Monitoring within an Autonomic Network: A GANA Based Network Monitoring Framework

Anastasios Zafeiropoulos[1], Athanassios Liakopoulos[1], Alan Davy[2], and Ranganai Chaparadza[3]

[1] Greek Research & Technology Network, Av. Mesogion 56, 11527 Athens, Greece
{tzafeir,aliako}@grnet.gr
[2] Telecommunications Software & Systems Group, Waterford Institute of Technology Cork Road, Waterford, Ireland
adavy@tssg.org
[3] Fraunhofer FOKUS Institute for Open Communication Systems, Kaiserin-Augusta-Allee 31, D-10589, Berlin, Germany
Ranganai.Chaparadza@fokus.fraunhofer.de

Abstract. The concept of self-managing of autonomic networks is a paradigm shift from today's management models, aiming at enabling networked nodes to self manage their behaviour within the constrains of the operator's policies and objectives. In this article, we present our approach for self-coordinating monitoring functions within such an autonomic network. This approach complies with the principles of a recently introduced *Reference Model* for autonomic network engineering/self-management within node and network architectures dubbed: the *Generic Autonomic Network Architecture (GANA)*, which aims to identify autonomic behaviours realised via hierarchical control loops among self-managing elements. The components of the proposed monitoring framework, the interactions among the identified elements and a complete use case scenario are described in detail.

Keywords: Autonomic network engineering, monitoring, self-management, GANA, Hierarchical Control-Loops (HCLs) framework.

1 Introduction

The vision of the Future Internet, is of an autonomic network whose nodes are engineered in such a way that all the traditionally so-called network management functions defined by the FCAPS (Fault, Configuration, Accounting, Performance, Security) management framework [1], as well as the fundamental network functions such as routing, forwarding and monitoring, are designed to automatically feed each other with information and effect feedback processes among the diverse functions. The feedback processes enable reactions in functions of the network and of individual nodes, in order to achieve and strive to maintain some well defined goals of the network.

Autonomic principles are required in order to accomplish management of dynamic, heterogeneous and complex networks, where each network entity needs to be able to take network optimisation decisions. FCAPS functions have to be intrinsically in-built

A. Dan, F. Gittler, and F. Toumani (Eds.): ICSOC/ServiceWave 2009, LNCS 6275, pp. 303–313, 2010.

into node architectures apart from being part of an overall network architecture - whereby traditionally, a separate management plane is engineered separately from the other functional planes of the network. This means that the functional planes of an autonomic network would need to be (re)-defined, re-factored or even merged.

In this paper, we start by briefly presenting the key principles of the Generic Autonomic Network Architecture (GANA) [2] that is proposed in the framework of the EU FP7-EFIPSANS IP Project[1] [11]. We focus on the realization of an autonomic monitoring framework within the EFIPSANS network based on GANA principles. GANA sets the principles and guidelines that need to be followed according to our vision of the Future Internet design. In contrast to any other of today's best known approaches, including clean-slate approaches, such as 4D [3], CONMan [4], a Knowledge Plane for the Internet [5], FOCALE [7], a Situatedness-based Knowledge Plane for autonomic networking [8], GANA captures in a holistic way, the generic principles required for a generic autonomic network architecture.

The paper is organised as follows; section 2 presents the fundamental concepts of the GANA architecture and a methodology for designing autonomic behaviours in compliance with GANA principles; section 3 describes the EFIPSANS network monitoring and information dissemination framework; section 4 details a use case scenario in which a number of autonomic behaviours are introduced and finally section 5 concludes the paper with a discussion of current challenges and future work.

2 A Brief Introduction to the Generic Autonomic Network Architecture: GANA

In contrast to existing approaches to autonomic networking, an appropriate model for an autonomic networked system is needed to be introduced that would allow us to reason and think of a particular autonomic function that implements the concept of a control loop at some particular level of abstraction. This means, we should then be able to talk about autonomic networking functions such as autonomic routing, autonomic forwarding, autonomic monitoring in the sense that the elements driving the control loops, use information learnt from its required information suppliers to control the behaviour of the functionality that is then considered to be autonomic.

Fig. 1 shows a model of an autonomic networked system and its associated control loop that we developed. It is derived and extended from the IBM-MAPE model [9] specifically for autonomic networking and is the basis for the derivation of the GANA architecture. The model is generic and illustrates the possibility of the distributed nature of the information suppliers that supply a so called Decision Element (DE) and the diversity of the information that can be used to manage the associated managed resources and elements. It can be applied to different levels of functionality and abstraction within node architectures and network architectures.

