

On the use of Policies for Ambient Networks Management

Carlos Kamienski, Joseane Fidalgo, Djamel Sadok, Jennifer Lima, Leonardo Pereira

Universidade Federal de Pernambuco, Brazil

Börje Ohlman, Johan Nielsen

Ericsson Research, Sweden

E-mail: {cak, joseane, jamel, jennifer, lafp}@gprt.ufpe.br,

{Borje.Ohlman, johan.nielsen}@ericsson.com

Abstract

The automation of management activities for highly mobile and dynamic environments, such as those envisioned by Ambient Networks, is presently a challenge. Users need instant access to a variety of different services, including basic seamless connectivity until advanced services that rely on security, quality of service and context-awareness features. In the last years, policy-based approaches have been proposed as effective mechanisms for the management of large dynamic networks, such as the IETF policy framework. In this paper, we present the design and implementation of PBMAN, a Policy-based Management Framework for Ambient Networks, based on a Peer-to-Peer infrastructure, aimed at providing scalability and self-configuration features. A proof-of-concept prototype has been implemented and a simple scenario of a video-on-demand service has been deployed, with QoS and cryptography features. We present our early insights and results of working with policies for the management of Ambient Networks.

1 Introduction

Wireless networks are increasingly becoming more ubiquitous and providing universal access to the Internet, through the convergence with fixed networks. The result is that a complicated structure of several overlapping but non-interoperable technologies has been created. In addition, users want to be provided seamless services, from simple connectivity to new advanced services independent of the networking technology used. This is one of the key motivations for the recent proposal of Ambient Networks (AN) [5], that aim the

cooperation and interoperation of networks built over a variety of technologies and belonging to different operators.

It is important that all these new features are managed in an integrated and flexible way. Policy-based Management (PBM) seems to adequately address it because it is an approach for the administration of complex network infrastructures in a largely automated way. Policies can be used, for example, to manage and control the access to network resources by high-level abstract levels of rules and decisions.

The PBM framework developed by the IETF [8] is a model for policy management comprised of Policy Decision Points (PDPs, also known as policy servers), Policy Enforcement Points (PEPs), policy repository and Policy Management Tool (PMT). The PDP is responsible for handling requests, querying the policy repository, making decisions and distributing them to the PEPs, which are the entities (e.g. routers) where the actions are actually implemented and/or enforced. The PMT support the specification and administration of policies, through a graphical interface. These policies are then stored in the policy repository. Some protocols are necessary within this framework, such as COPS for PDP and PEP interworking and LDAP for the PDP to be able to access policies in the policy repository.

The IETF PBM framework was not designed having in mind heterogeneous scenarios where frequent changes are the rule, such as dealing with mobile and wireless users with highly dynamic usage patterns and unpredictable service needs. In this paper, we advocate that a new policy framework should be specially designed for coping with the challenges of Ambient Networks. There are three main

motivations for proposing a new framework for policy management. First, the IETF framework is focused on specific policy areas, such as QoS and security, and on simpler problems from typical corporate networks. We are broadening the scope of applications for policy management considering service usage in the global Internet. Second, the 3G/4G scenarios targeted by our proposal consider a huge number of mobile wireless users with highly dynamic mobility and service usage patterns. Traditional PBM does not provide a way to share information dynamically among policy domains. And third, for complying to important technical requirements associated with those environments, which are provided by the p2p technology, such as scalability, fault tolerance and load balancing.

In this paper, we present the design and implementation of PBMAN, a Policy-based Management Framework for Ambient Networks. PBMAN is based on a Peer-to-Peer (P2P) infrastructure, aimed at providing scalability and self-configuration features. A policy environment was specified, comprised of information and data models, a simple policy language and a policy interpreter. A proof-of-concept prototype has been implemented and a simple scenario of a video-on-demand (VoD) service was deployed. We present our early insights and results of working with policies for the management of ambient networks.

The rest of the paper is structured as follows. Section 1.1 summarizes the main concepts of Ambient Networks. Sections 2 and 3 present the PBMAN framework and prototype respectively. In section 4, a deployed real scenario of using PBMAN is depicted. Finally, section 6 draws some conclusion and topics for future work.

