An Architecture for Autonomic Management of Ambient Networks

Marcos A. Siqueira[1,2], Fabio L. Verdi[2], Rafael Pasquini[2],
Mauricio F. Magalhães[2]

[1] CPqD Telecommunications R&D Center,
Rod. Campinas - Mogi-Mirim, km 118,5, 13086-902 Campinas, SP Brazil,
[2] State University of Campinas, UNICAMP, FEEC, DCA,
PO Box 6101, 13083-970 Campinas, SP Brazil
{siqueira,verdi,pasquini,mauricio}@dca.fee.unicamp.br

Abstract. This paper proposes an architecture for Ambient Networks (AN) management, which is based on components such as Autonomic Networking and Policy Based Network Management. The main goal is to allow AN management aligned to the network requirements, incorporating functionalities such as self-configuration, self-management, self-healing and self-protection. In the proposed architecture, all these functionalities are controlled by a distributed policy-based system, allowing network configuration based on business policies.

1 Introduction

With the advent of wireless technologies such as IEEE 802.11, 802.16, cellular 3G, 4G (all IP), the connection of users to networks and even the connections among different networks tend to be dynamic. An example of such scenario, presented in the context of the Ambient Networks (AN) project [AN06], is a train of passengers operating an internal WLAN (Wireless Local Area Network), communicating to the external world via wireless access networks available through the railway. The train might have different kinds of wireless interfaces, connecting to the better network available at each moment, maintaining the passengers connections. Therefore, the train has to negotiate parameters such as the connection conditions, even without having a previous knowledge of the resources provided by other networks.

These kinds of networks become very difficult to be operated by humans without the use of automated management tools, due to the quantity of protocols and logical entities that must be configured, the amount of interconnections and its dynamicity, the great quantity and widespread of network nodes, besides the heterogeneity of technologies and types of equipments from many vendors. This heterogeneity generates greater complexity in network operation, resulting in increase of OPEX (OPerational EXpenditures) since specialized engineering, operation and support teams are required for each solution employed.

It has been argued by the academy and some standardization bodies that network management automatization is the solution to: operation cost reduction,

resources optimization, security control, and mainly the responsiveness in troubleshooting. Besides, the application of the Autonomic Networking [Str05] concept hides the network complexity, by means of self-management, self-optimization, self-defense and self-configuration, in a manner aligned to the end business. For complete automatization of operation tasks, network management tools might act as integrated parts of the enterprise process model. As an example, the CRM (Customer Relationship Management) might be integrated to the services activation tools, that should be integrated to the NMS (Network Management System) allowing automatic services provisioning according to the CRM commands generated by client requests or others.

This paper discusses how the autonomic networking can help in the solution of the problems presented in the previous paragraphs, through the automatization and consequent simplification of network operation. The proposal is that the concept of network policies should be used to regulate the autonomic network operation. Specifically, for validating the proposed approach, it is proposed a distributed architecture that supports autonomic management and operation of Ambient Networks. For validating the proposal, an infra-structure of distributed policy agents is implemented using *web services*, which are very suitable for overlay networks, allowing automatic discovery of resources and open communication, both required by Ambient Networks.

The rest of this paper is organized as follows: Section 2 presents the concept of Ambient Networks and its requirements, Section 3 presents the concept of Autonomic Networks, Section 4 presents a brief about Policy-Based Network Management, Section 5 presents an architecture for autonomic management of ambient networks, describing its functional elements and the strategies for the system implementation in a distributed way, finally Section 6 presents the conclusion and future works.

2 Ambient Networks

Ambient Networks (AN) [AN06] is a project aiming at investigating future communication systems. One of the main goals is to allow dynamic composition of heterogeneous networks from different operators and/or different technological domains transparently to the users and services.