We define a Decision Element (DE) as a concept that is associated with some concrete resources or functional entities (e.g. protocols) that the Decision Element manages and

[1] The EFIPSANS-IP-Project aims at exposing the features in IPv6 protocols that can be exploited or extended for the purposes of designing or building autonomic networks and services, as necessitated by GANA.

drives its control loop. Information or views are being continuously exposed by its managed resources or functional entities, together with information coming from other required or potential information suppliers of the Decision Elements. We also adopt the concept of a Managed Entity (ME) to denote a managed resource or an automated task in general, instead of a managed element. As illustrated in Fig. 1, the actions taken by the Decision Element do not all necessarily have to do with triggering some behaviour or enforcing a policy on the Managed Entities but that, some of the actions executed by the Decision Element may have to do with communication between the Decision Elements and other entities. This is indicated by the extended span of the arrows: "Downward Information flow" and the "Horizontal Information flow", as well as the fact that a Decision Element also exposes *views* such as *events* to its "upper" Decision Element and receives *policies, goals and command statements* from its "upper" Decision Element.

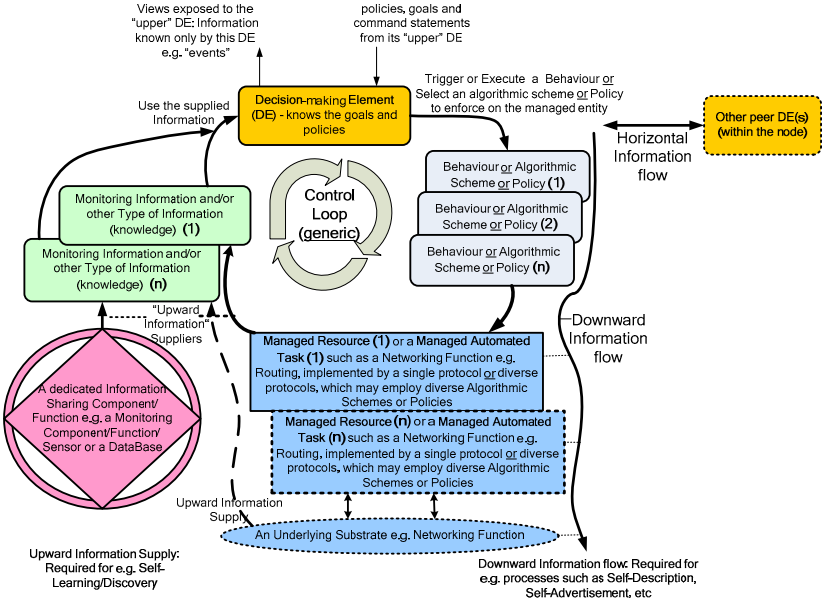

Fig. 1. A generic model for an abstract autonomic networked system

2.1 Types of Control Loops in GANA

GANA defines a Hierarchical Control Loops (HCLs) framework that reflects on the need for control loops operating at different levels of abstractions and complexity. Four levels of abstractions for which Decision Elements, Managed Entities and associated control loops can be designed are described below.

Protocol Level [level 1]: The concepts of a control loop, Decision Element, Managed Entity, as well as the related self-manageability issues may be associated with some implementation of a single network protocol or an automated task.

Function Level [level 2]: On a higher level of abstraction, concepts of a control loop, Decision Elements and Managed Entities may be addressed on the level of abstracted network functions, such as routing, forwarding, mobility management, QoS management, etc.

Node Level [level 3]: On a higher level of autonomic networking functionality, the concepts of a control loop, Decision Elements, Managed Entities, as well as the related self-manageability issues may be associated with a system or node as a whole. The node's main Decision Element has access to the "views" exposed by the lower level Decision Elements and uses its knowledge of the higher level to influence the lower level Decision Elements to take certain desired decisions, which may in turn further influence or enforce desired behaviours on their associated Managed Entities, down to the lowest level of individual protocols.

Network Level [level 4]: The highest level of control loops is the network level. There may exist a logically centralized Decision Element (DE) or DEs such as the ones proposed in the 4D network architecture [3] that knows the objectives, goals or policies to be enforced by the whole network. The objectives, goals or policies may actually require that the nodes' main Decision Elements of the network export "views" such as events and state information to the logically centralized Decision Element(s), in order for the network-level DEs to influence or enforce the Decision Elements of the nodes to take certain desired decisions. A distributed network level control loop is implemented following the above set-up, while another would involve the main Decision Elements of nodes working co-operatively to self-organize and manage the network without the presence of a logically centralized Decision Element(s).

