J. Basic. Appl. Sci. Res., 3(9)270-279, 2013

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

An Autonomous, Self-directed, Ant-optimized Adaptive Learning-based Intelligent Network Architecture (ASAALI)

Zeshan Iqbal[1], Adeel Akram[2]

[1,2]Department of Computer Engineering, University of Engineering and Technology, Taxila, Pakistan

ABSTRACT

Autonomic network management systems are currently in focus seen in the enterprises and are one of the prime areas of research in this field. Achieving the complete functionality from the self-management perspective is still unachievable and opens many research areas. In this paper, we introduce ASAALI management system, a comprehensive learning-based intelligent network management system model for autonomous, self-managed, adaptive and secure network management. This novel network management system model is more efficient in all aspects from existing architectures.
KEYWORD: Autonomous, network management system, self-directed system, self-managing, self-learning, self-protecting, secure system model

INTRODUCTION

In the present era of communication and IT, the corporate networks have evolved to become a mesh of several types of networks of diverse nature. Technologies like Optical Fiber, DSL, Ethernet, Wi-Fi, WiMAX and CDMA are commonly used in the same network as underlying core technology and sometimes even a combination of different networks joining together to provide a complete solution. But at the end, they all work together using generally accepted standard TCP/IP protocol stack. This makes all these technologies interoperable but also puts limitations on them, and thus makes network management more difficult and challenging for business organizations.

Autonomous network management (ANM) is the most appropriate approach for the reduction of complexity and cost of network management solutions. This approach not only used to manage network loops autonomously but also reduces human interaction in management tasks. This important effect of network management area has directed both academia and industry to focus their research in making this field mature enough to resolve the problems currently being faced by the existing network owners.

Autonomous and self-directed network management platform that are based on ant-optimization self-learning method [1] designed with proactive and predictive approach allows the devices to take decisions based on the previous analysis of data. Computation is done on each node through intelligent agent programs. These agents are self-managed in nature.

The aim of this research is to develop an autonomous and self-directed network management platform that is intelligent in nature to self-manage its network resources by itself.

Here we introduce a comprehensive multi-layered adaptive learning-based intelligent network management system which is an Autonomous and self-managed. This was done by careful evaluation of existing autonomous NMS architectures, discussed in literature to propose a novel network management system model. Our research work mainly focused on identifying the benefits and drawbacks of different autonomic network management systems by evaluating them through a qualitative method and comparing with our proposed model.

First section gives us the background knowledge of ANM system. Second section is about our model proposed. Third section validates our system by qualitative analysis access method. Fourth section is about the simulation of system that we developed for validation. Section 8 is the conclusion of paper.

BACKGROUND

The existing network management techniques have changed network management quite significantly; however people are still developing new techniques and technologies in this area. The focus of this work has recently started shifting towards autonomic management systems. IBM was the first to launch its initiative for autonomic computing [2]. Autonomic network management is the specialized area of autonomic computing and is much more complex since it needs more independent environment than the autonomic computing.

Several methods are designed by the researchers based on two approaches, *flat architectures*: autonomic managers over the network are working at same level thus forming a distributed network and *hierarchical*

*__Corresponding Author:__ Zeshan Iqbal, Department of Computer Engineering, University of Engineering and Technology, Taxila, Pakistan

architectures: managers are working at different layers thus communicating with each other and playing specific roll that is assigned to them. Various architectures based on these two approaches are found in literature.

Autonomic Internet (AutoI) is one of the FP7 European projects proposed for the management of future networks autonomously [3]. This project holds its uniqueness due to two properties [4] that is it was designed for heterogeneous networks and helps in the management of resources and services virtualization. AutoI uses 2-tier architecture in a distributed hierarchical manner [5].

DRAMA (Dynamic Readdressing and Management for the Army) is a hierarchical policy based network management system designed for MANETs using intelligent agents and is distributed in nature [6]. DRAMA also uses the concept of clustering when organizing the nodes.

CA-MANET (Context-Awareness of MANETs for the Autonomic Management) is also a hierarchical distributed policy based network management system. It applies three-tier architecture in its design. It also forms hyper-cluster in creation of autonomous networks [7].

Autonomous Decentralized Management Architecture (ADMA) is a distributed flat autonomous management architecture designed to provide MANETs with support of some autonomic regulations [8].

The *Autonomic Network Architecture (ANA)* [9] is one of the FP6 EU-IST funded project proposed for future next generation networks. The purpose of developing ANA is a simplified management architecture that support autonomic behaviors to connect multiple heterogeneous networks [10].

In-Network Management architecture within 4WARD project is designed as Wired and Wireless and World Wide Architecture and Design [11] and is one of the FP7 European project. It proposes developments to improve the process of distinct network architectures as well as the co-occurrence, connectivity and necessary interrelationship in heterogeneous environment.

