base and *subtree* for scope. Generated traffic was measured in both directions (c→h and c←h) to better evaluate the behaviour of devices at each role. This query was used by the client to retrieve the same total of 1000 active policies stored on the host. However, in spite of retrieving the same LDAP traffic, the total generated traffic was different for each topology. This is attributed to the different numbers and sizes of exchanged packets, incurring different overheads.

Figure 5-9 shows the total generated traffic of the four *ldapsearch* queries. The y axis uses a logarithmic scale for better readability. Measurements were taken for the aforementioned four topologies. These results confirm that an LDAP host on a laptop (c:M,h:M , c:S,h:M) generates less traffic than a host on a portable device (c:M,h:S), although the difference is noticeable for larger numbers of retrieved entries (4016). For fewer retrieved entries, the use of a portable device does not affect traffic overheads.

Figure 5-10 displays the total policy retrieval time for the same operations. These measurements indicate a bottleneck in terms of latency, caused by the limited processing capabilities of the portable device. This is mostly affecting the search operations of a portable client that retrieves large numbers of policies. However, it was observed that an LDAP host on a portable device can efficiently serve laptop clients, requiring 4.8s to complete the transfer of 100 policies, compared to 1.0s for the reverse topology. The increase of latency is acceptable, taking in mind the advantages from the use of portable devices to form the DPR overlay.

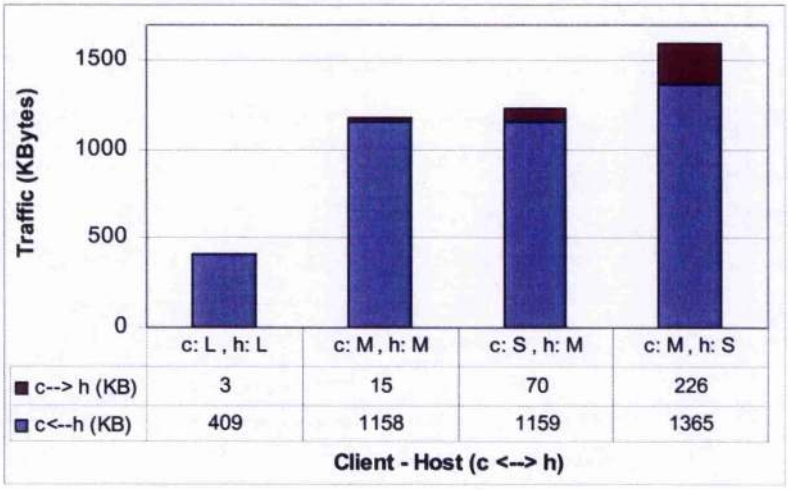

Figure 5-8. Traffic for retrieval of 1000 policies (4016 entries)

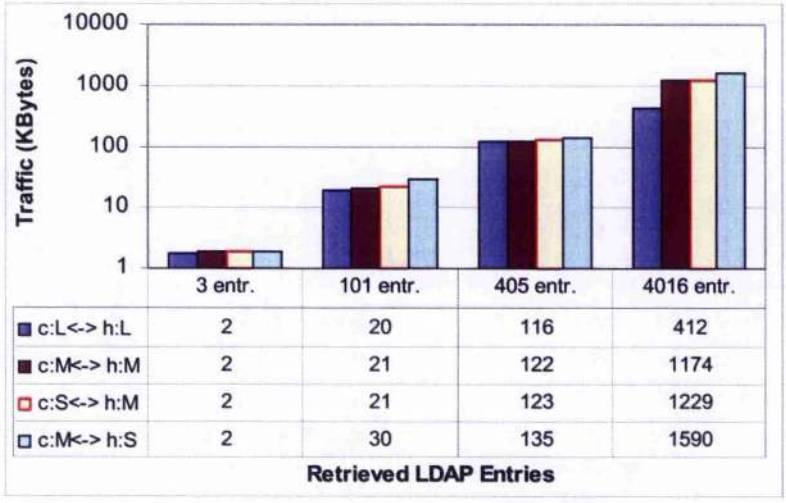

Figure 5-9. Generated traffic for policy retrieval

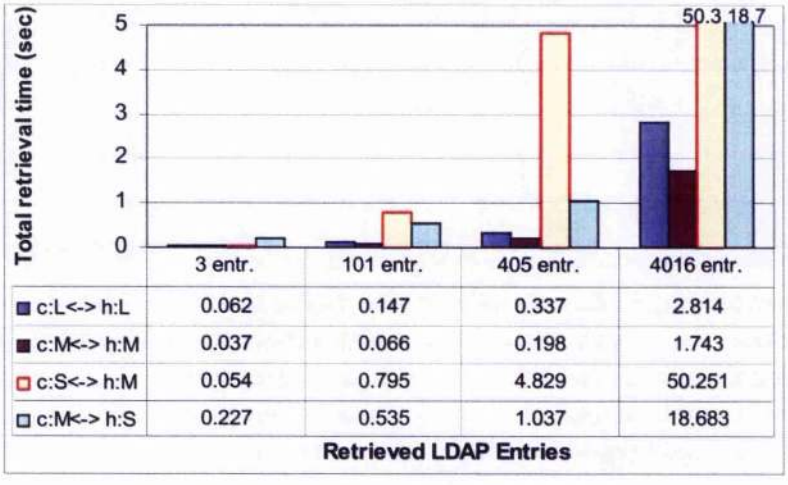

Figure 5-10. Time taken for policy retrieval

Measurements for DPR Policy Access using Replication

The use of LDIF files is one of the popular and convenient methods for offline distribution and storage of directory entries. This is due to their human-readable plain text format that is easily understood and debugged. Therefore, LDIF distribution is the method used for implementing offline replication. Once the files are transferred from a Master DSA to a remote host, they can be used to create an identical directory, either as a new Master DSA or as a Slave DSA. To confirm the viability of this method, LDIF files were transferred from a laptop computer to a portable device over 802.11b wireless links. Although a rigid security framework was out of the scope of these experiments, scp command line utility was used for transfers, i.e. a secure remote file copy command, part of OpenSSH connectivity tools [Table 5-2,pp.118]. Measurements in Table 5-4 show an expected traffic overhead compared to LDIF file size, which is acceptable. For large directories the overhead was 68KB and less than 8%, while the time taken was less than 1.5 seconds. The described offline replication method can be used for backup purposes in the Single replication state to maintain standby directories ready to return online.

Table 5-4. Measurements for secure remote transfer of policies

SSH Trasfer of LDIF	800 policies	200 policies
LDIF size (KB)	864	218
Traffic M-->S (KB)	913	234
Traffic S-->M (KB)	19	9
Total Traffic (KB)	932	242
Traffic Overhead (KB)	**68**	**24**
Traffic Overhead Incr. %	**7.8 %**	**10.9 %**

Beyond offline replication, the main implementation efforts were devoted to multiple online and synchronised directories, able to realise the designed Distributed Policy Repository (DPR). A series of experiments and methods are described below. To switch from Single replication state (i.e. one Master directory) to Selective or Full replication states (i.e. additional Slave directories), exact replicas of the online Master DPR component should be distributed in the network (among hypercluster nodes). The motivation is to use the DPR overlay for the coordination of distributed PDP. The proposed method to achieve this is to use updated LDIF files to re-construct a Slave DSA collocated with PDP and then use LDAP Replication engine to maintain synchronisation. Pull-based and push-based replication were implemented and evaluated. These methods were compared to the centralised retrieval of policies by each PDP, using the traditional LDAP Search operation.

Pull-based and push-based replication are based on *syncrepl* functionality of OpenLDAP (synchronization replication engine, RFC4533 [215], Appendix B). This is implemented by using appropriate *syncrepl* directives on Slave DSA in order to connect, authenticate and time their

operation with their Master DSA. Before starting a Slave DSA, its configuration file (slapd.conf) needs to be modified with correct replication parameters. An example partial configuration is shown in Table 5-5.

Table 5-5. Replication engine directives

slapd.conf (replication part)	**Explanation**	
syncrepl rid=110	Replica identifier	
provider=ldap://192.168.1.126:9000	Master LDAP DSA URI	
type=refreshOnly	Replication mode	1*
interval=00:00:01:00	Refresh period in dd:hh:mm:ss	2*
searchbase="cn=active policies,dc=ccsr,dc=com"	Base DN to bind for replication	3*
filter="(objectClass=*)"	Filter which objects to replicate	3*
scope=sub	Scope of replication (sub, one, base)	3*
attrs="*"	Filter which attributes to replicate	3*
retry="20 3"	Retry efforts if Master not available	4*
authentication	Authentication details for connection	
	1-4* : main configurable parameters	

A wireless node that has not hosted an active DPR component before is the worst case scenario, since it has to be informed of all selected policies. Beyond *ldapsearch* operation, the focus here is on initial directory replication and subsequently its maintenance. Sole use of *syncrepl* operation is a simple and straightforward solution. The new DRP host activates a blank Slave DSA, i.e. without any stored policies, and relies on *syncrepl* operation to retrieve defined policies. This method showed an increase in generated traffic, though its main drawback was the significant delay of 68.5 seconds in retrieving and processing new policies.

An alternative retrieval method involved the secure transfer of policies in LDIF files. This required a two step process, i.e. retrieval of updated LDIF data and start of replication engine. As before, LDIF files were transferred using secure file copy (scp/ssh). For measurements below, an online Master DSA was used and a Slave DSA was instructed to maintain all active policies (searchbase="cn=active policies,dc=ccsr,dc=com", filter="(objectClass=*)", scope=sub,attrs="*"). The Master DSA was populated with 200 policies (816 LDAP entries) from an LDIF file of 218KB.

Through measurements, it was noticed that using the same offline LDIF files to populate Slave directories was effective and both DSA contained exactly the same policies. However, once replication engine was started on Slave DSA, it required a significant amount of time and traffic for the first synchronisation attempt (404.12KB, 158.1 sec). This was because the same directory entries had different operational attributes (entryCSN, entryUUID) when created in different DSA. Therefore, these data where gradually retrieved and updated by the Slave DSA to fully synchronise with its Master. To avoid the unacceptable time delay, an updated "live" LDIF file

was extracted from the active Master DSA. The inclusion of operational attributes doubled the file size of LDIF (from 218KB to 442KB) and required 480KB total traffic over SSH (secure shell protocol). The new file transfer was completed in about one second (1.16 sec) and required 6.62 seconds processing time on the resource-constrained portable device. The importance of this method is attributed to the significant reduction of synchronisation time for new nodes participating in DPR overlay; in particular from 158.1 seconds to 7.8 seconds. Detailed measurements for initial policy retrieval are shown in Table 5-6. From TCP session analysis, the data communication time was separated from total session time, to better illustrate communication and processing time overheads. As expected, the alternative retrieval operation *ldapsearch* performed better and required almost half traffic to retrieve the same policies. The total time when using *ldapsearch* depends on implementation (i-d), because after the retrieval of entries these need to be processed locally. Further experiments and analysis have provided an improved view of both distributed and centralised policy retrieval and confirmed the added benefits of distributed policy replication.

Table 5-6. Measurements for Initial Policy Retrieval

Initial Policy Retrieval (200 policies) searchbase:"cn=active policies,dc=ccsr,dc=com"								
Master DSA (A) ←→ (B) Slave DSA [192.168.1.126:9000 ←→192.168.1.110:port]								
Method	**Bytes**			Packets			**Time (sec)**	
	Total	A→B	A←B	Total	A→B	A←B	Comm.	**Total**
Empty DPR	**404500**	383809	20691	688	378	310	54.2	**68.5**
Offline LDIF	**413823**	387204	26619	857	457	400	130.3	**158.1**
Live LDIF (ssh)	**491045**	479747	11298	481	342	139	1.2	**7.8**
Search operation	**246072**	234025	12047	370	189	181	9.5	**i-d**

Nodes that have hosted an active DPR before, but may have obsolete data due to inactivity, can synchronise their directory faster and with less traffic. Naturally, incurred traffic would be dependent on Slave inactivity time and policy changes at the Master during that time. During test measurements, a Slave DPR had been inactive for a few minutes and minor modifications were performed at its Master. Upon reactivation of Slave DPR, re-synchronisation was complete in 0.3 seconds and required a total traffic of only 1253 bytes. However, in a different experiment, an obsolete partial database was re-synchronised to a full replica in 76.1 seconds and required 383800 bytes. In such cases, it is preferable to first reconstruct the database from LDIF files and then connect to the Master DSA to maintain synchronisation.

Two replication types are supported by *syncrepl* engine, offering different features and traffic overheads. Their intended use with the DPR overlay has been explained in a previous section. Here the technical details are presented regarding their implementation and performance in a real

wireless environment. Pull-based replication (type=refreshOnly) is driven solely by the Slave DSA and attempts to connect to the provider DSA at configurable periodic intervals, e.g. every 1 minute (interval=00:00:01:00). If the provider (Master DSA) is available, a TCP connection is established for the synchronisation session and is closed once updates are made. A new TCP connection is made for every attempt. Any updated content is retrieved at those periodic intervals, i.e. is not immediately transferred to replicas. If no updates are available, the connection simply reconfirms provider's presence to the replicating DSA. If the provider is not available for any reason, the replicating DSA reschedules an attempt according to retry parameter. This configurable parameter is especially useful in a wireless environment where links show unpredictable intermittence. The number of retry efforts and their interval can be configured depending on network volatility and therefore can anticipate link breaks and topology changes.

Table 5-7 shows measurements taken during pull-based replication. Master DSA (A) was hosted on a laptop while the replicating Slave DSA (B) was hosted on a portable device connected via encrypted 802.11b ad hoc mode. Synchronisation traffic in bytes and packets was very low and periodically required less than 1.3KB. In addition, time taken for synchronisation sessions and processing was negligible.

Table 5-7. Measurements for Pull-based Replication

Pull-based Replication (refreshOnly, 200 policies) searchbase:"cn=active policies,dc=ccsr,dc=com"							
Master DSA (A) ←→ (B) Slave DSA [192.168.1.126:9000 ←→192.168.1.110:port]							
Synchronisation Attempt	**Bytes**			Packets			Time (sec)
	Total	A→B	A←B	Total	A→B	A←B	
First Sync.	**1187**	403	784	13	5	8	0.042
Periodic Sync.	**1253**	469	784	14	6	8	0.028
Lost Sync.	**128**	54	74	2	1	1	0.025

An important feature of the designed DPR overlay is the ability to deploy and maintain special purpose partial replicas of the repository. At a higher level, this behaviour is defined by actions of DPR Management policies, e.g. *deployDPR()*, Table 5-1. These actions modify the searchbase, scope, filter and attrs replication directives, defined in the local directory configuration file (slapd.conf). Optionally the database suffix parameter may be aligned with replication searchbase. With the above configuration changes, selective replication of directory content is possible. In practise it is sufficient to modify only the searchbase parameter, because of the hierarchical structure of LDAP DIT. Table 5-8 shows measurements for experiments where partial replication was used. By adding cn=CHANNELManagement to replication search base, the Slave DSA selectively replicates the directory branch for Channel Management policies.

Table 5-8. Measurements for Partial Pull-based Replication

Partial Pull-based Replication (refreshOnly, 100 policies) searchbase:"cn=CHANNELManagement,cn=active policies,dc=ccsr,dc=com"							
Master DSA (A) ←→ (B) Slave DSA [192.168.1.126:9000 ←→192.168.1.110:port]							
Synchronisation Attempt	**Bytes**			Packets			Time (sec)
	Total	A→B	A←B	Total	A→B	A←B	
First Sync.	**1340**	469	871	15	6	9	0.095
Periodic Sync.	**1274**	469	805	14	6	8	0.026
Lost Sync.	**128**	54	74	2	1	1	0.025

Push-based replication (type=refreshAndPersist) is driven mainly by the Master DSA (provider), that maintains some state information about the replica DSAs that needs to update. In this case, a single TCP connection is established upon first contact and is maintained for the whole duration of their synchronisation session. Using this connection, the Master DSA immediately pushes any updates to a Slave DSA, thus making them available to replicas faster. The maintenance of a single connection and Master's coordination reduce traffic overheads for synchronisation, although an uninterrupted TCP connection is not always possible in wireless ad hoc networks. Temporary link breaks are expected; therefore the retry mechanism is used as before to re-establish a new connection. For experiments, the same Master (A) and Slave (B) topology was used, in order to compare the results of both replication methods. For push-based replication, a single synchronisation session was opened with total traffic of 1.3KB and negligible latency. This session remained open for the whole duration of synchronisation and was used by the Master to immediately update Slave DSA.

A series of policy update operations was also performed to evaluate the synchronisation cost of replication. For this purpose, a web-based LDAP Management interface was used to connect to the Master DSA and perform policy updates. The graphical interface was provided by *phpLDAPadmin v.1.1.0.5* running on *Apache2 v.2.2.3* HTTP Server [Table 5-2,pp.118]. Ten new policies (41 LDAP entries) were added and subsequently were removed from the Master directory. The plain text definition of policies required 11113 bytes in LDIF without operational attributes. During updates, measurements were taken to evaluate the synchronisation cost. The main observation from experiments was that *pull-based* updates required a non-negligible time of about 5 seconds to complete. During these synchronisation sessions, a short traffic burst of updated entries (~0.25sec) was followed by a longer processing period on the portable device (~4.5sec), before closing each session. Update time for *push-based* replication was still negligible, with the added benefit of immediate receipt of updates. In terms of generated traffic, both methods produced similar total traffic as shown in Table 5-9. For policy additions, Pull replication required 49% more traffic than Push replication, but it was more efficient for policy deletions.

Table 5-9. Comparison of Replication Methods

Pull and Push -based Replication							
Master DSA (A) ←→ (B) Slave DSA [192.168.1.126:9000 ←→192.168.1.110:port]							
Master DSA Modification	Bytes			Packets			Time (sec)
	Total	A→B	A←B	Total	A→B	A←B	
Push-based Replication (refreshAndPersist)							
Add 10 pol.	**23738**	22220	1518	64	41	23	**0.429**
Delete 10 pol.	**10791**	9933	858	48	35	13	**0.324**
Pull-based Replication (refreshOnly)							
Add 10 pol.	**35447**	33937	1510	49	30	19	**4.766**
Delete 10 pol.	**2201**	1351	850	15	6	9	**5.055**

5.4.2 Comparison of Distributed and Centralised Policy Access Methods

The next series of conducted experiments aimed to compare the two proposed distributed replication methods to a centralised deployment without replication. In the centralised case, policies were transferred to Policy Decision Points (PDP) using an LDAP client and *ldapsearch* operation. Two subcases were examined: the first one (*best case*: *ls-best)* refers to PBM deployments where PDP can be notified about when policies change and which is their exact location (DN) in the centralised PR. This out-of-band notification directs PDP search operations and is excluded from presented measurement; however in practise it would mean additional overheads. A second more realistic subcase (*average case: ls-avg*) was also examined, where PDP needed to periodically check the PR for changes to discover changed or updated policies and their location.

The operations needed per case are shown in Table 5-10, according to a real life scenario. $\mathbf{T_p}$ is total policy access time, including initial policy retrieval (T_a), policy addition time (T_b) and policy deletion time (T_c). Incurred total traffic $\mathbf{C_p}$ was also measured. The following experiment scenario was used for evaluation:

t_0: Master DSA holds 200 policies (800 entries)
t_1: PDP requests all policies
t_2: PDP holds all policies
t_3: 10 policies added at Master DSA
t_4: PDP update completed
t_5: 10 policies deleted from Master DSA
t_6: PDP update completed
T_p : total policy retrieval and update time
$T_p = (t_2 - t_1) + (t_4 - t_3) + (t_6 - t_5) = T_a + T_b + T_c$

Table 5-10. Comparison of Distributed and Centralised Policy Access Methods

	Distributed PR Push Replication	Distributed PR Pull Replication	Centralised PR No Replication (best case)	Centralised PR No Replication (average case)
T	sr-push	sr-pull	ls-best	ls-avg
a	i)Transfer LDIF file ii) LDAP reads/stores LDIF iii) Open sync. connection	i)Transfer LDIF file ii) LDAP reads/stores LDIF iii) First sync. connection	i) *ldapsearch -s sub "all"* ii) Custom processing iii) Custom storage	i) *ldapsearch -s sub "all"* ii) Custom processing iii) Custom storage
b	i) n/a ii) *syncrepl* automatically updates DPR	i) Periodic t_i sync. messages ii) *syncrepl* periodically updates DPR	i) n/a *ii)ldapsearch -s sub "new"* iii) n/a iv) Custom processing v) Custom storage	i) Periodic *ldapsearch -s one "all"* ii) Custom processing iii) *ldapsearch -s sub "new"* iv) Custom processing v) Custom storage
c	i) n/a ii) *syncrepl* automatically updates DPR	i) Periodic t_i sync. messages ii) *syncrepl* periodically updates DPR	i) n/a ii) Custom processing	i) Periodic *ldapsearch -s one "all"* ii) Custom processing
	new DPR replica with an empty DSA	new DPR replica with an empty DSA (interval t_i)	PDP is notified when policies change and their exact PR location	PDP discovers when policies change and where is their location

Figure 5-11 compares the generated policy access traffic for each phase of the scenario, as well as a comparison of their total. It should be noted that both replication methods assume the worst case of a new replica with an empty DSA. Figure 5-12 provides a comparison of total policy access time T_p for all cases. This graph displays the time taken for policy communication and it excludes the incurred time delays for local processing and storage on devices. Therefore it is the total occupation time of the wireless transmission medium. An important observation from these measurements was that the use of push-based replication, significantly reduced the total latency for policy access, in the examined case to 4 sec. compared to 10 sec. for centralised access. Update times for pull-based replication appear increased, because the TCP session remained open while the portable device was processing the received updates. The faster performance can counterbalance the increase of traffic for distributed policy access using replication. The presented measurements refer to policy transfer and update operations relevant to the connection of a Master DPR component hosted on a laptop with a Slave DPR component hosted on a resource-constrained device. These conclusions confirm the applicability of the proposed DPR design and justify the inclusion of lightweight portable devices to the implemented DPR overlay. Slave DPR components are collocated with a PDP and, depending on replication state, they may also provision remote PDPs. As analysed in Chap.3, the organisational model of the whole network affects overall policy distribution costs. For distributed PDP deployment over wireless networks, these costs are significantly affected by the DPR overlay organisation, i.e. replicas' location. For

the centralised Policy Repository case and deterministic PDP allocation, costs are also affected by the average hop distance travelled by *ldapsearch* sessions between PDP and the central PR. For these reasons, the problem of *DPR replica placement* needs to be examined.

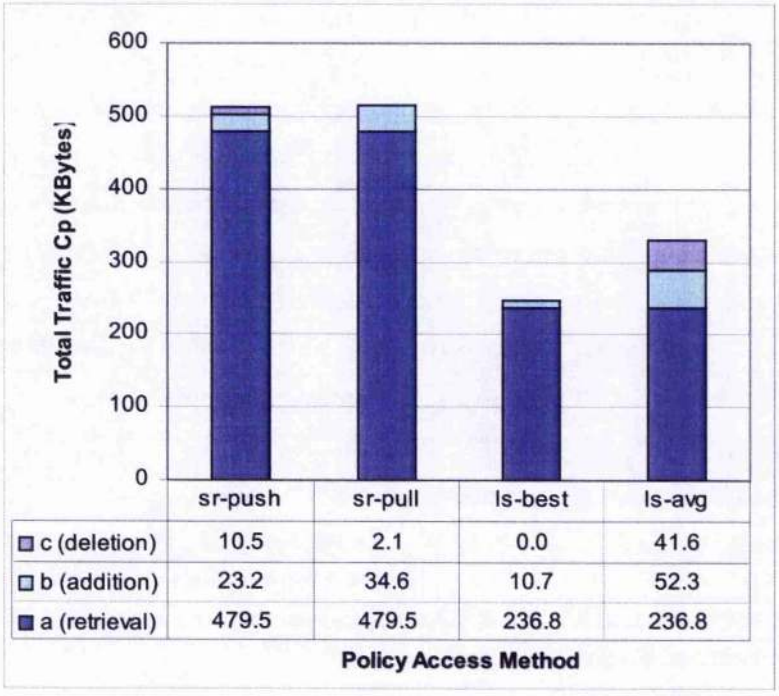

Figure 5-11. Comparison of Total Policy Access Traffic

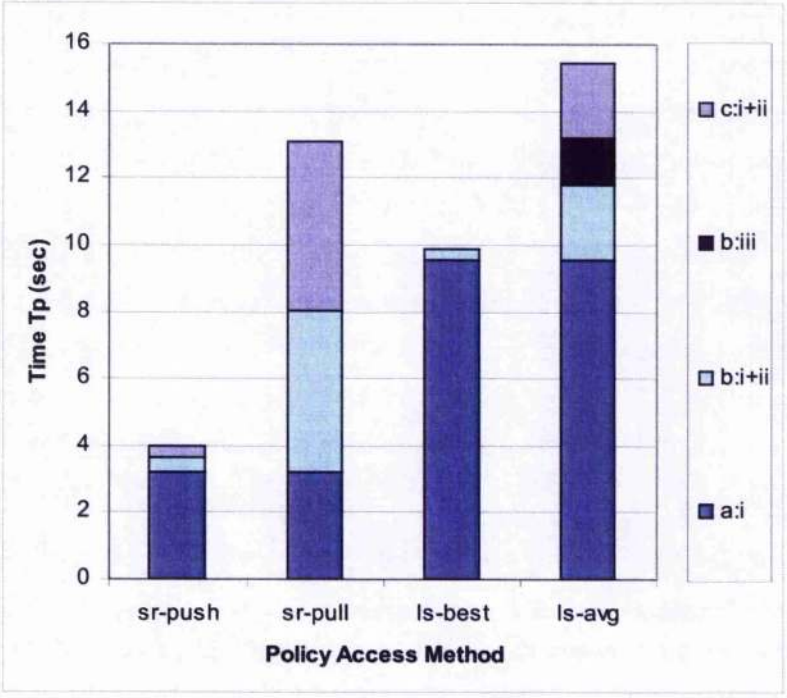

Figure 5-12. Comparison of Total Policy Access Time

5.4.3 Algorithms and techniques for DPR instance placement

The hybrid organisational model uses a *clustering strategy* to create a number of clusters and select a cluster representative (CH) to participate in collaborative management tasks. Selected CH and MN form the *hypercluster*. In §3.5 clustering strategies and algorithms were presented and it was explained how the hypercluster is algorithmically formed based on an adapted version of a distributed algorithm by Wu [78] (Appendix A). In this subsection, it is assumed the hypercluster has been constructed, to allow the examination of algorithms and techniques for *DPR instance placement* among the hypercluster nodes. Since all hypercluster nodes integrate a DPR component, the objective of these algorithms and techniques was to autonomously decide which DPR components should be active at any time. This would increase policy availability, while reducing resource consumption and traffic overheads. As mentioned, the goal is to balance the traffic cost of policy transfers from a logical PR to numerous distributed PDP, with the traffic cost of synchronising distributed PR instances. Based on the above, the DPR replica placement problem has been defined:

> **DPR replica placement problem**: *Given an arbitrary network G and a number M of Master DPR, select a number of N network nodes to place a Slave DPR, such as to minimise the cost of replicating the data of M to N and the cost of DPR access for the rest of the G-(M+N) nodes.*

As discussed in Chap.2, an optimal replica/cache placement solution is a computationally intensive task, hindered by the distributed nature of wireless systems. Solutions applied to the replica placement problem are not "feasibly computable" [22],[76] and have been proven to be at least NP-complete. For example the Dominating Set (DS) or the Connect Dominating Set (CDS) problems are NP-complete, while facility location problems like the connected facility location problem, have been proven to be NP-hard [76],[77],[78],[88]. The bottom line is that an optimum solution to such problems would require non-deterministic polynomial time to be computed. The adoption of heuristics is sensible for lightweight wireless devices, where processing power and resources are limited. For example, the adopted distributed algorithm by Wu [78] uses two heuristic rules to reduce an initial non-optimum solution for the Connect Dominating Set (CDS) problem (§3.6,pp.71). Similar heuristic approaches have been adopted for solutions of the DPR placement problem.

There is an indirect connection between the algorithms for hypercluster creation and DPR instance placement that lies in the nature of wireless networks and affects their effectiveness. The main aspect that needs to be taken in mind before investigating DPR instance placement is the connectivity properties of the hypercluster set of nodes. If this is ignored and a DPR overlay is transparently selected, it will be difficult to control hop distance between DPR instances. Hence

there would be a risk of creating arbitrary long paths that would make directory synchronisation inefficient.

While the examined optimal DPR replica placement problem has not been formally proven as a computationally infeasible task, the majority of the algorithms adopted for its solution are formally proven to be at least NP-complete, if not NP-hard. The problem of optimal replica or cache placement in wireless ad hoc networks is an area currently receiving intense research interest. Two solution families have been identified in literature (§2.2) and form the solution basis of the *DPR replica placement problem*: (1) based on node domination and (2) based on location analysis. As explained, solutions are outlined with emphasis on practical engineering aspects of replica placement specific to wireless ad hoc networks. A complete analytical proof of algorithmic solutions is out of the scope of this thesis and is reserved for future investigation.

Node domination based solutions:

An efficient distributed execution of the connected dominating set calculation was proposed by Wu [78] and has been widely used to create virtual backbones in MANET [79][80]. Virtual backbones create a connected sub-graph of a network which is used for traffic forwarding. Another distributed approach for connected dominating set creation is integrated to OLSR routing protocol and will be examined below. The creation of a Connected Dominating Set of nodes and the placement of DPR replicas on them ensures a connected overlay of DPR components where policy distribution and updates can be improved.

An adapted version of [78] has been already presented for the distributed hypercluster construction [2] based on context-aware heuristics (§3.5,pp.69). Experiments performed in [2] have showed good convergence times in a distributed manner. This motivates the association of hypercluster creation with DPR placement solution, a method which is mostly useful when the network is in *full replication* state. The main concept is to reduce algorithmic complexity by avoiding a duplicated selection process, since all nodes in the hypercluster are required to activate their DPR components. Therefore, if full replication policy is triggered, all hypercluster nodes self-configure their DPR components to start and acquire an updated replica of the policy repository. There are some important advantages from the linkage of hypercluster creation with DPR placement solution. The algorithm is likely to take in mind connectivity parameters and select nodes with relatively better capabilities depending on heuristics. In addition, quick selection and convergence can be achieved based on existing and tested algorithmic solutions. On the other hand, DPR placement results will be as good as the hypercluster algorithm. Inevitably, the high dependency on hypercluster selection algorithm creates some disadvantages for replica placement. The main issue is the limited method's applicability beyond full replication. As already examined

in Chapter 3, the large hypercluster population in certain conditions may lead to increased policy distribution costs.

