Autonomic Network Management for Next Generation Networks

A. Louca, A. Mauthe, D. Hutchinson
Computing Department
InfoLab 21
Lancaster University
Lancaster, LA1 4WA
Email: {a.louca, a.mauthe}@lancaster.ac.uk, dh@comp.lancs.ac.uk

Abstract—**With the exponential growth of current computer networks, the need for autonomic network management is clearly increasing. While there have been many attempts to tackle autonomicity in network management environments, the platforms are too specific and apply only to very specific environments. A different approach is presented in this paper whose the concept of a dedicated network management platform based on autonomic principles is adopted. This platform encourages a more generic and streamlined approach for targeted research in autonomic network management by providing a lightweight, platform-agnostic middleware which gives access to communication primitives, platform-specific augmented support, basic data storage support and a query language for information retrieval in a dynamically-composed atomic functional units environment.**

I. Introduction

The Internet has been an extraordinary success - it grew exponentially over the years, connecting literally hundreds of millions of users. However, despite success, its simplicity and transparency (due to the use of the current generation of Internet protocols) push rich functionality to the network edges, leaving the core dumb with out provision and design for extensive management features [1]. This has resulted in foes when it comes to management and fault diagnosis. This increasing growth of users and the lack of proper management methods integrated inside the network core has made modern computer networks hard to manage efficiently and reliably, affecting the dependability of mission critical network applications and services [2]. Additionally, it has become a real challenge to operate these networks, as the human operators capacity to manually manage, control and operate these networks are being exceeded [3]. To control this complexity and allow operators to be able to manage the even more complex future networks more efficiently and reliable, the need for a decentralised and autonomic control that will remove part of the managing burden from human operators has become more obvious . Autonomic Computing, firstly proposed by IBM [4], [5], seeks to introduce some intelligence in computer systems to be able to manage themselves with using just high-level guidance and objectives by their human operators. While this initiative focuses more on technology management as a general, the research introduced here focuses on managing computer networks.

To make the network management more useful, it is essential to make higher-level information available to it [6]. This can be achieved by introducing a novel platform to manage this information. Current networks are only packet-movers, with no extensive provision for management included in the original protocol design [1], and are without a network-wide perspective of the current status. This paper will discuss some requirements for such a platform, which will enable the network core to receive additional information regarding its purpose and goals. An overview of a potential prototype architecture for such a platform is also presented, with sole purpose of using it as a basis for research in autonomic network management.

The paper structure is as follows: section 2 gives an overview of the background in autonomic network management, section 3 discusses the requirements of such a platform and a possible prototype architecture for the purpose of a research platform. Section 4 discusses the next steps and future work towards the implementation and evaluation of the platform and section 5 concludes the paper.

II. Background

The fully decentralised and distributed administrative approach that is embedded in the Internet design has created high management overhead with a lot of manual work: configuration, fault diagnosis and design. It requires critical human intervention, which is also time consuming and error-prone. Additionally, the design of current Internet Protocols exacerbate this issue, as they move functionality from the network core to the network edges, leaving the network dumb [1], [7]. When an anomaly occurs in the network, the core cannot understand what is going wrong, as it does not *know* what it is doing – core networks are simply packet movers. Only the edge applications understand that something is not working as it should, but, however, the core can not provide any useful input, as it has no perception what the behaviour of edge applications or the network. Additionally, this creates another problem when it comes to network design and configuration: the human operators must manually configure and define routing or QoS policies per router, as there is no way of providing the network core devices with a higher level goal of the operators and a way to translate these high-level

ISBN: 978-1-902560-24-3 © 2010 PGNet

targets into low-level operations. When designing, however, a technique to give a solution to remediate the above described problems, must be careful in order not to lose the features that made the Internet a success in the first place, but devise it so that combines the virtues of the Internet with the potential solution [6].

Clark noted very early [1] that the next step in the Internet was to develop next-generation tools for management of resources in the context of multiple administrations. Later, he proposed to develop a network architecture that can understand high-level instructions and assemble itself, adapt as requirements change, automatically detects when something is not going according to plan and attempts to fix it or explain why it can not do so. In a subsequent paper [6], Clark states that the fundamental design of the current Internet architecture is causing problems when it comes to management, and proposes a network that has a high-level view of its goals and constrains and based on this view is able to act towards that target. This high-level framework is called *ïhe knowledge plane¨*, and its ultimate target is a network that can organise, evolve and repair itself and increase transparency to its operators.