MOTIVATION

The models so far designed do not completely cover the self-management issues. Each model focuses on one or some perspectives of the self-management areas defined by IBM [12, 13].

The model we proposed not only covers all the perspectives of self-management. It also focuses on the management of cloud networks. Autonomous management platforms so far proposed also focuses on the policy based management of networks. However we used an ant optimized learning-based system for the adaptation purpose in our management system.

PROPOSED MODEL

In the proposition of our model we used three-tier architecture. The names layers are Overlay Network Layer (ONL), Knowledge Network Layer (KNL) and Adaptation and Planning Network Layer (APNL) as shown in Figure 1**: An Autonomous, Self-directed, Ant-optimized Adaptive Learning-based Intelligent Network Architecture (ASAALI)** below.

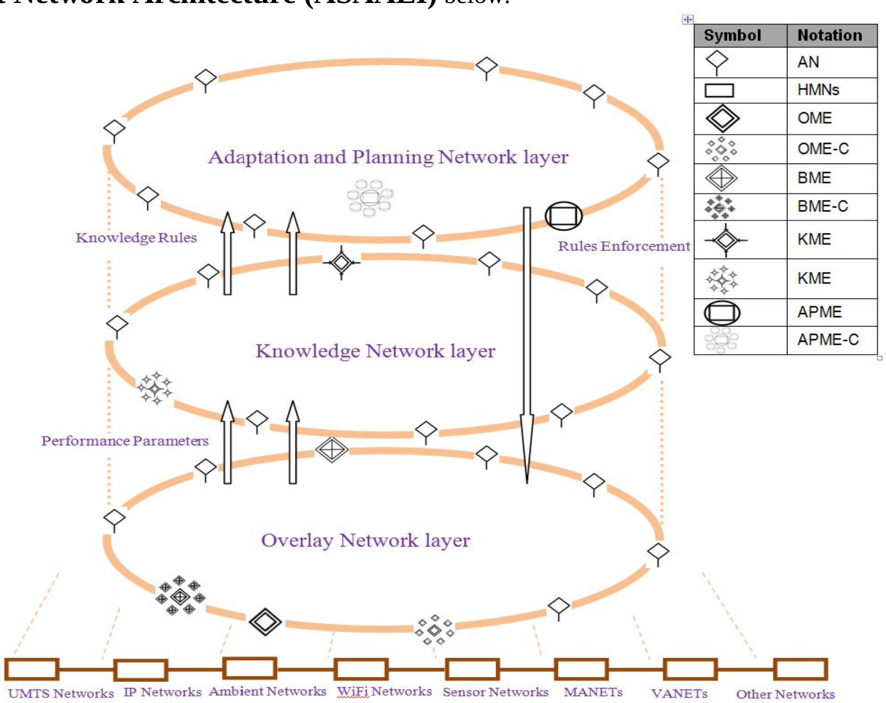

Figure 1: An Autonomous, Self-directed, Ant-optimized Adaptive Learning-based Intelligent Network Architecture (ASAALI)

The functionality and responsibilities of each layer are different. Since it is a three-tier autonomous network management model applied in hierarchical manner, at every layer, several Autonomic Entities (AE) are functioning to achieve each goals defined for the respective layer. The role of AE is defined as, Management Entity (ME) and Autonomous Node (AN) which will be further explained in the relevant layer later on. The architecture of each AN contain MAPE-K model and its functionality [14]. The MAPE-K model control loop communicates with the underlying networks by help of manageability interface (sensor and effector). IBM, the autonomic computing architecture initiator has already proposed manageability interface of an autonomic entities as per OASIS WSDM standard [15].

Based on the OASIS WSDM technical specs, the managing competencies attainable from the *manageability interface* include detection, configuration, metrics, status, procedures, and activities created through manageable entities regarding managing needs. The layers are explained in detail below:

1. Overlay Network Layer

The ONL is a base layer. Building an autonomous managed network depends on this layer. In the initial step at ONL all available networks are scanned, a virtual network is established by selecting nodes from the underlying Heterogeneous Managed Networks (HMN) found during scan, these nodes must have ability to compute for autonomous network management. Elections are conducted among the available nodes of respective network for which normalized weights are assigned to any performance parameters (e.g, Key Performance Indexes) set forth from that network. The node with highest cumulative weight is selected as AE and the value found as lowest cumulative weight can be selected as a threshold point for that network.