The hierarchies of Decision Elements in GANA, which allow for some decisions to be taken autonomously at different levels of control and complexity, may have the following forms: Hierarchical relationships between Decision Elements where lower level Decision Elements are managed by their upper level Decision Element and Peering relationships between Decision Elements, which facilitate communication between Decision Elements for exchanging information.

2.2 The Functional Planes of GANA

In GANA [2], we first take the position that the functional planes known in today's world of networking can be compressed - with merging and re-factoring some of the planes - into four functional planes that are still called: the Decision Plane, the Dissemination Plane, the Discovery plane and the Data plane, which we adopt from the 4D architecture [3], but, we re-define them for GANA because we introduce the concept of an autonomic Decision Element into the architecture of every network node/device, as opposed to the 4D, which assumes that network nodes/devices need not have Decision Elements within them.

3 EFIPSANS Network Monitoring Framework

Monitoring information is a fundamental part of a wide number of network functionalities, such as QoS Management, Routing, and Mobility. However there may be a

great deal of operational overhead involved in the configuration / re-configuration / optimisation of these operations to ensure that the information being supplied is of sufficient accuracy for the corresponding operations. In this section, we propose a Monitoring Framework that is aligned with GANA principles, extending autonomic monitoring principles presented in the literature. The monitoring framework describes the basic functions needed for management of monitoring activities within an autonomic network. By developing a Monitoring Decision Element responsible for the configuration of the monitoring protocols and mechanisms, other function level Decision Elements within a node and within the network can be guaranteed that relevant and sufficiently accurate information is available to drive their control loops.

3.1 The Monitoring Decision Element and Its Functionality

The Monitoring DE (Mon_DE) - shown in Fig. 2 - resides in the function level and controls all the actions of an autonomic node related to monitoring. It also interacts with the rest of the Decision Elements or the Managed Entities residing in the same or other autonomic nodes, whenever these entities require specific monitoring actions to be taken or monitoring mechanisms to be activated. It is possible to define multiple control loops at a function level, each of them implementing a specific monitoring function based on decisions made by the Mon_DE.

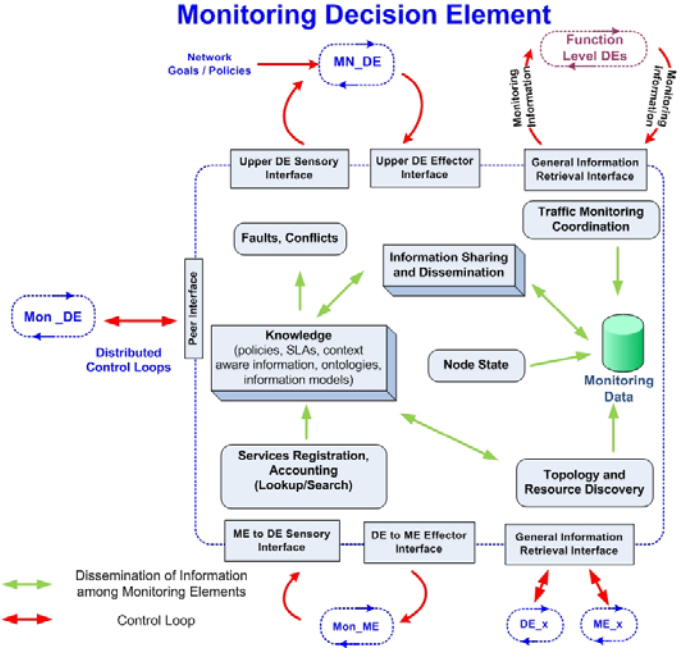

Fig. 2. The Monitoring Decision Element and its interfaces

The Mon_DE is responsible to orchestrate Monitoring Managed Entities (Mon_MEs) so as the node level monitoring policies are fulfilled and specific functions are realised. It is responsible for monitoring actions or mechanisms such as:

- *traffic monitoring coordination:* a) manage mechanisms for traffic monitoring and schedule monitoring measurements in such a way that they do not degrade other services –such as QoS or mobility services- or negatively affect collected results, b) activate active or passive measurements in the autonomic node in accordance to provided services and network context and c) store and publish any historical data collected via measurements for later analysis.
- *information sharing and dissemination:* collect information about stored data in node and network level and make it available to other Decision Elements or Managed Entities. According to the network topology, monitoring data may either be distributed across the network or stored in a centralized database. The Monitoring Decision Element is aware of the data collection and dissemination scheme that is selected and acts as an information supplier to other Decision Elements, providing them with guidelines for data acquisition and dissemination.
- *topology discovery:* collect information necessary for creating the topology of the network and publish topology data in a appropriate format to other Decision Elements or Managed Entities, e.g. the visualization Managed Entities.
- *resource discovery:* collect information regarding the available resources in network level. This can be proven very useful in ad hoc networks where nodes continuously join/leave the network and allocation and negotiation of different resources is dynamic.
- *monitor the state of the node:* monitor the available node resources and specific metrics such as available power, storage capacity and status of interfaces. The Monitoring Decision Element can interact with SNMP MIBs and request information through queries. SNMP alarms can also be generated in case of specific events.
- *provide context-aware information:* process any collected data according to the specified GANA ontology [6] in order to enable reasoning over the available information and extract meaningful events. By increasing the context-awareness level of monitoring data exchanged by autonomic nodes, it is possible to efficiently sense changes in the network and the provided services and proceed to corrective actions (self-healing).
- *fault and conflict management:* diagnose a) faults in node and network level, b) violation of guarantees for predefined performance metrics and c) firewalling and security alarms. The Monitoring Decision Element is able to identify conflicts in the policies imposed by other Decision Elements and trigger accordingly the responsible Decision Elements in order to resolve the conflicts.
- *accounting and registration to network services:* identify services supported in the autonomic network, enable subscription to services and access to specific information and provide related information to other Decision Elements.

3.2 Basic Interactions and Interfaces of the Monitoring Decision Element

The Monitoring Decision Element interacts with the following elements:

- The Main Node DE (MN_DE) residing at the node level establishing a hierarchical relationship. Within this relationship, the Mon_DE operates as a Managed Entity, i.e. only provides information and alarm events, and offers a management interface to receive node or network level policies.
- The various Decision Elements at the function level, such as the routing and mobility Decision Elements, establishing a peering relation where it operates as an information supplier.
- The various Monitoring Managed Entities (Mon_MEs) and various Decision Elements and Managed Entities at the protocol level establishing a parent relationship. The Mon_MEs implement specific (monitoring) functions required by other Decision Elements and Managed Entities to operate. The Mon_DE operates as a Decision Element within this relationship, i.e. implements the control loops.
- Mon_DEs at other autonomic nodes, establishing peering relationships, in order to form distributed control loops, exchange relevant monitoring information and coordinate network wide monitoring tasks.

All the Mon_MEs at the protocol level are under the control of the Mon_DE. Thus, when a specific Decision Element at the function level, such as the QoS_DE, requires a Mon_ME to be activated, it has to request it through the Mon_DE. These restrictions are imposed in order to avoid conflicting actions, as monitoring actions consume network or node resources and may also affect other services, e.g. bandwidth aggressive measurements may cause QoS degradation to legitimate traffic. It should be noted that any Managed Entity usually participates in more than one control loops under the authority of different Decision Elements as the role of information supplier, but it is managed by a unique Decision Element.

3.3 Monitoring Distributed Control Loops

The GANA architecture allows the realization of distributed control loops achieved through the interactions among multiple Decision Elements located in autonomic nodes or via a logically centralized overlay of Decision Elements operating on the network-level. In the first case the decision process is distributed across multiple Decision Elements while in the second case the decision is taken by a single or a small group of Decision Elements chosen via a delegation process.

A distributed control loop, either realized via a centralized entity or through interactions of multiple entities, has to be applied in order to take any decisions on how to monitor an autonomic network and its deployed services. This loop is responsible for providing information regarding the overall state of the network. The Monitoring Decision Elements at the function level participate in the network-wide distributed control loop. Each autonomic node advertises its capabilities to other nodes relating to monitoring functions and the network-wide control loop is responsible for directing how to monitor the supported services.

4 A Detailed Scenario: Traffic Monitoring and QoS Control

In the following scenario, we focus on autonomicity as a feature of traffic monitoring, coupled with Quality of Service (QoS) management functions of an ingress edge

router, complying with GANA principles. As network and traffic conditions continuously change, monitoring protocols and mechanisms must be appropriately re-configured in order to facilitate the efficient QoS management of the autonomic network. The objective of QoS control at the ingress point within a DiffServ domain is to ensure that the traffic admitted to the network is appropriately classified, policed and shaped to assure performance guarantees.