Current algorithmic and management techniques are not sufficient to solve this problem: a novel approach and a fresh start is required to target the management issue [6]. Network Management is an attempt to design and implement a management entity for Next Generations Networks that will address the above mentioned issues, by constructing a middleware which will support the dynamic composition of functional units, such as sensors, cognitive processing, learning techniques and configuration which will serve as building blocks for creating an autonomic management service to meet a set of high-level requirements, given by the human operators.

The autonomic computing initiative, launched by IBM, aims to reduce the human role in managing computer systems by a self-management approach, having computer systems anticipate requirements and resolve operational issues that arise without any human assistance [5], [8]. This reduces the human role by providing high-level requirements and delegating control tasks to the system to achieve these goals. This approach is generally called an autonomic or goal-oriented management approach [9], starting with the specification of high-level requirements from human operators. After they have been analysed and refined they are composed into lower-level, machine understandable statements, such as policies [9]. This approach has been the focus of several research projects [10], [11], which developed several analytical methods of transforming high-level goals into low-level policies.

The bio-networking architecture was developed within the BIONET [1] project at the University of California. This architecture takes principles from biological organisms and creates analogies to computer networks to allow them to adapt and survive [12]. It provides a middleware that supports components which are used to build applications and autonomic paradigms as design guides. On the middleware autonomous mobile agents, *the cyber-entities*, are used to implement network applications. This approach relies on mobile agent background and reuses some of its concepts.

[1]http://netresearch.ics.uci.edu/bionet/

The FOCALE Architecture (Foundation, Observation, Comparison, Action and Learning Environment) [13] assumes that any managed resource (resources can range from single devices or entire networks) can be turned into an Autonomic Component. FOCALE embeds an Autonomic Manager in each Autonomic Computing Element (ACE), making it possible to provide uniform management functionality throughout the managed autonomic system. Multiple Autonomic Computing Elements can join to form Domains or Environments.

A policy-based autonomic framework for next generation networks [14] which employs the DEN-ng model, providing extensible means to represent the policies as well as the behaviour of participating managed entities from a technology-neutral perspective. The framework is also attempting to satisfy adaptive functionality, such as self-healing and self-configuring, and support for translating high-level requirements into instance-specific policies that will govern network elements directly. Similarly, ANEMA [15] introduces an architecture which by using a continuum of policies leverages autonomic management in a multi-service IP network.

All of the above projects do introduce a novel approach of targeting the issue of increasing management complexity, however they target it with fundamentally different ways (policies, bionets etc.) with no de-facto generic platform for conducting academic research in the area. Highlighting this issue, this paper introduces a more generic platform, by providing only the fundamentals in order to facilitate the creation of autonomic management network entities, but not dictate a specific way of doing it. Researchers will be free to use these primitives to develop functional units that run on top of this middleware. This paper presents a novel approach to Network management, by adopting new algorithmic and techniques to address the problem. Autonomic

III. Autonomic Network Management Middleware - A management platform

As indicated above, the purpose of this work is to deliver a generic management platform for autonomic network management research, by providing a middleware which will run on all of the participating network devices in the network, extending the existing control plane and running in parallel with existing IP protocols. This remainder of this section discusses the proposed research platform, firstly by stating some necessary requirements it must fulfil in order to accomplish this role and finally by laying down an architecture for this platform.

A. Platform requirements

In order to introduce some intelligence in the management part of the network core, it would need some more knowledge and relative understanding of its purpose and high-level goals.

This extra knowledge and information will allow the development of algorithmic techniques on this platform, that will enable the management plane of the network to react, evolve and optimise given the necessary sensory input, effectively suppressing the effectiveness of the current management algorithms, which have been judged insufficient [6]. Autonomic Network Management points to a solution where the core network has a view of what its high-level goals and purposes are, as set by its human operators. The following section will discuss the perspective of this approach, and how this differs from current approaches, an architect and potential algorithmic solutions to some of the problems this solution is called to answer.

Additionally, these requirements are necessary to be fulfilled in order to differentiate this management platform from existing ones and truly offer something novel in the field.

The following list, briefly discusses some important targets and goals that this novel research platform will try to address, in order to enable the development of more advanced and intelligent autonomic network management algorithms and techniques.