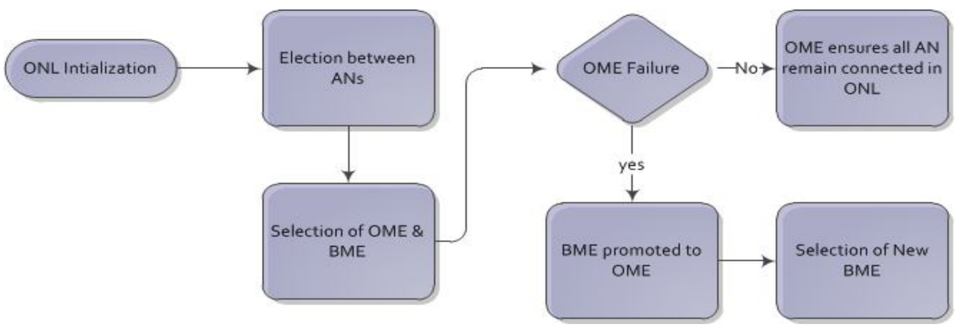

Figure 2: Work Flow of ONL

This criterion for node selection can vary from network to network and thus its measuring components will also vary in nature. After selection, the node will be monitored by ONME and if value of weight reaches to the threshold point defined earlier for that particular network, it will start scanning for nodes again and thus execute whole process again in the same manner for selection of next AE from respective network.

This layer contains following components with brief description of each component:

1.1 Overlay Network AN

Nodes during the creation of ONL are selected based on two properties, availability of computation ability and a part of nested heterogeneous network. If the performance of a node is declining with time, the node status immediately changes to standby node and start searching for other node. This Overlay Network Autonomous Nodes (ONAN) when selected from the HMNs make a virtual network, which here we call ONL. The responsibility of nodes is to implement the policies given to it from the upper layer to the network which it belong.

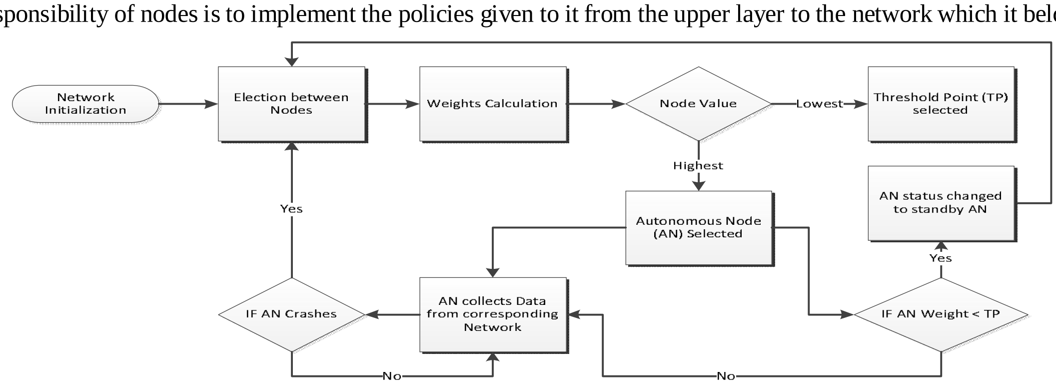

Figure 3: Selection of Autonomous Node

1.2 Overlay Management Entity

Among all the nodes selected from heterogeneous networks, there are few entities selected as Management Entities. The responsibility of these entities is to manage ONL. The role of OME is to manage the virtual network nodes. These OMEs will check periodically the status of nodes, if any node is not responding or is taking time in responding; OME will select anc as ONAN from the respective network based on previously defined criteria provided its cumulati exceeds the threshold point selected for ONL and give it a role of ONL node and share the concerned policies with it. In large or complex networks clusters of OMEs can be created so that they can balance the load with each other.

1.3 Backup Management Entity

For fault tolerance, Backup Management Entities (BME) will be created as if ME fails, system shall remain online. BMEs also create clusters in large networks so that they can easily replace ME clusters on failure.

2. Knowledge Network Layer

KNL layer is responsible for the knowledge gathering, building repositories and later analysis of these knowledge repositories. At this layer several Knowledge nodes will be working, each for different network. Every node will be playing its part in data collection. The creation of rules will be done on the basis of the data analysis for policy generation. These rules will then be passed on to the next layer for further decisions. Building knowledge based management system is highly acknowledged [16].