Another approach was investigated and focused on integration of DPR replica placement with MANET routing protocols. Proactive protocols, e.g. OLSR, may provide highly distributed solutions with minimum additional algorithmic complexity. OLSR [209] uses a fully distributed algorithm to select Multi-Point Relay (MPR) nodes that form a connected dominating set for efficient flooding and reduction of protocol overheads. The protocol aims to minimise the MPR set, through the use of heuristics. An MPR set is similar to a *virtual backbone*. The standardised status of OLSR and its wide support from wireless ad hoc networks, motivate its use for management purposes, i.e. for clustering and DPR instance placement. For wireless ad hoc networks using OLSR routing protocol, the creation of a management node set can be facilitated by inter-layer communication between the Application layer and the Network layer, i.e. the OLSR routing daemon of each node. This option significantly reduces overheads since DPR node selection is "outsourced" to the routing protocol. If OLSR is not used or inter-layer communication is not available, then it is possible to reproduce the MPR selection algorithm [209] at the Application layer. The benefits from the latter option need to be evaluated under different application scenarios, having in mind the overheads incurred due to the proactive nature of the algorithm and the need for updated neighbourhood information. Another consideration is the resulting size of the Connected Dominating Set and coincidently the number of distributed replicas in the network.

Intuitively, solutions for the DPR replica placement problem based on Dominating Sets are not expected to be optimal. This can be explained taking in mind the goal of DS creation, which is the creation of a virtual backbone for forwarding traffic and messages. Therefore, placement of a replica on every node of a DS may be redundant. This can be verified by reviewing Figure 3-11 and Figure 3-12 (pp.74), where hypercluster's population is plotted against the total network population. For example, in some cases of sparse deployments, about half of the nodes belong to the hypercluster (*Fix.Dens(1:27800)*). In such cases, full replication on all hypercluster nodes would be practically infeasible. On the contrary, in increasingly dense network deployments, the hypercluster population remains quite small, which also restricts the number of DPR replicas that serve the increasing network population (*Var.Dens(~1:625)*).

Obviously the dependence between network clustering and replica placement can become both an advantage and disadvantage depending on conditions. In order to optimise the number of replicas in the network while maintaining good connectivity among replicas, a dual stage process is suggested. First, an algorithm is used to create a connected dominating set that constitutes the hypercluster. In this case, either MPR or Wu's algorithm can be used. Subsequently, additional elimination heuristics can be executed among hypercluster nodes, to reduce the Connected

Dominating Set and select the nodes for DPR placement. The re-execution of Wu's algorithm among the created CDS is a promising solution, supported by its quick and efficient convergence in only two rounds. The dual execution the algorithm with context aware heuristics would create a new set of capable and well connected nodes. What is very important is that the complexity of creation time would remain bounded to just four rounds

Location analysis or facility location based solutions:

This solution family addresses the same problem of optimal replica placement by adopting concepts of Location Analysis and Operational Research (an interdisciplinary branch of applied mathematics) [87]. In general, facility location problems involve a given number of facilities that needs to be optimally located in an existing area and fulfil given requirements. Facility location problems are particularly attractive as solutions to the DPR replica placement problem because they follow similar requirements, e.g. cost minimisation or minimisation of facilities.

An algorithmic solution [90] of particular interest has been introduced in Chap.2. In [90], the authors elaborate on the "Efficient Cache Placement in Multihop Wireless Networks" and attempt to find the optimal cache placement which minimises the total cost, i.e. the incurred overheads from cache updates and requests to caches. They prove that the problem is equivalent to a special case of the NP-hard *connected facility location problem,* called the *rent-or-buy problem* [91]. The rent-or-buy problem is also NP-hard [91], therefore several approximation algorithms (heuristics) had been developed [90].

The *rent-or-buy* problem formulation as explained in [90],[91] is repeated here and is mapped to the DPR replica placement problem in parentheses: an existing facility (Master DPR) is given, along with a set of locations (hypercluster nodes) at which further facilities (Slave DPRs) can be deployed. Every location (hypercluster node) is associated with a service demand (acquirement of policies), which must be served by one facility (DPR instance). Authors described a polynomial-time algorithm based on heuristics that approximates the optimal (brute force) solution for arbitrary graphs within a factor of 6. Their solution allows for a distributed and asynchronous implementation suitable for wireless ad hoc environments.

As described above, location analysis solutions are particularly attractive to the DPR replica placement problem, because they can be mapped to specific solved problems and follow similar requirements, like cost minimisation. Further investigation and adaptation of location analysis solutions is part of future work.

5.5 Summary and Conclusion

In this Chapter, a step-by-step methodology was presented to realise policies for a pragmatic PBM system for wireless ad hoc networks. A realistic example guided the methodology, focusing on the definition of policies that control the Distributed Policy Repository (DPR) component. The critical DPR component of the proposed framework is controlled by policies in order to ensure maximum availability and increased survivability in the ad hoc network environment.

Furthermore, the mapping procedure was outlined from IETF's PCIMe [204],[207] Information Model representation to a solid implementation-ready Data Model format. Policies were mapped to appropriate LDAP classes using IETF's LDAP Schema mappings [211],[212]. In addition, LDAP schema extensions were implemented for the scenario-specific defined classes. The outcome of this methodology has been implemented on OpenLDAP DS in order to instantiate policies for the managed network.

For the purpose of PBM for wireless ad hoc networks, the presented straightforward methodology can implement complex functionality in a future-proof manner and at the same time, maintain interoperability by building on existing standards. These are significant benefits of using policies and PBM since they allow a transparent and technology-independent implementation to encapsulate management logic and objectives, separating their enforcement from implementation. As a result the management system can be easily updated and upgraded, keeping costs for software maintenance low.

After policy definition, the next task is to store new policies in the Distributed Policy Repository (DPR) and distribute them to respective Policy Decision Points (PDP). The introduced Distributed Policy Repository (DPR) is a physically distributed set of components consisted of interconnected directories hosted on selected hypercluster nodes. The coordination of distributed PDP in a wireless environment is quite hard and remains an open research topic [100]. In the proposed solution, this problem was transformed to the deployment and maintenance of the DPR by exploiting standardised LDAP operations and replication features. In this way, the DPR interconnects the distributed PDP and offers a logically uniform view of network management objectives through policies.

The proposed policy-based framework integrates a self-maintained DPR overlay, configured and maintained by special *DPR management policies.* The aim was to balance both the traffic cost of policy transfers from a logical PR to numerous distributed PDP and the traffic cost of synchronising distributed PR instances. In effect, DPR management policies created a closed control-loop that guided the DPR behaviour and replicas' distribution, ensuring both maximum repository availability (distributed copies) and a single logical view of the policies (replicated content). The DPR also implemented the ability to deploy and maintain special purpose *partial*

replicas, offering a customised view of network policies that can relate to a specific service or location. This feature can be employed when there is a need for localised control or bottlenecks to increase scalability and availability of wireless networks.

The DPR components were implemented for portable wireless nodes to confirm design applicability. Based on testbed deployment, measurements of traffic and latency were taken for different topologies, providing valuable performance indicators for large-scale deployment. Evaluation results of proposed distributed policy replication methods were favourably compared to those of centralised methods. The *DPR replica placement problem* was also investigated, aiming to minimise the cost of replicating the data from master DPR to slave DPR and the cost of DPR access for the rest of the nodes. With emphasis on practical engineering aspects, known problems and heuristics from Graph Theory were investigated and adapted for DPR replica placement. Algorithmic solutions based on node domination and location analysis were applied. The integration with proactive routing protocols was also suggested.

Chapter 6

Policy provisioning and selective enforcement for wireless ad hoc networks

6.1 Introduction

Policy provisioning is the process of communicating policy decisions and directives between a Policy Decision Point (PDP) and a Policy Execution Point (PEP) using a suitable protocol [202], [206]. As examined in §2.4.2 (pp.44), the interaction between PEP and PDP can be done based on two models (*outsourcing* and *provisioning),* which are combined in the proposed framework. To facilitate the communication between PEP and PDP, a lightweight policy provisioning protocol has been proposed, based on the Remote Procedure Call (RPC) paradigm. The protocol was implemented by using and extending XML-RPC protocol [157] and defining required procedures at both PDP and PEP components. Methods for Policy Objects (PO) management and their lifecycle were outlined, providing design guidelines on their implementation within the framework. The main innovation focused on mirroring the role-based and context-aware aspects of the proposed organisational model to PO management.

The next stage of policy-based operations is policy enforcement, i.e. the execution of a policy decision [206]. Since policy enforcement is tightly related to provisioning, similar requirements and obstacles also apply here. Using XML-RPC, implemented procedures would receive technology-independent parameters that were mapped to device-dependent execution. Departing from traditional uniform policy enforcement, new concepts for selective policy enforcement were proposed, to deal with consumers' increased concerns about the acquisition of their personal data. Hence, a twofold protection mechanism is integrated to the proposed PBM framework, offering user-centric control and integrating a policy-based regulation scheme.

6.2 Policy provisioning

Policy provisioning needs to transfer policy decisions to enforcement points and these decisions may range from device-specific parameter configuration to remote triggering of complex methods. Different granularity levels may coexist in the same system depending on implementation and the support provided by the used protocol. In the context of this work, both models for policy provisioning were combined, depending on the policy decision that needs to be made. This allowed for increased design flexibility and resulted in optimised resource utilisation. Because of the wireless environment and the wide use of lightweight devices, the *provisioning* model is favoured over the *outsourcing*. Due to its inherent asynchronous operation, it allows end-devices to operate mainly unsupervised based on the provisioned policy directives they have received. According to RFC2753 [197], the concept of a Local PDP (LPDP) is adopted in the sense that a provisioned PEP is able to make local decisions. However, the requirement that "this partial decision and the original policy request are next sent to the PDP which renders a final decision" [197] was relaxed, because in a wireless environment that would cancel any benefit of local decision making and introduce significant delays. Instead, the PEP may report its local decisions/actions to its PDP, to acknowledge a configuration change or event that can be used in cluster-wide or network-wide decisions. In addition, when needed, critical events are reported and the controlling PDP may provide new directives and decisions based on the *outsourcing* model.

Presented work was targeted on heterogeneous devices participating in wireless ad hoc networks; therefore a *middleware approach* was adopted for policy enforcement and provisioning. This approach is widely used for distributed objects programming. As it has been explained in §3.3.3, preinstalled software *modules* implement management functionality and use appropriate *components* depending on assigned role. Enforcement on provisioned nodes is implemented by CN's (Cluster Node) set of components, i.e. *PEP*, *CCP* and *CN Interface*. Provisioning can be either external (remote CH) or internal (encapsulating CH), depending on device role.

The middleware approach is beneficial because it allows the majority of developed software to remain device-independent and only requires development of device-dependent functionality to use special device API, operating system calls and internal device functionality. There is an obvious tradeoff between the software development/maintenance process and the range of supported devices. The middleware approach is applied to the introduced *policy provisioning protocol*, by mainly involving remote procedure calls from PDP to PEP and vice versa. To preserve system's extensibility and wider applicability, the provisioning protocol transfers technology-independent parameters that are mapped to device-dependent execution commands. These concepts assist towards satisfying an important requirement for policy provisioning in wireless ad hoc networks, which is to achieve uniform management in an environment of

increased heterogeneity. As a result, the transmission of device-independent parameters can be supported and standardised in a technology-agnostic provisioning protocol. Using such protocol makes the specification of PEP-PDP management interface easier to define, leaving vendor specific details for implementation.

6.2.1 Policy provisioning protocol

Based on an investigation of existing provisioning protocols and methods, a suitable off-the-shelf solution was not available. To suit the needs of the designed framework and the requirements of wireless ad hoc networks, the proposed protocol was based on a combination of existing solutions and protocols aiming to satisfy most requirements.

Policy provisioning is closely related to policy enforcement, because it needs to transfer the actual decisions to enforcement points (PEP). Therefore, the design of a provisioning protocol is firstly dependent on the actions the PEP can support, i.e. their management interface. For example, the majority of core network devices, e.g. routers, typically support SNMP by implementing the protocol stack in their firmware. Other devices may also support COPS protocol. Researchers have also suggested programmable routers, where their operating system can execute on demand plugins [158] and recently such routers have appeared in the market [159].

XML-RPC [157] was chosen as the basis of the provisioning protocol because it is lightweight, interoperable, easy to extend, easy to deploy and widely supported by devices. Its main requirement is HTTP and XML processing capability at enforcement points. XML-RPC was preferred over SOAP or fully-fledged Web Services for being less resource consuming, simpler to implement and less demanding in device capabilities. Regarding HTTP support, XML-RPC requires HTTP 1.0 [197] but is also compatible with HTTP 1.1 [199]. Therefore it is supported by virtually all networked devices, even legacy mobile phones based on the outdated and limited MIDP1.0/CLDC1.0 runtime environment for J2ME (Mobile Information Device Profile / Connected Limited Device Configuration for Java 2 Micro Edition) [160],[161]. In addition, the huge majority of wireless devices currently support XML capabilities, thus satisfying this requirement:

- The majority of mobile phones and smart-phones have embedded a lightweight version of Java (JME Java Micro Edition, formerly J2ME) and a mature choice of compact XML parsers has been used for many years, e.g. kXML, minML [162]. In addition, Java Community Process (JCP) has standardised XML capabilities for J2ME enabled devices through JSR172 "J2ME Web Services Specification" [163], defining specifications and providing a reference implementation. Therefore many devices already come with embedded XML processing capabilities.

- Lightweight devices like PDAs or wireless peripherals provide vendor-specific implementations for XML processing or bundled applications. Also, laptop, desktops and mainstream computing devices normally include XML processing software but it is also straightforward to install bundled applications integrating the required XML processing capability.
- Major vendors of infrastructure networked devices (e.g. routers) ship their products with integrated XML processing capabilities, providing suitable XML APIs. Although initial products used CORBA/IIOP as the XML transport mechanism [165] the trend towards Web Services and XML/HTTP-based management has continued to evolve [164],[166] and is currently embraced by network management community and industry.

6.2.2 Management and lifecycle of Policy Objects

Policy Objects' management and lifecycle is examined in this section, providing design guidelines on their implementation within the PBM framework. For this purpose the definition for *Policy Objects* (RFC2753,[202]) was adopted and clarified in §2.4.1 (pp.39). According to Object Oriented programming principles, PO for supported policies are implemented by *classes* and these classes are used for the runtime creation of respective *instances* when a new policy is introduced. The main innovation focuses on mirroring the role-based aspects of the proposed organisational model to PO management. PO management can exploit the properties of network organisation to increase PBM system scalability.

The types of PO for this framework follow the role-based hybrid hierarchy of policies, as described in Section 4.2. The enforcement scope of each policy classifies *runtime PO instances* in node, cluster, hypercluster, and network -wide PO. This classification is facilitated by the Roles attribute of each policy that determines its enforcement scope. The role-based classification of PO *instances* implicitly directs their execution location among network nodes. Each Cluster Node (CN) executes PO with local enforcement scope at its PEP, allowing it to take local decisions as explained earlier. Cluster Heads (CH) additionally execute PO with cluster and hypercluster wide enforcement scope at their PDP. Cluster-wide PO are used to provision decisions to all their cluster's PEP, while hypercluster-wide PO only provision decisions to their own PEP. Finally, Manager Nodes (MN) additionally execute network-wide PO at their PMT, used to provision decisions that need to be enforced to all network nodes.

Figure 6-1 displays the deployment stages of Policy Objects. The implementing code for PO can be created either with automated code generators that parse the defined policy types' specification and create appropriate *classes* [153]. Alternatively, the code can be manually implemented to realise the policy type it represents. The latter approach was followed for the needs of presented

examples. The implementation of PO should also provide appropriate interfaces for the manipulation of their runtime parameters, in order for policy updates to be facilitated. Testing and debugging of created classes may occur before their installation. Modern programming languages provide appropriate techniques for *object* management (e.g. Java Interfaces), as well as dynamic parameter and method discovery of *class* properties (e.g. Java Reflection API).

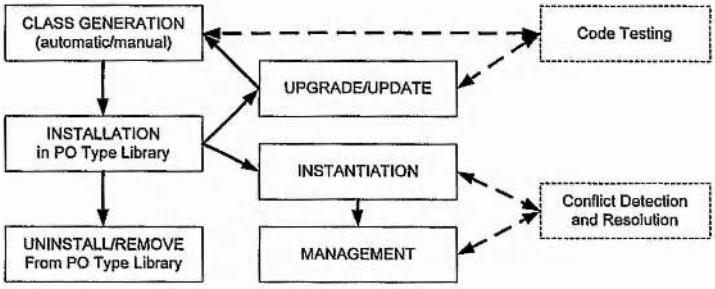

Figure 6-1. Deployment Stages of Policy Objects

Created *classes* are stored inside each device's software repository, the *Policy Objects Type Library*. Once again, *network formation* and *deployment* (§3.3, pp.56/65) affect the classes stored in each device. For example, if manager nodes (MN) are statically defined, then other nodes need not store PO with network-wide scope. Also, devices always in cluster node (CN) role (due to capabilities) need to carry only PO with local enforcement scope. The initial installation (storage) of PO inside the *policy objects type library* is normally done offline, before the system is up and running. In fact there is no operational reason restricting online installation of PO and this is one of the major benefits of policy-based paradigm. However, on the fly PO installation blurs the distinction between PBM and mobile-code techniques, bringing the drawbacks of mobile-code migration and distribution to PBM. In addition, PO classes implementation may be updated or upgraded, either because policy specifications changed or to improve and optimise code performance. Once a policy type is not needed anymore, the respective PO classes are removed from devices' software repository.

Once instantiated, PO instances enter the *policy objects lifecycle management* phase, as depicted in Figure 6-2 and explained below. Before instantiation and during their management the runtime parameters of PO need to be examined for possible static and dynamic policy conflicts respectively. This requires a policy conflict detection and resolution (CDR) mechanism to be in place, in order to prevent inconsistencies. Relevant issues of CDR have been investigated in Section 4.3, pp.87. Figure 6-2 suggests the states in the lifecycle of PO. Finite state machines and automata have been employed for managing the states and transitions in stateful PBM systems (e.g. DEN-ng [136], FAIN [169]). The proposed states' diagram is provided as a guideline, while thorough modelling of a stateful PO management protocol is out of the scope of this thesis.

Lifecycle management states guide the behaviour of an *instantiated* PO in the volatile memory of the hosting device. After instantiation, a PO is *active*, meaning it is enforcing the policy it implements. Depending on policy conditions the PO may become *scheduled,* i.e. remains in memory but not actively enforcing the policy. It may return to active state on scheduled intervals. Depending on implementation and runtime operating system or platform, a PO may become *dormant* meaning it remains in memory without enforcing the policy. Dormant objects can be used when object instantiation is more expensive than copying and/or modifying an existing in-memory object. This situation is considered for frequently accessed PO, having in mind lightweight network devices, where read access time from storage media is significantly higher than full-blown computing devices. In addition, the capability of storing PO instances inside an LDAP Directory (RFC 2713)[200], as used in Ponder Toolkit [108], can be also be exploited in combination with the high distribution degree of proposed DPR component (§5.3, pp.108). Once a PO instance in no longer needed, it enters *destroyed* state meaning it is removed from device's volatile memory.

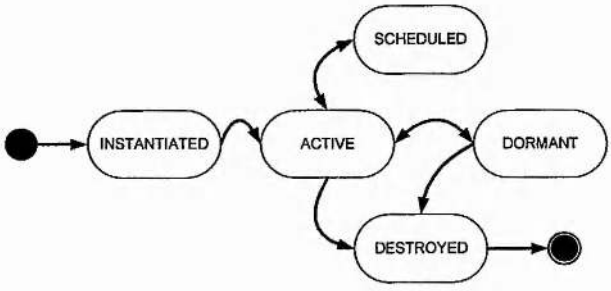

Figure 6-2. State diagram for Policy Objects lifecycle

Evidently, the most important state of a PO is when "*active*", since it is responsible for the runtime application of the policy it implements. Depending on the actual policy, PO functionality may include listening to events, evaluating conditions and executing actions. Periodic events cause a timely evaluation of conditions, typically involving averaging a metric or performing some time-based function. External events may be received asynchronously and cause condition evaluation. Conditions in PO receive contextual input from collocated or remote context-aware components. It is possible to realise respective Context Objects to allow the representation of more complicated contextual relationships (Context Modelling [2]).

6.2.3 Policy Provisioning Implementation Example

A proof of concept implementation is presented below, taking as an example the defined policy p3 for energy conservation of §4.2.2 (pp.84):

```
Policy Instances: { CN&&CH }[bp_event] if {BP=(0..33]} then {TransPow:= 2:Low Power)}
                  { CN&&CH }[ bp_event] if {BP=(33..100]} then {TransPow:= 1:Normal Power }
```

A simple cluster-wide object contains the aggregated average Battery Power (BP) of all cluster nodes. This context is reported to the policy p3, triggering conditions evaluation (bp_event). Based on the reported values, the Cluster Head decides which is the appropriate transmission power level and provisions this decision to all PEP of its cluster (cluster-wide enforcement scope).

As discussed, the presented implementation was based on XML-RPC specification which in practise allowed quick development of the provisioning protocol. As with most RPC implementations, the client-server model is adopted between two communicating parties, i.e. the PEP and the PDP. On the PEP, an embedded web server would listen for PDP requests that would remotely invoke PEP methods. Two types of methods/request were implemented: context retrieving and policy provisioning. On the PDP side, an XML-RPC client would prepare and dispatch an appropriate XML-RPC Request to the PEP address. The lightweight web server would process the request and execute requested actions. On completion, an XML-RPC Response would be returned to the PDP. Traffic measurements are presented in parallel with protocol functionality to better illustrate its operation.

PEP Context Retrieval: First, the PDP asks every PEP to report their battery status and based on replies it can calculate the average Battery Power of all cluster nodes (BP). The following Request is sent to each PEP to remotely invoke their PEP.get_battery() procedure. Each PEP replies with a Response message and includes its battery power in parameters, e.g. 88%:

```
POST /RPC2 HTTP/1.1
Content-Length: 103
Content-Type: text/xml
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.5.0_12
Host: 192.168.1.110:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive

<?xml version="1.0"?>
<methodCall>
        <methodName>PEP.get_battery</methodName>
        <params></params>
</methodCall>
```

```
HTTP/1.1 200 OK
Server: Apache XML-RPC 1.0
Connection: close
Content-Type: text/xml
Content-Length: 114

<?xml version="1.0"?>
<methodResponse>
        <params>
                <param>
                        <value><int>88</int></value>
                </param>
        </params>
</methodResponse>
```

PEP Policy Provisioning: The main functionality of the provisioning protocol is shown here. After a policy is triggered at a PDP and depending on the roles assigned to it, the PDP builds a list of enforcement targets where policy actions need to be transmitted. In the examined example, the cluster-wide enforcement scope of policies {CN&&CH} means all cluster PEP need to be enforcing appropriate transmission power. Naturally, the PDP notifies PEP only when there is a need for configuration changes or when a new PEP joins its cluster. A similar conversation is done to achieve PEP Provisioning, i.e. transfer of policy decisions to enforcement points. In this case, PEP.set_power(int) is invoked as an XML-RPC Request, defining in parameters the correct transmission power according to policies, e.g. TransPow:= 1(Normal Power). As before, PEPs reply with an XML-RPC Response message to confirm their transmission power:

```
POST /RPC2 HTTP/1.1
Content-Length: 143
Content-Type: text/xml
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/1.5.0_12
Host: 192.168.1.110:8080
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
<?xml version="1.0"?>
<methodCall>
        <methodName>PEP.set_power</methodName>
        <params>
                <param>
                        <value><int>1</int></value>
                </param>
        </params>
</methodCall>
```

```
HTTP/1.1 200 OK
Server: Apache XML-RPC 1.0
Connection: close
Content-Type: text/xml
Content-Length: 113
<?xml version="1.0"?>
<methodResponse>
        <params>
                <param>
                        <value><int>1</int></value>
                </param>
        </params>
</methodResponse>
```

The measurements taken during these conversations are shown in Table 6-1. This basic implementation provides insightful information about protocol's operation. Measurements demonstrate the most fine-grained example of conversation between a PEP and PDP, i.e. retrieval of a single context and provisioning of a single policy decision. High overheads were expected because of the use of XML with its verbose plain text encoding. In spite of the overheads, XML-RPC remains a promising solution because it is an extensible and interoperable solution. These advantages have made the successor of XML-RPC, i.e. SOAP, the de facto standard of emerging web-based management paradigms.

Table 6-1. Traffic Measurements for Policy Provisioning

	Protocol Headers	HTML Header	XML Content	TOTAL
PDP→PEP.get_battery	404	243	103	750
PEP.get_battery→PDP	404	111	114	629
Context Retrieval:				**1379 bytes**
PDP→PEP.set_power	404	243	143	790
PEP.set_power→ PDP	404	111	113	628
Policy Provisioning:				**1418 bytes**

Table 6-2. Software for Policy Provisioning Implementation

Name (type)	Full Name & Version (website)	Supported Java Version	File Size (KB)	Dynamic memory (KB)	Lic.
PEP					
cvm (virt.mach) +(library)	phoneME advanced MR2 phoneme.dev.java.net	Micro Edition J2ME/CDC/FP 1.1	3192 (+2356 lib)	3796-4480	GPL
jamvm (virt.mach)	JamVM V.1.4.3 jamvm.sourceforge.net	Standard Edition J2SE 1.4.2	184	7380-10440	GPL
classpath (library)	GNU Classpath 0.91 www.gnu.org/software/classpath	Standard Edition J2SE 1.4.2 (full) J2SE 1.5 (partial)	11264	n/a	GPL
jikes (compiler)	Jikes 1.22 jikes.sourceforge.net	Standard Edition J2SE 1.4.2 (full) J2SE 1.5 (partial)	1576	n/a	IPL
PDP					
java (virt.mach)	Java ™ Standard Edition 1.5.0 java.sun.com/j2se/1.5.0	Standard Edition J2SE 1.5	65076 (+92160 jre/lib)	13312-13516	GPL
Common					
xml-rpc (library)	Apache XML-RPC 2.0 ws.apache.org/xmlrpc/xmlrpc2	all the above	152	n/a	ASL

The software used for this implementation was based on Java and is shown in Table 6-2. In addition, Wireshark 1.0 Network protocol analyser (www.wireshark.org) was used to capture and analyse traffic incurred during reporting and provisioning. All software used was available under GNU General Public License (GPL) or equivalent [171]. IBM Public License (IPL) and Apache Software License (ASL) have similar licensing terms, equivalent to GPL in terms of software reuse. The measurements of the dynamic runtime use of memory (RAM) showed that compact Java implementations could be executed on resource-constrained portable devices (PEP).

6.3 Policy enforcement for wireless ad hoc networks

Policy enforcement is the execution of a policy decision [206]. A policy decision involves a series of triggered policy actions once a policy's conditions are evaluated as true. The actual actions can have varying granularity and different abstraction levels, ranging from device-specific parameter configuration to remote triggering of complex functions (RPC, remote procedure calls). Different levels may coexist in the same system depending on implementation.

Since policy enforcement is tightly related to policy provisioning, the requirements and obstacles mentioned earlier also apply here. Using XML-RPC for provisioning, implemented enforcement procedures generally receive technology-independent parameters that are mapped to device-dependent execution commands. The rational for this methodology is the aim for uniform management in an environment of increased heterogeneity. Device heterogeneity is also linked with the ownership relation between devices and users, who are increasingly concerned with the acquisition of their personal data.

In the following sections, further extensions are introduced to the PBM framework according to the aforementioned requirements of wireless ad hoc networks. Using the case study of *urban space networks* as described in §4.3, the introduced concepts are demonstrated through policy examples. The complexity of such environments and the vast numbers of devices provide a challenging environment where the deployment of a policy-based system can significantly simplify management tasks and accelerate device configuration. In order to effectively manage urban networks and wireless networks in general, special policy sets and types are needed. A small sample of these includes the following policy types: *Location-Based/Aware Services (LBS/LAS)* policies can provide a rich and customisable experience to a mobile user, depending on his/her physical location as well as his/her privacy settings. *Content delivery policies* can control the information that a user receives while at home or on the move. *Network-wide Preferences* policies can provide users with the recommended settings and the parameterisation of their controlled devices.

6.3.1 Selective policy enforcement for end-user privacy protection

When it comes to managing a network where the networked devices belong to individuals rather than organisations, issues like privacy and data protection should be considered. In European Union for example, strict legislation by the European Data Protection Supervisor (EDPS) mandates the processing and acquisition of personal data (Directive 95/46/EC, www.edps.europa.eu). National authorities have been established to monitor their enforcement, for example the Information Commissioner's Office (www.ico.gov.uk). Different regulations apply in the US, where a territorial approach is adopted, differentiating how personal data can be

processed in different states. This world-wide inconsistency is causing confusion and concerns to individuals who nevertheless expect their privacy to be respected. It is evident that the management of a network consisting of individuals' devices should or is legally obliged to respect the directives regarding the collection and processing of personal data. In spite of any regulatory directives, consumers are increasingly concerned with the acquisition of their personal data. These concerns place critical requirements in the design of a management framework for end-user devices: provide data protection, respect privacy and respect preferences. In order to tackle these issues, a twofold protection mechanism is incorporated in the proposed policy-based management framework:

- *User-centric control:* Individuals can set their privacy preferences to their controlled networked devices and explicitly restrict access to their personal data, regardless of network policies.
- *Policy-based regulation scheme*: The national or regional data protection authority has the ability to introduce appropriate policies to the managed system that will ensure users' personal data are not collected or exploited.

As it will be explained in the following subsections, the realisation of the described scheme is facilitated with the differentiation between managed objects to accommodate the needs of user-centric control and with the integration of Data Protection authorities in the policy definition and management process.

Policy Free and Policy Conforming Objects

The definition and differentiation between Policy Free Objects (PFO) and Policy Conforming Objects (PCO) is established in this subsection, by indicating the benefits and complications imposed to the system. The motivation behind this differentiation is also presented.

Network management can be seen as a set of operations on managed objects (MO) in order to achieve effective FCAPS management, as defined by ISO. Traditionally, a human network manager can control almost every MO in the system by setting or retrieving values, monitoring the status and reacting to reported events. In other words, a central administrative authority owns and controls the managed network. But as previously explained, the case of wireless ad hoc networking is fundamentally different from traditional networks, especially in environments like urban spaces. Individual users are reluctant to entrust the management of their devices to a central authority and demand more control over their owned devices. This contradiction has motivated the idea to differentiate MOs and introduce Policy Free Objects (PFO) and Policy Conforming Objects (PCO).