- **Network-wide perspective:** Currently, devices in the network, such as routers, switches, firewalls etc., have no network-wide perspective of what the status is. They only know what is going on within that device, in relevance with their local, manually provided configuration. They do not share their current status with their neighbouring devices in order to create a consistent network-wide perspective of the network. Lacking perspective is impossible to move towards achieving a high-level target, and the human operators must manually configure each individual machine and make sure it meets their requirements.
- **Applications involvement:** The network core has no involvement with the requirements of applications at the network edge. but applications do have an understanding what to expect from the network core and how they expect it to behave in terms of data transfer rate, quality of service etc. This understanding is not shared with the core as it currently has no such capability, and this knowledge is kept at the edges (the end-to-end argument). An autonomic management platform should allow limited (restricted by the network operators) interaction of the core with the edges, to communicate requirements and respond whether the service requested is possible or not, given a set of requirements (such as jitter, available bandwidth etc.).
- **Cognitive support:** In order for the network to start moving towards a high-level goal it must first understand the goal, and how that translates into low-level actions and configuration. Additionally, as time passes, it must have the ability to optimise and evolve with possible reconfigurations, to best fit the goal. In order to support this, the autonomic management platform must allow support for cognitive processing, which will provide the ability for the network to learn, evolve and take decisions towards the best fulfilment of its goal.
- **Sensory support:** To support cognitive functions, new adaptive sensors will be required to enable more advanced statistics and measurements that what is currently offered with current probing methods.
- **Human supervision:** Support for supervision from human operators must be present, to allow them to view the current status of the whole network. Operators should be given an extensive supporting toolkit, with the ability to probe and collect this extra information generated from above described activities. Additionally, this interface will be used for pushing new requirements and high-level goals to network devices, and it will act as a human-network interface.
- **Information Interoperability:** In order to deploy a network-wide management service, information and data should be in a specific format that it is unified and standard across the network. However, it is already difficult to achieve this, but it is necessary to have interoperability between semantically equivalent data across management domains [3]. A mechanism should be present to enable data sharing between various functional units and devices will have the support to convert data between different formats, given that they are in the same semantic domain.

Further, it is important to make assumptions regarding the platform the middleware is running on explicit [2]: assumptions about the environment and expected results of automated actions, and communicate these assumptions with the researchers working on the platform as well as human operators. This is important because it will be easier to tell when a system is operating under supported conditions or not. Additionally, the same applies for the goals given to the system: they must also be explicit. A problem that is unstated it is unlikely the solution to be either coherent or comprehensive, never mind communicating that to a machine.

B. Proposed architecture

The previous sections of this paper briefly discussed some proposed goals of a management platform for autonomic networks. This section will go on and discuss a possible architecture in order to achieve these goals. It is clear that this architecture needs to be distributed and all core devices in a network domain must participate in order for this to work and achieve a solid and consistent perspective of the network.

In addition to the high-level goals, however, it must also be able to provide support to some supplementary functions, which will be complementary in the sense of providing administrative and development support. The following list briefly discusses some of these items.

- **Support for Management Overlays:** The complexity of current networks dictates the support of management overlays, as a network might span across the world, most of the times within different networks, enabled from technologies like AoMPLS and VPLS. For example a multi-national company may have two networks: one in Europe and one in America, which are not directly

interconnected; however the administrators set the same targets and goals for both networks. In order for Autonomic Network Management to be able to enable management communication between these two physically distinct networks, but logically united, it must support management overlays over existing physical networks, so to keep a unified logical management.
- **Support for recomposition and functional extensibility:** As targets change and network evolves the management plane must be able to evolve itself as well. Newer hardware, software and protocol support must be able to be deployed easily across the network, so the whole management domain can evolve.
- **Trust and security:** Networks are rarely managed by one single domain, but they are rather split into multiple sub-domains. The reason for this is that one team of engineers might handle just application and web servers and the other might handle network devices; however, while both groups form the network, the two sub-domains might not share all information between them (i.e. customer sensitive data, different high-level targets etc.). Additionally, fake or untrusted devices must be able to be filtered out from this management plane, as they might interfere with the operations. The platform must carry support trust, security and privilege management across devices and domains.