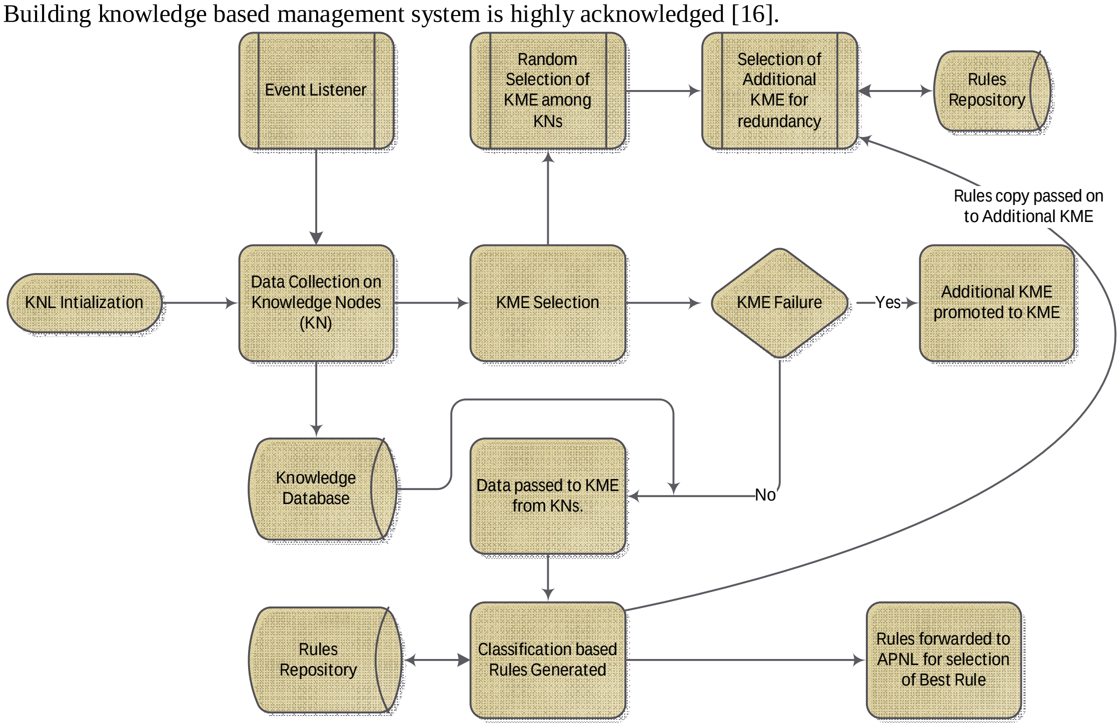

Figure 4: Work Flow of Knowledge Network Layer

2.1 Knowledge Management Entity

The Knowledge Management Entity (KME) is the one that is responsible for collection and analysis all the networks datasets repositories, that is, it will not only collect these repositories but also use them for analysis purpose. An efficient Ant Colony Optimization based classifier called AntMinner-CC is used for the classification of data thus making sure that a correlation can be built between the available data sets of heterogeneous systems [1]. This classifier creates rules from the data sets of each AN and optimize them to adapt best rules. KME will forward these rules created for decision making to the higher level. For redundancy purpose additional KMEs are used in network. In complex networks clusters of KMEs are built so that they can do the load distribution with in nodes. If any KME or KME cluster gets down, the redundant KME or KME cluster will take its place and start working so that network shall be working uninterrupted. System will create another redundant KME and pass its data to this next additional KME so that it can replace active one on failure.

2.2 Knowledge node

Knowledge node is in knowledge network layer. The role of node as in KNL is changed. Here it collects data from each network and transfer this data to KME for analysis.

3. Adaptation and Planning Network Layer

This layer is the final layer where actually decision making will be done. This layer opens gateway to the self-management abilities in autonomous management. This layer will be using the predictive techniques to predict the future issues, networks will be facing, based on the experience of other type of networks; it will pass the rules to them for uninterrupted routines. The ac ınd planning both can be achieved by this layer. It will have its own management entities.

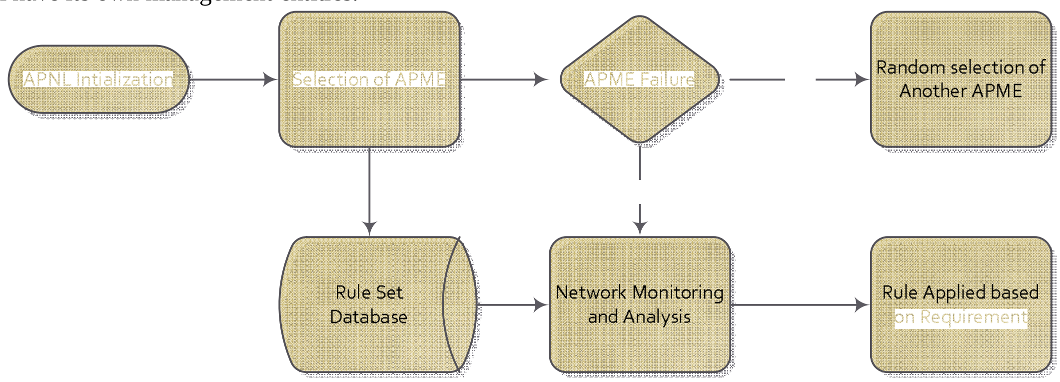

Figure 5: Work Flow of APNL

3.1 Adaptation and Planning Management Entity

The Adaptation and Planning Management Entity (APME) is the one that will be acting as Management Entity in the APN layer. APME will be running here the same classifier used by KME for the selection of best rule among the available options to implement [1]. This is a self-learning entity that will tell ANs what to apply on underlying network. APME also creates its own cluster for data planning and distribution and for adaptation in complex network scenarios.