A policy-based management system automates the control of network devices, by enforcing policies over their managed objects (MOs). *Policy Free Objects (PFO)* are defined as the MOs of a networked device which are directly controlled by device's owner and their values and/or status *are not* influenced by policy decisions. *Policy Conforming Objects (PCO),* similarly to traditional MO, are controlled by the PBM system, i.e. their values and/or status *are* influenced by policy decisions. Figure 6-3 presents conceptually the above definitions, while Table 6-3 shows the classification rules depending on user input. *Managed Objects* (MO) are grouped in two categories: *manageable* and *access control.*

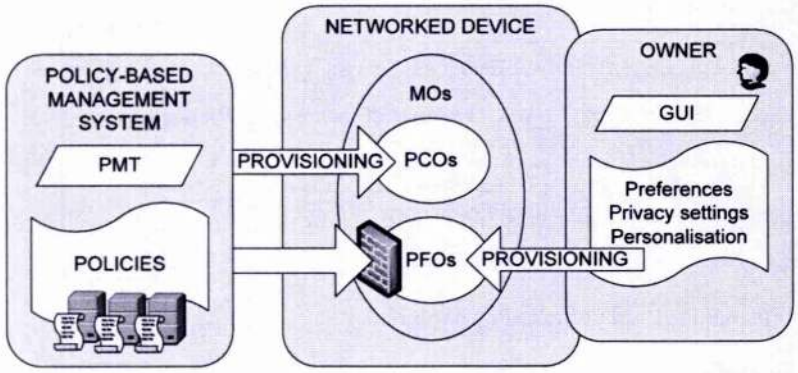

Figure 6-3. Policy Free and Policy Conforming Objects

The introduced separation significantly differentiates traditional enforcement and resulted in redesigning architectural aspects of devices and their PEP. Some optional elements are needed to facilitate selective enforcement and offer more user control. First a Graphical User Interface (*Node GUI*) provides to the device owner the ability to set privacy and preferences settings. User input is examined by the *Preferences and Settings Translator* element that classifies managed objects to policy-free and policy-conforming and through the *CN Communication Adaptor* transfers this information to *PFO/PCO Access Control* element for access control enforcement. Enforcement is carried out on device's *Managed Objects* as well as *Local Policy Objects* and *Context Objects.*

The values of *manageable* objects can be directly configured according to their allowed values or values range. Access control objects are related to a binary decision to allow or restrict access to their controlling data. Such data can either be information that has an external data provider (e.g. location data from GPS receiver) or another MO (e.g. one of the manageable ones). These values and classification decide on the read and write access (RA, WA) rights for the policy-based management system, i.e. whether policies can access and modify parameters on user devices. The concept is still applicable for devices directly managed by the network operator, as well as legacy devices. In these cases, all managed objects are considered as policy-conforming (PCO) and PBM is carried out as normal.

Table 6-3. Classification and access rights for Managed Objects

MO Type	*MO Value*	*Read Access (RA)/ Write Access(WA)*	*Classification*
Manageable Object	user specified	RA Allowed WA Restricted	PFO
	auto	RA Allowed WA Allowed	PCO
Access Control Object	yes	RA Allowed WA not applicable	PFO
	no	RA Restricted WA not applicable	PFO
	auto	RA policy-defined WA not applicable	PCO

6.3.2 Realisation of End-User Privacy Protection

This section provides details and implementation guidelines for integration of the End-User Privacy Protection scheme with the proposed framework. The twofold protection mechanism of user's privacy and preferences is described. First, the user-centric control scheme employs the defined Policy Free and Policy Conforming Objects with example policies. Next, the details of the policy-based regulation scheme are presented with applicability examples. For policy examples, events are omitted since policies are grouped under the same triggering events and a description of the event is provided for better understanding. Finally, the role assignment of these policies is to CN only, which means they are applied to cluster nodes and are triggered individually for each one of them (local enforcement scope).

User-centric control of privacy and preferences

As outlined earlier, individual users are reluctant to grant complete control of their devices to a central authority and demand more influence on owned device's behaviour and data disclosure. The presented idea of Policy Free and Policy Conforming Objects (PFO/PCO) can accommodate these demands and offer a way for users to set their privacy preferences and explicitly restrict access to their personal data, regardless of network policies.

As proof of concept, an example is presented based on the *urban spaces* case study. First, a limited set of MO grouped them in two categories: *manageable* and *access control*. Table 6-4 lists the defined MO and their allowed values for this case study. To accommodate user control, the devices' owners are allowed to set their preferences using a user-friendly interface (GUI) in order to set the values of selected MO. Depending on the users' input, MO are classified as *policy free* (PFO) if an explicit value has been set, or *policy conforming* (PCO) if their value was set to *auto*. The mapping is straightforward and lightweight allowing devices to automatically carry out the classification. As a result, read/write permissions are set by the users for the information they consider private, as well as their preferred values for device settings.

Table 6-4. Defined Managed Objects for Case Study

Manageable Object		
Code	**Meaning**	***Allowed Values***
DST	Device Status	on, off, auto
PWU	Power Usage	normal, low, sleep, auto
SBW	Shared Bandwidth	[0-100]% , auto
SMR	Shared Memory	[0-100]% , auto
Access Control Object		
Code	**Meaning**	***Allowed Values***
SL	Show Location	yes, no , auto
SB	Show Battery	yes, no , auto

To better illustrate the concepts, an example user configuration and related policies are explained. Based on the defined MO, a user decides on his/her preferences and desired privacy levels and using the GUI defines the values shown in Table 6-5. The two rightmost columns show the effect of user's decision, in terms of *read* and *write* access to MO and their respective classification as PFO or PCO. The MOs that had their values explicitly set by the user are classified as PFO and they will not be affected by network policies (DST, SMR, SL, SB). The ones with values equal to "auto" are classified as PCO and the PBM system can access and modify them (PWU, SBW). The management system can operate regardless of users' selection, but cannot override their preferences.

Table 6-5. Example privacy and preference settings of Managed Objects

Manageable Object		*User Input*	*Read Access (RA)/ Write Access(WA)*	*PFO/PCO*
Code	**Meaning**			
DST	Device Status	**on**	RA Allowed WA Restricted	PFO
PWU	Power Usage	**auto**	RA Allowed WA Allowed	PCO
SBW	Shared Bandwidth	**auto**	RA Allowed WA Allowed	PCO
SMR	Shared Memory	**30%**	RA Allowed WA Restricted	PFO
Access Control Object		*User Input*	*Read Access (RA)*	*PFO/PCO*
Code	**Meaning**			
SL	Show Location	**no**	RA Restricted for Location data	PFO
SB	Show Battery	**yes**	RA Allowed for Battery status	PFO

Table 6-6 contains management policies introduced by the network operator. Based on user's preferences, policies P1, P2 will affect the particular user, while policies P3 and P4 will not. For simplicity, example policies are not overly complex, yet useful enough to demonstrate the proposed concepts. The case study assumes a network consisted of personal devices owned by network users (mobile phones, PDAs etc), as well as devices controlled by the network operator (NO devices: information kiosks, wireless traffic cameras, etc). Some of the networked devices

may operate unsupervised and the management system must ensure their proper operation. The NO introduces the policies of Table 6-6 to the system, with the purpose of conserving the battery of managed devices (P1,P3) and to allocate shared resources according to device statistics and remaining battery (P2,P4). Statistics such as the average free bandwidth (avgFreeBW) and memory (avgFreeMR) are recorded by devices. These local statistics can be used in policy conditions to trigger policy actions with local effect.

Table 6-6. Network Operator Policy Examples

P#	Policy	affects example user
P1	if (SB=yes)^(Battery>30%) then setPWU(normal)	yes
P2	if (SB=yes)^(avgFreeBW>60%)^(Battery>80%) then setSBW(40%)	yes
P3	if (time=[2:00..4:00])^(avgFreeBW>90%) then setDST(off)	no
P4	if (Battery>50%)^(PWU:=normal)^(avgFreeMR>60%) then setSMR(50%)	no

The user of the example defines his/her preferences for the owned devices, by explicitly setting the device status to on and the shared memory to 30%. Also, the user restricts access to his/her location data but allows the PBM system to read the battery status. As a result, policies P3 and P4 do not apply to the user's device, while policies P1 and P2 do apply and configure the PCO objects, i.e. the shared bandwidth and the power usage profile. Regarding data protection, the disclosure of the user's current position is protected but he/she may not benefit from Location-Based Services (LBS) that utilise positioning details (Figure 6-4). The PFO/PCO scheme is not affected by the presence of devices owned by the NO even if those do not support PFO/PCO managed objects. Such devices operate as normal policy controlled devices, i.e. all their objects are set to PCO status, thus allowing their full configuration.

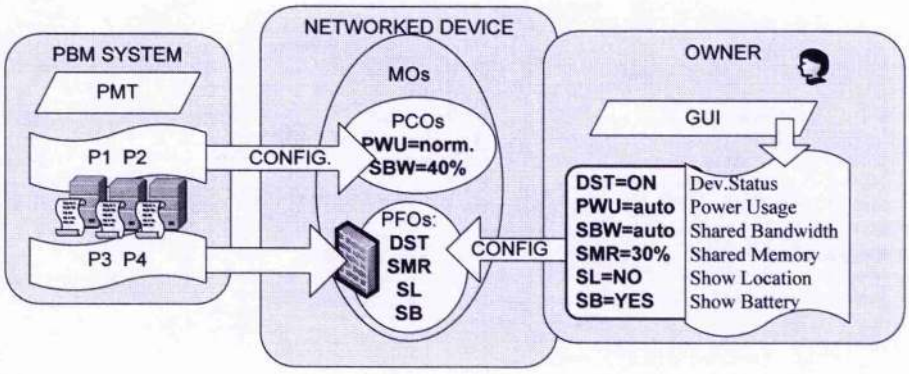

Figure 6-4. Device configuration with example user's privacy and preference settings

Returning to a previous example (§4.3.2, pp.91), employed earlier for investigating inter-manager conflict detection, those policies would need to be revised, taking in mind user's preferences and settings. The policy type below was used to configure the shared bandwidth of users when they

enter a location with limited bandwidth. However, the condition of the policy type should be adapted as shown to comply with the introduced scheme:

```
{CN} [newUser]   if (SBW==auto )^(SL==yes)^(locateUser(Stadium))
                 then setBW((SBW:=X%),(mngBW:=Y%),(p2pBW:=Z%))
```

By adding two additional conditions {(SBW==auto)^(SL==yes)}, policies of this type take in mind privacy settings of users about location disclosure (SL) and preference about permitting policies to alter the bandwidth the user is prepared to share (SBW). The values set by the example's user are such that policy conditions will evaluate as true, resulting in automatic configuration of shared bandwidth by policies, as agreed by the network and service providers. Some users though may choose not to reveal their location data by setting SL==no and as a consequence the policies above will not affect their devices.

An apparent question regarding the presented scheme is whether the providers actually include necessary conditions, in order to check user's preferences before accessing the managed objects on user devices. Assuming they may not, either because of wrong policy specification or because they attempt to commercially exploit these data, then an additional protection mechanism should be in place. For this purpose, a policy-based regulation scheme is proposed, to protect users from unfair data exploitation and enforce the regional data protection regulations.

Policy-based regulation scheme and privacy issues

In addition to the explicit user defined preferences, the PBM system has the ability to control unfair exploitation of user data by deploying a regulation scheme with appropriate policies. In §4.3, the rationale for multiple managers and the notion of *"eligible entities"* has been explained. Based on the same multi-manager case study, this subsection explains how the regulations of data protection can be enforced in the system and more importantly not overridden. For this example, a data protection agency is considered as an *eligible* entity that has the control of one manager node (MN), for example UK's national agency, the Information Commissioner's Office (ICO). Using the policy management tool (PMT) interface, the ICO has the ability to manage the lifecycle of policies and introduce appropriate policies to the managed system according to current regulations. In addition, it can review, edit or disable existing policies to ensure users' personal data are not collected or exploited by other "eligible entities"; in this case study, by the network operator or a service provider.

For example, users who are willing to reveal their location data (SL=yes) should be protected from services that can continually track their position. Tracking is possible by frequently polling the user location and comparing consecutive measurements, depending on the accuracy of the available positioning method and the user speed. With the increased penetration of high accuracy GPS-enabled devices in the consumer market and the improvement of indoor positioning

methods, this issue is becoming quite important. Let us assume that current regulations state that "tracking the position of civilians is allowed within a circular area of uncertainty that has a defined minimum radius", setting a minimum radius for pedestrians (min_rad) of 100m. The high-level policy in this case states:

> *Tracking the position of civilians is allowed within a circular area of uncertainty with a minimum radius of 100m*

Using simple physics equations (speed = velocity*time), the high-level policy can be translated in low-level directives. The polling interval of location data must have a minimum value (Min_poll_int) so that between consecutive polls, the user can be found in an area with high uncertainty, i.e. uncertainty radius>min_rad. Based on equation 6.1, Figure 6-5 graphically illustrates the uncertainty area between consecutive polls (t_o , t_o + poll_int):

$$uncertainty\ radius = accuracy + speed*polling\ interval \tag{6.1}$$

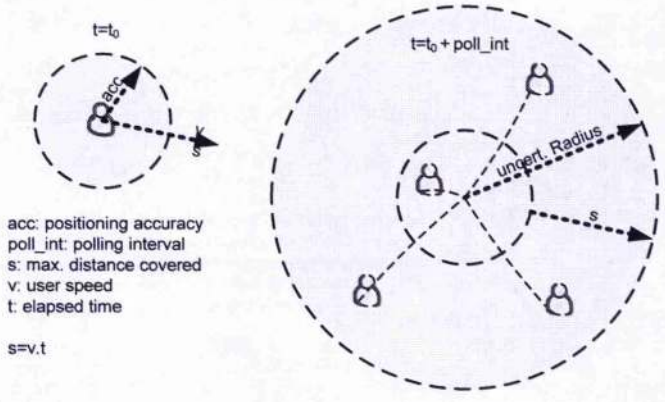

Figure 6-5. Graphic representation of user location uncertainty

Using equation 6.1, the ICO can formulate an appropriate low-level policy that will enforce the described regulation and high-level policy:

```
if    (SL=yes)^(0<Loc.speed<1.5m/s)^(Loc.accuracy<min_rad)
then  set_Min_poll_int((min_rad-Loc.accuracy)/Loc.speed)
```

Further than configuration policies, a regulatory body can use the policy-based system to monitor the collection of user data and gather information for offline processing. Simple policies can periodically log information about the services that retrieve user data. The logged details can be reviewed and analysed statistically to extract information about how service providers use the location data of users and further investigate their unfair exploitation.

Continuing on the topic of regulatory policies and their enforcement, the problems related to wireless ad hoc networks deployment will be examined later (§7.2,pp.160). Regional regulations may restrict the use of specific channels for ad hoc networks, hence ad hoc network users may involuntarily break the law, especially if using the default settings of their devices in a different

geographic region. However, end-users have no need to be aware of channels and regulations, as long as they are connecting to infrastructure-based WLAN, regardless of their geographic area. In managed WLAN, devices connect to infrastructure-based wireless access points (AP) and automatically tune to the correct channel, thus reducing the probability of misconfiguration. The problems described are bound to ad hoc networks, since it is up to the initiating device to select a channel for deployment. It would be useful to ensure that roaming users are conforming to regional regulations with minimal inconvenience. Therefore, a solution is proposed based on special regulatory policies, controlling the initial deployment of ad hoc networks. Further details are given in §7.2 (pp.160).

As presented, the flexibility of a PBM system allows complex policies to be formulated during runtime and be introduced to the system without disruption. This allows managing entities to adapt to changes and simplifies the complex task of configuring a large scale network, as in the examined case study. A change in regulations can be applied by editing existing policies or introducing new ones, without disrupting the operation of the network and affecting the users. From a business point of view that means less cost for software maintenance and less effort for manual configuration and updates of devices. However, from an administrative point of view, the system should incorporate sophisticated mechanisms to resolve policy conflicts in the multi-manager environment, as described in §4.3.2 (pp.91).

6.4 Summary and Conclusion

In this Chapter, two important operations of a policy-based network management system were examined and solutions were proposed specifically targeting wireless ad hoc networks. Policy provisioning and enforcement were adapted to their requirements, departing from traditional solutions that were deemed unsuitable. To facilitate the communication between PEP and PDP, a lightweight policy provisioning protocol was proposed, based on the Remote Procedure Call (RPC) paradigm and the use XML over HTTP as the transport mechanism. The protocol was implemented using and extending XML-RPC protocol [157], having in mind the wide support of XML/HTTP technologies by virtually all networked devices. A proof of concept implementation was presented, taking as an example a policy for energy conservation from §4.2.2 (pp.84).

The lifecycle and management of policy objects (PO) was also examined, focusing on mirroring the role-based aspects of the proposed organisational model to PO management in order to increase scalability. The enforcement scope of each policy was used to classify runtime PO *instances* in node, cluster, hypercluster, and network -wide PO, thus enabling task delegation and distribution among network nodes. Delving into aspects of policy enforcement in user-owned wireless networks, the increased concern of consumers about the acquisition of their personal data

was addressed. By departing from traditional uniform policy enforcement, new concepts for selective policy enforcement were introduced. For this purpose, a twofold protection mechanism was integrated to the proposed PBM framework, offering user-centric control and integrating a policy-based regulation scheme.

Concluding this Chapter, the aim for uniform management in an environment of increased heterogeneity was addressed and different innovative solutions were proposed. Regarding policy provisioning and enforcement, it was observed that the target wireless environment and the wide use of lightweight devices favoured the *provisioning* model over the *outsourcing* one. The main reason was its inherent asynchronous operation, which allowed end-devices to operate mainly unsupervised, based on the provisioned policy directives they received. A technology-independent *policy provisioning protocol* based on XML-RPC was implemented to transfer decisions between PDP and PEP. This middleware approach mainly involved remote procedure calls from PDP to PEP that were mapped to device-dependent execution commands. This has significantly preserved system's extensibility and wider applicability, because it allowed the majority of developed software to remain device-independent and only required development of device-dependent functionality on PEP.

Device heterogeneity was also linked with the ownership relation between devices and users. Based on the essential requirements differentiation between the management of wireless networks consisted of user-owned devices instead of organisation-owned, important issues like privacy and data protection needed to be addressed. The main requirement was to respect users' preferences and safeguard the unfair use of their personal data. Therefore a twofold scheme was proposed that prevented manager entities to acquire information against the users' will and offered more control to the device's owner. The examined case study referred to a trusted environment, assuming that the wireless network was always managed by trusted entities. The case of non-trusted environments with possibly compromised manager entities poses the requirement of rigorous security schemes and malicious node detection which are challenging aspects for future investigation.

Chapter 7

Validation Case Studies

7.1 Introduction

Self-management is a complex goal that has been closely related with *autonomic computing*, self-organising and self-maintained systems. As it has been explained, researchers have separated self-management operation into four desired capabilities, each of which is contributing to the overall goal of enabling fully self-managed autonomic systems [134]. By adopting a gradual transition towards self-management, two of the self-management capabilities were addressed in this Chapter, i.e. *self-configuration* and *self-optimisation*. Having in mind that a complete self-management prototype is not yet available, the design and implementation of a partially self-managed wireless system is presented and evaluated.

The first presented case study deals with the dynamic configuration of the communication frequency (channel) in a wireless ad hoc network based on IEEE 802.11. The solution addressed the *self-configuration* of ad hoc networks deployment by initiating communications using the best available wireless channel. The second issue addressed was the *self-optimisation* of wireless ad hoc communications by evaluating wireless channel conditions and dynamically switching to a new optimal channel. Currently, in dense deployments of WLAN (e.g. conferences, convention centres) users tend to manually initiate ad hoc networks without relying to any infrastructure support. The ad hoc nature of configuration and spontaneous network creation has resulted in poor performance and interference problems among WLAN, not to mention regulatory violations in some cases. The deployment of ad hoc networks and their coexistence with managed WLAN has not received enough research interest, since in most cases it is assumed that an area free of interference is available and all ad hoc stations communicate using the same channel. These assumptions had allowed research to focus on inter-station interference and MAC layer

performance, yielding fundamental theoretical background for wireless ad hoc networks and MANET in particular. On the other hand, industrial interest for MANET has been limited, mainly due to the lack of a compelling business model.

Taking a step further towards self-management, the second case study elaborates on the need for novel service management solutions, which would enable flexible and customisable service provisioning to users of wireless networks. The increasing numbers of wireless devices and the spontaneous nature of their interactions are not catered from current service management frameworks. On top of that, increased device heterogeneity further hinders service provisioning and fails to meet users' expectations. Both service providers and users can benefit from an open service market where user's preferences are better satisfied. Today's constant need for accessing any kind of information, anytime, anywhere, further motivate new management paradigms. Inevitably, a novel framework for service management is required, taking into account the diverse conditions and requirements of wireless networks. Industrial predictions mention that "extending the service portfolio is one of the best options for growth" [172], thus fuelling more research interest in novel solutions for mobile users. The proposed adaptive service management framework extends the presented PBM framework and builds on its hybrid organisational model for wireless ad hoc networks. A number of features of the PBM framework were deemed useful for wireless service management. By combining the benefits of hierarchical and distributed management schemes, the hybrid model offers the desired properties of policy-based management through multiple PMTs, distributed decision making by cooperating PDPs and distributed policy storage in DPR.

7.2 Self-management capabilities for wireless ad hoc networks

Self-Configuration and Self-Optimisation were the first capabilities investigated since they are closely interrelated in terms of functionality. A system's configuration needs to result in effective operation and high performance; therefore self-configuration needs to be oriented towards optimised solutions. Respectively, self-optimisation needs to discover the configuration settings that will improve and increase System's performance. This close relation and interaction has motivated efforts towards a first step for the implementation of fully self-managing wireless ad hoc networks. The described case study of wireless ad hoc networks was suitable to fully exploit the benefits of the aforementioned policy-based framework. For this purpose, necessary policies and algorithms were designed for the deployment of such networks, while their performance and applicability were evaluated through testbed implementation. By making appropriate policies available in the DPR, user devices were assisted by receiving guidelines that would transparently configure the ad hoc network, choosing the best available wireless channel to avoid interference

and dynamically switching channels if performance degrades. The presented solution effectively addressed the self-configuration and self-optimisation needs of channel assignment in wireless ad hoc networks, making an important step towards the implementation of fully self-managing systems.

By facilitating a predictable and controlled ad hoc network deployment, the performance of both wireless ad hoc networks and infrastructure-based WLAN can be significantly improved. One of the first issues that need to be addressed is channel assignment in wireless ad hoc networks. The proposed solution can be deployed on top of existing and future access networks using a technology-independent policy-based management layer. The solution spans among different layers of the protocol stack, exploiting context and cross-layer principles, while preserving layers modularity at the same time. This paradigm was deemed necessary, since the applicability domain of ad hoc networks is based on a majority of off-the-shelf end-user devices and normally includes only a few special purpose devices, e.g. mesh routers or programmable access points. In addition, standards conformance is an important aspect for the applicability of any to solution.

Cross-layer communication was used between 802.11 MAC sub-layer [183] and Application layer, aiming to make the PBM system aware of wireless channel conditions. This specialised context collection method provides a feedback mechanism for policies. Based on specified application events (e.g. reduced goodput), the triggered policies can initiate relevant procedures that after the inspection of MAC headers, provide feedback to the system and possibly trigger further policies to correct the problem or report unresolved issues to the user or the network manager. As already explained in §3.2 (pp.54), a closed control loop is formed that adds a degree of self-management to the network. There are two important advantages with the adoption of this approach:

- By using a policy-based design, the system is highly extensible and easily configurable. Policies can change dynamically and independently of the underlying technology.
- By implementing decision logic, based on policies and extracted inter-layer context at the Application layer, modularity is preserved without modifying the MAC protocol.

Two potential obstacles have been already identified and need to be overcome in order to make the deployment of ad hoc networks easy, efficient and safe:

1. Interference between wireless ad hoc and existing WLAN networks

   The main reasons for the disappointing performance of ad hoc networks are interference between newly formed ad hoc networks and existing infrastructure-based WLAN, as well as interference with already deployed ad hoc networks in the same area. These can lead to severe problems in the throughput and coverage of collocated infrastructure-based WLAN. As

already mentioned, devices operating in unlicensed ISM bands can arbitrarily use any of the defined channels and should be able to cope with interference from devices competing to access the same unlicensed bands. The MAC sub-layer can be fairly tolerant against interference and noise at the cost of speed and performance. Choosing a random deployment channel is likely to have a detriment effect for the ad hoc network performance. The above problem has been verified by testbed measurements. To tackle this problem, policies P1 to P8 (Table 7-1) were designed to exploit context extracted from MAC sub-layer, firstly for initial channel configuration of new wireless ad hoc network and secondly for the dynamic adaptation of the wireless channel of deployed ones

Table 7-1. Wireless Ad Hoc Networks Self-Management Policies

P#	*Event*	*if {Conditions} then {Actions}*
1	Init_new_adhoc	if {ready} then {scanChannels()},{generateScanComplete(results)}
2	ScanComplete(results)	if{otherWLANdetected=true}^{FC:=freeChannels(results),FC=true}^{PC:= preffered(FC,*ch_list*),PC=true} then{optimizeChannel(PC,$algorithm_1$($criteria_1$))}
3	ScanComplete(results)	If {otherWLANdetected=true}^{FC:=freeChannels(results),FC=true} ^{PC:= preffered(FC,*ch_list*),PC=false} then{optimizeChannel(FC,$algorithm_2$(criteria$_2$))}
4	ScanComplete(results)	if{otherWLANdetected=true}^{FC:=freeChannels(results),FC=false} then{optimizeChannel(all, $algorithm_3$ ($criteria_3$))}
5	NewWLANdetected	if {*dyn_adapt*=true} then {generateStartAdapt(newWLANinfo)}
6	LinkQualityCheck	if{LinkQuality<thr_a}^{*dyn_adapt*=true} then{generateStartAdapt(cachedWLANinfo)}
7	StartAdapt(WLANinfo)	if {channel_distance(WLANinfo,current)<*dist*} ^{*app_specific_metric*<thr_b) then{scanChannels()},{generateAdaptChannel(results)}
8	AdaptChannel(results)	if {results_evaluation()=true} then{channel_switch(all,$algorithm_4$($criteria_4$))},{verify_switch()}
9	SystemBoot	if {region=FCC} then set_criteria(approvedChannels[$list_1$])
10	>>	if {region=ETSI} then set_criteria(approvedChannels[$list_2$])

2. Regulatory conformance of ad hoc networks deployment

Although this issue is rarely addressed, it is indirectly affecting the popularity and usability of wireless ad hoc networks. Users attempting to deploy wireless networks may be breaking the law, especially if their devices have been configured with the default ad hoc network settings of a different geographic region than their current. Taking for example 802.11b technology in 2.4GHz ISM band, according to IEEE Std. 802.11-2007 [183]: "Channel 14 shall be

designated specifically for operation in Japan" ([183]:pp.566,674). This means that the regulatory domain of Japan allows the use of all 14 defined channels of the 802.11b standards for the deployment of WLAN. For most devices used in this region, the default channel for ad hoc deployment is channel 14. However, the rest of the regulatory domains, e.g. Europe (ETSI) or Americas (FCC), explicitly forbid the use of Channel 14 for 802.11b WLAN. In FCC domain, Channels 12 and 13 are also forbidden. Adding to the confusion of ad hoc network users, France and Spain further forbid different channels ([183]:Tables 15-7, 18-9). To prevent such problems, additional policies (Table 7-1:P9,10) can be introduced by the regional network managers, which in turn influence the criteria for the policy-based channel selection described later (Table 7-1:P2,3,4,8). For example, in America (FCC) policy P9 applies with *list1*={1..11} and in Europe (ETSI) policy P10 applies with *list2*={1..13}. Similarly, additional policies can be defined for current and future technologies, e.g. 802.11a [184] or 802.11n [185].

To illustrate the proposed solution, wireless networks based on IEEE 802.11 [183] were investigated, since currently this standard is the most widely deployed technology for WLAN and offers support for ad hoc networks (§2.3.5,pp.33). Once a user initiates an ad hoc network using a device supporting 802.11b/g, the device is set in IBSS mode (ad hoc/peer mode) and device-dependent software and hardware configure the transmission parameters. The device assumes the role of the wireless Access Point and its wireless interface begins to emit beacon messages advertising the existence of an ad hoc network on the statically defined channel. Other parameters are also advertised, like the beaconing interval and any encryption methods used, thus enabling nearby in-range devices to join the ad hoc network in a peer-to-peer manner. Additional details can be found in §2.3.5 (pp.33). Assuming a realistic deployment in a populated area and not in an anechoic chamber, such deployment would imply the coexistence of various WLAN (either ad hoc or infrastructure-based) and inevitably their interference. Choosing the default channel or even a random channel is likely to have a detriment effect for the ad hoc network performance. Unwanted side effects will also be noticed in the operation of nearby infrastructure WLAN or ad hoc networks. The problems arise from the access to the wireless medium and three cases can be identified during the deployment of an ad hoc network on a specific channel:

a) The channel is already in use by other WLAN

b) Adjacent or nearby channels are in use by other WLAN

c) No nearby channels are in use by other WLAN

In practice, cases b) and c) are difficult to be separated since co-channel interference depends on unpredictable environmental factors and is also technology-dependent. Term "nearby" implies the channels that are closer than the next adjacent non-overlapping channels and is also technology-

dependent. The above cases were examined on an experimental testbed and measurements were taken. The policy-based solution was deployed, aiming to dynamically assign the best available channel and autonomously adapt to changes in the wireless environment.

To prevent the detrimental effects of interference, context information was used, extracted from the headers of Layer 2 frames. This can be achieved by two methods explained below. Either method can be used depending on the scenario and hardware support:

1. The device is using the wireless interface to passively monitor all packets it can hear (also known as "rf-monitor") and forwards them to the monitoring policies for processing of the 802.11 MAC headers. Therefore the device can extract useful information about the Data Link Layer performance of its one-hop neighbours and by processing this information can trigger appropriate adaptation policies. The advantage of this method is that it fully exploits management frames and headers of 802.11 without associating to a wireless access point (AP) or network. If the device has more than one wireless interfaces it can also assess its own performance. The drawback of this method is that the monitoring interface cannot be used for communication.
2. The device is using the wireless interface in "promiscuous" mode and associates to a wireless network as normal. The traffic packets received by the device are examined and information can be extracted from them. In this case not all packets transmitted on the channel are captured, since the device cannot overhear the channel while transmitting. This may be a drawback since the device cannot have a complete view of the neighbourhood and may continue to cause interference to other devices without being able to detect that. However, the apparent advantage is that the device can still use the interface for communication, which is important in the case of devices with a single wireless interface.

In order to assess the performance of the designed policy-based approach, a wireless testbed was used for evaluation, implementing critical aspects. In addition, the testbed was used to measure the effects of interference between devices using the same channel or devices with varying channel distance. Experiments were performed in a confined indoor space, matching the typical conditions of the described case studies.

The experimental testbed was consisted of 10 nodes: 2 laptops, 8 portable wireless devices, namely 4 PDAs and 4 Internet Tablets. All devices were equipped with internal 802.11b wireless interfaces, while the two laptops had an additional PCMCIA external wireless card. Table 7-2 includes more information on the used equipment. For the configuration of the wireless interfaces, Linux scripts were used with *wireless-tools v28*. The source code of *airodump-ng* was modified for monitoring the wireless channel (a popular open source 802.11 packet capturer, part of the

aircrack-ng suite www.aircrack-ng.org). These modifications allowed the inspection and dynamic use of captured information within the policy-based interface. Communication between nodes was done either by SSH (secure shell protocol) or by HTTP.