Nowadays, Internet Protocol is the *lingua franca* to enable information exchange through networks. Nevertheless, there is divergence in the network control layer, since different mechanisms are needed, for instance, to implement VPNs, security, QoS, Multicast, and others. This diversity of control mechanisms is a barrier to reach the level of integration desired in the Ambient Networks. Therefore, ANs present as challenge, the definition of universal essential control functions aiming at reaching a scenario that allows dynamic network composition and enhanced mobility control. These functions might take into account heterogeneous environments with different network techonologies and services as well as varied network management functions. A conceptual framework, formed by three basic principles was defined by [SEPP05]. These principles are:

1. **ANs must be built over open platforms:** the goal is to eliminate architectural constraints about what or who might connect to what or who, enabling network services not only for nodes, but also to whole networks. This strategy surpasses the limitations inherent to node centric architectures, mainly in scenarios with PANs (Personal Area Networks), mobile networks, ad hoc networks, connected to one another.
2. **ANs are based on self-composition and self-management:** the composition of networks so that packets could be forwarded would be simple, but the composition of networks in a managed way, maintaining advanced functions such as QoS, security and mobility is a complex task. The goal is ANs to have self-composition and self-management functions as basic premise.
3. **AN Functions can be added to existent networks:** the philosophy of ANs is not to create a totally new network, but aggregate the wished functionalities to current networks, allowing them to integrate not only at the packet forwarding level, but also at the control plane, enabling the operation of services end-to-end through the cooperation of the different networks.

The main concept defined by the AN project is the ACS (Ambient Control Space), which manages the functionalities related to control and data transport through a set of interfaces for services and applications. The ACS is composed by a number of interdependent complex control functions that are being developed in the context of the AN project. Examples of these functions are the support to multi-radio accesses, mobility, security, management and smart media routing context management.

An Ambient Network has well defined control interfaces to other ANs, service platforms and applications. Aiming at allowing cooperation between different ANs, it is defined the ANI (Ambient Network Interface), which allows the connection of the ACS functions of a given AN to the ACS of connected ANs. On the other hand, the access to the services of a given AN is performed via ASI (Ambient Services Interface). Together, the ANI and ASI might support: plug and play connection and composition of ANs, network reconfigurations, support to mobile networks and a single interface to external entities, even in the case of cooperation between two or more ANs. Figure 1 shows the architecture defined for Ambient Networks, including the Ambient Control Space, ANI and ASI interfaces.

2.1 Ambient Network Management

Ambient Networks require management systems able to act in a dynamic and automatic way, allowing the networks to perform tasks such as auto-composition without the need of human intervention. To be able to perform these tasks, AN management systems might be dynamic and auto-managed, acting actively in the network. Other important requirement presented by [NGA+04] is the ability to connect the management systems of two ANs consistently in the occurrence of AN composition, or even in the event of separation of these systems in a predictable and consistent way. Among the techniques already proposed for AN management, the following are worth listing:

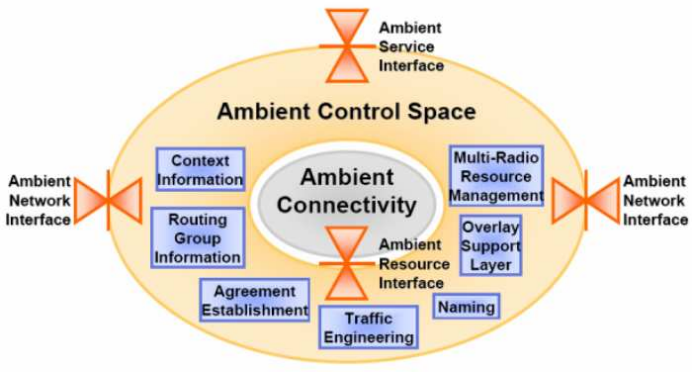

Fig. 1. *Ambient Network Architecture*

1. **Point-to-point model:** it is a distributed management model in which the main characteristic is the possibility of interaction among peer elements or networks, allowing for instance, dynamic composition and decomposition of networks.
2. **Pattern-based model:** the paradigm of pattern-based management deals with algorithms for processing and aggregation of management information in a distributed system. A key feature is the separation of the tasks semantics from the management operations (management control flow).
3. **Plug and play model:** the main objective of this model is to enable automatic configuration of new network elements, allowing them to join an AN with support to the domain features. Other strategies include application of traffic engineering for resources optimization in the AN, besides integration of management strategies.

The next section presents Autonomic Networking scheme as a possible architecture for distributed, policy-based management of Ambient Networks.