4. Our Approach for Knowledge Management

In our system first of all we have identified the type of errors to handle data. Four types of errors are identified in this process, that is, *the hardware error*, *the software error*, *the communication error and intrusion error*, based on these errors system builds its knowledge repositories. But first of all we will explain the nature of these errors.

The *hardware errors* are the most common errors that any machine in the heterogeneous network faces. These errors will be recorded by the AE of concerned network. The AE will then transform these errors to KNL for policy making process. The format of data for each network will be different but at KNL the data will be analyzed so that right format for policy creation shall be adapted.

The *Software errors* cover the errors in any software application, relevant services and configuration files /even registries (in case of few operating systems like Windows). These errors also have diverse nature as these errors belong to different applications running over different operating systems in heterogeneous network environment.

The *communication errors* are errors recorded at network level. These errors are about the communication line error, protocol error or any parameters that can identify the performance issues.

The *intrusion errors* are handled by introducing IDS in our system for protection of network management system and will be discussed later.

For each type of error, system will have a separate repository. These repositories will be the first step towards rule creation.

5. Self-Management of our model

As mentioned earlier the self-management includes 4S strategy. We tried to cover all the aspects in design, the details about these characteristics are as follows:

5.1 Self-Configuration

The KNL whenever gathers information to create set of rules from the member node entities, it refines this specific knowledge for the higher level. The higher layer utilizes these knowledge repositories for rule generation. The decision for the effective use of rule is dependent on Adaptation and Planning Network Layer.

The APNL specifically passes guidelines to the groups where they are required for usage thus making networks self-configurable in nature.

As an illustration let's discusses an example of an Ethernet network, where a Wi-Fi router can be installed that links the Wi-Fi nodes to IP network. This router requires an initial set up for adjustment with the existing network. Now as administrator installs the router, he applies certain setup instructions designed for installation purpose, hence unintentionally creating a new rule for the Wi-Fi router installation. This rule is passed to APNL where the guidelines can be prepared for any network that needs to install new Wi-Fi router in it without administrator assistance. All needed is to get a fresh router and connect it to any network.

5.2 Self-Optimization

Self-Optimization is applied at two different levels in our model. At first ONL layer needs to be optimized. At this layer when AEs are selected from different heterogeneous networks, a criterion is defined for their selection as explained earlier is used and therefore optimizing the ONL for further use in network management. Here optimization of resources by the layer components itself shows Self-Optimization ability of network.

The second level of Self-Optimization is achieved at the APNL layer where policies are created for same type of problem but for a different type of network.

The priority and order of any rule may vary for same type of problem in different networks. Subsequently, relevant decisions to be taken at different levels, makes the system very complex to handle these rules and to decide that which rule is most appropriate for which scenario. Thus selection of best rule ensures the success of our system.

5.3 Self-healing

Identification of errors by KNL and taking measures against these errors by the APNL thus make system self-heal in nature.

5.4 Self-Protection

The Self-Protection can be managed by introducing Intrusion Detection System (IDS) in ONL. Since ONL is the layer where nodes are creating a virtual network so detecting and countering intrusion attack at this layer will be more feasible. Here Snort, an open source network intrusion detection and prevention system (NIDS, NIPS) is used as part of our application. Since Snort can perform protocol analysis, content searching and content matching over real time traffic, which makes it as the best application to integrate in our system than any other application available. Snort might be set up in three different modes: sniffer, packet logger and NIDS. In sniffer mode, it examines network packets and shows them on the console. In packet logger mode, it records packets to a raw file. In NIDS mode, it keeps track of network traffic and evaluates this with a procedure set described by the autonomous system. It then executes particular measures depending on exactly what is discovered. Thus Self-Protection can also be achieved by the help of our KNL layer that builds a knowledge repository for the ONL to protect the system from intruders.

EVALUATION AND COMPARISON OF OUR MODEL WITH OTHER AUTONOMIC NETWORK ARCHITECTURES

For evaluation purpose *Qualitative method of assessment* is used that comprises with evaluating distinct systems features or even components using a non-numeric as well as individually distinct scale (e. g. classifications, intensities, etc.).We here tend to initial determine the primary qualitative assessment factors causing the degree of Autonomicity associated with ANM architectures. Furthermore, we tend to evaluate as well as assess the actual attributes regarding surveyed architectures using the recognized list of qualitative standards that will help us in elaborating the taxonomy of autonomic network approaches.

Comparing ANMs with each other through quantitative assessment is not possible. It is observed that not all of these architectures are open-source and restrict us to stay on the figures provided by the authors. Developing them again is impossible for any research group as it needs a lot of time and money to invest. Thus measurement of performance of all the architectures is impossible to achieve.