1.1 Ambient Networks

Ambient Networks (AN) is a new networking concept, which aims to enable the cooperation of heterogeneous networks belonging to different operator or technology domains [5]. This cooperation should be transparent and “plug-and-play”, i.e., no previous configuration or negotiation is required between network operators. Formally, an Ambient Network is a collection of networks and/or devices sharing a common control plane, called Ambient Control Space (ACS). The ACS is comprised of a collection of functional entities (FEs), each one

reflecting different control and management tasks, such as composition, mobility, security and QoS.

Network composition is the key architectural concept and the main challenge of Ambient Networks, aimed at enabling control-plane interworking and sharing of control functions among networks. Intuitively, composition can be thought of as a mechanism for automatic negotiation of roaming and/or service level agreements (SLAs), which today are done manually. When individual network completely merge and create a unique new ACS, the composition is called Network Integration. When networks partially merge during a composition, it is called Control Sharing, where only a subset of the services/resources of the constituent individual networks will be part of the new composed network. Composition also encompasses legacy (current) forms of interactions between networks (i.e., connectivity), which are classified as Network Interworking.

2 PBMAN Framework

PBMAN (Policy-based Management for Ambient Networks) is aimed at designing and implementing a management infrastructure for Ambient Networks. The technique adopted in PBMAN is Policy-based Management and the main underlying enabling technology is Peer-to-Peer. In fact, PBMAN itself is an instantiation of a more abstract framework, called P4MI (Peer-to-Peer Policy Management Infrastructure) [4]. A primary design principle adopted in PBMAN is to keep the architecture general and simple. As new experience is gained with designing and implementing the framework, new features and functionalities will be added.

The general PBMAN architecture (depicted in Figure 1) is focused on the role and implementation of the ACS on the various networks in an Ambient Networks scenario. In Figure 1 we can clearly identify three types of implementations for the ACS: PDN ACS, User ACS and PEP ACS. Both User and PEP ACS may be part of a combined Agent ACS.

The picture shows that both users and PEPs are considered special cases of small networks, as far as the framework is concerned. There are many users and PEPs that are connected to each

other and to the PDN by means of a wired network AN₁. On the other hand, AN₁ has also some wireless users connected to its PDN, who may be local users or visitors. AN₁ must transparently provide access and services to those users according to their profiles, regardless of whether they are wired or wireless, local or visitors. In order to be able to perform this task, AN₁ must exchange information with other ANs, and therefore composition and decomposition is required.

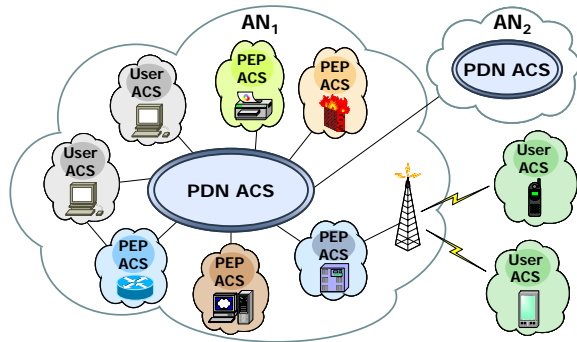


Figure 1 – PBMAN General Architecture

2.1 Policy Decision Network and Agents

The Policy Decision Network (PDN) is the heart of PBMAN, comprised of two main entities, Decision Points and Repositories. A Decision Point, also called PDN Node or P-Node, is a policy server, which accepts some part of the whole PDN work. There is a significant difference between a PDP in the IETF PBM client/server model and a PBMAN P-Node. The former is designed to interact with a set of PEPs and is not intrinsically aware of the existence of other PDPs, e.g., for load balance and fault tolerance purposes. The latter is able to interwork with other P-Nodes by design via a p2p network, based on Distributed Hash Tables (DHT) [1], called PDN-ring.

Repositories store policies according to some requirements, such as making easier the process of searching and retrieving them. PBMAN does not specify a particular storage technology, such as LDAP, as long as different implementations are able to interoperate. Also, there is a need for keeping information about entities associated to policies, such as user profiles.