Table 7-2. Wireless Testbed Specifications

	Operating System (Linux Kernel)	*Processor (MHz -family)*	*Ram (MB)*	*Wi-Fi support*
Sony Vaio Z1XMP	Debian R4.0 (2.6.18)	1500 - Intel	512	802.11bg (x2)
HP iPAQ H5550	Familiar v0.8.4 (2.4.19)	400 - ARM	128	802.11b
Nokia N800	IT OS2007 (2.6.18)	330 - ARM	128	802.11bg

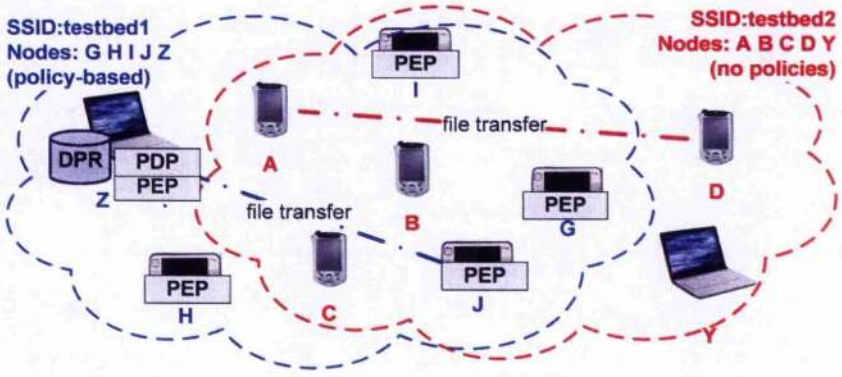

Figure 7-1. Wireless Ad hoc network testbed deployment

For the purpose of performed experiments, the devices were organised in two independent clusters of five nodes as seen in Figure 7-1. The clusters were setup using different SSID (Service Set Identifiers) in IBSS (ad hoc) mode. The manufacturers default channel for ad hoc networks creation was found to be Channel 1 (2412MHz). The network speed (rate) was set to 11Mbps, to allow comparable results among nodes. One of the clusters (testbed1) integrated policy-based (PBM) support and the cluster head employed a PDP for the needs of its cluster. After the PDP had retrieved policies 1 - 8 (Table 7-2) from the nearest DPR (in this case collocated), it had accordingly instantiated policy objects (PO) for monitoring conditions and provisioning actions among cluster nodes. For evaluation purposes the PBM support was selectively used to measure its effect on network performance.

7.2.1 Channel Selection Algorithm

The implemented PBM system, integrates channel selection algorithms, used in the actions of policies P2, P3, P4 and P8. Triggered actions *optimizeChannel(channel_set,algorithm$_n$(criteria$_n$))* and *channel_switch(channel_set, algorithm$_n$(criteria$_n$))* are called using as parameters the monitored measurements of a channel set (e.g. FC: free channels, PC: preferred channels) and the

algorithm with criteria to be used for channel selection. For the purpose of this case study an algorithm was used based on the weighted average (WA) of a channel metric.

$$WA(x)= \left[\sum_{i=1}^{n} (w_i x_i)\right]\left[\sum_{i=1}^{n} (w_i)\right]^{-1} \quad (7.1)$$

Elaborating on the algorithm, the *criteria*$_n$ parameter of each policy specifies the channel metric (x) and weights (w_i) to use for the calculation of the WA, for each candidate channel. The flexibility of a PBM design is evident, since different algorithms, metrics and criteria can be used to achieve the desired management objectives.

Policies P2, P3 and P4 have similar functionality, which is to select the best available channel during initial ad hoc deployment. The triggering of *optimizeChannel(_,_)* method is controlled by the scan results of P1 and specifically the availability of preferred and/or free wireless channels. Candidate channels are included in the *channel_set* parameter, together with the *algorithm*$_n$*(criteria*$_n$*)* to use. Currently all algorithms (1-4) are based on the calculation of the weighted average (WA) of a channel metric, while customisation and fine-tuning of policies is achieved by differentiating *criteria*$_n$ that specify the channel metric (x) and weights (w_i).

By using the source code of wireless packet capturer *airodump-ng* (www.aircrack-ng.org), the developed custom version allowed the extraction of valuable cross-layer information, without breaking layers modularity. Some of the available metrics can be calculated internally by *aerodump-ng* for existing SSID (Service Set Identifiers) occupying each available channel. SSID can be advertised by infrastructure-based (BSS/ESS) or ad hoc wireless networks (IBSS):

- moving average of signal power, using a configurable period
- signal link quality, as calculated by the percentage of captured beacons
- amount of captured or missed frames and respective frames/second
- number of data packets and data packets/second

In addition, after the initial deployment, application specific metrics could also be collected. As will be explained later, such metrics are more useful for triggering policies involved in dynamic adaptation, e.g. the use of the moving average of goodput measurements in policy conditions.

For metric (x), the monitored average packet/sec metric was used to calculate the WA for all allowed channels and select the one with the minimum value. Linear weights decrease arithmetically as the measurements of less interest are included in a weighted averaging process. For example, to evaluate Channel 1 and calculate the WA of metric *x* on Channel *i*, it is expected that nearby channels (e.g. 2 or 3) to cause more interference than the distant ones (e.g. 4 or 5). It is also noted that Channels 1 and 6 are considered as non-overlapping hence significant interference is not expected (Figure 2-2,pp.36). The initial assumption was that frequency distance affects

weights in an inversely proportional relation. By considering N=5 channels (i.e. for channel distance i = [0,N]), the linear weights are shown in Figure 7-2. Weights are a convex combination, i.e. normalised so their sum is 1. Due to the symmetric distances between channels, these weights are extended towards negative channel distances, aiming to cater for cases where channels can experience interference from both sides of their central frequency. E.g. channel 8 can experience significant interference both from channels 7 and 9. By mirroring weight values to their negative i values (i.e. channel distance i = [-N,N]) and recalculating their convex combination, these new weights are shown in Figure 7-3.

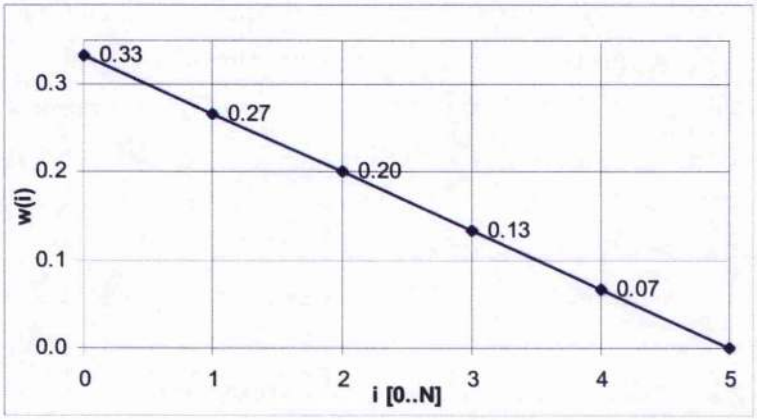

Figure 7-2. Arithmetic (linear) weights' distribution

The performed static channel measurements significantly affected the calculation of weights (w_i), differentiating this approach from the initial assumption of linear weight distribution. Having identified the detrimental performance of consecutive channel deployments, as verified by measurements of Table 7-3, the performance degradation of goodput was normalised, to produce the new empirical weights distribution. A graphic representation of weights is depicted in Figure 7-3, compared to the initial mirrored linear distribution.

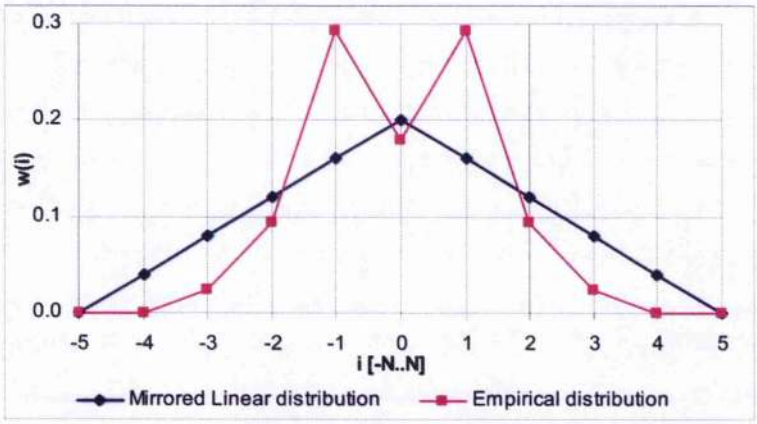

Figure 7-3. Mirrored linear and empirical weights' distributions

This empirical weight assignment has the advantage of using real performance measurements and can be used to dynamically adapt the weights and effectively the selection of a better channel, either for initial wireless ad hoc deployment or for dynamic channel switch. As will be detailed below and shown by testbed measurements, the described algorithm and parameters have identified a better channel to avoid interference.

7.2.2 Self-Configuration for Initial Channel Assignment

Experiments first involved static measurements to evaluate wireless channel performance in the presence of multiple ad hoc networks with varying channel distance. According to this scenario, the two clusters would simultaneously attempt to initiate file transfers among peers of the same cluster, as shown in Figure 7-1. First, the two ad hoc networks were formed on the default channel (channel 1). Using the same channel for both clusters was made possible by using different network names (SSIDs), namely "testbed1" and "testbed2". Afterwards, the same networks were deployed in different channels and file transfers were performed. While "testbed2" was always deployed on the default channel 1, "testbed1" was deployed on channels 1,2,4 and 6 to vary channel distances and evaluate the effect of interference. Figure 2-2 (pp.36) shows available channels and spacing for 802.11b/g, where a total of 13 central frequencies is defined with a 5MHz spacing and a required channel bandwidth of 22MHz. As it has been explained, inevitably channels interfere with each other due to small frequency spacing (§2.3.5,pp.32).

Initially, cluster node J (CN J) of cluster "testbed1" downloaded media files from cluster head Z (CH Z), taking measurements of the received data download throughput (goodput) and download completion times. The results of the average goodput for each channel combination (T1,T2) are shown in Table 7-3, where T1 is the deployment channel of "testbed1"and T2 that of "testbed2". What is worth noticing is that the goodput performance of ad hoc deployment in consecutive channels is even worse than deployment on the same channel by approximately 13%. This can be explained by considering the 802.11 MAC layer functionality, where while on the same channel, all devices hear for Request To Sent (RTS) frames and back-off from using the channel and thus can avoid collisions and excessive MAC frames retransmissions. On the contrary, when nearby channels are used, frames from different channels are perceived as interference and increased channel noise, causing the MAC layer to retransmit lost frames and possibly reduce transmission rate to avoid excessive BER. As recorded by measurements, this effect is reduced the furthest apart the channels are, although is still noticeable even when "non-overlapping" channels are used (e.g. 6,1). This can be explained because of the proximity of most devices which results in the near-far effect. The problem is encountered due to 802.11 PHY/MAC operation that aims to achieve fairness in channel throughput and utilisation based on channel sensing measurements (CSMA/CA) [74],[75]. The mentioned observations regarding how channel spacing affects

performance measurements has affected the replacement of the mirrored linear weights' distribution with the empirical weights' distribution as previously shown in Figure 7-3.

Table 7-3. Initial Channel Assignment Measurements

testbed1, 2 (T1,T2) channel #	Goodput (*testbed1*)		Goodput decrease (%)	Downl.Time increase (%)
	KByte/sec	Mbps		
1,1	445.61	3.48	-20.38	+20.00
2,1	373.47	2.92	-33.27	+46.67
3,1	499.96	3.91	-10.07	+10.67
4,1	544.69	4.26	-2.68	0.00
6,1	559.69	4.38	---	---

Additional measurements of missed and sent frames, further confirm the detrimental effects of randomly assigning channels to deployed ad hoc networks. All measurements displayed in Figure 7-4 were taken from the node Z, the CH of "testbed1", using its second wireless interface in *rf-monitor* mode, i.e. capturing all packages transmitted on a specified channel. The purpose was to verify how the device perceives the wireless channel while transmitting using its first interface. Two sets of measurements are shown:

1. For same channel deployment of both clusters:(T1,T2)=(1,1). Both ad hoc deployments and node Z monitoring was done on channel 1 (shown as two leftmost graph bars for each monitored node)
2. For consecutive channel deployment of clusters:(T1,T2)=(2,1). The ad hoc deployment of *testbed1* and node Z monitoring was done on channel 2, while *testbed2* remained deployed on channel 1 (shown as two rightmost graph bars for each node)

Frame measurements provide a good indication of channel utilisation and the level of occurred collisions (missed frames). Therefore, these results highlight the detrimental effect of cross-channel interference for wireless networks deployed on nearby channels. Two points worth noticing are mentioned here:

1. Missed frames are in both cases increased, even exceeding sent frames. Focusing on node J, these measurements indicate the high levels of interference, causing a significant amount of frames to be corrupted. It is also noted, that in case (2,1) missed frames are increased by approx. 15% compared to case (1,1), confirming that sequential channel deployment is worse than same channel deployment in terms of MAC layer performance.
2. Node Z can hear a significantly increased number of frames sent from nodes (A,D) of a competing ad hoc network. Focusing on frames sent from nodes A and D, as measured

from node Z, it is noticed that for consecutive channel deployment, node Z captures and decodes 10 times more sent packets on its operating channel (channel 2) in spite of the fact nodes A and D operate on a different (channel 1). The small channel distance of 5MHz results in the increased effect of cross-channel interference.

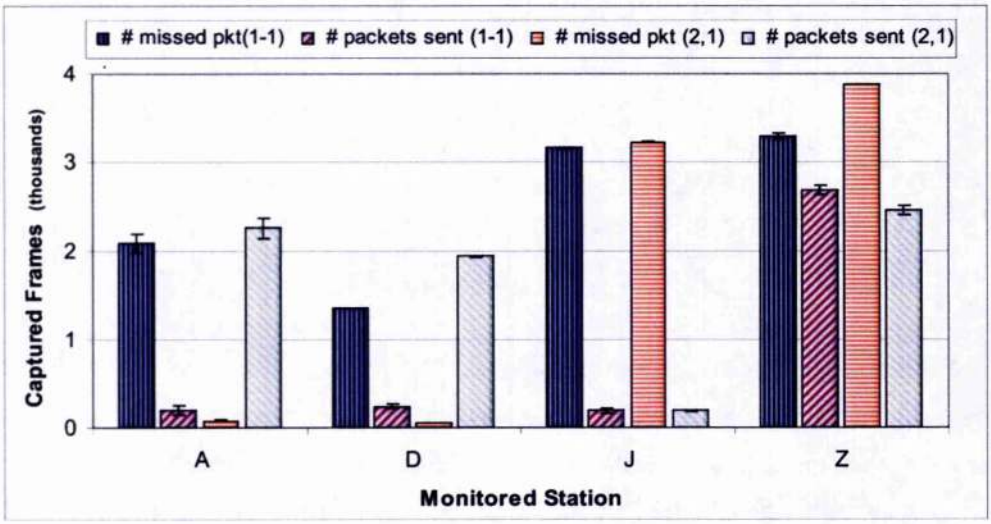

Figure 7-4. Frame measurements at Cluster Head (Node Z) for same channel deployment (1,1) and for consecutive channel deployment (2,1)

To alleviate aforementioned problems, PBM support is enabled for *testbed1* and the cluster head (node Z) ensures that policies (P1-4,Table 7-1,pp.160) are applied during the initial phase of ad hoc deployment. The aim is to select the most suitable channel in order to avoid cross-channel interference. After P1 had scanned channels, P2 detected the presence of testbed2 on channel 1 and the scan results indicated channels 2-10 as free (FC=true,PC=true). Channel 11,was found occupied by an infrastructure WLAN. Hence, the conditions of policy P2 were true, triggering action *optimizeChannel* with parameters the preferred channels (PC=1,6,11). Since channel 6 of the preferred (non-overlapping) channels list was free and nearby channels were not interfering, as expected the aforementioned algorithm had selected it. Therefore, the ad hoc network is initiated on the selected channel by node Z and the rest of the cluster nodes join using SSID *testbed1* on the same frequency.

As confirmed by the measurements shown in Table 7-3, the policy-based initial channel configuration results in the optimum configuration (T1,T2)=(6,1). These measurements show that cluster self-configuration of its initial ad hoc channel deployment, results in a 20.4% increase of average goodput when compared to using default channels (1,1) and up to 33.3% increase for random channel assignment (2,1). File download completion time was accordingly improved, avoiding a 46% increase for random channel assignment.

7.2.3 Self-Optimisation for Dynamic Channel Switch

The second implemented scenario investigates the dynamic adaptation of wireless ad hoc networks, aiming to anticipate interference and real-time throughput degradation. Based on the topology of Figure 7-1, the coexistence of two separate ad hoc networks on the same channel was initiated (*testbed1* and *testbed2* on channel 1). At first, no traffic transfers were performed between nodes. The scenario execution had two phases:

- Phase 1: ad hoc network *testbed1* initiates a file transfer between nodes, with cluster node J downloading a 46MB file from cluster head Z
- Phase 2: ad hoc network *testbed2* initiates another file transfer between nodes A and D

To evaluate the implemented solution, two experiment sets of the described scenario were executed, one set with the PBM solution enabled and enforcing policies (P5-8, Table 7-1,pp.160) and another set without any PBM functionality. Every effort was made to maintain the same execution conditions during all experiments, to allow comparison of taken measurements. A representative extract of measurements is presented in Figure 7-5 and explained below.

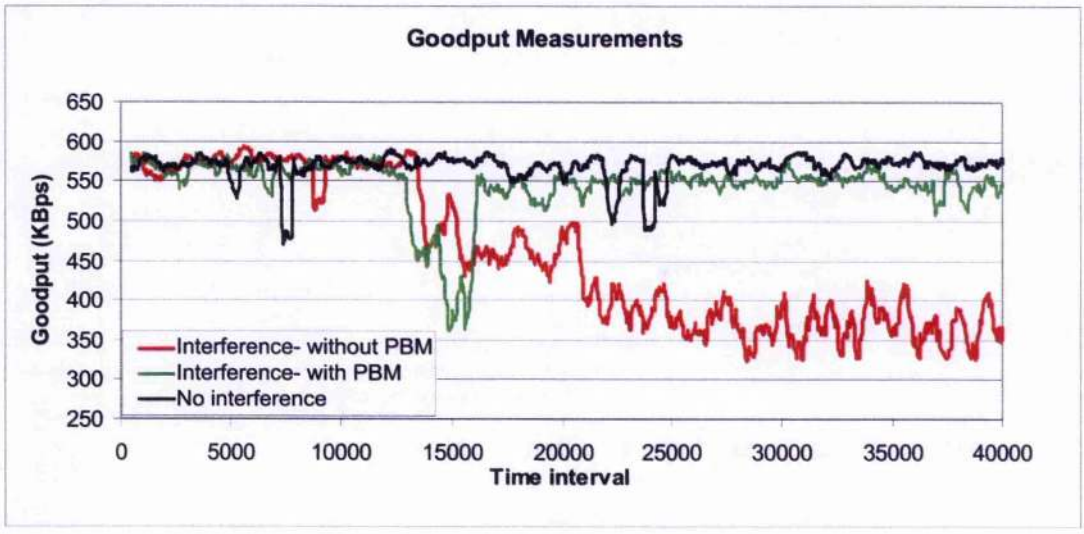

Figure 7-5. Policy-Based Channel Assignment Measurements of File Transfer Goodput

The measured results demonstrate a significant improvement in network performance when the proposed PBM solution is used (Figure 7-6). The ad hoc cluster *testbed1* is self-optimising by monitoring events and conditions, resulting in reconfiguration of the transmission channel to avoid interfering WLAN. When the competing ad hoc network (*testbed2*) initiated a file transfer (phase 2), this resulted in increased collisions and missed frames for both clusters, which was reflected in reduced Link Quality reported by the wireless interface at node Z. Policy P6 was triggered by *LinkQualityCheck* event and had evaluated the moving average of *LinkQuality* as less

than 50% (thr_a). As a result, it executed action *generateStartAdapt*, thus initiating the adaptation process for channel optimisation. In turn, policy P7 was triggered and monitored the specified application metric, in this case the moving average of goodput measurement for the file download between nodes Z and J (*app-specific-metric*). The measurements of this metric are shown as bold lines in Figure 7-6 (top), while thin lines show instantaneous goodput measurements (bottom). Comparing the two graphs of Figure 7-6, it is verified that the use of a moving average smoothes goodput fluctuations and prevents false triggering of adaptation policies. Once policy P7 detected the reduction of goodput below 3.67Mbps (thr_b), it acted by scanning the wireless channel, triggering policy P8 and passing scan results (event *AdaptChannel*). Policy P8 acted by executing *channel-switch* method using the weighted average algorithm ($algoritm_3$) with specified weights ($criteria_3$) as described earlier. The method indicated that a better channel was available and initiated dynamic switch of the ad hoc network *testbed1* to channel 6. A channel switch period took place, causing temporary disconnection of nodes from their cluster head Z. The measurements show that L2 disconnection and connectivity loss occur, however the effect on the ongoing file transfer between J and Z was temporary goodput reduction with a quick recovery to significantly higher goodput. In fact, when compared to the execution without PBM support, the described self-optimisation resulted in a 33.5% peak increase of goodput with an average increase of 20.3%. Also, average download time for a 46MB file dropped from 116sec to 50sec.

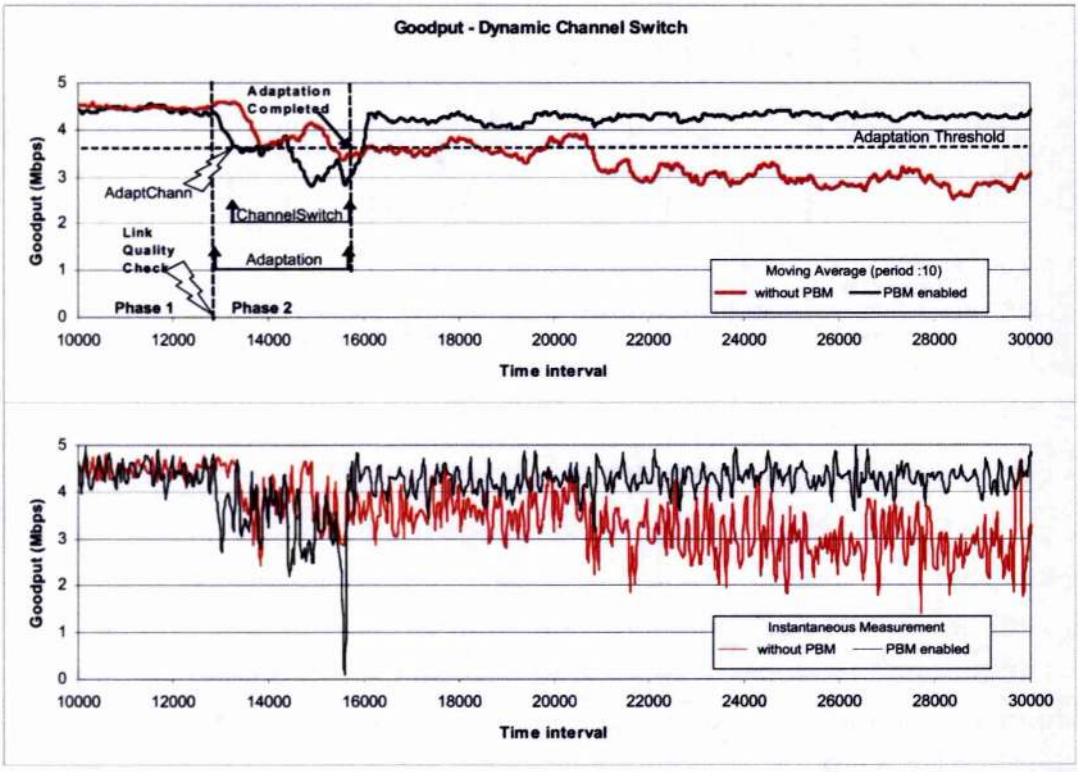

Figure 7-6. Testbed measurements of goodput using dynamic channel switch

7.2.4 Case Study Summary

Today the deployment of wireless ad hoc networks is becoming a popular and convenient solution for quick network setup and spontaneous or opportunistic networking. Unfortunately, user experiences have been disappointing, mostly because of difficulties in setup and poor performance. Therefore, solutions were proposed for two potential obstacles that need to be overcome in order to make the deployment of ad hoc networks easy, efficient and safe:

1. interference between newly created ad hoc networks and existing WLAN
2. regulatory conformance of ad hoc networks deployment.

Based on the introduced policy-based management framework, self-configuration and self-optimisation were integrated as a first step to implement a truly Self-Managing solution. By facilitating a predictable and controlled ad hoc network deployment, the performance of both wireless ad hoc networks and infrastructure-based WLAN can be significantly improved. One of the critical issues that need to be addressed is the channel assignment of wireless ad hoc networks. The proposed solution can be deployed on top of existing and future access networks using a technology-independent policy-based management layer. At the same time, inter-layer communication is used between 802.11 MAC sub-layer and Application layer, aiming to make the PBM system aware of wireless channel conditions.

The proposed solution was investigated for wireless networks based on IEEE 802.11, due to their support for ad hoc operation and increased market penetration. The experimental testbed was consisted of 10 nodes equipped with internal 802.11b wireless interfaces. A set of policies was designed, aiming to alleviate the two issues mentioned above. For the purpose of the case study, policies used channel measurements in order to evaluate the best possible channel for ad hoc initiation or dynamic switch. An algorithm based on the weighted average (WA) of a channel metric was introduced and explained. Having identified through measurements the detrimental performance of consecutive channel deployments, the performance degradation of goodput was normalised, to produce a new empirical weights distribution for the algorithm.

The policy-based initial channel allocation resulted in optimum configuration, as confirmed by measurements. The ad hoc cluster self-configured its initial channel deployment and this resulted in a 20.4% increase of average goodput, compared to using default channels and up to 33.3% increase to random channel assignment. Additional experiment sets investigated the dynamic adaptation of wireless ad hoc networks, aiming to anticipate real-time interference and throughput degradation. Using the PBM solution, ad hoc clusters were self-optimising by monitoring events and conditions, resulting in reconfiguration of the transmission channel to avoid interfering WLAN. Measurements showed a 33.5% peak increase of goodput with an average increase of 20.3% and reduction of average download time for a 46MB file from 116s to 50s.

7.3 Service Management for Wireless Networks

7.3.1 Policy-based Framework for Adaptive Service Management

In order to manage a complex set of services and offer the expected Quality of Service (QoS) and Experience (QoE) to users, a service provider (SP) has to take into account several parameters and constraints. But for a service to be successful, a certain degree of control must be given to the end-user. Preferences offer some control to users and allow for the customisation of available services. Users' preferences may express general device settings or access rights to integrated hardware (e.g. power profile, GPS receiver). These are referred to as *basic preferences,* so as to differentiate from *service-specific preferences*. The latter refer to user options aiming to customise a specific service. Both preferences and device capabilities affect the adaptation process of deployed services.

The architecture presented in Figure 7-7 is based on the aforementioned PBM framework, which is extended and customised by introducing the *Se*rvice *A*daptation *L*ogic (SEAL) and *U*ser *Pr*eferences *C*ontrol (UPRC) components. The novel features introduced, together with detailed policy design, facilitate a flexible and extensible service management framework. The Service Adaptation Logic (SEAL) component accepts users' requests and provides a customisable, adaptive service management framework by taking into account device capabilities and service-specific preferences. SEAL interacts with the UPRC on a user's device, aiming both at the enhancement of user's experience and the optimisation of offered services.

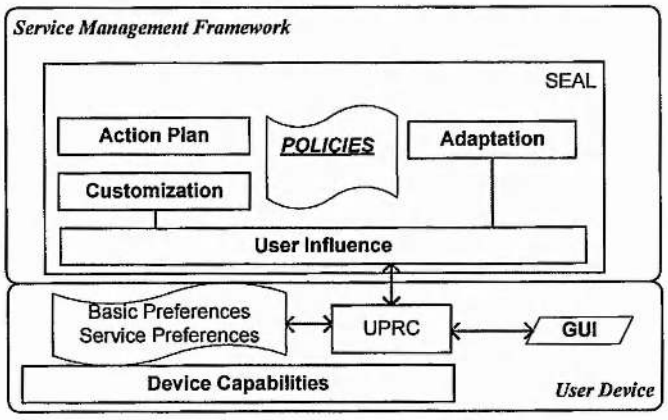

Figure 7-7. System Architecture for Adaptive Service Management

7.3.2 Service Adaptation Logic

The *Se*rvice *A*daptation *L*ogic (SEAL) component is a network-side entity responsible on one hand for adapting offered services according to specific user's preferences and on the other hand

for influencing these preferences in order to optimise service utilisation. These tasks are policy-driven, enabling a flexible and extensible service creation and execution environment. Service customisation and adaptation are directed by users' service requests. Each request contains necessary information for the operation of the service, such as device capabilities and service-specific preferences. Before a service is provided, SEAL performs a three-level customisation procedure, according to respective *Customisation policies*. The first level is based on the requesting device capabilities. In addition, two extra levels of customisation are introduced, which depend on the users' preferences, differentiating between basic and service-specific preferences. These parameters are examined by relevant policies and result in device and service-specific provisioning. The final stage of service provisioning is the enforcement of *Action Plan policies,* that take as input the results from the triple layer customisation procedure and execute the actual provisioning based on user preferences.

With the aim of service provisioning optimisation, SEAL may attempt to influence user's preferences. This task can be executed directly by the service provider (*proactive influence*) or can be triggered during the service customisation task (*reactive influence*). The latter refers to the notification of a user during service initiation with the purpose of improving the requested service. Based on customisation policies, the user is informed about available service improvements and prerequisites, i.e. which preferences should be changed to allow the SP to offer the improved service. While users' preferences need to be respected at all times, a service provider may need to proactively influence them for certain services to operate smoothly. For example, a file sharing service cannot operate, if all users choose not to share any files in their sharing preferences. In these cases, the SP needs to influence users (proactive influence) to change their preferences.

Service adaptation can be achieved by statistical analysis of the service-specific users' preferences and device capabilities. By analysing these data, SEAL may identify current trends in service requests and profile the capabilities of users' devices. Based on the extracted information, SEAL enforces *Adaptation policies* to dynamically change the provisioned service, aiming to satisfy more users' requests with less overhead.

On the client-side, the User Preferences Control (UPRC) entity communicates with SEAL, in order to visually notify the user and handle necessary device configuration changes. This lightweight entity assists in preferences management and based on user input replies to the influence notifications from SEAL. As shown later in Figure A-6 (pp.222), the new UPRC internal component is integrated within "*CN interface*" component, allowing it to interact with collocated *Node GUI* and *Preferences and Settings Translator*. Through the Graphical User Interface, the device owner can set privacy and preferences settings that classify managed objects to policy-free and policy-conforming (PFO/PCO, §6.3, pp.148). This interaction aims to influence the user to change these settings and effectively alter the access control restrictions on managed

objects. The incentive for the user can simply be the possibility to receive an enriched service, e.g. a user connected via Bluetooth is offered higher bitrate if he or she switches transmission to WiFi. In addition, the SP may operate a user reward/incentives scheme [92], e.g. offer free songs downloads for users that accept to share unused bandwidth for traffic forwarding. A detailed case study is presented below, demonstrating the functionality of SEAL. Evaluation results through simulation demonstrate the effectiveness of the proposed adaptation process.

7.3.3 Adaptive Service Management for Media Delivery

Media Service Scenario for Wireless Ad Hoc Networks

To demonstrate the introduced ideas, a detailed scenario is presented elaborating on policy definitions for an adaptive service management framework. The increasing popularity of music downloads and video-sharing activities among the Internet and mobile users have motivated the selection of a media service for experimentation. In this scenario, as depicted in Figure 7-8, users can have access to media services (audio, video etc.) while travelling on trains, where normally user connectivity is limited. This scenario is particularly attractive in the case of underground train networks or interstate/intercity train services through sparsely populated areas.