The Autonomic Network Management platform needs to be distributed: each participating network device will be required to run an instance of the middleware and join the rest of the administrative domain. It will run aside from the data plane of network devices, i.e. it will not be part of the high-speed data plane, rather, it will be the evolution to the current established control plane. This will allow the Internet to continue to function as it currently does, adding the additional benefits from autonomic management -networking platform. The service itself will be designed to be as lightweight as possible, with the ability to strip components if they are not required for a specific device.

To support evolvability and dynamic composition based on high-level targets, the platform will take a *bottom-up* approach for translating those targets into low-level components. To do this, functions are required for a set of high-level goals, will be split into atomic processing elements, called functional units. Functional units will be dynamically composed to reassemble the high-level goal functional requirement. This will also maximise reusability of functional units that collect or process information.

Multiple functional units will have the ability to dynamically compose, with the assistance of the platform middleware, into a *container*. *Containers* will reassemble the low-level equivalent of a high-level goal (or part of it). Multiple containers may be stacked in layers together, increasing complexity as the container layers move upwards. This is required so to achieve capabilities by assembling multiple lower-level processing units together, for example one container might keep statistics for traffic, while a higher-level container might do profiling based on these statistics. A higher-level container might provide decision logic to act upon a change in the profiler results.

Functional units must support at least one of the following three categories, in order to support expose a set programming interfaces that would enable it to run within the middleware:
- **Sensory:** It will support collecting data for a specific data source type only. They will support variable granularity level and the collection will be co-ordinated from cognitive units in the same composition group. Sources of sensory might be directly sources on the host machines, or other cognitive processing units in a different container.
- **Cognitive:** These functional units will provide basic cognitive functions. They will be responsible for processing the information collected from one or more sensory units and produce a decision whether or not to act based on the high-level target they are assigned to.
- **Configurators:** These units will have the ability to adapt, modify or create new low-level configuration for a specific function on the host device. How they are going to configure the function depends on the output of the cognitive functional units.

Communication between functional units and/or containers will be exclusively based on the *query language* that will be created for this purpose. Each functional unit will expose one or more *information points*, each support a specific combination of the below features. The query language will be standard for all functional units, and will have the following features:
- **Permanent query:** This will determine if the query is permanent, i.e. like a standing order. A permanent query will continue to execute on functional units on set intervals. Each interval tick will cause an upstream event, so that the requestor can collect the data produced.
- **Output format:** This will set the output format of the query.
- **Detail:** This will set the detail level. Higher detail level means greater data resolution.
- **Request type:** This will set what type of information is being requested from the functional unit. If the unit does not support this type of requests, it will respond with an error message.
- **Context and Semantics:** This is required as in order to achieve interoperability between various management domains and nodes, context and semantics information will be required.

High-level targets and goals will be provided to the platform via the human supervision interface in a machine readable format. A translation algorithm will identify atomic components needed to perform the goal provided, and it will dynamically compose the necessary containers so that it fulfils to the best possible way the goal. To support this dynamic composition, each of the functional units running within a management entity will have to publish the supported request types, detail and output formats so that it will support the dynamic composition. To increase the speed of the execution, containers will be dynamically compiled together during run-time, however still allowing the dynamic replacement when a different target is set or a newer unit is available in the pool.

Permanent queries between functional units and containers will allow the formation of multiple control loops, which will target the continuous optimisation of the system towards its given high-level target.

The platform middleware will expose a series of supporting functionality that will allow the functional units to organise and perform towards a target. The following mechanisms will be exposed by the middleware:

- **Communication primitives:** Will provide basic communication, based on the IP protocol. These communication primitives will allow devices to organise into management domains and create network management overlays (which can be mapped 1:1 to the physical network or span to multiple physical networks) to support the organisation. These domains will be created under the direction of the human operators: they will instruct the devices which domains to join. If a new management domain is to be created, a high-level description must be given to the first device upon the creation of the domain, in a machine readable format. This description will be automatically replicated to new devices upon the direction and authorisation to join an existing management domain.
  Multiple management domains might be joined into a bigger super-domain, while still maintaining their local identity and scope. This is useful in large corporate and commercial networks, where they device their bigger network into smaller counterparts for additional flexibility. The middleware will also support access-levels per information point, so domain administrators might expose just the information they want to other networks or domains.
- **Data Storage:** The ability to provide cloud-based data storage for the functional units. The data storage must be able to span across multiple participating nodes and have the ability to support distributed queries to relieve load on participating nodes by redirecting queries to more lightly-loaded or higher capacity nodes.
- **Query support:** Middleware will implement the query language, and will have the ability to execute queries against the data kept by the targeted functional unit.
- **Trust and security:** The role of the security officer in this dynamic network will be the burden of the middleware. It will regulate and filter the queries and communications that originate from functional units in order to meet a specific security policy enforced by the human operators.
- **Composition:** Given a high-level requirement, the middleware must be able to decompose it into lower-level components that if put together will be able to accomplish the high-level objective.