3 Autonomic Networking

When the idea of Autonomic Computing (AC), introduced by IBM in 2001 [KC03] is applied at the level of computer networks, the result is the concept of Autonomic Networking. The application at the network level of the autonomic systems requirements such as self-management, self-configuration, self-optimization, self-healing and self-protection enables drastic simplification and automatization of the network operation. Therefore, is perceived that the concept of Autonomic Networking is a superset of the Policy-Based Network Management (PBNM) concept. While PBNM is focused in network auto-configuration based on certain conditions, the concept of Autonomic Networking includes broader functions such as self-protection, self-optimization and self-healing. Fol-

lowing, some possible solutions for the implementation of the features needed for the construction of an autonomic network are discussed:

1. **Self-management:** it is a desirable feature, but it is not commonly available in current network management tools. Self-management plays a very important role due to the complexity and distributed nature of autonomic systems. Self-management of autonomic systems is an issue to be more deeply investigated by the academy.
2. **Self-configuration:** a possible way to perform self-configuration in networks is through the application of the PBNM concept, in the sense that network events and conditions can be determinant to trigger automatic reconfiguration of elements, ruled by pre-configured policies. The following references discuss about the auto-configuration of autonomic systems using policies: [KY03] proposes a language named JSPoon aiming at integrating autonomic functions in autonomic agents which might be installed at the systems to be managed; [BTBR04] proposes an autonomic middleware for control of the service used to orchestrate distributed and self-healing applications; and [BGJ+04] proposes a policy-based framework for managing autonomic database systems based on business goals, allowing automatic data management based on events.
3. **Self-optimization:** network optimization is an issue in continuous research by the academy. The most relevant initiative is the deployment of traffic engineering, facilitated by MPLS (Multiprotocol Label Switching) [DOA02]. Notwithstanding, the network optimization can be reached through traffic control and congestion prevention. Specifically, [Aib04] presents a new paradigm focused in self-optimization according to high level business goals, such as profit maximization. This paradigm replaces traditional optimization schemes, based on metrics related to IP as resources availability. This work proposes an autonomic process responsible for such optimizations, and a set of logical entities and its communication interfaces to implement this process.
4. **Self-healing:** it is evaluated that self-healing is the more difficult feature to be implemented in the point of view of a computer system. Systems can be designed with some degree of redundancy being, therefore, fault tolerant. Nevertheless, self-healing of failed components might not be performed automatically without human intervention.
   Currently, mechanisms used for fault recovery in networks are very developed. In the core layer, [LRP05] presents a functional description of the extensions needed by GMPLS (Generalized Multi-Protocol Label Switching) for adding the functionalities of protection and restoration. The IETF working group BFD (Bidirectional Forwarding Detection) is in period of specification of a protocol (named BFD) for fast detection of failures in links. RFC4090 [PSA05] specifies a mechanism for fast reroute of LSPs (Label Switched Paths) within MPLS networks. In the distribution layer, metro network technologies also support several mechanisms for fault tolerance, comprising: RPR (Resilient Packet Ring) [RPR06], RSTP (Rapid Spanning

Tree Protocol) [RST] and EAPS (Ethernet Automatic Protection Switching) [SY03]. In the access layer, the following mechanisms are suitable for fault tolerance: RF diversity, dial backup with DDR (Dial on Demand Routing), access link redundancy, CPE router duplication, protocols for redundancy such as VRRP (Virtual Router Redundancy Protocol) [KWW+98] and GLBP (Gateway Load Balancing Protocol) [GLB06].

5. **Self-protection:** the vendor Cisco Systems launched recently a bundle named "Cisco Self-Defending Network" [SD06]. The promise is that the implementation of this strategy allows the network to identify, prevent and adapt to threats. The first phase of the project includes integration of security features to network elements including routers, switches, wireless access points, and others network appliances. The second phase includes a mechanism named NAC (Network Admission Control) which allows network elements enabled for security to intercommunicate in a collaborative way, allowing security functions to be extended to equipments of individuals which eventually connect to other networks bringing risk to the corporate network.
   Adaptive defense from threats is the last phase, which helps to minimize security risks, dealing with threats dynamically in multiple layers, allowing a more strict control of traffic, endpoints, users and applications. This last strategy aims at protecting every packet and flow that pass throughout the network. Therefore, it is noted that the initiative Self-Defending Network contributes to the self-protection feature wished for autonomic networks.

4 Policy Based Network Management

In the last years, it has been developed architectures, information models and policy protocols aiming at reaching the goals of PBNM systems, such as mapping high level business rules into network configurations, according to the occurrence of a set of events, and even the analysis of applicable conditions.