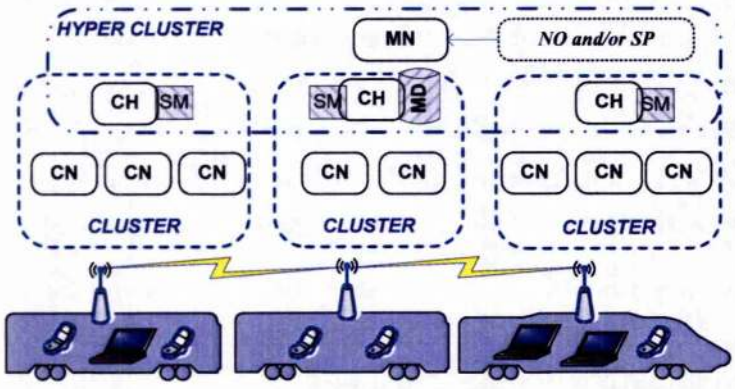

Figure 7-8. Case study scenario

A network operator (NO) deploys Cluster Heads (CH) onboard trains and offers the infrastructure to different service providers (SP). A multiple manager (MN) environment is possible, where policies orchestrate manager interaction (§3.4, pp.66 and §4.3,pp.87). A device acting as CH can be a wireless access point with processing and caching capabilities. Depending on physical dimensions and passenger density, each train carriage can be considered as a separate cluster managed by a Cluster Head (CH). Economic considerations affect the hardware specification of CH, where trade-offs between cost and user coverage need to be made. As discussed, CHs are interconnected (forming the *hypercluster*) and for this case study they share a common media database, physically located in the middle of the train. CHs also interact with the Manager Node (MN) controlled by the SP, to update policies and report critical events. Naturally MNs are not on

trains and for this scenario their communication with CH need not be uninterrupted. A fixed network support is implied, to allow CH contact their MN when needed. For example, CH onboard public transport may stop communication while in transit between stations and resume once they have arrived to dedicated synchronisation points, e.g. station platforms, central stations or maintenance locations. The important advantage of the proposed design is the fact that CH carry the required management logic within their PDP something that allows them to operate autonomously. Provided that policies do not change very often, CHs maintain an update view of the DPR and can provision their cluster devices with appropriate service settings.

The media service users are able to request media items available on the shared main database, as well as items shared by other users. All service requests are made to the CH and the latter maintains a list of all available media items either on the network-wide Media Database (MD) or the cluster-wide Shared Media (SM) table. Apart from identification keywords and source location, this list describes items in terms of media/content type, quality and operational requirements. To access the media service a user presents the CH with a request message that consists of three main attributes: the *device capabilities*, user's *media preferences* and user's *basic preferences*, reflecting the three-level customisation process. For this case study, the procedure can be viewed as a filtering process on matching media items where policies are used to guide the selection decisions of the CH. *Media* and *basic preferences* are optional and depend on the user's demand for personalised media delivery. In subsequent sections, these policies are described, along with their specification and usage. User requests can follow the format below:

```
mediaRequest( userID,
        devCabs[codecs, freeSpc],
        mediaPrefs[  type{audio, video, picture, any},
                     quality{high, medium, low, any},
                     content{news,sports,entertain, any},
                     source{MD, SM} ],
        basicPrefs[  connect{bluetooth, wifi},
                     battStrgy{norm, pwrSave}]
        );
```

Policy-based Service Customisation and Adaptation

Once a request is received by a CH, a triple level customisation process is initiated aiming to satisfy the user's request. It should be noted that the service is able to recommend changes to user preferences in order to provide alternative media when current settings fail to return any results. The process allows a fully customised and tailored media service delivery to the users, depending on their request. At the same time, the service adapts to the users' demands. This process is graphically represented in Figure 7-9 and is further explained in this subsection.

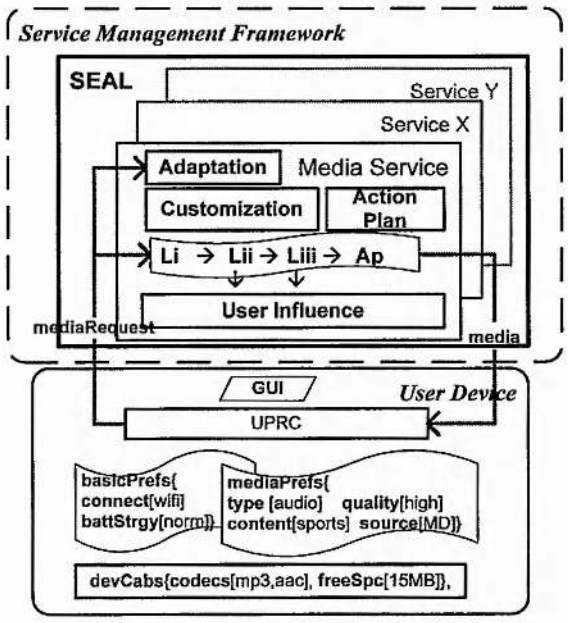

Figure 7-9. Media Service and Service Adaptation Logic (SEAL)

Capabilities and Preferences customisation

Initially, the CH searches the Media Database (MD) and Shared Media (SM) list for media items matching the criteria by keyword, content type and media type. Besides the usual media selection based on device capabilities (Li), two additional levels of customisation (Lii,Liii) are introduced, which make use of the *service* and *basic* user preferences respectively. The initial generated list (*mediaList*) contains all matching media along with their metadata and triggers the first level of customisation according to the requesting device capabilities. Three sequential policies (Table 7-4, $LiP_{1,2,3}$) apply here, aiming to determine media items on the generated *mediaList* with matching codecs and free memory space. These policies are triggered by *chkDevCabs* event, signalling the first filtering level.

Policies LiP_1 and LiP_2 check for media in the list that match the supported codecs of the user device and additionally satisfy free space requirements. Policy LiP_3 applies only to audio and video media (event: *chkStream(mediaList[name, _])* and is triggered only if there is a match for codecs but the available space does not satisfy the requirements of that media. The output of the first filtering level is an updated *mediaList* of items matching the requesting user's device capabilities. It should be noted that this list also includes items that the user cannot download because of limited free space and are marked as possible streaming media.

Policies guiding the customisation based on user's requested media service preferences are shown in Table 7-4, $LiiP_{1,2,3}$. These policies aim to determine media items that fit to the quality and source preferences. If a match is not found, then the user is informed to change the media preferences so as to result in alternative options.

Table 7-4. Triple level customisation policies

P#	*Event*	*if {Conditions} then {Actions}*
First customisation level policies – device capabilities		
LiP1	chkDevCabs()	if {supportCodecs(devCabs[codec,_], mediaList[name,codec,_])} then { selectCodec(mediaList[name, _])}
LiP2	chkDevCabs()	if {supportSize(devCabs[_, freeSpc], mediaList[name, size, _]) ^ selectedCodec(mediaList[name, codec, _])} then {selectItem(mediaList[name, _])}
LiP3	chkStream()	if { source(mediaList[name, _]) = = MD ^ mediaType(mediaList[name, _]) == (audio v video) } then { selectStream(mediaList[name, _]),selectItem(mediaList[name, _]) }
Second customisation level policies – service preferences		
LiiP1	chkServPrefs()	if {supportQuality(mediaPrefs[_, quality, _], mediaList[name, quality, _]) ^ supportSource(mediaPrefs[_, source], mediaList[name, source, _])} then { selectItem(mediaList[name, _]) }
LiiP2	noMatch(qual, src)	if { usrFlag(mediaPrefs, not_informed) } then { informUsr(options[]) , setUsrFlag(mediaPrefs, informed) }
LiiP3	usrReply(qual, src)	if { timeout = = FALSE } then { chkServPrefs(quality, source) }
Third customisation level policies – basic preferences		
LiiiP1	chkBasicPrefs()	if {supportConnect(mediaPrefs[_, connection, _] ^ mediaList[name,connection, _]) } then {selectItem(mediaList[name, _])}
LiiiP2	noMatch(connect)	if {usrFlag(connection, not_informed)} then {informUsr(options[]) , setUsrFlag(connection, informed)}
LiiiP3	usrReply(connect)	if timeout = = FALSE then chkBasicPrefs(connection)

Policy $LiiP_1$ will be invoked at the second customisation level, with triggering event *chkServPrefs*. Its action selects media items from the list, if matching quality and source are found. If this is not the case (event *noMatch((quality, source), mediaList[_])*), then policy $LiiP_2$ notifies the user about failing to match his/her media service preferences and suggests changes to these preferences aiming to provide alternatives. The next policy ($LiiP_3$) processes the user's reply (event *usrReply(mediaPrefs[quality, source])*) and checks the new preferences. Note that the

action of this policy acts as a trigger for the first ($LiiP_1$) indicating that the process starts again with alternative user preferences. The condition of the second policy ($LiiP_2$) checks if the user has already been informed once, so as to avoid looping when he/she does not change any preferences or the notification expires.

Similarly, a third customisation level aims to satisfy the basic user preferences. The policies of (Table 7-4, $LiiiP_{1,2,3}$) select a media item if matching connectivity between the user and the media source is found and notify the user about failing to match his/hers connectivity preferences. For example, when a WiFi user requests media found on another user who uses only Bluetooth connectivity, then the system suggests a change to the first user's connection preferences to allow him/her to receive the desired media. The initiating event for this customisation level is *chkBasicPrefs(connection)*.

Action plan

After the customisation process, the user will be presented with a list of media items to choose from. The user's reply will serve as the trigger for the action plan policies (event *userSelect(medialist[name,_], userID)*). Based on these policies (Table 7-5,$ApP_{1,2,3}$), the Cluster Head decides whether to stream the selected media to the user, provided the first customisation level had marked that media for streaming. Otherwise, depending on its source, the media is downloaded on the user's device from the CH's database or from another user (*sourceUserID)*.

For clarity, the above policies are simple; however the service provider has the ability to change the action plan by editing existing policies or introducing new ones, taking into account more parameters or operational conditions. For example, ApP1 could include conditions like link quality or utilisation between the user and the Cluster Head, in order to avoid significant packet losses that would degrade streaming media quality [4]. In addition, as technology evolves, the option of P2P streaming media between users can be easily integrated to the PBM system with the introduction of a few new policies, instead of fully upgrading the media service software.

Service Adaptation

An important task of SEAL is to adapt existing services according to statistical analysis of users' prevailing service-specific preferences and device capabilities. This adaptation improves both service performance as well as users' experience. For this case study and the simulation presented in the next section, SEAL monitors (1) requested *quality* based on media-specific preferences and (2) availability of *codecs* based on device capabilities. By calculating the Weighted Moving Averages (WMA) of certain request parameters, SEAL can identify the trends in media requests and device capabilities among the served users. Using the flexibility of the underlying policy-based system, the service provider can anticipate users' demands and accelerate the processing of their requests. The following policy example illustrates the benefits of the designed framework.

The adaptation process takes place at the Cluster Heads (CH) using the aggregated parameters of their cluster requests. A periodic event (*calculateWMA(qualityCnt[],codecCnt[])*) triggers the above adaptation policy. The Weighted Moving Average is a statistical formula used to analyse time series data in order to smooth out short-term fluctuations, thus highlighting longer-term trends. By counting the occurrences of *low (L)*, *medium (M)* and *high (H)* for the media *quality* preference, the highest WMA value (*popQualityWMA*) identifies the most popular quality (*popQuality)* requested. In the same way, the most popular *codec (popCodec)* can be identified, i.e. the one available on the majority of the devices during the examined period.

According to policy SaP1, if the average occurrences of the popular formats exceed the ones defined by thresholds (thr1,thr2) and the Cluster Head processing load (*chLoad*) is below 25%, then the CH begins the adaptation action, i.e. transcodes the most requested (*mostReq[]*) media files within its cluster using these two parameters (quality q, codec c). As a result, available media options can be significantly increased for the majority of users. In addition, conditions prevent CHs to start the resource-consuming transcoding process, if they are already busy serving users' request (higher *chLoad*).

Table 7-5. Action Plan and Service Adaptation Policies

P#	*Event*	*if {Conditions} then {Actions}*
Action Plan policies		
ApP1	userSelect(medialist[],userID)	if {streamSelected(mediaList[name,_])==TRUE} then {setupStream(mediaList[name,_]) , streamTo(userID)}
ApP2	userSelect(medialist[],userID)	if { (streamSelected(mediaList[name,_])==FALSE) ^ (source(mediaList[name, _]) = = MD) } then { setupFileTransfer(mediaList[name,_]) , downloadTo(userID) }
ApP3	userSelect(medialist[],userID)	if { (streamSelected(mediaList[name,_])==FALSE) ^ (source(mediaList[name, _]) = = SM) } then {setupFileTransfer(mediaList[name,_], sourceUserID), downloadTo(userID)}
Service Adaptation policy		
SaP1	calculateWMA()	if (popQualityWMA > thr1) ^ (popCodecWMA > thr2) ^ (chLoad < 25%) then transcodeItems(mostReq[],popQuality,popCodec)

Evaluation of Service Adaptation

The adaptation process was simulated with the enforcement of policy SaP1, measuring its effect over time on the described media service. A custom-made simulator was developed, integrating a random service request generator and the adaptation functionality of SEAL. The metrics used for evaluation were the number of requests per interval for a combination of user preferences *quality* and *codec*, and the estimated difference of media availability. Media availability is defined as the ratio of available media of specified preferences and/or device capabilities combination over the total number of available media. The relative media availability difference was measured, compared with the media availability of the same service without adaptation. Higher ratios reflect a higher probability of a user's request being satisfied and a wider range of media options for that combination. The availability improvement was depicted as a positive relative difference.

A request generator has been developed using Java (Java SE 1.4.2) and was programmed to send 100 requests per time interval for a total of 50 intervals. This generator produced random requests, except during specified intervals where request parameters were deliberately biased. The purpose of the bias was to simulate the increase in media requests for a specific combination of quality and codec (q,c). In real life, this would happen when the passengers of one carriage have a common behaviour that differs from the average user of the service. For example, a group of students using their mobile phones try to download low quality tracks while on the train, resulting in increased (L,aac) requests. Or commuters of a first class carriage try to access high-quality video news on their laptops during peak hours, resulting in increased (H,mp4) requests. These behaviours are simulated with a bias in the request generator during interval periods 11-20 and 31 to 40 respectively. For the purpose of the simulation, three codec formats were chosen, namely aac, mp3 for audio and mp4 for video, while three options for quality can be available (Low, Medium, High). Figure 7-10 shows part of the simulation results for the number of requests for the mentioned combinations ((L,aac),(H,mp4)), plus an additional random one (M,mp3) for comparison. The peaks on the graph are the result of the generator bias.

For every time interval, policy SaP1 is triggered and the popularity thresholds are checked. If both thresholds are exceeded, indicating a very popular quality and codec combination, then the action of transcoding is enforced. This results in an increased number of available media for that combination, thus resulting in increasing media availability during those periods (Figure 7-11). The use of a weighted moving average ensures that adaptation is not triggered for a sporadic increase in requests. This is also reflected in the delayed triggering of the adaptation process (after interval 14 and 34), ensuring that a trend in users' requests has been established. Effectively for the examined measurements snapshot, a 12.2% average increase in requests for low quality tracks (L,aac) results in a 3% increase in media availability for that combination (period 11 to 20).

Similarly a 15.2% average increase in requests for high quality video (H,mp4) results in a 2.9% increase in media availability (period 31 to 40). For the total period, the effect on media availability for a random combination of media requests (M,mp3) is minimal (-0.7%).

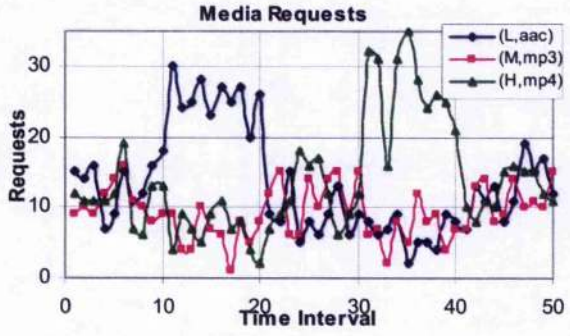

Figure 7-10. Media requests over time

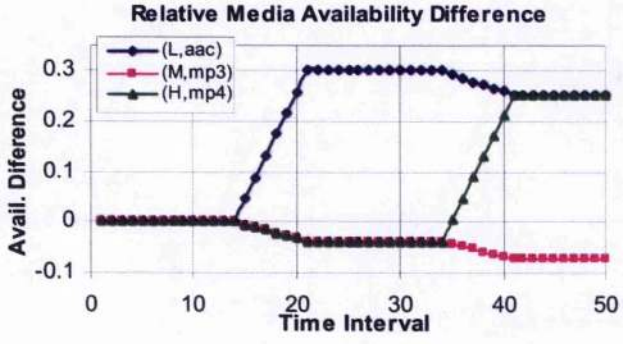

Figure 7-11. Adapting behaviour of media availability

Media Transfer Performance Evaluation

In order to demonstrate how policies can improve network performance, an example is presented to evaluate their effect to a wireless network. The described clustering model and media service are considered for management of media transfers among wireless network users. Based on the previous case study scenario, the adaptive service management framework has identified the candidate media that fit the user's request. Continuing with this evaluation example, the aim is to transfer those media files between two devices within a cluster. The responsible CH uses policies with cluster-wide scope to make decisions based on local events and conditions. In other words, the management system uses policies to examine local conditions and decide the best way to transfer a file, i.e. whether to download the file locally or stream it from the source. Beyond the described case study, the applicability of introduced policies extends to various wireless ad hoc networks where clusters can be formed, e.g. within a house or among users visiting an attraction.

The service provider defines a set of policies that are enforced whenever a media file transfer is requested within a cluster (Table 7-6). The conditions of these policies use two introduced metrics that express the current conditions in the cluster:

- *network utilisation (NU):* expresses the average bandwidth utilisation between the source and destination based on the maximum real bandwidth of each device

$$NU=(1/2)*[avgBW_s/maxBW_s + avgBW_d/maxBW_d]$$

- *media capacity (MC):* provides a metric of how the minimum free bandwidth between source and destination devices compares to the bitrate of the requested media. A bigger MC shows better bandwidth availability for media streaming

$$MC=[min(maxBW_s - avgBW_s,\ maxBW_d - avgBW_d)]/mbr$$

For the equations above, *avgBW* is the average value of a device's utilised bandwidth over time, *maxBW* is the maximum real bandwidth of a device and *mbr* is the bitrate of requested media. Subscript s and d refer to source and destination devices respectively. Using these metrics to hide the complexities of policy conditions, three media transfer policies are specified:

Table 7-6. Media Transfer Policies

P#	Policy
P1	if (NU<0.3) then download(file)
P2	if(NU>0.3)^(MC>1) then stream(file)
P3	if(NU>0.3)^(MC<1) then stream_reduced(file)

The action of P1 is to download the file, if the conditions between source and destination are good (NU<0.3). When NU>0.3, i.e. the average availability of bandwidth is reduced, policies P2 and P3 decide on the action by evaluating MC. If media capacity is sufficient (MC>1) the file is streamed to the user (P2). However, when MC<1 streaming the file at the original bitrate would cause bad media quality as well as further network congestion. Therefore, the action of P3 is to reduce the bitrate of the file before streaming. Bitrate reduction may be achieved by providing an alternative medium format with lower bitrate, so as to avoid resource-consuming transcoding.

The defined metrics offer a comparable way to describe the local conditions between source and destination devices. The cluster head evaluates the policy conditions by calculating NU and MC in order to enforce the appropriate action. Although these metrics take into consideration the conditions only at source and destination, this should be sufficient when proper network organisation is employed, e.g. using the proposed organisational model and algorithmic cluster creation (§3.6.pp.71). This can ensure that formed clusters are either relatively small or dense so that created multihop paths are short. This is necessary in order to avoid the severe bandwidth reduction over multiple wireless hops and reduce management overheads within clusters. An

additional measure to increase the reliability of these local metrics is to change local calculation of *avgBW* and *maxBW*, taking in mind bandwidth measurements recorded during multihop connections only.

In order to evaluate the effect of the above policies to the network performance, the discrete event network simulator *ns-2* (www.isi.edu/nsnam/ns) was used. The purpose of the simulations was to measure the performance of a clustered wireless ad hoc network based on IEEE 802.11 with or without the presence of the aforementioned policies. Transfers were setup over a static multihop MANET. In order to create a controlled simulation environment which could be also deployed on the real experimental testbed of Figure 7-1, cluster size was restricted to 5 nodes. An FTP traffic generator emulated file download and a UDP generator emulated media streaming, while additional TCP/UDP traffic flows were created to affect the avgBW values. The effective bandwidth of 802.11 networks is much less than their maximum data rate, as confirmed by previous testbed experiments (Table 7-3,pp.167). Therefore maxBW is set to 1Mbps for calculations. Simulation parameters are listed in Table 7-7. The simulation scenario included the transfer of different file types (Table 7-8) between two users under various network conditions. For the case of streaming media, these file attributes were used for the bitrate and duration of the CBR (constant bitrate) traffic generators. Several tests were performed for each file type and performance characteristics were measured for downloading or streaming the same media file. The simulated media files had the same duration, so as to illustrate the option of streaming different versions of the same file. For each test, the values of NU and MC were calculated, in order to be used in policy conditions and decide which action to enforce.

Table 7-7. Simulation parameters

Time	Area	MAC	Routing	File Download		Media Streaming		Background Traffic	
600s	1000m x400m	IEEE 802.11	AODV	TCP Agent	FTP App.	UDP Agent	CBR App.	UDP Agent TCP Agent	CBR App. FTP App

Table 7-8. Media Table

	Size(Kb)	Bitrate(Kbps)	Dur.(s)	Popular Formats
M1	2880	96	240	MP3 podcasts, 3GPP video
M2	24000	800	240	MPEG4 video

Figure 7-12 shows the downloading throughput from source to destination with respect to the network utilisation (NU). It was observed that the enforcement of P1 ensures that when downloading (NU<0.3) the throughput remains sufficient. These measurements have identified the bandwidth saturation for higher value of NU and showed how P1 can prevent this from happening. In addition, the download time remained reasonable as demonstrated in Figure 7-13, where the download time ratio is low for NU<0.3. This ratio is the download time of each test

over the minimum time for NU=0. A ratio=2 means the user has to wait twice as much as if the same file was downloaded for NU=0. For NU>0.3 the PBM system decides to stream media, in order to avoid excessive download times and user dissatisfaction. Based on media capacity value (MC), policy P2 or P3 is enforced.

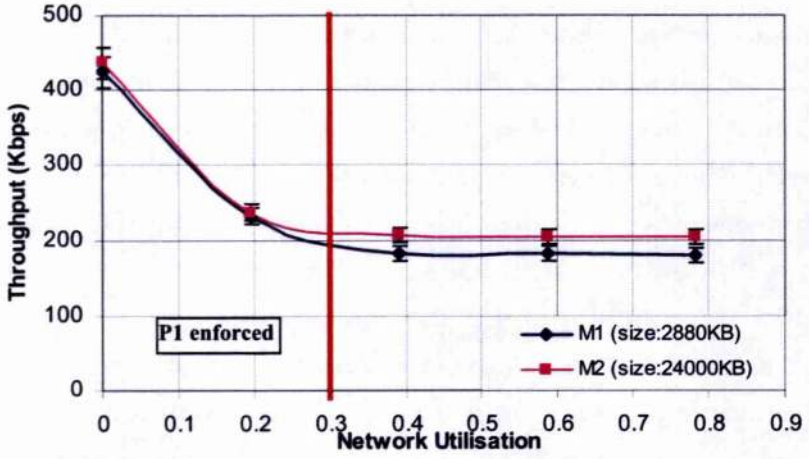

Figure 7-12. Throughput for downloading between source and destination

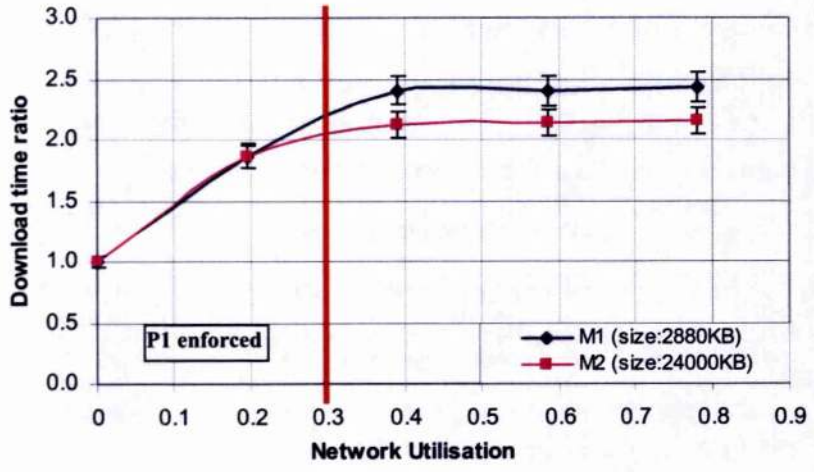

Figure 7-13. Download time ratio

Streaming tests were performed for the same network conditions as in the previous simulations and the same media were used. For streaming media, a representative metric of the quality is the end to end delay of the received packages. As expected, the smaller the MC the bigger the delays observed. The long delays while streaming M2 (800Kbps) can be avoided with the enforcement of P3, since in those cases MC<1. As shown in Figure 7-14, by streaming the alternative version M1 (96Kbps), delays are significantly reduced and MC remains above 1. In addition, the throughput ratio was calculated as the transmitted throughput over the actual media bitrate.

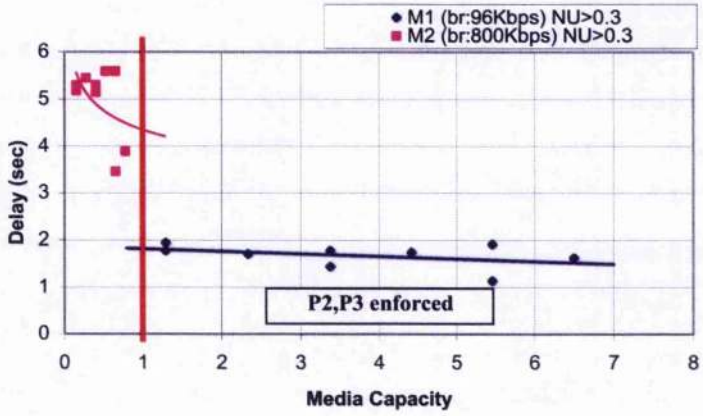

Figure 7-14. Received packet delays for streaming media

The measurements presented in Figure 7-15 indicate that high bitrate media (M2) cannot be transmitted under the current conditions and the degraded ratio translates to bad media quality. Streaming low bitrate media (M1) is possible and the ratio is near 1, demonstrating excellent media quality. Again, the value of MC reflects the local conditions and the enforcement of policies P2 and P3 prevents the initiation of a high bitrate transmission when the conditions do not allow for satisfactory media transfer rates.

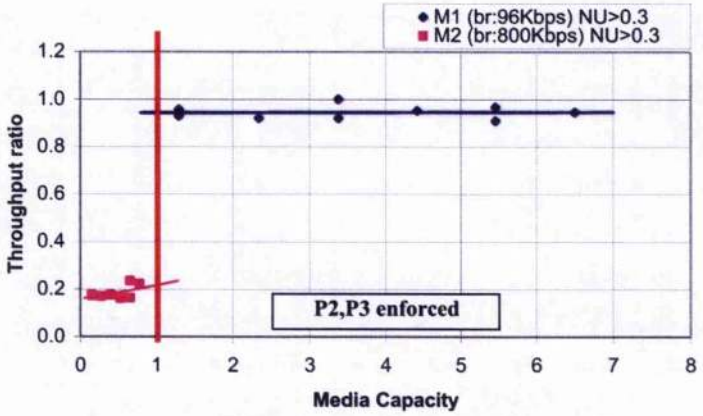

Figure 7-15. Throughput ratio for streaming media

The main challenge in this case study is how to define the best possible thresholds for NU and MC. Due to the continually changing conditions of wireless ad hoc networks and real life testbed experiences, it is practically infeasible to calculate their optimum value in advance. A possible solution would be to select initial values based on empirical measurements and by collecting feedback after each transfer, correct these thresholds. In the case study, a CH may selectively query some of the setup transfers to collect information (achieved throughput, throughput ratio, calculated NU, calculated MC) in order to evaluate the effectiveness of current thresholds.

Based on the presented results, tangible benefits to the network performance can be obtained using the proposed PBM solution. A significant improvement can be achieved since policies control the creation of media traffic flows and prevent further congestion. From the users' point of view, the experience in sharing media is improved. Although user's experience is subjective, measurements of packet delays and download times offer an objective metric to evaluate the quality of media delivery. These metrics show reduced packet delays with the deployment of appropriate policies and improved quality of delivered media.

7.3.4 Case Study Summary

An adaptive policy-based service management framework for wireless networks was presented in this section. The framework accommodates a level of control from end-users through generic and service-specific preferences. While these preferences can guide the provider towards a fully customised service, they can also be influenced to achieve optimised service utilisation. Another important feature of the framework is the support for service adaptation. This functionality was based on statistical and contextual information and as demonstrated through simulation it can potentially enhance service performance and user experience. The overall concept of adaptive and customised service provisioning was driven by policies, which facilitated a flexible and extensible service creation, enhancement and deployment environment.

The various components and functionality of the framework were demonstrated through an extended media service scenario and simulation of the adaptation procedure. Service management was supported with the specification and description of policies influencing the different levels of processing required, from service creation to service delivery. The examined scenario and simulation results validate the applicability and potential of the proposed approach, despite the relative simplicity of the introduced policies. Additional service provisioning policies were used to set up media transfers and based on local metrics decide on most appropriate transfer method. Further simulation results confirmed the performance improvement from the automated policy-based decision-making.

7.4 Conclusions

Concluding this Chapter, the benefits of using policies in wireless ad hoc networks were verified. The contradiction of ad hoc network creation and pre-provisioned policies was alleviated by the distribution of policies among capable nodes using the Distributed Policy Repository. The voluntarily enforcement of policies can provide autonomous wireless devices with the logic to guide their *self-management*. As it was demonstrated through testbed experimentation, by adopting a pragmatic view towards the management of wireless networks and a policy-based

design, a system with self-management capabilities can be realised. The wireless ad hoc system demonstrated self-configuration and self-optimisation capabilities, significantly improving its performance by dynamically switching channel to avoid interfering WLAN.

In addition, the applicability of introduced self-management concepts has been broadened from wireless ad hoc networks to cover the area of service management for wireless networks. Various concepts have found applicability in the new area and provided a novel adaptive framework for service creation, customisation and delivery. Significant scope for further research and integration of self-managing capabilities for next generation wireless networks was also identified.

Chapter 8

Summary and Conclusion

8.1 Summary

Wireless ad hoc networks pose major research challenges because of their diverse nature and their ubiquity. *Motivated by the deficiencies of current management frameworks in a rapidly evolving wireless landscape, and the increasing users' demand for unrestricted spontaneous communication; the objective of this thesis was to propose a novel management framework specialised for wireless ad hoc networks.* The new framework attempts to leverage the potential of wireless ad hoc networks as an emerging communication paradigm. For the purpose of this thesis, a realistic research approach was adopted towards ad hoc networking, disengaging from the limitations of the MANET paradigm. *Wireless ad hoc networks consist of a majority of end-user devices, capable of multihop communication, and optionally supported by limited infrastructure.* The presented framework aimed to facilitate their efficient and scalable management, combining design and theory with testbed implementation and simulation studies.