1. Architecture based Categorization

This specifies the overall architectonics of previous autonomous approaches along with our approach. ANM architectures are being split in to two approaches, that is, flat and hierarchical. These approaches may be used being an assessment metric given that they represent two different school of thoughts associated with network management, each one of these featuring its pros and cons. The hierarchical approach works on the traditional administration style where a main server supervises and handles routines. The system is being converged for an optimum solution as all the choices are created with a centralized supervision by having a general look at whole network. Nevertheless, this method is affected with scalability problems within big networks and the issue of the solitary stage results in failure of management system. To handle this problem, the recommended solution is to form a cluster of managed nodes managing different networks. Nevertheless, another level of management is required for the cluster management. In contrast to the hierarchical architectures, there are few ANM architectures that follow the other approach of flat architecture. The primary concern in flat approaches is always to determine the system that can combine the choices and activities of every manager. The effectiveness of architectures discussed here depends upon the system designed to overcome its own constraints.

It was found that the ADMA, ANA and INM are the examples of flat architectures. On the other hand AutoI and CA-MANET falls in the distributed hierarchical network architecture, and only DRAMA follows a centralized hierarchical approach as it defines a central point of management. This is due to the scalability issue that a centralized architecture face thus creates an immense traffic over the network. It is also observed that these hierarchical architectures are designed as 3-Tier architectures. We also used the 3-tier distributed hierarchical management architecture approach due its high advantage over the other designs.

2. Interest based focus

Designing a complete and comprehensive autonomous architecture remains a challenge for researchers. However with the passage of time, researchers divided this goal into steps to achieve it. These goals later defined comprehensively by IBM as self-management characteristics. ANM architecture discussed in literature uses a subset of self-management tasks and we here refer it as *interest based focus*. For comparison it is necessary that each ANM shall be using the same self-management tasks to implement Autonomicity.

The AutoI, DRAMA, CA-MANET and ADMA are designed with primary focus on self-configuration perspective. However ANA is self-configuring as well as self-optimizing management architectures designed for the underlying networks. In INM self-healing is also implemented that makes it even better than the other architectures. It is observed that every architecture implements self-configuration characteristics, this is because it is the fundamental step towards the self-management and all other characteristics of self-management are influenced by this process.

We followed complete 4S approach of self-management as discussed above. Our designed architecture is a self-configuring, self-optimizing, self-healing and self-protecting in nature thus finishes the Autonomicity of network management.

3. Intended Situation

Network management architectures are designed with an approach that they target specific environment. Thus any management architecture should be evaluated based on the network for which it is designed. For example, flat architecture will be quite helpful for ambient network management but if we select a LAN with hundreds of nodes, it will still be manageable efficiently with a centralized hierarchical management system. It is also observed that flat architectures are selected in cases where network are dynamic and large scale in nature.

It is observed that AutoI and INM are designed for both static and dynamic networks however DRAMA, CA-MANET and ADMA are considered best for only dynamic networks. CA-MANET is best for low mobility based environments, DRAMA on the other hand are tested and designed for group mobility pattern based networks. For large scale network, ANA is considered as best among all the others as is tested for large scale environment and is a flat architecture [17]. We have particularly designed management system with an approach that it should work best with in the cloud network based environment.

4. Adaptation methodology

Generally, adaptation is referred as a procedure used by any node to make it familiar and acceptable by surrounded environment [18]. Autonomic networks refer adaptation as knowledge of network used to take decision about what action shall be performed and in what scenario it will be the best solution to adapt. As far as MAPE-K model perspective, adaptation consist of analyze and plan components.

For self-adaptation mechanism, Policy based methodology is used by mostly autonomous management systems as ANA, ADMA, DRAMA, CA-MANET, INM and AutoI apply it in their architectural designs. Two reasons are observed for this adaptation selection. First of all the adaptation plan is very simple; it uses an incident, condition and achievement method for adaptation. Secondly it modifies previously build policies at run time and applies them on network without any interruption thus reconfigures network on fly [19]. We have used an ant-optimization based learning mechanism for self-adaption purpose. The main purpose of learning-based approach is that policy-based adaptation fails in heterogeneous network environment where an adaptation rule works quite well in one network but completely fails in the other however learning-based mechanism design rule on previously gathered knowledge that help them creating the different versions of same rule for different networks [20].

5. Monitoring methodology

As discussed previously, monitoring of network can be done by capturing all kind of data (hardware, software, communication etc.) that is helpful for network management autonomously of any heterogeneous network. Two main approaches are followed in this context:

- First approach spreads knowledge to all nodes in network. Gossip-based aggregation [21], selective broadcasting [22] and situated view [23] approaches belongs to this group.
- Second approach spreads knowledge to specific group of nodes in network. This type contains distributed context repository [20] and tree-based aggregations [24] etc.

It was found that AutoI uses Information Model Ontologies for monitoring purpose, CA-MANET uses XML-RPC, ANA uses tools supported by MFB, INM uses Gossip-based and tree-based aggregations. However the granularity observed for monitoring is semi-network wide for all these above architectures. This is due to avoiding an extra overhead of replicated policies f[illegible]em.