Policy Agents represent hosts, equipments or devices used by users or by the network for providing services and enforcing policies. The interaction between agents and the PDN is

based on the hierarchical p2p DHT-based adopted approach. Agents may be comprised of two parts, which may be simultaneously present or not: PEPs and Users. PEPs are agents aimed at enforcing policies, such as routers, firewalls and remote access servers. PEP agents are also software and hardware for providing services, which must enforce policies of right of use, security, accounting, etc. Examples of this type of PEP are gaming and printing servers. Users represent devices or networks of connected devices that a given real user is using for accessing AN services.

2.2 Policy Information Model and Language

The information model (Figure 2) presents the abstract representation of managed entities and how they relate to each other. PBMAN information model comprises three main types of management entities: policies, targets and associations. Targets are entities which policies may be associated to, such as services or users (policy agents and targets). One important step in a policy management system is the association between policies and targets. Currently, PBMAN targets are: user, group, network (AN), service, service bundle and (technology) access class. Policy Sets are groups of policies, useful for organization purposes.

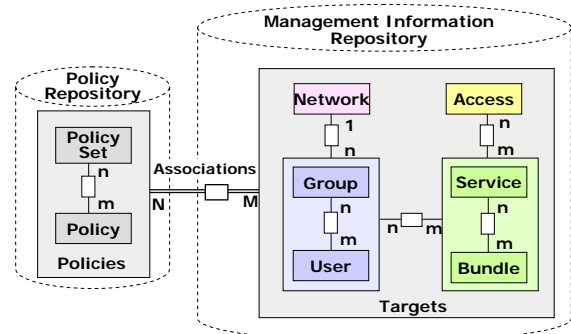


Figure 2 – PBMAN Information Model

This PBMAN information model has been used in our prototype and the video scenario of section 4 is based on a subset of it. We are not advocating this model as a final and general solution to ambient networks, since various other models may be suggested.

As far as scope is concerned, there are two types of services in PBMAN: local services and Well Defined Services (WDS). The former are used for organizing users into services (e.g.,

subscribers into plans for a provider) and is a key concept in PBMAN for allowing automatic service understanding and provision when a composition happens. The latter are used in order for guaranteeing that a common understanding about service semantics will be achieved by all participating networks.

Following the principle of simplicity adopted in PBMAN, we have chosen to design a simple policy language from scratch, called LPBMAN. Although the literature contains well-specified and complete policy languages, such as Ponder [3], after a careful analysis of the existing policy languages, we decided that a new language was the best option for PBMAN. The real challenge is how to deal with network composition and how to write policies for managing the whole process. Had an existing language been chosen, important extensions would be needed anyway. Using a new language has allowed us to not include more complexity than really needed.

2.3 Network Composition in PBMAN

Network composition is expected to be extensively used in ambient networks. Therefore, it should be performed as fast and efficiently as possible, in such a way to provide seamless services to users, as well as not imposing a heavy processing and communication burden on the management system. PBMAN general architecture is aimed at facilitating operations such as the merging of PDNs into a single one and dividing and joining separate PDNs.

By focusing on the goals of efficiency and performance, PBMAN identifies some different types of network compositions, in order to be able to optimize each one as much as possible. One single general composition process would be ideal. However, the efficiency of the overall management system would be far from ideal. Composition is currently classified in PBMAN according to two criteria: type of ACS involved and mobility pattern. Since in PBMAN there are two types of ACSs (PDN and Agent), three different types of composition are possible according to this criterion: PDN/PDN, PDN/Agent and Agent/Agent.

Figure 3 depicts a typical PDN/PDN composition. The situation before the composition is shown in Figure 3a. Both PDN_A and PDN_B are single PDNs, each one with four P-Nodes. During the composition, a new PDN

ring (PDN_{AB}) is created and two P-Nodes of each PDN are chosen as members of PDN_{AB} .