The *policy-based management (PBM) paradigm* provided the means to integrate self-management capabilities, with *policies* capturing the high-level management objectives to be autonomously enforced to devices. A layered policy hierarchy was combined with a hybrid organisational model to create three adaptation layers. In parallel, context was extracted from network nodes and was used as feedback to the policy-based components in a closed loop. As a result, policy-based management provided *controlled programmability* in the highly dynamic environment of wireless ad hoc networks, helping to automate management operations.

The proposed framework attempted to facilitate distributed deployment of the policy-based functionality over wireless ad hoc networks. The availability of policies was increased with the

design and implementation of a Distributed Policy Repository (DPR). The DPR enabled the distribution of policy provisioning and enforcement functionality, targeting lightweight heterogeneous devices. The selective enforcement of policies was also addressed, aiming to offer control to users and protection of their privacy. Finally, two case studies were presented to validate the proposed framework. First, the deployment of wireless ad hoc networks was investigated, by facilitating their self-configuration and self-optimisation with the assistance of policies. A second case study extended the policy-based framework for adaptive service management, based on user preferences and statistical processing of service requests.

8.2 Contribution Overview and Conclusions

The contribution of this thesis focused on the design and implementation of novel concepts towards a framework for the management of wireless ad hoc networks. The composition of those distinct concepts adds to the value of an integrated framework that provides a controlled environment for the deployment of wireless ad hoc network and ensures their scalable and efficient performance. The overall thesis contribution can be identified in three areas:

1. *Design of a policy hierarchy and a network organisation model for self-management*

   The combination of a role-based hybrid organisational model with a context-aware policy hierarchy has provided a controlled degree of distribution regarding the PBM tasks and responsibilities. Under certain deployment conditions, the algorithmic creation of a loosely tiered clustered network increased scalability by reducing policy retrieval traffic. The overheads from policy replication were compared to equivalent centralised deployments without replication, showing adaptive behaviour according to the network population. Adaptation was also facilitated with the introduction of the policy's enforcement scope and context interaction. The integration of context-aware counterparts to the PBM elements provided contextual feedback to policies at three different organisational levels. This has enabled the creation of a closed control loop at each level, forming the basis for localised and network-wide self-management.

2. *Deployment of distributed PBM functionality for wireless ad hoc networks*

   The management of wireless ad hoc networks was possible with the distribution of PBM functionality and elements, thus decentralising the traditional design of PBM systems. Based on the developed technology-independent policy specification, policies were oriented to resource-constrained wireless devices and aimed to maintain interoperability with full-fledged PBM systems with adequate power. Decentralisation was based on the design and implementation of a Distributed Policy Repository (DPR) which facilitated a variable degree

of policy distribution and replication using LDAP directories. The overlay of replicated DPR directories had assisted in the coordination of distributed nodes, responsible for collaborative management. Coordination was possible by facilitating the distribution and synchronisation of dispersed wireless policy decision points (PDP) and pushing them closer to the enforcement points they control. In addition, DPR offered a logically uniform view of management objectives through policies, helped avoid a single point of repository failure, distributed traffic load and provided alternative access options for PDP. The DPR also supported the ability to deploy and maintain special purpose partial replicas, offering a partial view of network policies that can relate to a specific service or location. The feature of partial policy replication was designed to anticipate the need for localised control or bottlenecks, aiming to increase scalability and availability.

The implementation and testbed deployment on lightweight wireless nodes confirmed the feasibility of the DPR design. The evaluation results of the proposed distributed policy replication methods were compared to those of centralised methods without replication, demonstrating that with the cost of increased traffic overheads, policy retrieval time can be significantly reduced. It was also shown that, improved DPR organisation using DPR replica placement algorithms can potentially reduce traffic overheads further, as in the case of network organisation.

Finally, a lightweight technology-independent policy provisioning protocol was implemented to transfer policy decisions for enforcement on distributed wireless nodes. Selective enforcement was integrated to satisfy the privacy requirements of users who participate in the management framework with their personal devices. The importance of this functionality lies in the differentiation of the policy enforcement strategy, from the traditional uniform and mandatory enforcement to the proposed user-oriented and selective enforcement.

3. Validation of PBM functionality for self-management on a real network

The investigated case studies have contributed towards the validation of the designed policy-based and self-management concepts. The first case study addressed the issue of actual deployment of a real wireless ad hoc network, attempting to overcome the lack of central coordination and the occurrence of interference. With the use of policies, the implementation of a self-configuring and self-optimising ad hoc network was possible, controlling the dynamic assignment of its wireless channel. The benefits from self-management capabilities were measured and quantified, improving the performance of wireless ad hoc networks and also facilitating their easier deployment. Finally, the proposed framework was extended for service management, implementing adaptable service provisioning and offering service customisation to end-users. The value of this case study is attributed on one hand to the

validation of the framework's flexibility and on the other hand to the integration of policy-based service adaptation functionality. Adaptation was achieved through statistical processing of users' service requests. Both case studies assisted in validating the proposed concepts, while testbed deployments made a first step towards the implementation of self-managed networks.

The three aspects of this thesis contribution have been combined under a unified policy-based framework for the self-management of wireless ad hoc networks. Throughout the thesis, partial contributions were identified based on the different operations of a PBM system and its respective functional components. As a final conclusion, Figure 8-1 indicates the applicability of each partial contribution to the functional elements of IETF's reference model. This figure illustrates a high-level view of the revised PBM framework presented in this thesis. In general, the proposed framework is highly suitable and customisable for networks with an accentuated ad hoc element in terms of nodes participation and communications initiation. Naturally, a number of open issues remain to be addressed and some of these have been identified in Figure 8-1 and are discussed in the following subsection.

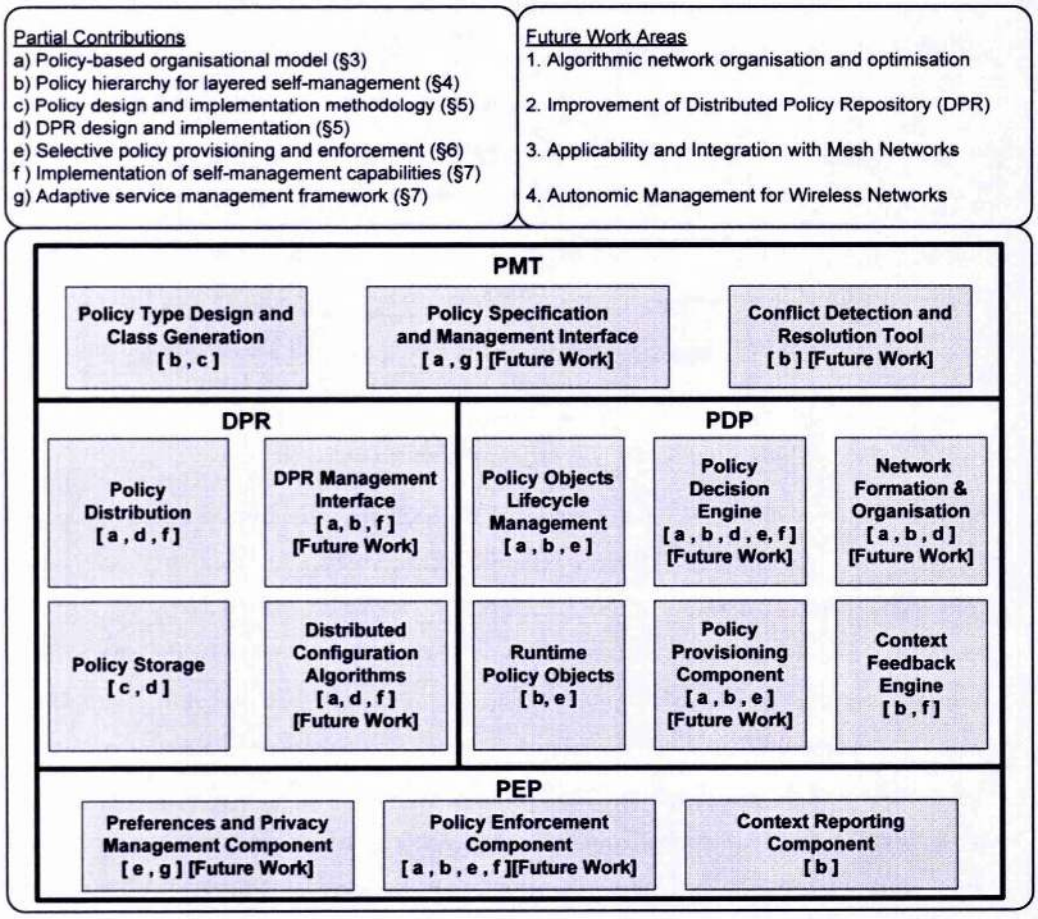

Figure 8-1. Adapted PBM framework with contributions and open issues

8.3 Future Work and Open Issues

Evidently, the work in the scope of this thesis has focused on addressing research goals as accurately and rigorously possible. That said, the completeness of this work can not be claimed and manifold possibilities for enhancement and future work remain open:

Algorithmic network organisation and optimisation

An important issue for improvement and optimisation is the algorithmic network organisation. The distributed creation of clusters by selecting the most capable nodes to form the hypercluster was based on an adapted version of Wu's algorithm. Some of its deficiencies were identified earlier for scenarios of very dense or very sparse wireless networks. The availability of a series of new distributed algorithms can be investigated, aiming to offer better performance in wider range of scenarios. Cross-layer design is also promising, especially for multihop networks using a proactive routing protocol. Piggybacking OLSR has been suggested and the effectiveness of Multi-Point Relay (MPR) selection could be exploited.

In the prospect of large scale deployment, probabilistic management can also be considered to reduce the number of managed nodes and guarantee their effective management. Another aspect to consider is the adoption of mobile peer-to-peer technologies for hypercluster self-organisation and the liberal network organisation without clusters. The parallel increase of processing capabilities of wireless devices and the adoption of mobile broadband access can relax the initially strict overhead requirements to facilitate more demanding solutions.

Algorithmic solutions were also suggested to solve the *DPR instance placement problem*. Optimisation of DPR problem solutions employ the challenging issues of cache/gateway placement is wireless multihop networks. Solutions based on node domination provide fast decisions for replica placement and can be combined with the network clustering process. While node domination solutions avoid operation duplication and expedite DPR node selection, unavoidably they link two separate functions with potentially different objectives. Their relation and interdependence need further investigation to confirm feasibility. The dual execution of Wu's algorithm with context aware heuristics was suggested to create a new set of capable and well connected nodes to host DPR replicas. Analytical modelling and simulation of this proposal need to be investigated as part of future work.

Problems and solutions from Location Analysis were mapped to the DPR replica placement problem and their potential was identified. Specific problems, like the facility location problem or the rent-or-buy problem, follow similar requirements to the DPR optimisation, like cost minimisation, therefore their further investigation and adaptation is part of future work.

Improvement of Distributed Policy Repository (DPR)

The deployment flexibility of DPR has been a significant contribution and opens the scope for further investigation and innovation. Beyond the open aspects of algorithmic optimisation, important design and implementation issues can be addressed in future work. The definition of a DPR Management Interface and its full integration with PBM is an important step towards flexible and reusable specification of DPR management policies.

An important feature of the designed DPR overlay is the ability to deploy and maintain special purpose partial replicas of the repository. Accordingly, special PDPs attached to partial DPR replicas are responsible only for the enforcement of a policy subset (related to a service) and can be dynamically deployed to provision time-based events or local conditions. To provision additional policies (for services), redirection via LDAP referrals to other partial DPR can be employed. Regarding Multiple Manager Replication (MMR), this feature of DPR implementation needs to be tested in large scale networks. The deployment of these features in dynamic real life scenarios is another topic for further investigation.

Applicability and Integration with Mesh Networks and the Internet

New paradigms of multihop wireless communication are under development with mesh networks being the most mature. Mesh networks can increase coverage in remote sparsely populated areas based on multihop routing and limited infrastructure support. The shift from closed proprietary equipment to open standards is expected to boost penetration of mesh networking and expand them from niche markets to mass market. Mesh network formation closely resembles the definition of wireless ad hoc networks in the context of this thesis. In fact, some of the examined scenarios, e.g. "urban spaces" or "on-train wireless services", can be directly mapped to mesh networking. These issues reveal an immense applicability potential of the proposed policy-based framework and its enhancement for mesh networks in a promising directions for future work.

At the same time, a continuous evolution of the Internet is witnessed with myriads of wireless devices connecting with a variety of access methods. With the advent of "Web2.0" and the proposals for a "Semantic Web", there is a vibrant open discussion about the "Future Internet". The fact is that the proliferation of user-generated content, online collaboration wikis and social networking websites have been thriving and the first steps towards their mobile/wireless deployment are being made. The adoption and customisation of current and future Internet trends for wireless networks is a major challenge. The spontaneous nature of user-generated content is naturally bound to mobile/wireless users and inherently has an ad hoc element. This could be the next milestone for wireless ad hoc networks, since they can offer the initiative to users and provide them with on-demand connectivity.

Autonomic Management of Future Wireless Networks

The evolution and composition of heterogeneous technologies is reforming the wireless landscape, evermore increasing complexity. To anticipate complexity and leverage the ad hoc networking paradigm, a policy-based framework with self-management capabilities has been proposed. The further investigation of adaptive service provisioning is another future direction, aiming to elaborate on appropriate policies to facilitate service management. A critical topic of future work in self-management is the stability of the integrated closed control loop and the definition of a valid operating region to avoid instability and oscillating behaviour. At the same time, the partial realisation of self-managing capabilities and their testbed deployment have verified the potential of self-management. Through future work in self-protection and self-healing, the concept of a system integrating all four capabilities is a long-term research plan. An extended self-management framework could become an implementation of the envisioned Autonomic Manager [133], as shown in Figure 8-2.

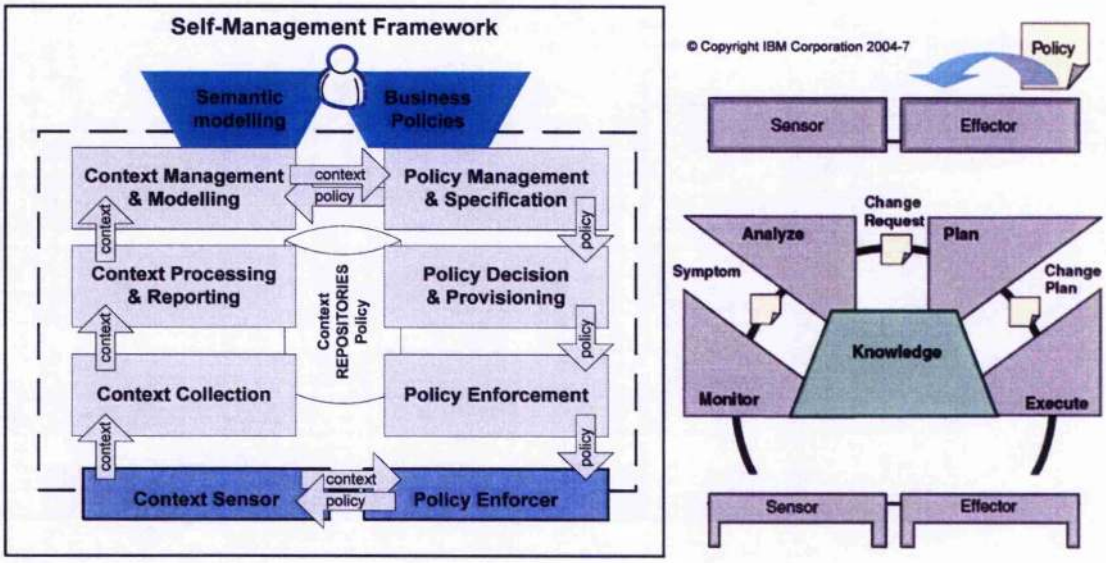

Figure 8-2. Self-Management framework and the Autonomic vision

"If I have seen further it is by standing on the shoulders of giants"

- Sir Isaac Newton

Bibliography

[1] A.M. Hadjiantonis, M. Charalambides, G. Pavlou, "An Adaptive Service Management Framework for Wireless Networks", *IEEE Vehicular Technology Magazine, Special Issue on Policy-based Management for Wireless Multimedia Services*, Vol.2, Iss.3,pp.6-13, Sep.2007.

[2] A. Malatras, A.M. Hadjiantonis, G. Pavlou, "Exploiting Context-awareness for the Autonomic Management of Mobile Ad Hoc Networks", Springer *Journal of Network and System Management* (JNSM), *Special Issue on Autonomic Pervasive and Context-aware Systems*, Vol. 15, No.1,pp.29-55, Mar.2007

[3] A.M. Hadjiantonis, G. Pavlou, "Policy-based Self-Management of Hybrid Ad hoc Networks for Dynamic Channel Configuration", *Proc. 11th IEEE/IFIP Network Operations and Management Symposium* (NOMS) , Salvador, Brazil, Apr.2008

[4] A.M. Hadjiantonis, M. Charalambides, G. Pavlou , "A Policy-based Approach for Managing Ubiquitous Networks in Urban Spaces", *Proc. IEEE International Conference on Communications* (ICC), Glasgow, UK, pp.2089-2096, Jun.2007.

[5] A.M. Hadjiantonis, A. Malatras, G. Pavlou, "A Context-aware Policy-based Framework for the Management of MANETs", Proc. *7th IEEE Int. Workshop on Policies for Distributed Systems and Networks* (POLICY), Ontario, Canada, pp.23-32, Jun.2006.

[6] F. Liu, A.M. Hadjiantonis, H.M. Tran, M. Amin, "An Architecture for Supporting Network Fault Recovery Management", *Proc. 2nd Int. Conf. on Autonomous Infrastructure, Management and Security* (AIMS), Bremen, Germany, Jul.2008, to be published.

[7] A.M. Hadjiantonis, G. Pavlou, "Self-management of wireless ad hoc networks using a policy-based paradigm and context-awareness", in *Context-Aware Computing and Self-Managing Systems*, W. Dargie, Ed, CRC Studies in Informatics Series, USA, to be published.

[8] A.M. Hadjiantonis, P. Flegkas and G. Pavlou, "A management model for ad hoc networks", *Proc. London Communications Symposium* (LCS), London, Sep.2005.

[9] R. Chadha, L.Kent. *Policy-Driven Mobile Ad hoc Network Management.* Wiley, 2008.

[10] S. Basagni, M.Conti, S.Giordano, I.Stojmenovic (Eds). *Mobile Ad Hoc Networking.* IEEE Press, 2004.

[11] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks.* Wiley, 2005.

[12] J. Sarangapani. *Wireless Ad Hoc And Sensor Networks.* CRC Press, 2007.

[13] I. Stojmenovic (Ed). *Handbook of Wireless Networks and Mobile Computing.* Wiley 2002.

[14] S. K. Sarkar, T. G. Basavaraju, C. Puttamadappa. *Ad Hoc Mobile Wireless Networks.* Auerbach Publications, 2008.

[15] A.S. Tanenbaum. *Computer Networks.* 4th Ed., Prentice Hall, 2003.

[16] A. Clemm, *Network Management Fundamentals.* Cisco Press, 2007.

[17] D. Verma. *Policy-Based Networking, Architecture and Algorithms.* Pearson, 2000.

[18] J. Strassner, *Policy-Based Network Management: Solutions for the Next Generation.* Morgan Kaufmann, 2003.

[19] M. Butcher, *Mastering OpenLDAP: Configuring, Securing and Integrating Directory Services.* PCKT Publishing, UK, 2007.

[20] T.A.Howes, M.C.Smith, S.G.Gordon, *Understanding and deploying LDAP directory services*, 2nd ed., Addison-Wesley Professional, 2003.

[21] M. T. Ozsu, P.Valduriez, *Principles of Distributed Database Systems* (2nd Ed.). Prentice Hall Inc.,1999.

[22] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, Ed. 1979.

[23] G. Pavlou, "On the Evolution of Management Approaches, Frameworks and Protocols: A Historical Perspective", *Journal of Network and System Management*, Vol.15,Iss. 4, pp.425-445, Dec.2007.

[24] J. Schoenwaelder, A. Pras, J-P. Martin-Flatin, "On the future of Internet management technologies", *IEEE Commun. Mag.*, Vol.41, Iss.10, pp.90-97, Oct 2003.

[25] J-P. Martin-Flatin, S. Znaty, J-P. Hubaux, "A Survey of Distributed Enterprise Network and Systems Management Paradigms", *Journal of Network and Systems Management*, Vol.7,Iss.1, pp. 9–26, 1999.

[26] G. Pavlou, P. Flegkas, S. Gouveris, A. Liotta, "On Management Technologies and the Potential of Web Services", *IEEE Commun Mag.*, Vol. 42, No. 7, pp. 58-66, 2004.

[27] A.Pras, T. Drevers, R. van de Meent, D. Quartel, "Comparing the Performance of SNMP and Web Services-Based Management", *IEEE Trans. Network And Service Management*, Fall.2004.

[28] Y. Yemini, G. Goldszmidt, S. Yemini, "Network management by delegation", *Proc. 2nd Int. Symp. on Integrated Network Management (ISINM)*, pp.95-107, Elsevier, 1991.

[29] A.Fuggetta, G.P. Picco, G. Vigna, "Understanding code mobility", *IEEE Trans. Software Engineering*, Vol.24, No.5, pp.342-361, May 1998.

[30] J. Schoenwaelder, J. Quittek, C. Kappler, "Building Distributed Management Applications with the IETF Script MIB", *IEEE Journal of Selected Areas in Communications* (JSAC), Vol. 18, No. 5, pp. 702-714, 2000.

[31] A. Bieszczad, B. Pagurek, T. White, "Mobile Agents for Network Management", *IEEE Communication Surveys and Tutorials*, Vol. 1, No. 1, 1998.

[32] A.Liotta, G.Pavlou, G.Knight, "Exploiting agent mobility for large-scale network monitoring", *IEEE Network*, Vol.16, No.3, pp.7-15, May/Jun 2002.

[33] J. Schoenwaelder, "Traditional Approaches to Distributed Management", *Presentation at 19th NMRG Meeting*, KTH Stockholm, Jan.2006.

[34] T.F.Franco et al, "Substituting COPS-PR: an evaluation of NETCONF and SOAP for policy provisioning", *7th IEEE Int. Workshop on Policies for Distributed Systems and Networks* (Policy), pp.10-16, June 2006.

[35] P.Gupta, P.R.Kumar, "The capacity of wireless networks", *IEEE Trans. Information Theory*, Vol.46, No.2, pp.388-404, Mar 2000.

[36] W. Haitao, C. Shiduan, P.Yong, L. Keping, M. Jian, "IEEE 802.11 distributed coordination function (DCF): analysis and enhancement", IEEE Int. Conf. on Communications, ICC 2002, Vol.1, pp.605-609, 2002.

[37] S. Kurkowski, T. Camp, M. Colagrosso, "MANET simulation studies: the incredibles", *ACM SIGMOBILE Mobile Computing Commun. Review* (MC^2R), Vol.9,Iss.4, pp.50-61,2005.

[38] ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). [online].Available:http://www.sigmobile.org/mobihoc

[39] T.R. Andel, A. Yasinsac, "On the credibility of manet simulations", *IEEE Computer*, Vol.39, No.7, pp. 48-54, July 2006.

[40] H.Xiaoyan, X.Kaixin, M.Gerla, "Scalable routing protocols for mobile ad hoc networks", *IEEE Network*, Vol.16, No.4, pp.11-21, Jul/Aug 2002.

[41] I.F. Akyildiz, W. Xudong, "A survey on wireless mesh networks", *IEEE Commun. Mag.*, Vol.43, No.9, pp. S23-S30, Sept. 2005.

[42] R. Bruno, M. Conti, E. Gregori, "Mesh networks: commodity multihop ad hoc networks", *IEEE Commun. Mag.*, Vol.43, No.3, pp.123-131, Mar.2005.

[43] ACM SIGMOBILE 1st Workshop on Vehicular Ad Hoc Networks (VANET 2004), Philadelphia, Oct.2004.

[44] R. A. Berry, E. M. Yeh, "Cross-layer wireless resource allocation", *IEEE Trans. Signal Processing*, Vol. 21,pp. 59–68, Sep. 2004.

[45] X. Qiuyan, J. Xing, M. Hamdi, "Cross Layer Design for the IEEE 802.11 WLANs: Joint Rate Control and Packet Scheduling", *IEEE Trans. Wireless Communications*, Vol.6, No.7, pp.2732-2740, July 2007.

[46] M. Conti, S. Giordano, "Multihop Ad Hoc Networking: The Theory", *IEEE Commun. Mag.*, Vol.45, No.4, pp.78-86, Apr.2007.

[47] M. Conti, S. Giordano, "Multihop Ad Hoc Networking: The Reality", *IEEE Commun. Mag.*, Vol.45, No.4, pp.88-95, Apr.2007.

[48] R. Sahoo et al, "Critical event prediction for proactive management in large-scale computer clusters", *Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '03, Washington D.C., Aug.2003.

[49] L. Pelusi, A. Passarella, M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks", *IEEE Commun. Mag.*, Vol.44, No.11, pp.134-141, Nov.2006.

[50] I.F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, E. Cayirci, "A survey on sensor networks", *IEEE Commun. Mag.*, Vol.40, No.8, pp. 102-114, Aug 2002.

[51] W. Chen, N. Jain, S. Singh, "ANMP Ad hoc network management protocol", *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol.17, Iss.8, pp.1506-1531, Aug.1999.

[52] C. Shen, C. Srisathapornphat, C. Jaikaeo, "An adaptive management architecture for ad hoc networks", *IEEE Commun. Mag.*, Vol.41, Iss.2, pp.108-115, Feb. 2003.

[53] R. Chadha et al, "Policy-based mobile ad hoc network management for drama", *IEEE Military Commun. Conf.*, MILCOM 2004. Vol.3, pp.1317 – 1323, Nov.2004.

[54] R. Chadha et al, "Policy Based Mobile Ad hoc Network Management", *5th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY'04)*, New York, USA, pp.35-44, Jun.2004.

[55] K.S. Phanse, "Policy-Based Quality of Service Management in Wireless Ad-hoc Networks", PhD thesis, Fac. Virginia Polyt. Inst. & St. Univ., Aug.2003.

[56] K.S. Phanse, L.A. DaSilva, "Protocol support for policy-based management of mobile ad hoc networks", *9th IEEE/IFIP Network Operations and Management Symposium (NOMS2004)*, Seoul, Korea, Vol.1, pp.3-16, Apr 2004.

[57] R. Badonnel, R. State, O. Festor, "Management of mobile ad-hoc networks: evaluating the network behaviour", *9th IFIP/IEEE Int. Symp. on Integrated Network Management (IM2005)*, pp.17 – 30, May 2005.

[58] R. Badonnel, R. State, O. Festor, "Probabilistic Management of Ad-Hoc Networks", *10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, Vancouver, Canada, pp.339-350,2006.

[59] R. Chadha et al, "PECAN: policy-enabled configuration across networks", *Proc. IEEE 4th Int. Workshop on Policies for Distributed Systems and Networks (POLICY'03)*, pp.52-62, Jun.2003.

[60] C.-Y.J Chiang, A. McAuley, D. Chee, L. Wong, "Generic protocol for network management data collection and dissemination", *IEEE Military Communications Conference* MILCOM 2003, Vol.2, pp. 971-976, Oct. 2003.

[61] K. Manousakis, A. McAuley, R. Morera, J. Barasj, "Routing Domain Auto-configuration for more efficient and rapidly deployable mobile networks", *23rd Army Science Conference*, USA, Dec. 2002.

[62] K.C. Young et al, "Ad hoc mobility protocol suite for the MOSAIC ATD", *IEEE Military Commun. Conf.*,MILCOM 2003, Vol.2, pp.1348-1352, Oct.2003.

[63] M.Burgess, G.Canright, "Scalability of peer configuration management in logically ad hoc networks", *e-Transactions on Network and Service Management*, Vol.1 No.1 Second Quarter 2004.

[64] S. A. Brueckner, H. Van Dyke Parunak, "Self-Organizing MANET Management" *Engineering Self-Organising Systems*. Springer, 2004.

[65] H. Van Dyke Parunak, S.A. Brueckner, R.Matthews, J.Sauter, "Pheromone learning for self-organizing agents", *IEEE Trans. on Systems, Man and Cybernetics,* Part A, Vol.35, No.3, pp.316-326, May 2005.

[66] A. K. Dey, "Understanding and using context", *Springer Journal of Personal and Ubiquitous Computing*, Vol.5,No.1, pp. 4-7, 2001.

[67] A.Malatras, *Context-Awareness for the Self-Management of Mobile Ad Hoc Networks*. PhD Thesis, Univ. of Surrey, 2007.

[68] Y.Kun, O.Shumao, A.Liotta, I.Henning, "Composition of context-aware services using policies and models", IEEE Global Telecommunications Conf., 2005. GLOBECOM '05., Vol.2, No., pp. 5 pp.-, 28 Nov.-2 Dec. 2005.

[69] A. Jayasuriya, et al, "Hidden vs. Exposed Terminal Problem in Ad hoc Networks", *Proc. of Australian Telecom. Networks and Applications Conf.*, Dec 2004.

[70] O.F. Gonzalez-Duque, M. Howarth, G. Pavlou, "Detection of Packet Forwarding Misbehaviour in Mobile Ad hoc Networks", *Proc. 5th Int. Conf. on Wired/Wireless Internet Communications* (WWIC'2007), Portugal, Springer, May 2007.

[71] Y. Hao, J. Shu, M. Xiaoqiaom L. Songwu , "SCAN: self-organized network-layer security in mobile ad hoc networks", *IEEE Journal on Selected Areas in Communications* (JSAC), Vol.24, No.2, pp. 261-273, Feb. 2006.

[72] I. Koutsopoulos, L. Tassiulas, "Joint Optimal Access Point Selection and Channel Assignment in Wireless Networks", *IEEE/ACM Trans. Networking*, Vol.15, No.3, pp.521-532, Jun.2007.

[73] J.B. Punt, D. Sparreboom, F. Brouwer, R. Prasad, "Mathematical analysis of dynamic channel selection in indoor mobile wireless communication systems", *IEEE Trans. Vehicular Technology*, Vol.47, No.4, pp.1302-1313, Nov 1998.

[74] P. Fuxjager, D. Valerio, F. Ricciato, "The myth of non-overlapping channels: interference measurements in IEEE 802.11", 4th Annual Conference on Wireless on Demand Network Systems and Services, WONS'07 , pp.1-8, 2007.

[75] M. Heusse, F. Rousseau, G. Berger-Sabbatel, A. Duda, "Performance anomaly of 802.11b", 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM'03, pp. 836-843 Vol.2, 2003.

[76] M. R. Garey, D. S. Johnson, L. Stockmeyer, "Some simplified NP-complete problems", *Proc. 6th Annual ACM Symposium on Theory of Computing*, Seattle, USA, pp.47-63, 1974.

[77] J. Blum, M. Ding, A. Thaeler, X. Cheng, "*Connected Dominating Set in Sensor Networks and MANETs*" in *Handbook of Combinatorial Optimization.* Springer, pp. 329-369,2005.

[78] J. Wu, H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks", in *Proc. 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Washington, USA, pp.7-14, 1999.

[79] J. Wu, F. Dai, "Virtual Backbone Construction in MANETs Using Adjustable Transmission Ranges", IEEE Trans. Mobile Computing (TMC), Vol.5, No.9, pp.1188-1200, Sept. 2006.