Figure 1 shows how functional units are organised by the Autonomic Network Management middleware into containers. Containers are organised together in order to achieve and work towards a high-level objective.

IV. Future Work

The platform implementation is currently under development, with an early prototype for testing concepts available. As a first stage, a more thorough requirements and specification document is going to be produced, alongside a prototype reference platform that will support the platform document. When a stable platform is achieved that can support to a satisfactory degree the requirements and goals defined in high-level in this paper, the development will focus on implementing specific functional units to support high-level goals such as resilience, based on a given scenario.

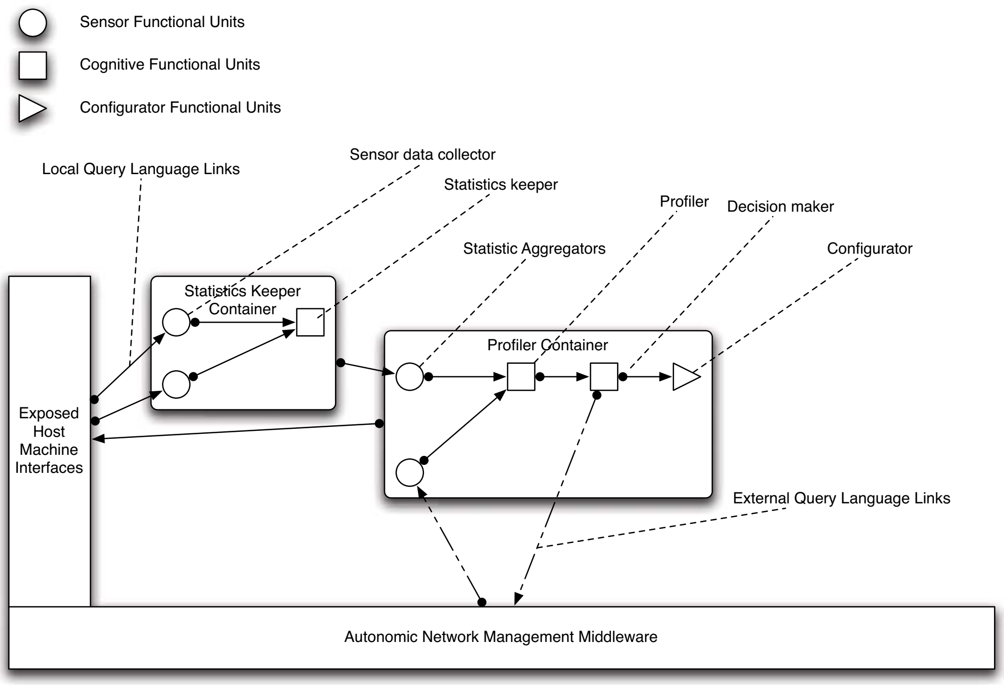
The first revision of this reference research platform for Autonomic Network Management will be written in Google's Go [1], in order to support rapid prototyping and take advantage of new, novel features introduced in this programming language, such as remote transparent communication channels, which will ease the development and reduce software's complexity.

Evaluation of the platform will target firstly its performance, making sure that it meets its lightweight target. Later, when a pool of functional units are available, subsequent evaluation targets will be defined, relevant to the targets of these units and the high-level goals of the user.