4.1 Description of Decision Elements, Managed Entities and Their Interactions

Effective Bandwidth ME (EB_ME): This Managed Entity is a process at the protocol level of GANA that implements the Effective Bandwidth estimation algorithm proposed in [10]. The process collects a packet trace from the network, performs a number of processing activities on it and reports an estimation of effective bandwidth for a particular QoS target of packet delay. The EB_ME is managed by the Mon_DE and can act as information supplier to other MEs and DEs.

Bandwidth Availability in Real-time ME (BART_ME): This is an active probing process, used to generate and inject probe packets into the network towards a destination node. The BART_ME on the destination node estimates the amount of available bandwidth along the path between the two nodes. The management and configuration of the BART_ME is done by the Mon_DE on each node.

Quality of Service DE (QoS_DE): This is a function level Decision Element that participates to a node-local control loop. It aims to configure the mechanisms - such as queue management, queue scheduling, marking, policy, admission control, etc - to support service guarantees provided by the network. The QoS_DE is responsible for configuring the node-level mechanisms and interacts with the MN_DE and other Function level Decision Elements such as the Mon_DE.

Admission Control ME (AC_ME): This Managed Entity ensures that traffic admitted into a node or network will not violate specified QoS performance guarantees. This process is based on the admission control algorithm presented in [10]. The AC_ME depends on measurements of effective bandwidth of the incoming traffic, realized by the EB_ME and is managed by the QoS_DE.

QoS Violation ME (QoS_V_ME): It produces estimations of performance violations for traffic exiting the network through an egress router. The violations are estimated by analysing short packet traces and results are compared with particular QoS targets. This mechanism is a modification of the EB_ME and is managed from the Mon_DE.

4.2 Control Loops and Entities Interactions

The autonomic behaviour instilled within the ingress edge node is the ability to control the incoming traffic into a domain while maintaining a high degree of confidence in the admission decisions. This is provisioned by an interaction between the Mon_DE and the QoS_DE, where the traffic monitoring requirements of the AC_ME change and the associated EB_ME must be re-configured dynamically in order to conform to these changes. There is a dependency relationship between the lowest level MEs, i.e. EB_ME and AC_ME, of the ingress router, necessary for MEs to operate in an optimal manner.

It is the responsibility of the interacting control loops, managed from the QoS_DE and the Mon_DE, to drive the re-configuration of these lowest level MEs.

The dependency relationship is defined as follows; the AC_ME requires effective bandwidth estimations on currently admitted traffic in order to make admission decisions. These measurements are supplied by the EB_ME that is managed by the Mon_DE and can be configured to supply these metrics with varying degrees of accuracy and frequency. This information along with other network state related information can help the QoS_DE to manage the AC_ME. However, in cases where the requests imposed by the QoS_DE and the Mon_DE are conflicting, interactions are established with the corresponding DEs that manage the conflicting hierarchical control loops and priorities are provided according to the specified policies. If priorities cannot be established, then the Main Node DE is responsible to resolve the conflict.

4.3 Information Flow and Associated Behaviours

In Fig. 3, we present the flow of information between Decision Elements and Managed Entities within the context of the GANA Monitoring Framework. The EB_ME supplies effective bandwidth information to the AC_ME, thus acting as an information supplier of that Managed Entity. If the QoS_DE detects that effective bandwidth estimation on admitted traffic needs to be updated quicker to compensate for the increased number of service request arrivals, it requests from the Mon_DE to reconfigure the EB_ME accordingly.

The BART_ME supplies information regarding the bandwidth availability between the ingress node and the egress node to the QoS_DE on the ingress node. As there is best effort traffic traversing the network, congestion can occur within the core network. If congestion reaches a limit, this can affect the services that can be admitted while maintaining QoS targets of the admitted traffic flows. The QoS_DE therefore has a policy to reconfigure the AC_ME to prioritise higher revenue services during such times of congestion within the network.