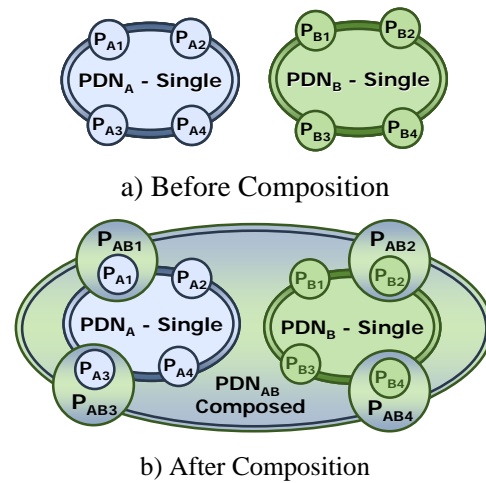


Figure 3 – Composition of PDN_A and PDN_B

An important aspect about PDN/PDN compositions is that they may take some considerable time to be performed¹, but they will typically happen only when the first user tries to access services of a remote network. This is necessary in order for the P-Node of a local network to be able to perform authentication and authorization based on the remote network policies, upon a user's request. For all subsequent accesses, the P-Node will have instant access to the remote network's information and the response therefore should be immediate.

Network composition requires several decisions to be made, which are controlled by policies in PBMAN. Composition policies may be used to: a) indicate conditions for accepting a composition request; b) determine composition model (integration, control sharing, network interworking); c) decide which services/resources to share; d) decide how to deal with conflicting policies; e) yield composition policies for the new composed network; f) decide which P-Nodes will take part of the new PDN; g) determine rules for decomposition, such as triggering event. A composition agreement in PBMAN is the set of all policies (user, support and composition), targets and their associations.

¹ As a matter of fact, we observed this long delay in our prototype.

3 X-PBMAN Prototype

We developed a simplified proof-of-concept prototype implementation of PBMAN, called X-PBMAN, implemented on top of the X-Peer [6] p2p middleware, described in the next section.

3.1 The X-Peer Middleware

X-Peer is a middleware designed and implemented for supporting P2P applications, based on a hierarchical architecture where super-peers (X-Peer nodes) communicate to each other through a DHT network. The main advantage of this proposal is the assurance of information location in a distributed and hierarchical network. Also, X-Peer is aimed at supporting various different P2P applications in a single middleware platform, thus differing from current solutions that usually use a new network for each new application. The current implementation of X-Peer is based on the Pastry DHT [7].

In the X-Peer architecture (Figure 4) there are three levels of communications: X-Peer nodes communicate through DHT; Applications communicate directly to each other via a simple P2P protocol; applications communicate to X-Peers by means of a specific protocol.

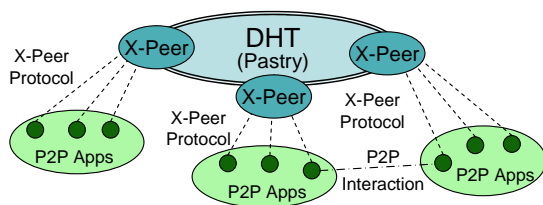


Figure 4 – X-Peer Architecture

This middleware is sufficiently flexible in that it can be configured to run as centralized system when using a single super node all the way to a fully distributed P2P system when an X-Peer node is co-located with each peer application. In other words, adding a new X-Peer node is very simple, so that load-balancing, fault-tolerance and scalability can be easily achieved.

This characteristic of the X-Peer architecture makes it easy to use for implementing the PBMAN framework. Policy agents can be mapped to applications, whereas P-Nodes can be mapped to X-Peer nodes.

3.2 Software Architecture

Figure 5 depicts the P-Node architecture designed and implemented for the X-PBMAN prototype, which performs some of the functionalities of the PDN ACS. The picture also contains some external entities a P-Node has to interact with. The current version of the architecture is comprised of six modules, representing the Functional Entities (FE) of the ACS. Each FE is implemented by a specific software module of X-PBMAN (name inside brackets). Some of these modules run in separate address spaces (processes) whereas others are bundled together with the main system.