Study areas of PBNM systems can be classified into: policy languages, policy information models, PBNM architectures, conflict detection and resolution strategies, policy protocols and policy edition strategies. Many works related to PBNM have been published. Works related to policy languages include [LSDD00], [AMN02] and [LK03]; works related to policy information models include [Str02], [MESW01] and [SMFdC04], works related to conflict detection and resolution include [LS99] and [Dun02]; works related to PBNM architectures include [SMC02] and [SNRLM05]. The references listed above are a small sample of the number of academic works related to PBNM systems. As a result to a simple search for papers at the IEEE repository, more than 50 papers related to application of PBNM in specific technologies or scenarios can be found.

From the commercial systems perspective, vendors of network management tols such as Hewlett-Packard, Micromuse, Extreme Networks, Cisco Systems, have launched their own PBNM modules, coupled to the NMS systems, mainly for specific goals such as policy-based QoS and security management.

The next section presents a strategy for managing and controlling Ambient Networks, employing Autonomic Networking and PNNM strategies.

5 Autonomic Management of Ambient Networks

This work proposes an approach for Ambient Networks management, applying Autonomic Networking concepts aiming at implementing support for mobility, network auto-composition, distributed management and other control functionalities. In order to develop the system architecture, the following premises are assumed: (1) The operation scenario will be a set of Ambient Networks, part of them nested, part of them intercommunicating through ANI and ASI interfaces. These ANs might compose or decompose with other ANs at any moment, and also might accept the connection of new nodes with full support to autoconfiguration; (2) The network devices might support a common GLL (Generic Link Layer) in order to allow connectivity at the link layer level.

The architecture proposed has the following main features: (1) The ACS has an Autonomic Management Layer (AML) which might be distributed through one or more ANs. The control messages are exchanged throughout the ANI, allowing seamless connection of two or more ANs in the management point of view; (2) The AML might be implemented totally distributed through agents installed at the network elements, providing support to communication with other ANs, allowing establishment of AN control functions; (3) The AML might operate automatically, without the need of human intervention in runtime, nevertheless being possible to perform tuning in the configurations that affect the behavior of the AN; (4) The operation of the AML might be policy-based, preferably being ruled by high level business policies.

Figure 2 presents a scenario composed by four ANs, in which AN1, AN2 and AN3 are composed, interconnected by ANI and ASI interfaces, while AN4 is in composition with AN2. According to the figure, each AN might have or not a management element named APS (Autonomic Policy Server). In the example shown, only AN2 doesn't have a APS. Each existent APS can instantiate APAs (Autonomic Policy Agents) and prosecute some control over them in the scope of one AN or other connected AN, if the last doesn't have its own APS. In this scenario, the proposed architecture requires the occurrence of information exchange among APAs in the same AN and among APAs of different ANs through the ANI interface. As far as AN2 doesn't have a APS, the agents APA2a, APA2b and APA2c are instantiated and partially controlled by the APSs of the connected ANs.

Comparing the presented network and management architecture with common PBNM architectures, is noted that the function of PDP (Policy Decision Point) is performed in a distributed way by the Autonomic Policy Servers and by the Autonomic Policy Agents. On the other hand, the PEP (Policy Enforcement Point) functions are performed only by the Autonomic Policy Agents. The following subsections detail the functions of the APS and APA.

5.1 Autonomic Policy Agent

The APAs might be instantiated manually by the network operator or by the APS, allowing the AN to manage its resources in a distributed and autonomic

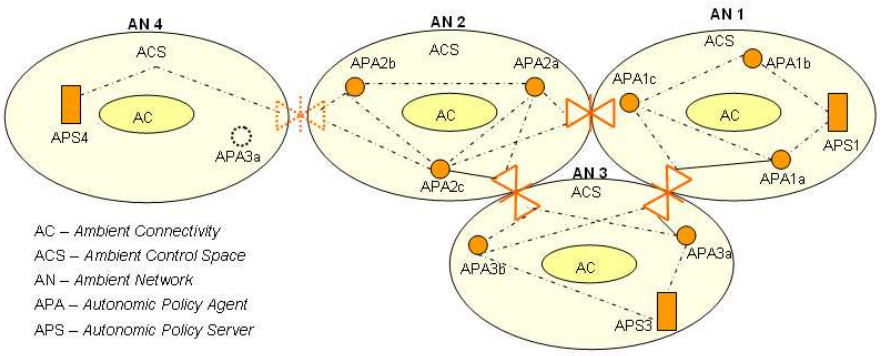

Fig. 2. Ambient Network Distributed Management

fashion. The APAs of a given domain will always establish logical communication channels for exchanging control messages. The logical paths might be configured with a fault tolerant strategy, as seen in Figure 2, where in the AN1 network, each APA has at least two logical routes to every other APA or APS in the domain, or to other ANs connected via ANI.