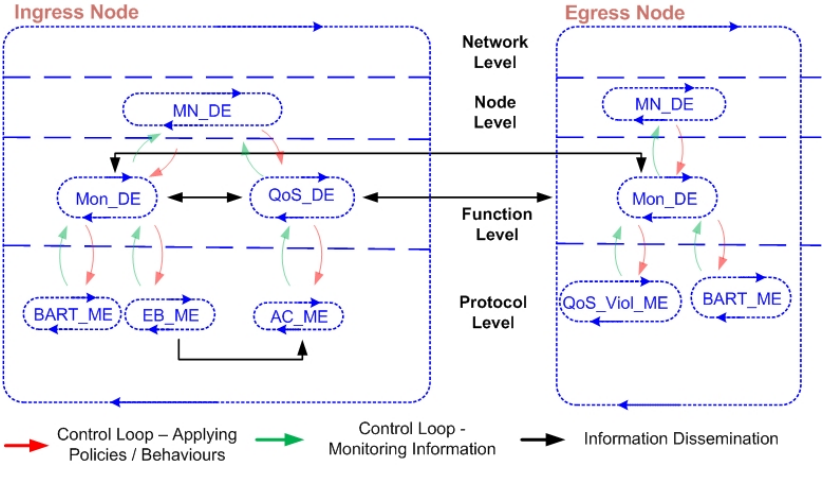

Fig. 3. Placement of DEs in different levels and their HCLs

Finally, the QoS_V_ME supplies information to the Mon_DE on the ingress node for any QoS violations that occur. If, for example, QoS performance violations are experienced on one-way delay metrics, the Mon_DE can inform accordingly the QoS_DE, which can configure the AC_ME to be more conservative in its admission decisions, ensuring adequate bandwidth is being reserved for admitted services. However such a configuration can impede on the AC_ME's performance in reacting to variations in traffic. This trade off decision is the responsibility of the QoS_DE to make and not the AC_ME.

5 Conclusions and Future Work

In this paper we presented the Generic Autonomic Network Architecture (GANA), as the basis for producing autonomic behaviour specifications for selected diverse networking environments. We proposed a network monitoring framework based on GANA principles and described how the different elements and high-level entities should interact, apply policies and process information. Specific autonomic behaviours were elaborated through a detailed scenario.

In our future work, the presented scenario is going to be implemented and the designed network monitoring framework will be evaluated in the EFIPSANS testbed. Focus will be given on the instantiation of the monitoring functions and the effectiveness of the monitoring mechanisms in diverse networking environments. Furthermore, GANA will be extended by focusing on interfaces that facilitate interactions among autonomic nodes and allow administrators to define network level objectives and policies.

Acknowledgement. This publication is based on work carried out and funded under the framework of the European Commission ICT/FP7 project EFIPSANS.

References

1. ITU-T, TMN Management Functions, Telecommunication Standardization sector of ITU, M.3400 (2000), http://www.itu.int/rec/T-REC-M.3400-200002-I/en
2. Chaparadza, R.: Requirements for a Generic Autonomic Network Architecture Suitable Requirements for Autonomic Behavior Specifications of Decision-Making-Elements for Diverse Networking Environments. In: International Engineering Consortium (IEC) Annual Review in Communications, vol. 61 (2008)
3. Greenberg, A., Hjalmtysson, G., Maltz, D.A., Myers, A., Rexford, J., Xie, G., Yan, H., Zhan, J., Zhang, H.: A Clean Slate 4D Approach to Network Control and Management. ACM SIGCOMM Computer Communication Review 35(5), 41–54 (2005)
4. Ballani, H., Francis, P.: CONman: A Step Towards Network Manageability. ACM SIGCOMM Computer Communication Review 37(4), 205–216 (2007)
5. Clark, D., Partridge, C., Ramming, J.C., Worclawski, J.T.: A Knowledge Plane for the Internet. In: Proc. of the 2003 conference on Applications, Technologies, Architectures and Protocols for Computer Communications, pp. 3–10 (2003)
6. Zafeiropoulos, A., Liakopoulos, A.: Context Awareness in Autonomic Heterogeneous Environments. In: INGRID 2009 (2009)

7. Jennings, B., Van der Mer, S., Balasubramanium, S., Botvich, D., Foghlú, M.O., Donnelly, W.: Towards autonomic management of communications networks. IEEE Communications Magazine 45(10), 112–121 (2007)
8. Bullot, T., et al.: A Situatedness-based Knowledge Plane for Autonomic Networking. International Journal of Network Management 18(2), 171–193 (2008)
9. IBM, Understanding the Autonomic Manager Concept, http://www.ibm.com/developerworks/library/acconcept
10. Davy, A., et al.: Revenue Optimized IPTV Admission Control using Empirical Effective Bandwidth Estimation. IEEE Transactions on Broadcasting 54(3), Part 2, 599–611 (2008)
11. The EFIPSANS project, http://www.efipsans.org/