DRAMA uses YAP Protocol, ADMA uses DPMP Protocol and both have a full network wide scope for monitoring purpose.

The monitoring approach we used in Autonomous management system designed is semi-network wide. We monitor any network by reading log files created over the system instead of using any protocol.

6. Convergence mechanism of Network

Convergence is required in distributed architectures since the centralized are already converged networks in nature. In distributed architecture each manage node is running its own MAPE-K loop thus each loop needs to be updated with the other manager decisions for optimized data.

AutoI defines its own distribution and negotiation mechanism for network convergence. However DRAMA is a centralized architecture so its autonomic manager uses distributes the extracted policies by using DRCP/DCDP. CA-MANET is the distributed architecture and so use distributed managers to coordinate with each other using uniform distributed DPRs. ADMA, ANA and INM are the flat architectures and therefore each one has its own mechanism to define network convergence.

Our architecture is also distributed in nature and therefore managed nodes of each network in heterogeneous environment disseminate rules based on uniform distributed DPRs.

7. Discovery mechanism

Discovery mechanism is the ability of system here defines a mechanism to determine learning required for its decision process in network management. These abilities are used mainly to congregate the optimum composition of data for future use.

8. Security mechanism

Security mechanism determines how to defend system in case of any external attack. If such mechanism is defined then system needs to identify processes involved in protecting management system. So far we have seen that there is currently no network architecture available that covers this aspect. However we have used a security mechanism for our autonomous management system called Snort, an open source system. The main reason for its selection is that it is a rule based IDS.

9. Heterogeneous System

Any autonomous system when manages several heterogeneous networks, it needs a mechanism to translate the language of one network to other in order to develop the interaction between different networks. Since each network has different attributes defined from other networks thus dealing this issue is quite complex and due to this problem of heterogeneity [5] several methods are designed in autonomous architectures, some architecture ignores this characteristic and emphasis on the self-management of networks thus making their focus more on homogenous systems. Only AutoI, ANA and INM are designed with considering this factor that they should support the heterogeneity in management system. Others ignored this factor. We have also adopted this approach and designed with this consideration that system should be heterogeneous in nature and support different systems all at once.

10. Open Architecture

Several architectures follow the concept of open system so that people give there feedback for the improvement of future applications of their architecture. AutoI, INM and ANA are the open-source architectures, except these others are the proprietary products of different research institutions. Even some parts of INM are also proprietary. We also respect the open system approach and our architecture is to some extent available as open-source but few portions are the proprietary product of University.

11. Adaptive Evolution

The property of any architecture to support its evolution so that supplements can be added and improvements can be made in future. This property also makes the architecture scalable reference to new technologies coming in the market. Architecture is considered evolvable if its core components are designed in such a way that middleware software shall be compatible thus any new algorithm or enhancement is architecture can be integrated effortlessly. AutoI, ANA and INM are evolvable architectures. We also considered evolvability in our architecture and design with this perspective that it can evolve when more new networks come forward and join the heterogeneous environment.

12. Validation

Every model is needed to be validated so that the assumed features can be tested. The architectures so far designed are validated either through mathematical r[illegible] imulations. For validation of the systems, they are also tested in different networks which also testify t[illegible] ıbility. None of the above identified Autonomous architecture used comparison method with the other existing architectures to validate their model or if compared, they used conventional management system for comparison.

SIMULATION

We have simulated our architecture by designing a real time application solution using Visual C# that gathers data from different network and after analysis passes rules to top hierarchy distributed servers. We designed three types of applications to apply our model. The *client*: running over the AN after its selection for data collection, *network level autonomic manager*: in KNL, this node is called KME, used to manage rule sets after data sets are passed from classifier to create rule sets and an *adaptation layer manager*: that implements the appropriate rule to each node in heterogeneous system. This is a first prototype so it only covers the basic features of our autonomous management system.

CONCLUSION

Autonomic networks are highly in focus of research community but still achieving the complete and optimize way the self-management functionalities is a long way ahead. We here provided a holistic view of research in the area of autonomic network management (ANM) architectures, with the focus on identifying the advantages and disadvantages of different autonomic management systems by accessing it through a qualitative method.

We noted that the existing architectures focus on policy based system instead of learning mechanisms. Also security features are also not included by any architecture which is a requirement to any model autonomic solution. We therefore used learning and security mechanism to improve the performance towards the optimized level.

We tried to cover the aspects of cloud networks but still needs an improvement as cloud networks are still moving towards there maturity.

REFERENCES

[1] A. R. Baig, W. Shahzad, and S. Khan, "Correlation as a Heuristic for Accurate and Comprehensible Ant Colony Optimization Based Classifiers," *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 2012.