The P2P FE is the core function of the P-Node, implemented directly by the X-Peer node, taking care of DHT-based policy location, routing, search and retrieval. The Composition FE is in charge of managing PDN/PDN composition in PBMAN. This module is called X-Peer Multi-ring Manager (X-MM), because a composition implies in creating a new PDN ring (DHT network) and managing two or more rings simultaneously. The Policy FE is the entry point for service requests and is responsible for processing policies to be enforced by PEPs. It is called X-Peer Policy Processing Module (X-PP)

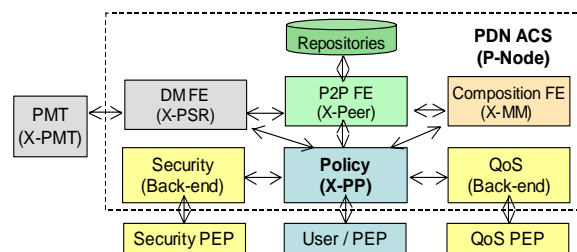


Figure 5 - P-Node Architecture

The DM FE implements a layer that extends the data storage capabilities of the DHT-network, in order to make it able to deal with more complex data structures, such as lists and tables. A known limitation of DHT-based systems is that they only support exact-match lookups. The QoS FE and Security FE are part of the policy system back-end, whereas the QoS and security PEPs are specific modules usually running on network devices in charge of configuring support services. The Repositories are implemented by the basic X-Peer storage method. The PMT (Policy Management Tool) provides functionalities for editing policies, management information and its relationships.

4 PBMAN Scenario: Video Service

A video on demand (VoD) service was chosen as the scenario for analyzing the effectiveness of the PBMAN approach. This scenario has been implemented, modeled, deployed and tested using the X-PBMAN prototype. It is comprised of two not directly attached networks, AN₁ and AN₂, both providers of the video service. Users subscribed to the video service in one network can watch videos from the other network, as long as they are composed.

There are three Well-Defined Services (WDS): video, QoS and cryptography services. Commercially, the combination of them may be sold by network providers in four different service levels, i.e., video with or without QoS and cryptography. For instance, services may be called Basic Video, Premium Video (with QoS), Secure Video and Secure Premium Video. Policies control the rights of accessing the video service and the configuration of QoS and security functions.

4.1 Video Service Transaction

Figure 6 depicts a typical transaction for the Premium Video Service (with QoS), beginning when the user accesses the web page of the video provider until he/she actually gets the video stream. Entities responsible for security (cryptography) functions are not represented in the picture, in order to improve its understandability. A typical video transaction when a remote user of AN₁ accesses the video server in AN₂ is made up of twelve steps:

1. The user accesses the VoD provider web page.
2. The request is forwarded to the video PEP.
3. The video PEP sends a request to the PDN.
4. PDN₂ detects the user is subscribed to AN₁ and it negotiates a composition with PDN₁.
5. In case the negotiation is successful, PDN₂ and PDN₁ proceed to creating a new PDN ring and republishing all relevant information.
6. The PDN performs a policy selection process and the Premium VoD service is granted.
7. The PDN sends a response to the video PEP.
8. The PDN tells the QoS PEP for marking packets of to this video stream to an appropriate QoS class.

9. The video PEP responds to the web server that resumes its pending session with the web client.
10. The web server presents a list of videos.
11. Upon selecting a video channel, the web client automatically launches the video client.
12. The video client contacts the video server and the video streaming session is initiated.

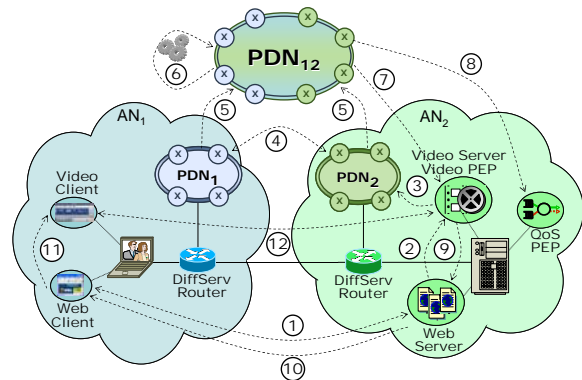


Figure 6 – Video Service Transaction

A more complex scenario would be that of a mobile user that is roaming in a third provider. This would require an Agent/PDN composition for providing the basic connectivity service to the user – this scenario is left as future work.