[80] J. Wu, F. Dai, M. Gao, I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks", *Journal Of Communications And Networks,* Vol.4, No.1, March 2002.

[81] P.-J. Wan, K. M. Alzoubi, O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks", in *Proc. 21st Ann. Joint Conf. of IEEE Computer and Communications Societies*, INFOCOM'02, New York, USA, 2002.

[82] U. Kozat and L. Tassiulas, "Network layer support for service discovery in mobile ad hoc networks", *in Proc. 22nd Ann. Joint Conf. of IEEE Computer and Communications Societies INFOCOM'02*, San Francisco, USA,2002.

[83] B. Das, V. Bhargavan, "Routing in ad-hoc networks using minimum connected dominating sets", in *Proc. IEEE Int. Conf. on Communications ICC'97,* Montreal, 1997.

[84] B. Das, E. Sivakumar, V. Bhargavan, "Routing in ad-hoc networks using a virtual back-bone", *Proc. 6th Int. Conf. on Computer Communications and Networks IC3N'97*, Las Vegas, USA, pp.1-20, Sept. 1997.

[85] P. Krishna, M. Chatterjee, N. H. Vaidya, D. K. Pradhan, "A cluster-based approach for routing in ad-hoc networks", *Proc. 2nd USENIX Symp. on Mobile and Location-Independent Computing MLICS'95*, Ann Arbor, USA, pp. 1-10, April 1995.

[86] C.Ragusa, A.Liotta, G.Pavlou, "An adaptive clustering approach for the management of dynamic systems", IEEE Journal on Selected Areas in Communications, Vol.23, No.12, pp.2223-2235, Dec. 2005.

[87] R.C. Larson, A.R. Odoni, "Applications of Network Models", in *Urban Operations Research*. Prentice-Hall,1981.

[88] R. Friedman, M. Gradinariu, G. Simon, "Locating cache proxies in MANETs", in *Proc. Fifth ACM Int. Symp. on Mobile Ad Hoc Networking and Computing*, MobiHoc 2004 Tokyo, Japan, pp.175-186, 2004.

[89] T. Bin, G. Himanshu, R. Samir, "Benefit-Based Data Caching in Ad Hoc Networks", *IEEE Trans. Mobile Computing*, Vol. 7, No.3, pp.289, Mar.2008.

[90] P. Nuggehalli, V. Srinivasan, C-F. Chiasserini, R.R. Rao, "Efficient Cache Placement in Multihop Wireless Networks", *IEEE/ACM Trans. Networking*, Vol.14, No.5, pp.1045-1055, Oct. 2006.

[91] C. Swamy, A. Kumar, "Primal-dual algorithms for connected facility location problems," Proc. 5th Int. Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'02), Rome, Italy, pp. 245–269, Sep. 2002.

[92] P.Obreiter, J.Nimis, "A Taxonomy of Incentive Patterns - The Design Space of Incentives for Cooperation", Technical Report No. 2003-9, Faculty of Informatics, Univ. of Karlsruhe, 2003.

[93] Cisco Systems, "Channel Deployment Issues for 2.4-GHz 802.11 WLANs". [online]. http://www.cisco.com

[94] M. Sloman, "Policy Driven Management For Distributed Systems", *Journal of Network and Systems Management*, Vol 2, No. 4, pp. 333-360,Dec. 1994.

[95] M.Sloman, E. Lupu, "Policy Specification for Programmable Networks", *Proc. 1st Int. Working Conference on Active Networks* (IWAN'99), Berlin, Jun.1999.

[96] P.Flegkas, P. Trimintzios, G. Pavlou, "A policy-based quality of service management system for IP DiffServ networks", *IEEE Network*, Vol.16, Iss.2, pp.50-56, Mar.-Apr.2002.

[97] P.Flegkas, P. Trimintzios, G. Pavlou, A. Liotta, "Design and implementation of a policy-based resource management architecture", *IFIP/IEEE 8th Int. Symposium on Integrated Network Management* (IM'2003), pp.215 – 229, Mar.2003.

[98] D.C. Verma, "Simplifying network administration using policy-based management" , *IEEE Network*, Vol.16,Iss.2, Mar-Apr.2002.

[99] R. Boutaba, S. Omari, A. Virk, "SELFCON: An Architecture for Self-Configuration of Networks", *Journal of Commun. and Networks*,Vol.3. No.4, pp.317-323, 2001.

[100] D. Chadwick et al, "Coordination between distributed PDPs", 7th IEEE Intl. Workshop on Policies for Distributed Systems and Networks (POLICY'2006), Canada, 2006.

[101] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan, T. Freeman, "A Multipolicy Authorization Framework for Grid Security", 5th IEEE Int. Symp. Network Computing and Applications (NCA), pp.269-272, 2006.

[102] R. Boutaba, I.Aib, "Policy-based Management: A Historical Perspective", *Journal of Network and Systems Management* (JNSM), Vol.15, No. 4, Dec. 2007.

[103] J.Lobo, R. Bhatia, S. Naqvi, "A policy description language", *Proc. 16th National Conf. on Artificial Intelligence and 11th Innovative Applications of Artificial Intelligence Conf.*, pp.291–298, USA, 1999.

[104] S. Gouveris, S.Sivavakeesar, G. Pavlou, A. Malatras, "Programmable middleware for the dynamic deployment of services and protocols in ad hoc networks", *9th IFIP/IEEE Int. Symp. Integrated Network Management* (IM 2005), pp. 3-16, May 2005.

[105] A. Malatras, G. Pavlou, S.Gouveris, S. Sivavakeesar, V. Karakoidas, "Self-Configuring and Optimizing Mobile Ad Hoc Networks", *Proc. 2nd IEEE Int. Conf. on Autonomic Computing*, ICAC'2005, pp.372-373, Jun.2005.

[106] M.Sloman, E. Lupu, "Policy Specification for Programmable Networks", *Proc. 1st Int. Working Conf. on Active Networks (IWAN'99)*, Berlin, Jun.1999.

[107] N. Damianou, N. Dulay, E. Lupu, M Sloman, "The Ponder Specification Language", *Workshop on Policies for Distributed Systems and Networks (Policy2001)*,Bristol, Jan 2001.

[108] N.Dulay, E.Lupu, M.Sloman, N.Damianou, "A policy deployment model for the Ponder language", *Proc. IEEE/IFIP International Symposium on Integrated Network Management*, pp.529 – 543,14-18 May 2001.

[109] G. Fitzpatrick, S. Kaplan, T. Mansfield, D. Arnold, and B. Segal, "Supporting public availability and accessibility with Elvin: Experiences and reflections," *Journal Collaborative Computing*, pp. 15–51, Oct. 2000.

[110] L.Lymberopoulos, E.Lupu, M.Sloman, "Using CIM to realize policy validation within Ponder Framework", DMTF Website Academic Alliance.[online]. Available: http://www.dmtf.org/education/academicalliance

[111] A.Westerinen, J.Schott, "Implementation of the CIM Policy Model using PONDER", *Proc. 5th IEEE Int. Workshop on Policies for Distributed Systems and Networks*, POLICY'04, pp.207 – 210, Jun.2004.

[112] E. Lupu et al, "AMUSE: Autonomic Management of Ubiquitous e-Health Systems." *Concurrency and Computation: Practice and Experience*, Vol.20,Iss.3, pp.277-295, 2007.

[113] S.L.Keoh , E.Lupu , M.Sloman, "PEACE: A Policy-Based Establishment of Ad-hoc Communities", *20th Ann. Computer Security Applications Conf.*, pp.386-395, Dec.2004.

[114] B. Delcourt A. van Lamsweerde, A. Dardenne, F. Dubisy, "The KAOS project: Knowledge acquisition in automated specification of software", *AAAI Spring Symposium Series*, Stanford University, pp. 59-62, March 1991.

[115] A.K. Bandara et al, "Policy refinement for DiffServ quality of service management", IEEE eTransactions on Network and Service Management (eTNSM),Vol3,No.2, 2nd Q.2006.

[116] J.Rubio-Loyola et al, "Using linear temporal model checking for goal-oriented policy refinement frameworks", *Proc. 6th IEEE Int. Workshop on Policies for Distributed Systems and Networks*, POLICY'05, pages 181–190, 2005.

[117] J. D. Moffett, M. S. Sloman, "Policy conflict analysis in distributed system management", *Journal of Organizational Computing*, Vol.4,No.1, pp.1–22, 1994.

[118] E.C. Lupu, M.S. Sloman, Conflicts in policy-based distributed systems management, *IEEE Trans. Software Engineering*, Vol.25, No.6, pp.852-869, Nov/Dec.1999.

[119] M. Charalambides et al, "Dynamic Policy Analysis and Conflict Resolution for DiffServ Quality of Service Management", *10th IEEE/IFIP Network Operations and Management Symposium, (*NOMS 2006), pp.294-304, 2006.

[120] M. Charalambides et al, "Policy conflict analysis for quality of service management", *IEEE 6th Int. Workshop on Policies for Distributed Systems and Networks* (POLICY'05), pp.99-108, Jun.2005

[121] R.Sahita (2002), "COPS Protocol Provides New Way of Delivering Services on the Network", Intel Website.[online].Available: http://www.intel.com/technology/magazine/communications/nc05021.pdf

[122] M. Burgess, "An approach to understanding policy, based on autonomy and voluntary cooperation", *16th IFIP/IEEE Distributed Systems: Operations and Management Workshop* (DSOM'2005), Vol.3775/2005, pp.97-108, Oct.2005.

[123] V. Koutsonikola, A. Vakali, A. "LDAP: framework, practices, and trends", *IEEE Internet Computing*, Vol. 8, Iss.5, pp.66 – 72,Sept.-Oct. 2004.

[124] Sun Microsystems (2003), "A Technical Overview of the Sun ONE Directory Server 5.2" http://www.sun.com/software/products/directory_srvr_ee/wp_directorysrvr52_techoverview.pdf

[125] E.J.Thornton, D.Mundy, D.W.Chadwick, "A comparative performance analysis of seven LDAP Directories".[online].Available: http://tnc2003.terena.org/programme/papers/p1d1.pdf

[126] W.Dixon, T.Kiehl, B.Smith, M.Callahan, "An Analysis of LDAP Performance Characteristics", *GE Global Research, Technical Information Series*, tech. report TR-2002GRC154, June 2002.

[127] X.Wang, H.Schulzrinne, D.Kandlur, D.Verma, "Measurement and Analysis of LDAP Performance", *Int. Conf. Measurement and Modeling of Computer Systems* (SIGMETRICS'2000), Santa Clara, CA, pp. 156-165, Jun. 2000.

[128] A.Vakali, B. Catania, A. Maddalena, "XML Data Stores: Emerging Practices", *IEEE Internet Computing*, Vol.9, Iss.2, pp.62 – 69,March-April 2005.

[129] A.Matheus, "How to Declare Access Control Policies for XML Structured Information Objects using OASIS' eXtensible Access Control Markup Language (XACML)", *Proc. 38th Ann. Hawaii Int. Conf. System Sciences 2005,* HICSS '05, Jan. 2005.

[130] M.Lorch, D.Kafura, S.Shah, "An XACML-based policy management and authorization service for globus resources", *Proc. 4th Int. Workshop Grid Computing*, pp.208–210, Nov.2003.

[131] OpenLDAP Foundation, "FAQ:Does slapd(8) support multi-master replication?". [online].Available: http://www.openldap.org/faq/data/cache/1240.html

[132] Isode Ltd. (Jul.2005), "Distributed Directory in support of large scale PKI".[online]. Available: http://www.isode.com/whitepapers/dist-dir-pki.html

[133] IBM Inc.(2004), "An architectural blueprint for autonomic computing".[online]. Available: http://www-03.ibm.com/autonomic/pdfs/ACBP2_2004-10-04.pdf

[134] J.O.Kephart, D.M.Chess, "The vision of autonomic computing", *IEEE Computer*, Vol.36, Iss.1, pp.41 - 50, Jan. 2003.

[135] R.Boutaba, J. Xiao, "Self-Managing Networks", in *Cognitive Networks: Towards Self-Aware Networks,* Qusay Mahmoud Ed., Chapter 4, pp.77-97, Wiley, 2007.

[136] J. Strassner, D. Raymer, "Implementing Next Generation Services Using Policy-Based Management and Autonomic Computing Principles", *10th IEEE/IFIP Network Operations and Management Symposium,* NOMS 2006, Vancouver, Canada, pp.1-15, 2006.

[137] R.Mortier, E. Kiciman, "Autonomic network management: some pragmatic considerations", *ACM Proc. 2006 SIGCOMM Workshop on Internet Network Management*, Pisa, INM '06., pp.89-93, 2006.

[138] G. Pavlou, A. Hadjiantonis, A. Malatras (Eds) (Dec.2006),"D9.1:Frameworks and Approaches for Autonomic Management of Fixed QoS-enabled and Ad Hoc Networks", *EMANICS Network of Excellence, Deliverable.*[online].Available: www.emanics.org

[139] J. Mäntyjärvi, P. Huuskonen and J. Himber, "Collaborative Context Determination To Support Mobile Terminal Applications", *IEEE Wireless Commun.*, Vol. 9, No. 5, pp.39-45, Oct.2002.

[140] M.Frodigh, P.Johansson, P.Larsson (2004), "Wireless ad hoc networking – The art of networking without a network", Ericsson publications.[online]. Available: http://www.ericsson.com/about/publications/review/2000_04/files/2000046.pdf

[141] C.S. Murthy, B.S. Manoj, "Ad Hoc Wireless Networks, Architectures and protocols", Prentice Hall PTR, ISBN: 013147023X, 2004.

[142] S.Sivavakeesar, G. Pavlou, C.Bohoris, A. Liotta, "Effective management through prediction-based clustering approach in the next-generation ad hoc networks", *IEEE Int. Conf. on Communications*, Vol.7,pp.4326 – 4330, Jun.2004.

[143] L.M.Feeney, B.Ahlgren, A.Westerlund, "Spontaneous networking: an application oriented approach to ad hoc networking", *IEEE Commun. Mag.*, Vol.39, Iss.6, pp.176-181, Jun.2001.

[144] LDAP RFC List.[online].Available: http://search.cpan.org/perldoc?Net::LDAP::RFC

[145] L.A.DaSilva et al, "Network mobility and protocol interoperability in ad hoc networks", *IEEE Commun. Mag.*, Vol.42, Iss.11, pp.88 – 96, Nov. 2004.

[146] Policy Research Group, DoC, Imperial College Website.[online]. Available: http://www-dse.doc.ic.ac.uk/Research/policies/index.shtml

[147] L.Lymberopoulos, E.Lupu, M.Sloman, "An adaptive policy based management framework for differentiated services networks", *3rd Int. Workshop on Policies for Distributed Systems and Networks*, pp.147 – 158, Jun.2002.

[148] DMTF website, CIM Policy Model White Paper for CIM v2.7.0.[online]. Available: http://www.dmtf.org/standards/documents/CIM/DSP0108.pdf

[149] R.Montanari, E.Lupu, C.Stefanelli, "Policy-based dynamic reconfiguration of mobile-code applications", *IEEE Computer*, Vol.37, Iss.7, pp.73–80,Jul.2004.

[150] M. Sloman, E.Lupu, "Security and management policy specification", *IEEE Network, Special Issue on Policy-Based Networking*, Vol.16, Iss.2, pp.10-19, March-April 2002.

[151] Mi-Jung Choi, Hyoun-Mi Choi, J.W. Hong, Hong-Taek Ju, "XML-based configuration management for IP network devices", *IEEE Commun. Mag.*, Vol.42,Iss.7,pp.84-91, Jul.2004.

[152] R.Natarajan, A.P. Mathur, P. McKee, "A XML based policy-driven management information service", *IEEE/IFIP Int. Symposium on Integrated Network Management*, pp.277-280,14-18 May 2001.

[153] N.Damianou, N.Dulay, E.Lupu, M.Sloman, T.Tonouchi, "Tools for domain-based policy management of distributed systems", *IEEE/IFIP Network Operations and Management Symposium*, NOMS'02,pp.203–217, 15-19 April 2002.

[154] A.Hadjiantonis, Personal Website.[online].Available: URL/research/schemas.zip

[155] IANA, Private Enterprise Numbers, http://www.iana.org/assignments/enterprise-numbers

[156] Kaspersky Lab, Field measurements from wardriving in major cities, Hannover 2006: http://www.viruslist.com/en/analysis?pubid=182068392, Sao Paolo 2008 pubid=204791997, Paris 2006: pubid=204791912, London 2007 pubid=204791945.[online]. Available: http://www.viruslist.com/en/analysis

[157] XML-RPC protocol specification. [online]. Available: http://www.xmlrpc.com/spec

[158] D. Decasper, Z. Dittia, G. Parulkar, B. Plattner, "Router plugins: a software architecture for next-generation routers", *IEEE/ACM Transactions on Networking*, Vol.8, No.1, pp.2-15, Feb.2000.

[159] Cisco Systems Inc., Cisco ASR 1000 Series Aggregation Services Routers.[online]. Available: http://www.cisco.com

[160] Q. Mahmoud (2000), "*MIDP Network Programming using HTTP and the Connection Framework*". [online]. Available: http://developers.sun.com/mobility/midp/articles/network/

[161] M. Pawlan (2001) "*Introduction to Wireless Technologies*", [online]. Available: http://developers.sun.com/mobility/getstart/articles/intro

[162] J. Knudsen (2002), "*Parsing XML in J2ME*", [online]. Available: http://developers.sun.com/mobility/midp/articles/parsingxml/index.html

[163] Java Community Process (Mar.2004) "*Java Specification Requests JSR 172, J2ME Web Services Specification*". [online]. Available: http://jcp.org/en/jsr/detail?id=172

[164] Juniper Networks, Inc., JUNOScript Application Programmer Interface.[online]. Available: http://www.juniper.net/support/xml/junoscript

[165] Cisco Systems Inc., "*Cisco IOS XR XML API Guide, Rel.3.2, Chap.12 XML Transport and Event Notifications*". [online]. Available: http://www.cisco.com

[166] Cisco Systems Inc. (Feb.2007), "*Cisco Announces Agreement to Acquire Reactivity*", Press Release. [online]. Available: http://newsroom.cisco.com/dlls/2007/corp_022107.html

[167] Cisco Systems Inc. (2006), "Living the Connected Life", Cisco Whitepaper. [online]. Available: http://www.cisco.com

[168] A.Yew, A.Liotta, G.Pavlou, "Applying a policy-based framework to manage quality of service requirements in the virtual home environment", *IEEE Intl. Conf. on Communications* (ICC2002), Vol.4, pp. 1985-1990, 2002.

[169] C. Tsarouchis et al, "A policy-based management architecture for active and programmable networks", *IEEE Network*, Vol.17, No.3, pp. 22-28, May-June 2003.

[170] R. T. Fielding, R. N. Taylor, "Principled design of the modern Web architecture", *ACM Trans. Internet Technology*, Vol.2, Iss.2, pp.115-150, May. 2002.

[171] Free Software Foundation, GNU Project, GNU General Public License.[online]. Available: http://www.gnu.org/licenses/gpl.html

[172] Deloitte TMT Industry Group (Jan 2007), "Technology, Media & Telecom. Trends: Predictions 2007".[online]. Available: http://www.deloitte.com

[173] IBM Corp., "Policy Management for Autonomic Computing", PMAC v1.2, 2005. [online]. Available: http://www.ibm.com/autonomic/

[174] IBM Corp., "Autonomic Computing Policy Language (ACPL)", 2005.[online]. Available: http://www.ibm.com/autonomic/

Standards

[175] Object Management Group, "The Common Object Request Broker: Architecture and Specification (CORBA)", Version 2.0, 1995.

[176] World Wide Web Consortium (W3C), "Web Services Activity".[online]. Available: http://www.w3c.org/2002/ws

[177] World Wide Web Consortium (W3C), "SOAP Version 1.2 Recommendation", 2005. [online]. Available: http://www.w3.org/TR/soap12

[178] OASIS XACML Tech.Com., "eXtensible Access Control Markup Language v2.0", Feb.2005.[online]. Available: http://www.oasis-open.org/committees/xacml

[179] 3rd Generation Partnership Project (3GPP) TS 29.207, "Policy control over Go interface", V1.0.0 for Release 5, 2002

[180] DMTF, Common Information Model (CIM).[online]. Available: http://www.dmtf.org/standards/cim

[181] DMTF, CIM Simplified Policy Language (CIM-SPL), (DSP0231 v1.0.0a, 2007).[online]. Available: http://www.dmtf.org/standards/published_documents/DSP0231.pdf

[182] DMTF, CIM Event Model White Paper, CIM Version 2.7 (DSP0107 v2.1, 2003).[online]. Available: http://www.dmtf.org/standards/documents/CIM/DSP0107.pdf

[183] IEEE Std 802.11-2007, IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[184] IEEE Std 802.11h-2003, "Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz band in Europe", (Amendment to IEEE Std 802.11).

[185] IEEE Draft Std P802.11n/D4.00, "Amendment 4: Enhancements for Higher Throughput", IEEE Unapproved Draft Std P802.11n/D4.00, Mar 2008, (Amendment to IEEE Std 802.11).

[186] ITU-T Rec.X.500, "The Directory: Overview of Concepts, Models and Service", 1993.

[187] ITU-T Rec.X.700, Information Technology - Open Systems Interconnection, "OSI Management Framework", 1992.

[188] ITU-T Rec.X.701, Information Technology - Open Systems Interconnection, "Systems Management Overview", 1992.

[189] ITU-T Rec. X.711, Data Communication Networks - Open Systems Interconnection, "OSI Management -Common Management Information Protocol Specification", 1991.

IETF Activities & RFC

[190] IETF Mobile Ad-Hoc Networks Working Group (MANET), IETF Website http://www.ietf.org/html.charters/manet-charter.html

[191] IETF Ad-Hoc Network Autoconfiguration Working Group (AUTOCONF), IETF Website http://www.ietf.org/html.charters/autoconf-charter.html

[192] IETF Policy Framework Working Group (POLICY), concluded in 2004, IETF Website http://www.ietf.org/html.charters/OLD/policy-charter.html

[193] IETF Resource Allocation Protocol Working Group (RAP), concluded in 2005, IETF Website, http://www.ietf.org/html.charters/OLD/rap-charter.html

[194] IETF Network Configuration Working Group (NETCONF), IETF Website http://www.ietf.org/html.charters/netconf-charter.html

[195] I. Chakeres, J. Macker, T. Clausen, "Mobile Ad hoc Network Architecture", Internet-Draft, (draft-ietf-autoconf-manetarch-07, exp. May 2007), work in progress, Nov.2007.

[196] **RFC1157**, J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", Standards Track, May 1990.

[197] **RFC1945**, T. Berners-Lee, R. Fielding, H. Frystyk, "Hypertext Transfer Protocol -- HTTP/1.0", Informational, May 1996.

[198] **RFC2501**, S. Corson, J. Macker, "Mobile Ad hoc Networking (MANET):Routing Protocol Performance Issues and Evaluation Considerations", Informational, Jan. 1999.

[199] **RFC2616**, R. Fielding et al, "Hypertext Transfer Protocol -- HTTP/1.1", Standards Track, Jun.1999.

[200] **RFC2713**, V. Ryan, S. Seligman, R. Lee, "Schema for Representing Java(tm) Objects in an LDAP Directory", Informational, Oct.1999.

[201] **RFC2748**, D. Durham (Ed) et al, "The COPS (Common Open Policy Service) Protocol", Standards Track, Jan.2000.

[202] **RFC2753**, R. Yavatkar, D. Pendarakis, R. Guerin, "A Framework for Policy-based Admission Control", Informational, Jan.2000.

[203] **RFC2849**, G.Good, "The LDAP Data Interchange Format (LDIF) - Technical Specification", Standards Track, Jun.2000.

[204] **RFC3060**, B. Moore et al, "Policy Core Information Model-Version 1 Specification", Standards Track, Feb.2001.

[205] **RFC3084**, K. Chan et al, "COPS Usage for Policy Provisioning (COPS-PR)", Standards Track, Mar.2001.

[206] **RFC3198**, A. Westerinen et al, "Terminology for Policy-Based Management", Informational, Nov.2001.

[207] **RFC3460**, B. Moore, "Policy Core Information Model (PCIM) Extensions", Standards Track, Jan.2003.

[208] **RFC3561**, C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", Experimental, Jul.2003.

[209] **RFC3626**, T.Clausen, P.Jacquet (eds), "Optimized Link State Routing Protocol (OLSR)", Experimental, Oct.2003.

[210] **RFC3644**, Y. Snir, Y. Ramberg, J. Strassner, R. Cohen, B. Moore, "Policy Quality of Service (QoS) Information Model", Standards Track, Nov.2003.

[211] **RFC3703**, J. Strassner et al, "Policy Core Lightweight Directory Access Protocol (LDAP) Schema", Standards Track, Feb.2004.

[212] **RFC4104**, M. Pana et al, "Policy Core Extension Lightweight Directory Access Protocol Schema (PCELS)", Standards Track, Jun.2005.

[213] **RFC4511,** J. Sermersheim (Ed.), "Lightweight Directory Access Protocol (LDAP): The Protocol", Standards Track, Jun.2006.

[214] **RFC4515**, M. Smith Ed., T. Howes, "Lightweight Directory Access Protocol (LDAP): String Representation of Search Filters", Standards Track, Jun.2006.

[215] **RFC4533**, K. Zeilenga, J.H.Choi, "The Lightweight Directory Access Protocol Content Synchronization Operation", Experimental, Jun.2006.

[216] **RFC4741**, R. Enns (Ed.), "NetConf Configuration Protocol", Standards Track, Dec.2006.

Appendix A. Deployment Issues

This Appendix first provides a short description of a distributed algorithm for the creation of a Dominating Set. The rest of this Appendix deals with deployment issues, demonstrating different implementation possibilities for the proposed framework and its components.

The goal is to dynamically create and maintain an appropriate set of hypercluster nodes that are eventually assign distributed management tasks based on roles. The motivation for using the presented algorithm was explained in Chapter 3. For a comprehensive treatment of algorithms for the calculation of CDS, the reader is referred to [77]. The algorithm by Wu [78] is simple and effective method to calculate a Connected Dominating Set of a graph in a fully distributed, decentralised manner. The algorithm is executed in two stages: the marking round and the optimisation round. The first round creates a possibly redundant Connected Dominating Set (CDS) and the second round reduces that set to become closer to a Minimum Connected Dominating Set (MCDS). Heuristics are used to provide a near optimum solution, since the calculation of a MCDS is a known NP-Hard problem [9] [77].

First it is assumed that $G= (V,E)$ is an undirected graph representing a wireless ad hoc network, where vertices V represent nodes and edges E represent wireless links between nodes. $N(v)$ denotes the open neighbour set of the vertex $v \in V$ if and only if $N(v) = \{u \mid \{v, u\} \in E\}$. The set $N[v]$ denotes the closed neighbour set of v, if and only if $N[v]=N(v)\cup\{v\}$. Each node v has a marker $m(v)$ to indicate whether it belongs to the CDS [m(v)=T] or not [m(v)=F]. In addition, each node v has an arithmetic identifier, $id(v)$.

a. Marking process
   1. Initially assign marker F to every v in V.
   2. Every v exchanges its open neighbour set N(v) with all its neighbours.
   3. Every v assigns its marker m(v) to T if there exist two unconnected neighbours
b. Optimisation rules (heuristics)

   Rule 1: Consider two vertices v and u in G'. If $N[v] \subseteq N[u]$ in G and $id(v) < id(u)$, change the marker of v to F if node v is marked, i.e., G' is changed to G'- {v}

Rule 2: Assume u and w are two marked neighbours of marked vertex v in G'. If $N(v) \subseteq N(u) \cup N(w)$ in G and $id(v) = \min\{id(v), id(u), id(w)\}$, then change the marker of v to F.

An attractive feature of this algorithm is its ability to dynamically anticipate topological changes of wireless ad hoc networks in an autonomous and decentralised way. Authors of [78] identify and provide solutions for three different types of dynamic changes, i.e. mobile host switching on, mobile host switching off, and mobile host movement.

Having adopted the described algorithm, a series of modifications were performed to allow its integration to the presented policy-based and context-aware framework. The main modification involved the substitution of the arbitrary arithmetic node identifier, with a scalar Capability Function CF. CF expresses two aspects of a node's capabilities, i.e. its computing attributes and its mobility. Nodes with higher CF values are preferred during the optimisation round. For example if a node moves quite often and is responsible for link breaks with its neighbours, then its CF is reduced and is less likely to remain in the CDS. In addition to their CF, each node has three more markers, indicating with 1 its current role: CN(v),CH(v), MN(v). These markers facilitate the dynamic role assignment process and can be used in combination with any static predefined role assignment of manager nodes. The executed distributed algorithm is able to identify the most capable nodes to participate in the hypercluster by creating and maintaining a connected dominating set. Nodes that have been marked as m(v)=T assume the role of a Cluster Head, i.e. set their marker as CN(v)=1.

Effectively, CHs together with MNs form the hypercluster and collectively manage the wireless ad hoc network. Nodes that have m(v)=F assume the role of a Cluster Node. Every CN registers itself to its CH neighbour with the highest CF value. Depending on the application use of the wireless ad hoc network, MNs are either dynamically introduced or statically configured upon the initial construction of the network. In the latter case these nodes are explicitly assigned to the MN role and thus to the hypercluster, whereas m(v)=T always and MN(v)=1 by default. In the former, case the described algorithm can be executed again only among the selected set of CH, thus creating a dynamic set of MN. The result is a clustered MANET with nodes in all three of the defined roles.

To evaluate the behaviour and efficiency of this algorithm for hypercluster creation and role assignment, a series of simulations was carried out. After the execution of the algorithm on a static MANET, the hypercluster size was measured. Random MANET topologies were created using the *ns-2* simulator (www.isi.edu/nsnam/ns) and the *setdest* utility, based on the simulation parameters listed in Table A-1. These parameters were chosen to resemble the original algorithm evaluation in [78], in order to confirm the correctness of obtained results. Additional details and evaluation results regarding the modified algorithm can be found in published work [2],[5].

Table A-1. Simulation parameters and results

Fix.Dens(1:1600)				
Network Population	Simulation Area (m^2)	Area Side (m)	Density Ratio	Hypercluster Size (average)
25	40000	200	1:1600	4.1
50	80000	283	1:1600	9.2
75	120000	346	1:1600	14.9
100	160000	400	1:1600	22.1
225	360000	600	1:1600	52.6
400	640000	800	1:1600	99.2
Fix.Dens(1:27800)				
Network Population	Simulation Area (m^2)	Area Side (m)	Density Ratio	Hypercluster Size (average)
25	695000	834	1:27800	12.9
50	1390000	1179	1:27800	26.3
75	2085000	1444	1:27800	38.7
100	2780000	1667	1:27800	54.6
225	6255000	2501	1:27800	124.1
400	11120000	3335	1:27800	221.2
Var.Dens.(~1:625)				
Network Population	Simulation Area (m^2)	Area Side (m)	Density Ratio	Hypercluster Size (average)
25	250000	500	1:10000	5.1
50	250000	500	1:5000	8.0
75	250000	500	1:3333	9.3
100	250000	500	1:2500	10.2
225	250000	500	1:1111	10.3
400	250000	500	1:625	10.3
Var.Dens.(~1:2500)				
Network Population	Simulation Area (m^2)	Area Side (m)	Density Ratio	Hypercluster Size (average)
25	1000000	1000	1:40000	11.2
50	1000000	1000	1:20000	21.4
75	1000000	1000	1:13333	29.9
100	1000000	1000	1:10000	35.3
225	1000000	1000	1:4444	47.1
400	1000000	1000	1:2500	60.4
Transmission radius r= 250 m				

Deployment issues and examples: Based on the presented motivation for module differentiation (§3.3.3,pp.65), three module design examples are presented below to realise the proposed framework and implement roles' functionality. Beyond the presented options, other combinations of components-modules are possible, maintaining the appropriate component set for each role.