Each APA is responsible for performing the following tasks: (1) Auto-configuration of network elements by which the given APA is responsible; (2) Take local policy decisions about configurations, and adjusts to be performed at the network elements, mostly related to QoS and mobility control; (3) Participate actively in authentication and authorization of new nodes and new ANs trying to join the network, enhancing scalability and security of the whole system; (4) Keep neighbor relationship and logical control paths to other APAs and the APS; and (5) Have the ability to maintain operation even in the occurrence of failure in communication with the APS.

5.2 Autonomic Policy Server

The APS is responsible for: (1) Keep a centralized copy of the whole policy repository of its respective domain of control, formed by one or more ANs; (2) Instantiate or activate APAs in a given AN aiming at optimizing the management functions of this AN; (3) Receive requests from APAs for decisions about complex situations or conditions, which require a broader vision than is available at the APA level; and (4) Provide an interface to the network administrator for entering policies at the whole system.

5.3 Policy Construction

In the context of this project, it is designed an advanced mechanism for interpretation and execution of policies. The policy engine allows for complex policy

structure, modeled not only by traditional ECA (Event-Condition-Action), but also considering the concept of situation, introduced by [AE02]. The policy engine designed for the PDP is shown in Figure 3.

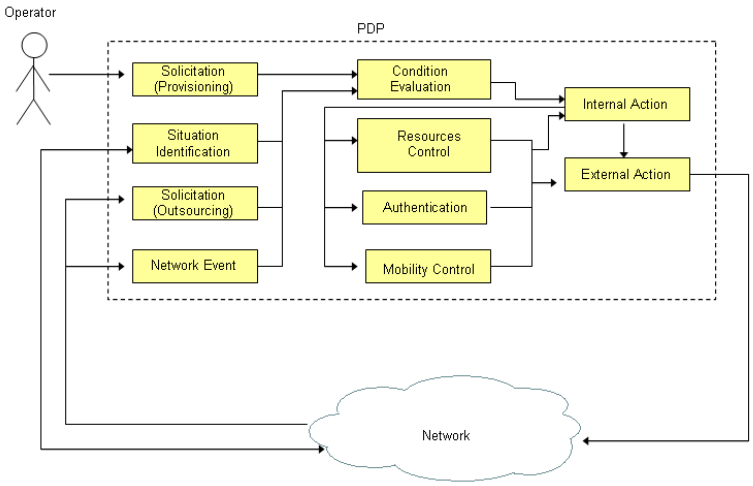

Fig. 3. Policy evaluation and execution flow

The PDP is formed by three main stages. The first one is responsible for receiving solicitations, events or for identifying a situation. The second stage is responsible for analyzing conditions, authentication, analysis of parameters related to mobility control, resources management (eg.: available bandwidth). The third stage is responsible for performing internal and external actions. Internal actions are executed when the analysis of further conditions is needed and external actions consist of performing specific configurations in the network elements.

5.4 Distributed Policy Engine Implementation

Figure 3 has shown the policy engine in a centralized fashion. Nevertheless, the AN architecture requires a distributed management system that suits its requirements. Therefore, the components of the policy engine were designed as independent Policy Agents, communicating to each other through a *web services* infrastructure. The use of *web services* as integration technology between different applications facilitates the development of future systems. It has been argued that distributed technologies such as CORBA, DCOM, RMI and others already exists and could be employed, but all of them adopt distribution mechanisms strongly coupled, synchronous communication and weak interoperability among different technologies. On the other hand, using XML and HTTP, *web services*

tend to be weakly coupled. Besides, *web services* facilitate the development of business solutions among enterprises due their ability for publishing enterprise solutions as services available throughout the network.

The distributed implementation was designed using the concept of policy flow. The APS is responsible for throwing the policy activation. According to Figure 04, the first step performed by the APS is determining the set o APA that will take part of the ring, creating the logical paths between them, then in step 02 the policy is distributed throughout these agents and activated. From this moment on any of these agents is responsible for triggering policy analysis and execution in the occurrence of a given event, solicitation, or identification of a situation. The policy flow is comprised by each agent running a set of methods of the policy and sending the policy to the next agent in the policy ring. Each agent receives a policy in a given state of the evaluation, indicating the next method to be executed. The local agent performs the tasks related to that method and dispatches the executing policy to the next agent, according to the decisions made. If a given agent doesn't know for any reason where to send the policy, it appeals to the APS which has a global view of the policy flow, and is able to perform the needed tasks of evaluating and executing the policy. The APS is published as a *web service* so that the PAs can find its available resources.