4.2 Scenario Modeling

This scenario was modeled using the PBMAN policy information model and policies written in the LPBMAN language (section 2.2). However, for the sake of simplicity (and space constraints) only a subset of the model, for the Premium Video service, is presented in this section. The following tables contain all targets, associations and policies for offering this service on both networks.

Table 1 contains a list of selected targets for AN₁, identified by *eov*. Two WDSs are needed, video and QoS, identified as such by the @WDS suffix. In AN₁, target Group is used for representing user subscription to a given service. Both group Executive and user José are local to AN₁, identified by the *jose@eov*.

Table 1 – AN₁ Selected Targets

Type	Name	Identifier
Network	Extreme On-line Video	eov
Service	Video Service	video@WDS
	QoS Service	qos@WDS
Group	Executive	exec@eov
User	José	jose@eov

Table 2 shows a set of associations between targets in AN₁. Group Executive is associated to both Video and QoS services, which means that all users associated to this group will be subscribed to Premium Video (including José). José will be able to use Premium Video in AN₂, as long as networks are composed and AN₂ offers Video and QoS services.

Table 2 – Selected target associations in AN₁

Type (A→B)	Target A	Target B
Group→Service	Executive	Video Service QoS Service
User→Group	José	Executive

Table 3 shows three selected policies for AN₁, involved in providing the Premium Video service. Policies p1@eov and p3@eov are associated to the video service, whereas policy p2 is associated to the QoS service. Policy p1@eov specifies that if the service is video@WDS and the requesting user belongs to the Executive group (**ct** means “contains”), then two actions are executed. First, additional policies associated to qos@WDS are retrieved from the PDN, which is policy p2@eov. Second, the video PEP is informed that the user was granted access to Premium Video. In this example, the string “service=premium_video” is sent directly to the video PEP, due to simplicity reasons. However, dealing with protocol response messages is a tradeoff between complexity in the policy specification, in the policy FE parser or in the PEP.

Table 3 – Selected AN₁ policies

<p>policy p1@eov set 1 by User if (service == video@WDS; usergroup ct exec@eov) use qos.service=qos@WDS; qos.action=configure do return(“service=premium_video”)</p>
<p>policy p2@eov set 1 by QoS if (qos.service == qos@WDS; qos.action == configure) set qos_peg=[map QOSPEP(req_peg_id)] do configure_qos_peg (qos_peg, client_addr)</p>
<p>policy p3@eov set 0 by User if (service == video@WDS) then do return (“service = deny”)</p>

Policy p2@eov is valid when service is qos@WDS and action is “configure”. In that case, the QoS PEP address associated to the video PEP is retrieved from the PDN and a backend command for provisioning QoS is executed. The structure [map

QOSPEP(req_peg_id)] means that a mapping information associated to the constant QOSPEP (previously stored) with the parameter req_peg_id is retrieved from the PDN.

Policy p3@eov is needed for denying access to non-subscribed users. In other words, when processing a request, policies p1@eov and p3@eov will be selected because both are associated to the video service. In case the user is associated to the Executive group, policy p1 will be executed, due to its higher priority (set 1). Otherwise, policy p3 will be executed.

Table 4 contains a list of selected targets for AN₂, identified by svod. Similar to AN₁, there are two WDSs defined, video and QoS. Notice that AN₂ does not use a group for associating users to this service, but a local service called Premium Video (pv@svod). User Karin is associated to this service.

Table 4 – AN₂ Selected Targets

Type	Name	Identifier
Network	Super Video-on-Demand	svod
Service	Video Service	video@WDS
	QoS Service	qos@WDS
	Premium Video	pv@svod
User	Karin	karin@svod

The set of policies needed for AN₂ are not shown, because they are similar to those defined for AN₁². Two important differences are worth mentioning. First, instead of writing “**usergroup ct exec@eov**” in the condition of policy p1@eov, AN₂ uses “**userservice ct pv@svod**”. Second, the backend method for configuring the QoS PEP has a different name than “configure_qos_peg” of p2@eov, because it refers to a local implementation that may differ among networks.