[2] P. Horn. (2001, 15th October 2001). *Autonomic Computing: IBM's Perspective on the State of Information Technology*. Available:

http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf

[3] *Autonomic Internet*. Available: http://ist-autoi.eu/

[4] A. Bassi, S. Denazis, A. Galis, C. Fahy, M. Serrano, and J. Serrat, "Autonomic internet: A perspective for future internet services based on autonomic principles," in *Proceedings of 2nd IEEE International Workshop on Modelling Autonomic Communications (MACE)*, 2007.

[5] Z. Movahedi, M. Abid, D. F. Macedo, and G. Pujolle, "A policy-based orchestration plane for the autonomic management of future networks," presented at the 6th International Workshop on Next-Generation Networking Middleware (NGNM), 2009.

[6] R. Chadha, Y. H. Cheng, J. Chiang, G. Levin, L. Shih-Wei, and A. Poylisher, "Policy-based mobile ad hoc network management for drama," in *Military Communications Conference, 2004. MILCOM 2004. 2004 IEEE*, 2004, pp. 1317-1323 Vol. 3.

[7] A. Malatras, A. M. Hadjiantonis, and G. Pavlou, "Exploiting Context-Awareness for the Autonomic Management of Mobile Ad Hoc Networks," *Journal of Network and Systems Management,* vol. 15, pp. 29-55, 2007/03/01 2007.

[8] M. Ayari, Z. Movahedi, G. Pujolle, and F. Kamoun, "ADMA: autonomous decentralized management architecture for MANETs: a simple self-configuring case study," in *Proceedings of the 2009 international conference on wireless communications and mobile computing: Connecting the world wirelessly*, 2009, pp. 132-137.

[9] *Autonomic Network Architecture (ANA) FP6 project*. Available: www.ana-project.org/

[10] M. Sifalakis, A. Louca, A. Mauthe, L. Peluso, and T. Zseby, "A functional composition framework for autonomic network architectures," in *Ne[illegible]erations and Management Symposium Workshops, 2008. NOMS Workshops 2008. IEEE*, 200[illegible]-334.

[11] *4WARD FP7 project*. Available: http://www.4wardproject.eu/

[12] H. Schemeck, "Autonomic computing - vision and challenge for system design," in *Proceedings of International Conference on Parallel Computing in Electrical. Engineering*, 2004, pp. 3–3.

[13] M. J. Khan, M. M. Awais, and S. Shamail, "Achieving self-configuration capability in autonomic systems using case-based reasoning with a new similarity measure," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, ed: Springer, 2007, pp. 97-106.

[14] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing - degrees, models, and applications," *ACM Comput. Surv.,* vol. 40, pp. 1-28, 2008.

[15] H. Kreger; and T. Studwell. (2005). *Autonomic Computing and Web Services Distributed Management*. Available: www.ibm.com/developerworks/autonomic/library/acarchitect

[16] J. Strassner, "Knowledge management issues for autonomic systems," in *Proceedings of Sixteenth International Workshop on Database and Expert Systems Applications*, 2005, pp. 398-402.

[17] G. Bouabene, Q. Zhang, and M. Scholler, "Final design, evaluation and prototype implementation of content-based routing mechanisms, Deliverable d.2.14," Tech. Rep.January 2009 2009.

[18] P. Oreizy, M. M. Gorlick, R. N. Taylor, D. Heimhigner, G. Johnson, N. Medvidovic, A. Quilici, D. S. Rosenblum, and A. L. Wolf, "An architecture-based approach to self-adaptive software," *Intelligent Systems and Their Applications, IEEE,* vol. 14, pp. 54-62, 1999.

[19] N. Samaan and A. Karmouch, "Towards Autonomic Network Management: an Analysis of Current and Future Research Directions," *Communications Surveys & Tutorials, IEEE,* vol. 11, pp. 22-36, 2009.

[20] A. M. Hadjiantonis, A. Malatras, and G. Pavlou, "A context-aware, policy-based framework for the management of MANETs," in *Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks, 2006. Policy 2006.* , Washington DC, USA, 2006, pp. 23-34.

[21] F. Wuhib, M. Dam, R. Stadler, and A. Clem, "Robust monitoring of network-wide aggregates through gossiping," *Network and Service Management, IEEE Transactions on,* vol. 6, pp. 95-109, June 2009 2009.

[22] W. Berrayana, S. Lohier, G. Pujolle, and H. Youssef, "Local View's dissemination for a network wide optimization to improve QoS in ad hoc networks," in *First International Global Information Infrastructure Symposium*, 2007, pp. 207-210.

[23] T. Bullot, R. Khatoun, L. Hugues, D. Gaïti, and L. Merghem-Boulahia, "A situatedness-based Knowledge Plane for autonomic networking," *International Journal of Network Management,* vol. 18, pp. 171-193, 2008.

[24] F. Wuhib, M. Dam, and R. Stadler, "Decentralized detection of global threshold crossings using aggregation trees," *Computer Networks,* vol. 52, pp. 1745-1761, 2008.