In the step 03 shown in Figure 4 a new Policy Agent appears. In this case, a situation is identified by the policy configured at the Policy Agent APA1, which carries out the process of policy evaluation, performing condition evaluation, which results in the execution of an internal action. The internal action delegates the authentication of the new APA to the APS. Then, the APS performs step 04 shown, authenticating APA3, and reconfiguring the policy ring with the new agent. The example presented is very simple, but the policy engine shown in Figure 4 allows the deployment of complex policies by the system.

6 Conclusion and Future Works

This paper presented a proposal of architecture for Ambient Networks management. As a new approach, Autonomic Networking and Policy-Based Network Management concepts are employed forming the base of the proposed architecture. The paper presented the requirements and concepts of Ambient Networks, discussing about propositions available at the literature and available tools that can fit these requirements. Finally, the implementation architecture of the distributed autonomic policy-based management system is described.

The work described in this paper is part of the author's Doctorate thesis [Siq06]. As future works, is proposed to be explored the strategies for APS discovery, algorithms for APA instantiation, strategies for creation and maintenance of logical paths between PAs in a domain and inter-domains, and the refinement of the policy flow. Finally, the system will be validated over a real testbed AN network.

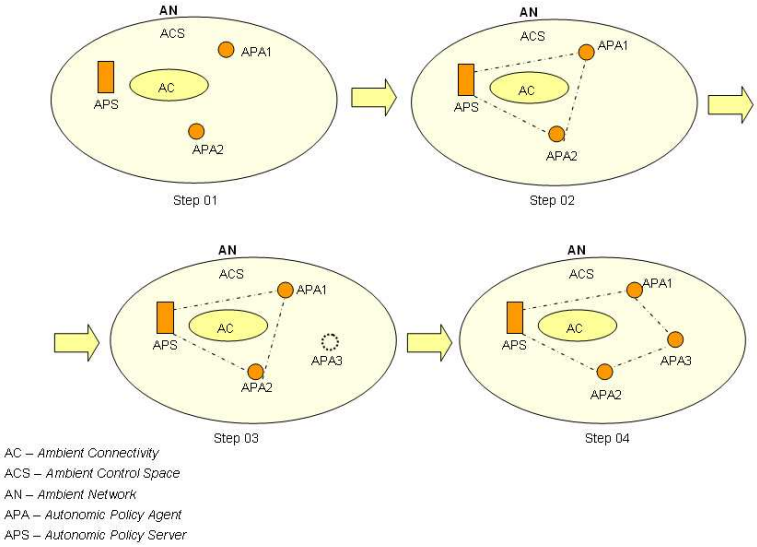

Fig. 4. Sample of the policy flow operation

References

[AE02] A. Adi and O. Etzion. The Situation Manager Rule Language. In *Proceedings International Workshop on Rule Markup Languages for Business*, 2002.

[Aib04] S. Aiber. Autonomic Self-Optimization According to Business Objectives. In *Proceedings of the International Conference on Autonomic Computing (ICAC04)*. IEEE, 2004.

[AMN02] X. Ao, N. Minsky, and T. D. Nguyen. A Hierarchical Policy Specification Language and Enforcement Mechanism, for Governing Digital Enterprises. In *Proc. of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, June 2002.

[AN06] AN. Project: EU FP6 Program Ambient Networks, April 2006.

[BGJ+04] M. Bhide, A. Gupta, M. Joshi, M. Mohania, and S. Raman. Policy Framework for Autonomic Data Management. In *Proceedings of the International Conference on Autonomic Computing (ICAC'04)*. IEEE, 2004.

[BTBR04] N. Badr, A. Taleb-Bendiab, and D. Reilly. Policy-Based Autonomic Control Service. In *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY04)*. IEEE, 2004.

[DOA02] B. Jabbari D. O. Awduche. Internet Traffic Engineering Using Multiprotocol Label Switching (MPLS). *Journal of Computer Networks (Elsevier Science)*, 40(1), September 2002.