V. Conclusion

This paper discussed briefly the current state of evolution in computer networks, and the problems with that rapid expansion. It discusses the solution based on IBM's Autonomic Computing Initiative and the subsequent works on creating autonomic network management platforms that reassemble part of these ideas. However, there is no de-facto platform for autonomic network management. This paper introduces a more generic platform that could be used for a testbed of prototypes and development of future concepts and ideas with regards to this area. A set of requirements of this platform have been briefly discussed, namely the requirement for a network-wide perspective, applications involvement, cognitive support, advanced sensory support, increased transparency so to ease human supervision and finally information interoperability between running instances and network devices. Finally, an architecture is presented, which will provide a set of basic functionality to facilitate the creation of management containers, which will reassemble a high-level goal. This middleware will fulfil the requirements discussed, as well as provide some necessary support to developers such as: management overlays, recomposition and functional extensibility and finally trust and security. The software that runs on this platform is decomposed into atomic functional units, which are then recomposed dynamically to achieve a high-level goal. The communication between these functional units will be in the form of a query language, which will also be the interface to storage as well as making sure that information is interoperable and has all extra meta-data attached. Finally, the future work

[1]The Go Programming Language, available online at: http://golang.org



Sensor Functional Units
Cognitive Functional Units
Configurator Functional Units
Local Query Language Links
Sensor data collector
Statistics keeper
Profiler
Decision maker
Statistic Aggregators
Configurator
Statistics Keeper Container
Profiler Container
Exposed Host Machine Interfaces
External Query Language Links
Autonomic Network Management Middleware

Fig. 1. Autonomic Network Management Architecture

was briefly outlined, with a more comprehensive specifications document and a reference prototype implementation, on which future evaluation will be targeted.

ACKNOWLEDGMENT

The authors would like to thank Frankie Garcia from Agilent Technologies and Manolis Sifalakis for their constructive discussion on this management platform and the members of IU-ATC [1].

REFERENCES

[1] D. Clark, "The design philosophy of the DARPA Internet protocols," in *Symposium proceedings on Communications architectures and protocols*. ACM, 1988, p. 114.

[2] R. Mortier and E. Kiciman, "Autonomic network management: some pragmatic considerations," in *Proceedings of the 2006 SIGCOMM workshop on Internet network management*. ACM, 2006, p. 93.

[3] N. Agoulmine, S. Balasubramaniam, D. Botvich, J. Strassner, E. Lehtihet, and W. Donnelly, "Challenges for autonomic network management," in *Proc. of the 1st IEEE International Workshop on Modelling Autonomic Communications Environments (MACE), Dublin, Ireland*. Citeseer, 2006.

[4] J. Kephart and D. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–50, 2003.

[5] P. Horn, "Autonomic computing: IBMs perspective on the state of information technology," *IBM Corporation*, vol. 15, 2001.

[6] D. Clark, C. Partridge, J. Ramming, and J. Wroclawski, "A knowledge plane for the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, p. 10.

[7] D. Isenberg, "Rise of the stupid network," *Computer Telephony*, pp. 16–26, 1997.

[8] M. Parashar and S. Hariri, "Autonomic computing: An overview," *Unconventional Programming Paradigms*, pp. 257–269, 2005.

[9] X. Li, H. Kang, P. Harrington, and J. Thomas, "Autonomic and trusted computing paradigms," *Autonomic and Trusted Computing*, pp. 143–152, 2006.

[10] P. Flegkas, P. Trimintzios, and G. Pavlou, "A policy-based quality of service management system for IP diffservnetworks," *IEEE network*, vol. 16, no. 2, pp. 50–56, 2002.

[11] J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, and G. Pavlou, "A methodological approach toward the refinement problem in policy-based management systems," *IEEE Communications Magazine*, vol. 44, no. 10, p. 60, 2006.

[12] M. Wang and T. Suda, "The bio-networking architecture: A biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications," in *Symposium on Applications and the Internet (SAINT)*, 2001, pp. 43–53.

[13] J. Strassner, N. Agoulmine, and E. Lehtihet, "FOCALE–a novel autonomic networking architecture."

[14] S. Davy, K. Barrett, S. Balasubramaniam, S. Van Der Meer, B. Jennings, and J. Strassner, "Policy-Based Architecture to Enable Autonomic Communications–A Position Paper," *Proceedings of IEEE CCNC, special session on Autonomic Communications, Las Vegas, USA, January*, 2006.

[15] H. Derbel, N. Agoulmine, and M. Sala
"un, "ANEMA: Autonomic network management architecture to support self-configuration and self-optimization in IP networks," *Computer Networks*, vol. 53, no. 3, pp. 418–430, 2009.

[1]India-UK Advanced Technology Centre: Available online at http://www.iu-atc.com/