1. Single module

This design is the simplest option and relies on the implementation of a single module, integrating all functional components. As already mentioned, the simplicity of this design comes at the cost of increased minimum specification requirements for participating devices. This implies that a number of lightweight devices cannot participate in the wireless ad hoc network since they will not be able to host the demanding software. The single module (Cluster Manager) is able to assume all roles by activating and deactivating the respective components subset (Figure A-1).

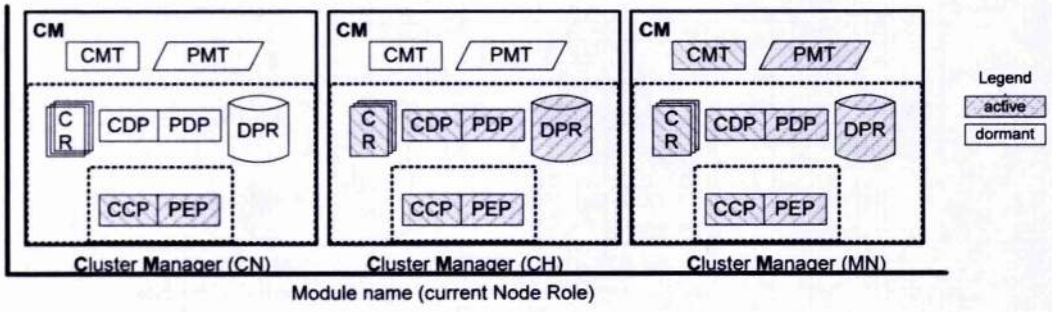

Figure A-1. Single module deployment of roles

2. Dual module

For a dual module design, a fully functional module (Cluster Manager) is designed, able to assume all three roles. A second module (Terminal Node) is also designed to enable the participation of a plethora of devices, e.g. mobile phones, media players, networked white goods etc. Figure A-2 shows how these two modules implement all three roles. It should be noted that the fully functional CM module can also assume the least demanding role (CN), if network composition and density allow that. An example of dual module deployment is presented later.

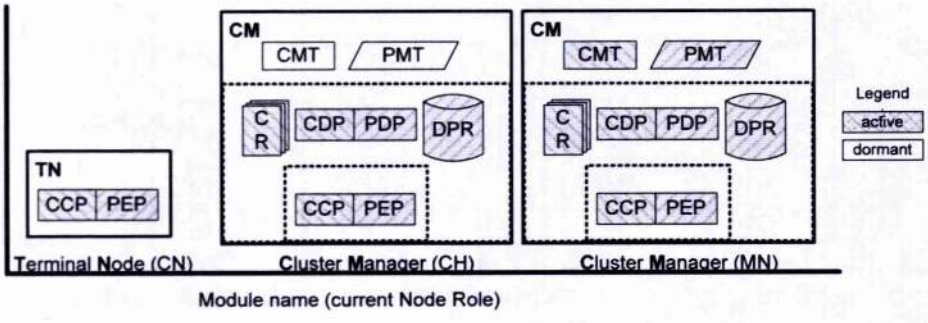

Figure A-2. Dual module deployment of roles

3. Triple module

A third design option is to match the component subsets of each role to a separate module, hence the triple module deployment of Figure A-3. The third module introduced is the Cluster Leader and can assume the roles of CH and CN (CN role not shown). Although more complex, the value of this design can be appreciated in mesh deployments of wireless ad hoc network, where managing entities deploy a limited number of dedicated management nodes, realising the Cluster Manager module and permanently assuming the MN role (Manager Nodes). The rest of the participating devices can carry either the Cluster Leader or the Terminal Node module. That will depend on their capabilities but also on the decision of the device owner. For instance, a laptop owner may decide to install the lightweight Terminal Node module, in order to preserve battery power and avoid resource-consuming operations. On the other hand, managing entities may introduce an incentives scheme, encouraging users to install the more demanding Cluster Leader module and contribute to the collaborative management tasks. Examples of such deployments were described in relevant sections, based on previous work in [4] and [1].

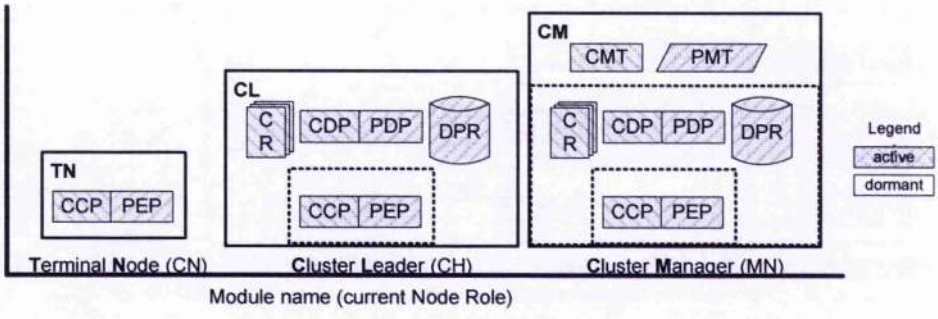

Figure A-3. Triple module deployment of roles

Example of dual module deployment

Since module separation was deemed necessary to accommodate a wider range of node capabilities in MANET, the dual module design is adopted for this example. A fully functional module (Cluster Manager or CM) was designed, able to assume all three roles. A second lightweight module (Terminal Node or TN) was also designed to enable the participation of lightweight devices. Thus TN can only be assigned to the CN role. On the other hand CM modules have full PBM functionality and context processing capability, therefore they are collaboratively responsible for MANET management by their assignment to any one of the three roles. The selection of the appropriate module for each network device depends mainly on its capabilities. A set of minimum requirements offers a prescribed guideline and indicates whether a device can efficiently host the CM module.

The two designed node modules are depicted in Figure A-4, where their respective policy and context related components are also shown. Depending on the assigned role of a cluster manager (CM), the respective components are either active or dormant.

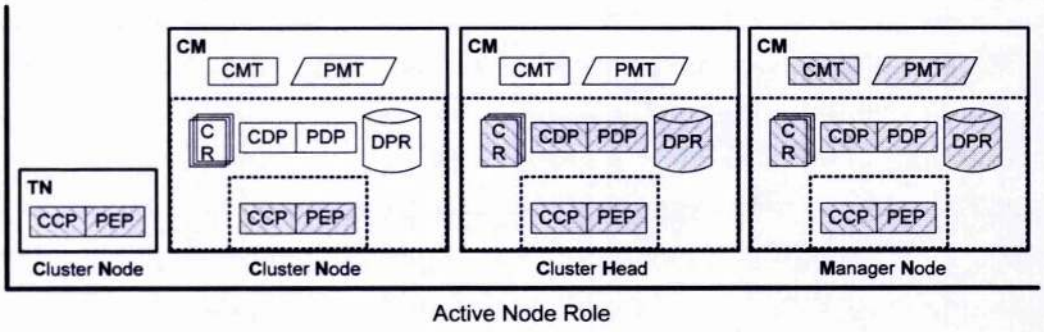

Figure A-4. Node roles and modules

A possible role and module deployment is shown in Figure A-5, to further elaborate on the applicability of the aforementioned dual module design. A deployment example is depicted in Figure A-5, matching the organisational model shown in Figure 3-5 pp.60.

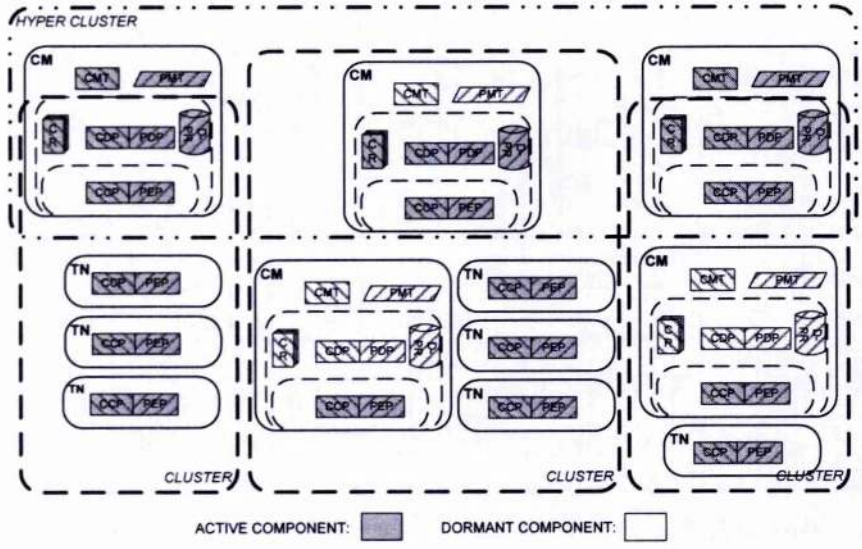

Figure A-5. Example deployment and node modules

Summary for example Modules: eligible roles and included components

Table A-2 summarises the properties of previous example modules by showing which roles the modules can host. Phrase "role N/A" indicates that the role cannot be supported (because of missing components). In addition, the constituting components and their activity status are shown. Phrase "comp. N/A" indicates that the component is not included with the specific module. Finally, phrase "policy dep." for DPR-CR components indicates that their status is dependent on DPR management policies, as explained in §5.3.

Table A-2. Summary Table for Example Modules

Module / Roles	Cluster Manager		Cluster Leader		Terminal Node	
MN Manager Node	Yes		role N/A		role N/A	
CH Cluster Head	Yes		Yes		role N/A	
CN Cluster Node	Yes		Yes		Yes	
Components	Curr. Role	Active (if avail.)	Curr. Role	Active (if avail.)	Curr. Role	Active (if avail.)
PMT-CMT	MN	Yes	MN	comp. N/A	MN	comp. N/A
	CH	No	CH	comp. N/A	CH	comp. N/A
	CN	No	CN	comp. N/A	CN	comp. N/A
DPR-CR	MN	Yes	MN	role N/A	MN	comp. N/A
	CH	policy dep.	CH	policy dep.	CH	comp. N/A
	CN	No	CN	Yes	CN	comp. N/A
PDP-CDP	MN	Yes	MN	role N/A	MN	comp. N/A
	CH	Yes	CH	Yes	CH	comp. N/A
	CN	No	CN	No	CN	comp. N/A
PEP-CCP	MN	Yes	MN	role N/A	MN	role N/A
	CH	Yes	CH	Yes	CH	role N/A
	CN	Yes	CN	Yes	CN	Yes

Detailed Internal Architecture of Components

This subsection provides a detailed view of the designed *components*, emphasising on their composition and interactions to form *modules*. In §3.3.3, a "*module*" has been defined as the preinstalled group of software *components* of a node, needed to realise the management functionality and operations of the proposed framework. The concept of "*roles*" was also introduced to achieve a role-based organisational model. In fact, this separation between roles and modules refers to the differentiation of the organisational role of an entity in the network as opposed to the actual software capabilities it carries. Based on the above, three role entities and their high-level components and interactions were introduced in §3.3. This subsection, provides additional information about the internal architecture of *components* for the framework aiming to serve as *module* implementation guidelines. In addition, a number of new internal components were introduced to the generalised framework components, realising the needed functionality for the presented case studies.

For continuity, components are presented according to the same component sets required for each role (§3.3: CN pp.61, CH pp.62, MN pp.64). This option implies a triple module design, but without loss of generality is adopted for presentation clarity. Different module design and

deployment options can be considered, as described earlier. For completeness, the corresponding context-aware components are also presented below, adopting a simplified technology-independent architecture and extending the original design presented in [2],[5] to suit a wider range of wireless ad hoc networks.

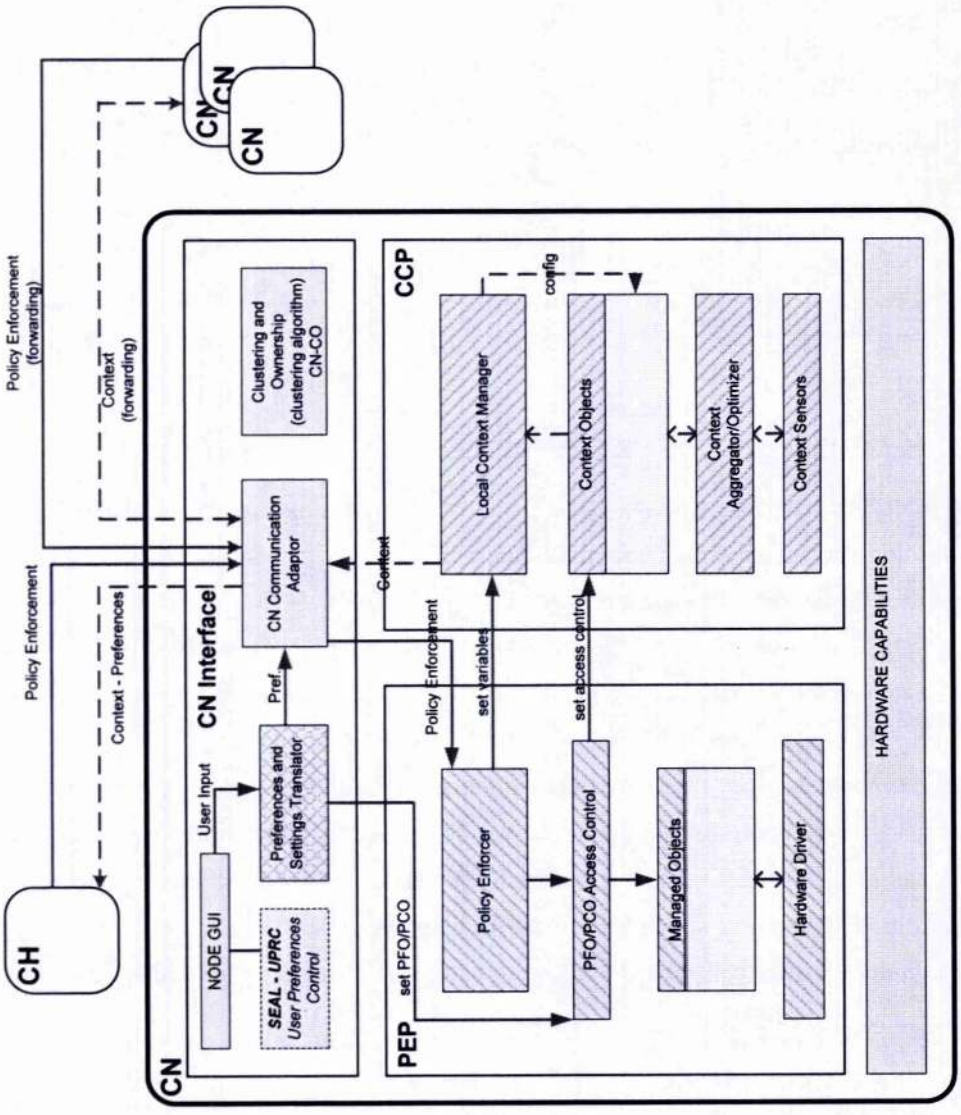

Figure A-6. Internal Architecture of *PEP*, *CCP* and *CN Interface*

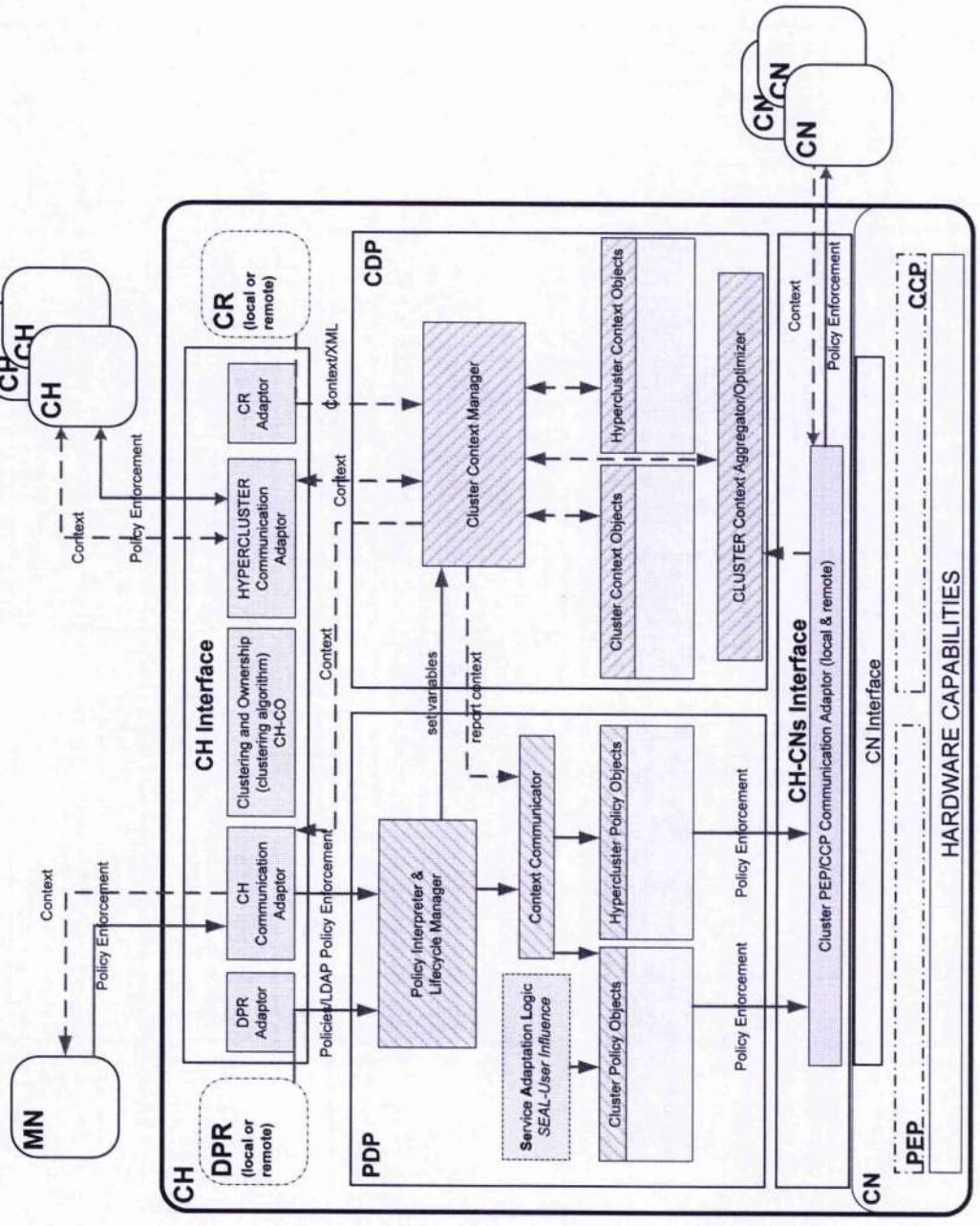

Figure A-7. Internal Architecture of *PDP*, *CDP*, *CH Interface* and *CHs-CNs Interface*

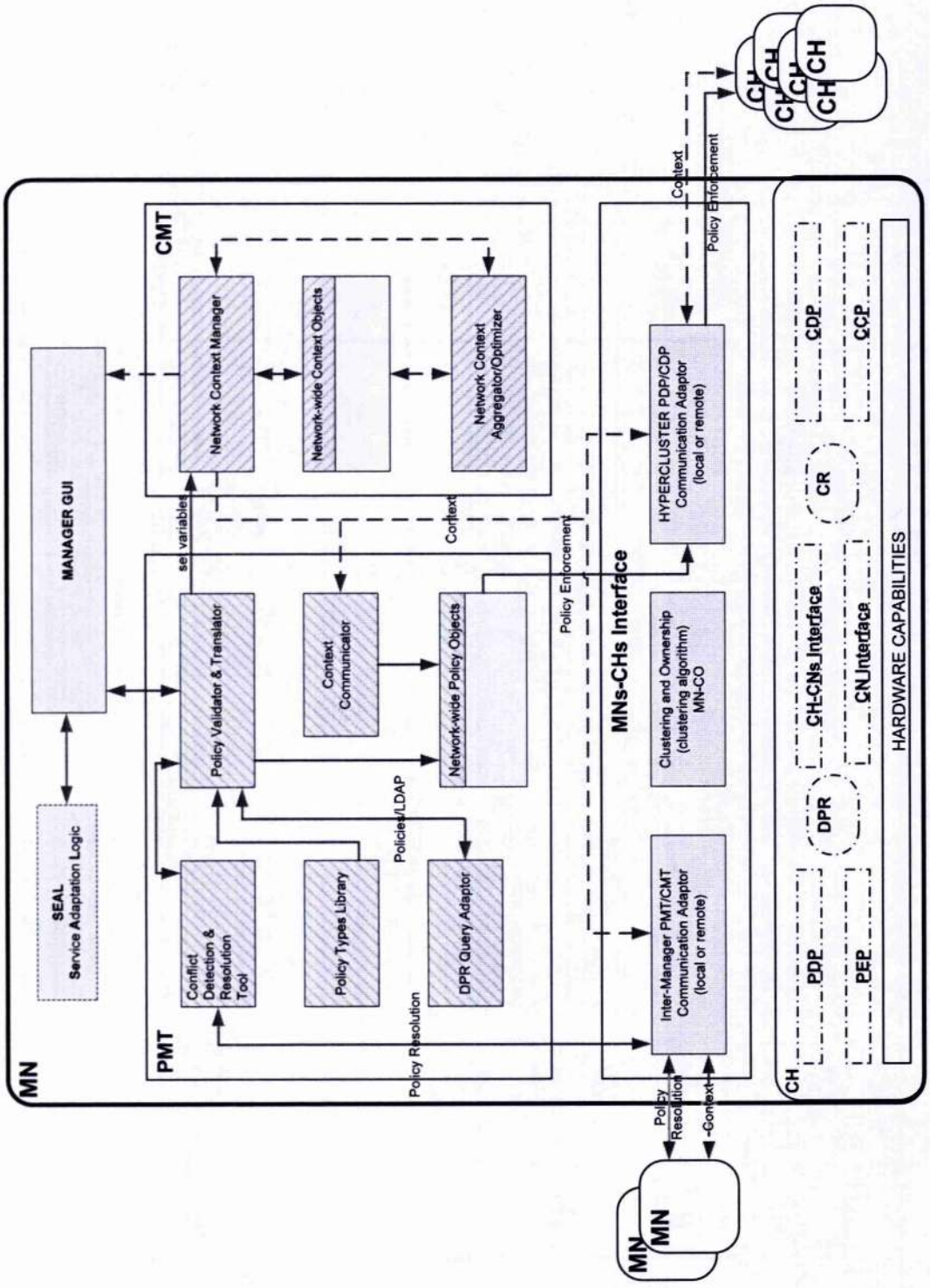

Figure A-8. Internal Architecture of *PMT*, *CMT* and *MNs-CHs Interface*

Appendix B. Introduction to LDAP (Lightweight Directory Access Protocol)

LDAP is a standardised protocol defined by IETF in a series of Request For Comments (RFC). The protocol's current version is LDAP v.3 [213] and as of June 2006, the RFCs defining the core protocol are RFC 4510 to 4519, available from IETF website (http://www.ietf.org).

According to IETF, the LDAP protocol is designed to provide access to directories supporting the X.500 models, while not incurring the resource requirements of the X.500 Directory Access Protocol (DAP). This protocol is specifically targeted at management applications and browser applications that provide read/write interactive access to directories. "Lightweight" means that the protocol is efficient and less demanding compared to the ITU-T X.500 DAP [186]. It uses a simplified set of encoding methods and runs directly on top of TCP/IP, contrary to DAP which requires the complete OSI network stack. A Directory Server Agent (DSA) including its directory content (e.g. policies) is simply referred to as a *Directory*. At the moment, LDAP v3 is established as the primary mean of accessing *Directories* over the Internet. There are four LDAP models:

1. Information model: defines the kind of data the directory can store. The basic building blocks of the directory are *entries*. Entries are composed of *attributes*, which are composed of an *attribute type* and one or more *values*. A *directory schema* contains all the information needed about the required and allowed attributes in a directory. An entry can be abstract, structural or auxiliary. Abstract entry classes can not be instantiated, but can be extended to structural or auxiliary ones. Structural classes are the main building blocks of a directory as they represent distinct entities and must follow the directory schema limitations. Auxiliary classes carry additional information and can be attached freely to structural ones to enrich their content. For the purpose of describing directory information, LDAP defines LDIF (LDAP Data Interchange Format, RFC2849) which is a text-based description of a set of directory entries or a set of updates to apply.
2. Naming model: defines how directory data are organised and refer to, i.e. how entries are structured and placed in a directory and how you each entry can be accessed. This model

specifies that entries should be arranged in an inverted tree structure. Each entry has its unique identifier, called Distinguished Name (DN), which refers to it unambiguously. The DN is formed by the entry's Relative Distinguished Name (RDN) and the position of the entry in the information tree separated with commas. The RDN of an entry is usually its name. All entries comprise the Directory Information Tree (DIT). For example: { *RDN*: cn = Ann Smith } for entry { *DN*: cn= Ann Smith, ou = Staff, dc= ccsr, dc= ac, dc= uk }.

3. Functional model: defines how information in the directory can be accessed and updated. There are three operation categories which group the nine basic protocol operations:

    a. Read operations allow to read and query directory's contents. These are the *Search* and *Compare* operations.

    b. Update operations allow to alter the directory's information. These are the *Add, Delete, Modify* and *ModifyDN* operations.

    c. Control operations allow the initiation and termination of the LDAP client/server communication. These are the *Bind, Unbind* and *Abandon* operations.

    In addition to these basic operations, LDAPv3 offers protocol extensibility using LDAP *extended* operations, LDAP *controls* and *SASL* (see Security model). The *extended* operation takes a request as an argument and returns a response. The pair of *extended* operation request/response is called an *extension* and can be used to define new operations in this way. Controls are additional information carried by LDAP operations which can alter the operation's behaviour.

4. Security model: defines how information in the directory can be protected from unauthorised access. LDAP supports the Simple Authentication and Security Layer (SASL) authentication framework (RFC 4422) to allow different authentication mechanisms to be used with LDAP. Several SASL mechanisms are currently defined, e.g. Kerberos V5 (RFC 4752), while new mechanisms can also be introduced. Also the connection-oriented nature of LDAP allows additional security mechanisms to be implemented using TLS and HTTPS.

These models promote interoperability between different implementations while allowing enough implementation freedom to fit specific needs. Together they constitute the LDAP protocol itself and direct its implementation and applicability.

LDAP Synchronisation-Replication engine

The *LDAP Content Synchronization Operation* is defined as a set of controls and other protocol elements which extend the LDAP search operation. The operation allows a client to maintain a copy of a fragment of the Directory Information Tree (DIT) and it supports both polling for changes and listening for changes. Full details are provided in the experimental RFC4533 [215]. This operation is fully supported by OpenLDAP Directory Server v.2.3 and later. An overview of

the operation and the replication engine functionality is extracted here from OpenLDAP Administrator's Guide. Additional details can be found in [19] and in RFC4533 [215]:

The LDAP Sync replication engine, syncrepl for short, is *a consumer-side replication engine that enables the consumer LDAP server to maintain a shadow copy of a DIT fragment. It provides a stateful replication which supports both pull-based and push-based synchronisation and does not mandate the use of a history store*. Syncrepl supports both pull-based and push-based synchronisation. *In its basic refreshOnly synchronisation mode, the provider uses pull-based synchronisation where the consumer servers need not be tracked and no history information is maintained.* The information required for the provider to process periodic polling requests is contained in the request itself. *In its refreshAndPersist mode of synchronisation, the provider uses a push-based synchronisation. The provider keeps track of the consumer servers that have requested a persistent search and sends them necessary updates as the provider replication content gets modified.*

With syncrepl, *a consumer server can create a replica without changing the provider's configurations and without restarting the provider server*, if the consumer server has appropriate access privileges for the DIT fragment to be replicated. The consumer server can stop the replication also without the need for provider-side changes and restart. *Syncrepl supports both partial and sparse replications.* The shadow DIT fragment is defined by a general search criteria consisting of base, scope, filter, and attribute list. The replica content is also subject to the access privileges of the bind identity of the syncrepl replication connection.

Multi-master replication

A special replication feature of LDAP DS is known as Multi-Master Replication (MMR). Some initial concerns from OpenLDAP Foundation mentioning "MMR is considered harmful" have been resolved (IETF draft-zeilenga-ldup-harmful), therefore OpenLDAP DS supports MMR since version 2.4, as most of DS vendors (e.g. Fedora DS). OpenLDAP provides two implementation options for MMR:

1. N-Way Multi-Master replication uses *syncrepl* (Content Synchronization Operation) to replicate data to multiple Master Directory servers.
2. Mirror Mode replication is a hybrid configuration and is not strictly a Multi-Master solution since all write requests are forwarded to one of the mirror nodes at a time.

Obtaining an OID for LDAP Schema development

In [19] ("Chapter 6 LDAP Schemas",pp.265-348), the author describes the full procedures to create and deploy a custom Schema for LDAP Directory Servers. A Private Enterprise Number (PEN) or OID can be obtained from IANA, which also maintains a list with assigned OIDs. The

author of [19] suggests that "the practice of using someone else's OID is called OID hijacking, and is frowned upon because it compromises the assumption that OIDs are globally unique", therefore interested developers should register a new OID if their organisation does not have one already.

The prefix for Private Enterprise Numbers (SMI Network Management Private Enterprise Codes) is : iso.org.dod.internet.private.enterprise (1.3.6.1.4.1)

Additional LDAP Resources:

Books:

[19] M. Butcher, *Mastering OpenLDAP: Configuring, Securing and Integrating Directory Services.* PCKT Publishing, ISBN-10: 1847191029 UK, 2007.

[20] T.A.Howes, M.C.Smith, S.G.Gordon, *Understanding and deploying LDAP directory services*, 2nd ed., Addison-Wesley Professional, ISBN-10: 0672323168 , 2003

Websites:

Full list of current RFC related to LDAP	http://search.cpan.org/perldoc?Net::LDAP::RFC
Direct access to RFC WXYZ from IETF website	http://www.ietf.org/rfc/rfcWXYZ.txt http://tools.ietf.org/html/rfcWXYZ
Internet Assigned Numbers Authority	http://www.iana.org
List of Private Enterprise Numbers (PEN)	http://www.iana.org/assignments/enterprise-numbers
Application for OID or PEN	http://iana.org/cgi-bin/enterprise.pl
OpenLDAP 2.4 Administrator's Guide	http://www.openldap.org/doc/admin24
OpenLDAP 2.4 Multi-Master and syncrepl Replication	http://www.openldap.org/faq/data/cache/1240.html http://www.openldap.org/doc/admin24/replication.html
OpenLDAP 2.3 Sync Replication	http://www.openldap.org/doc/admin23/syncrepl.html