[Dun02] N. Dunlop. *Dynamic Policy-Based Management in Open Distributed Environments*. PhD thesis, Department of Computer Science and Electrical Engineering of the University of Queensland, September 2002.

[GLB06] GLBP. Gateway Load Balancing Protocol. http://www.cisco.com search GLBP, 2006.

[KC03] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *IEEE Computer Magazine*, 36:41–50, January 2003. ISSN:0018-9162, Issue 1.

[KWW+98] S. Knight, D. Weaver, D. Whipple, R. Hinden, D. Mitzel, P. Hunt, P. Higginson, M. Shand, and A. Lindem. Virtual Router Redundancy Protocol. IETF RFC2338, April 1998.

[KY03] A. V. Konstantinou and Y. Yemini. Programming Systems for Autonomy. In *Proceedings of the Autonomic Computing Workshop Fifth Annual International Workshop on Active Middleware Services*, ISBN 0-7695-1983-0/01. IEEE, 2003.

[LK03] A. Joshi L. Kagal, T. Finin. A Policy Language for a Pervasive Computing Environment. In *Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY03)*, 2003.

[LRP05] J. P. Lang, B. Rajagopalan, and D. Papadimitriou. Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification. nternet Draft draft-ietf-ccamp-gmpls-recovery-functional-04.txt, April 2005.

[LS99] E. C. Lupu and M. Sloman. Conflicts in Policy-Based Distributed Systems Management. *IEEE transactions on software engineering*, 25(6), November 1999.

[LSDD00] E. Lupu, M. Sloman, N. Dulay, and N. Damianou. Ponder: Realising Enterprise Viewpoint Concepts. In *In Proceedings 4th Int. Enterprise Distributed Object Computing*, 2000.

[MESW01] E. Moore, E. Ellesson, J. Strassner, and A. Westerinen. Policy Core Information Model – Version 1 Specification. IETF RFC 3060, February 2001.

[NGA+04] J. Nielsen, A. Galis, H. Abrahamsson, B. Ahlgren, M. Brunner, L. Cheng, J. A. Colas, S. Csaba, A. Gonzalez, A. Gunnar, G. Molnar, and R. Szabo. Management Architectures and Approaches for Ambient Networks. In *12th WWRF meeting*, Toronto, Canada, November 2004. NLE-PR-2004-65.

[PSA05] P. Pan, G. Swallow, and A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. IETF RFC4090, May 2005.

[RPR06] RPR. Resilient Packet Ring Alliance. http://www.rpralliance.org, 2006.

[RST] RSTP. IEEE Standard - 802.1w - Rapid Reconfiguration of Spanning Tree, supplement to ISO/IEC 15802-3:1998.

[SD06] Self-Defending. Cisco Self-Defending Network. http://www.cisco.com search self-defending networks, 2006.

[SEPP05] Andreas Schieder, L. Eggert, N. Papadoglou, and F. Pittmann. Components and Concepts of the Ambient Networks Architecture. Wireless World Research Forum WWRF, 2005.

[Siq06] Marcos A. Siqueira. *Policy-based Autonomic Management (work in progress)*. PhD thesis, Unicamp, 2006.

[SMC02] M. A. Siqueira, M. F. Magalhães, and E. Cardozo. A Policy Management Architecture for MPLS Networks. In *Proceedings of the 15th International Conference on Parallel and Distributed Computing Systems (PDCS 2002)*, 2002.

[SMFdC04] M. A. Siqueira, M. F. Magalhães, L. J. L. Farias, and M. C. de Castro. A BGP/MPLS PPVPN Management Information Model and a J2EE-based Implementation Architecture for Policy and Web-Based Configuration Management Systems. In *Proceedings of IEEE ICN'04 - 3rd International Conference on Networking*, 2004.

[SNRLM05] M. A. Siqueira, N. A. Nassif, R. A. Resende, and M. Lima-Marques. Policy-Based Architecture for QoS Management in Enterprise IP Networks. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, 2005.

[Str02] J. Strassner. DEN-ng: Achieving Business-Driven Network Management. In *NOMS*, 2002.

[Str05] J. Strassner. Autonomic Networking - Theory and Practice. Tutorial 4, IM2005, Nice-France, May 2005.

[SY03] S. Shah and M. Yip. Extreme Networks' Ethernet Automatic Protection Switching (EAPS) Version 1. IETF RFC 3619, October 2003.