Currently, X-PBMAN prototype implements only the network integration composition type, which means that all information (policies, targets and associations) will take part of the new composed network. Furthermore, all P-Nodes of a given PDN take part of the new created PDN. Therefore, when AN₁ user José accesses the video service in AN₂ (or conversely when Karin accesses AN₁) during the step 6 of section 4.1 the Policy FE will be able to find all information needed for evaluating and taking decisions on

² At least the policies we have written for this scenario are similar for both networks. However, there are different ways of modeling and writing policies in PBMAN.

that request. Since both video and qos services are WDSs, they will be understood by both networks. Consider the case when policy p1@eov is executed by the composed network upon José's request. When interpreting keyword "use" (that indicates a new policy processing cycle), policies from AN₁ and AN₂ associated to QoS@WDS will be selected, but only pollicy p2@eov will be executed, since it contains a local backend command.

5 Related Work

Since the concept of Ambient Networks is very new, there is almost no related work in this area. The most close to PBMAN is PAP (P2P ACS Prototype) [2], which has been developed to demonstrate network management concepts, including policy based network composition and hierarchical management of P2P overlays. However, compared to PBMAN, PAP may be seen as a complementary work, since it addresses only network level composition among wireless users for obtaining basic connectivity. PBMAN addresses mainly the composition of more stable networks (PDN/PDN) and virtual compositions of agents (Agent/Agent).

6 Conclusions

In this paper we presented PBMAN, a Policy-based Management framework for Ambient Networks based on the P2P technology. Also, we presented the design and implementation of X-PBMAN, a proof-of-concept prototype implementation of PBMAN. At last, we modeled a scenario according to the PBMAN framework and implemented in the prototype.

Purposefully, our approach took two important directions. First, we did not adopt existing policy languages, due to their unwanted complexity and lack of flexibility to deal with novel challenges of Ambient Networks. Second, we adopted a spiral methodology (framework-prototype-framework), for being able to considerably improve the design of the framework based on experience. We avoided going through specifying a comprehensive and complex framework and only implementing it afterwards. We have been able to keep the framework as simple as possible, while fulfilling all theoretical and practical requirements identified so far.

As we expected, network composition is the main complication for policy writing and processing due to the requirements of strong cooperation from different management domains. The key challenge is to understand and fulfill the needs of remote users as they were local ones and implement it in an efficient way.

As future work we intend to keep improving both the framework and prototype implementation, including deploying scenarios that address specific needs of wireless users. Also, we expect to improve the information model, the policy language and its processing algorithm, as experience is gained with more complex scenarios.

7 Acknowledgements

This work was supported by the Research and Development Centre, Ericsson Telecomunicações S.A., Brazil.

8 References

- [1] Balakrishnan, H., et al., "Looking Up Data in P2P Systems", CACM, February 2003.
- [2] Brunner, M., "Ambient Network Management – Solution Design and Functions", Ambient Networks Project Deliverable D 8.2, June 2005.
- [3] Damianou, N., et al., "The Ponder Specification Language", Workshop on Policies for Distributed Systems and Networks, January 2001.
- [4] Kamienski, C., et al., "On the Use of Peer-to-Peer Architectures for the Management of Highly Dynamic Environments", 3rd Workshop on Mobile Peer-to-Peer Systems (MP2P'06), March 2006.
- [5] Niebert N. et al., "Ambient Networks: An Architecture for Communication Networks Beyond 3G", IEEE Wireless Communications, April 2004.
- [6] Rocha Jr., J., et al., "X-Peer: A Middleware for Peer-to-Peer Applications", 1st Brazilian Workshop on Peer-to-Peer, May 2005, (in Portuguese).
- [7] Rowstron, A., et al., "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", 18th IFIP/ACM Intl. Conf. on Distributed Systems Platforms, October 2001.
- [8] Yavatkar, R., Pendarakis, D. & Guerin, R., "A Framework for Policy Based Admission Control," RFC 2753, January 2000.