A Multi-Level Resilience Framework for Unified Networked Environments

Angelos K. Marnerides*†, Akshay Bhandari ‡, Hema Murthy‡ and Andreas U. Mauthe†
*School of Computing & Mathematical Sciences, Liverpool John Moores University, Liverpool, UK
a.marnerides@ljmu.ac.uk
†InfoLab21, School of Computing & Communications, Lancaster University, Lancaster, UK
a.marnerides2,a.mauthe@lancaster.ac.uk
‡Department of Computer Science & Engineering, Indian Institute of Technology Madras, Chennai, India
akshayb,hema@cse.iitm.ac.in

***Abstract*—Networked infrastructures underpin most social and economical interactions nowadays and have become an integral part of the critical infrastructure. Their continuous and uninterrupted operation relies heavily on the performance of the various networked elements that compose the internal and external functionality of the overall ICT system. Thus, it is crucial that heterogeneous networked environments provide adequate resilience in order to satisfy the quality requirements of the user. In order to achieve this, a coordinated approach to confront any challenges is required. However, there is additional complexity since challenges manifest themselves under different circumstances in the various infrastructure components. The objective of this paper is to present a multi-level resilience approach that goes beyond the traditional monolithic resilience schemes that focus mainly on one infrastructure component. The proposed framework considers four main aspects, i.e. users, application, network and system. The latter three are part of the technical infrastructure while the former profiles the service user. This paper illustrates how an integrated approach coordinating knowledge from the different infrastructure elements allows a more effective detection of challenges and facilitates the use of autonomic principles employed during the remediation against challenges. In order to demonstrate the feasibility of the framework we analyse two scenarios of unified networked environments.**

***Index Terms*—Resilience, Autonomic Networks, Network Architectures, Anomaly Detection, Security**

I. Introduction

Computer networks constitute the backbone of today's information society by providing connectivity between people as well as ICT (Information Communication Technology) systems. Consequently, they are increasingly mission-critical, especially when used as part of always-on services and applications (e.g., Web-services, Internet Television, Cloud applications, etc.), domain specific safety-critical services (e.g., Air Traffic Control (ATC) networks), critical management services for operators (e.g., Utility networks), and critical real-time financial services (e.g., stock-market systems). The security and resilience of such infrastructures is therefore paramount but at the same time becomes increasingly difficult to achieve.

Hence, the development of resilience mechanisms has to be a prime objective within the design and engineering process of any system or network [1], [3]. However, in the past, availability was the main concern in the design and operation of computer networks [1] and less emphasis was placed on resilience aspects. Moreover, within the actual deployment of networks and ICT systems resilience aspects have also often been treated as add-on and resilience mechanisms have been implemented without reference to a generic resilience framework [1]. Trying to increase system resilience later by deploying such a generic resilience framework leads to monolithic solutions that mainly consider a part or a particular communication layer only. Thus, they usually focuses on a particular resilience sub-domain (e.g., fault-tolerance [5], security [2], [3], or survivability [4]) and do not look at the overall system resilience. More advanced resilient schemes that propose cross-layering methods (e.g. [6], [7]) tend to neglect higher-layer features that express the explicit requirements and characteristics of service users. Hence, their formulation results in one-dimensional performance-oriented solutions that strictly focus on traditional network performance metrics (e.g. throughput, delay, jitter) but avoid mapping these metrics onto the overall end-user QoE and QoS. Therefore, such approaches lead to what we consider "single-level" approaches.

In order to provide a better overall resilience it is therefore necessary to integrate and co-ordinate the provisioning of resilience across different layers and system components. This should also help to provide more adequate end-user QoE whilst satisfying the QoS requirements of the different system elements. In order to achieve this we propose a generic resilience framework that considers the overall impact of simultaneous challenges to different infrastructure elements and cases where the same challenge manifests themselves differently in interdependent systems. With such a framework it is possible to develop a resilience architecture that uses autonomic properties to ensure the adaptability of different networks and ICT systems and helps to improve overall resilience.

In this paper we first present a *multi-level* resilience framework that allows the construction of case-specific resilience architectures that consider the various infrastructure levels and further allowing the construction of user related metadata to better control and identify challenges. User-specific requirements are considered in order to ensure that QoE objectives related to a given resilience strategy are met. Due to the *multi-level* persona of this framework it overcomes

the drawbacks of the traditional monolithic, single system approaches as currently employed in some ICT infrastructure. This is achieved by the joint analysis of challenge indicators and co-ordinated detection actions that also help to coordinate the remediation process at the different system elements.

The remainder of this paper is structured as follows: Section II describes the requirements that such a framework should comply with. Section III is dedicated at presenting the concepts behind our resilience framework and Section IV illustrates the practical aspect of our framework within two case studies. Finally Section V concludes and summarises this work.

II. Requirements for Resilience in Unified Networked ICT Environments

Based on the argumentation developed earlier, we conclude that a generic resilience framework should confront the following requirements:

1) Due to the ubiquitous and user-centric persona of todays ICT environments, a resilience framework should consider various levels of communication; thus, it should be a *multi-level* resilience framework.
2) A *multi-level* resilience framework should allow the composition of practical resilience architectures that enable the direct description of a user over a service that is provided within a networked environment.
3) Users should be represented as a meta-feature by three core levels: the application/service, network and system. In this way, any analysis over the framework would assess the end-user QoE and QoS.
4) Such a framework should manage, process and intelligently analyse heterogeneous data gathered from multiple sources of information within a common knowledge plane.
5) The knowledge plane of such a framework should be in perfect synchronisation with the control, data and management planes.
6) An efficient synergistic deployment of the knowledge plane with the rest will enable a given resilience architecture the autonomic properties of *self-awareness*, *self-management*, *self-optimization* and *self-defence*.
7) Apart from its collaborative behaviour with the remaining planes, the knowledge plane via its self-awareness and self-defence capabilities should be in charge of initially detecting and characterising a challenge (e.g. attack, link failure). Subsequently it should inform the control plane of which remediation actions should be triggered based on the components that initiate the properties of self-management and self-optimization.

Therefore we propose a generic *multi-level* resilience framework that aims to match the majority of the requirements specified. Its description is provided in the next section.

III. Multi-Level Resilience Framework

The design of a robust multi-level resilience framework had to fully comply with the requirements provided earlier

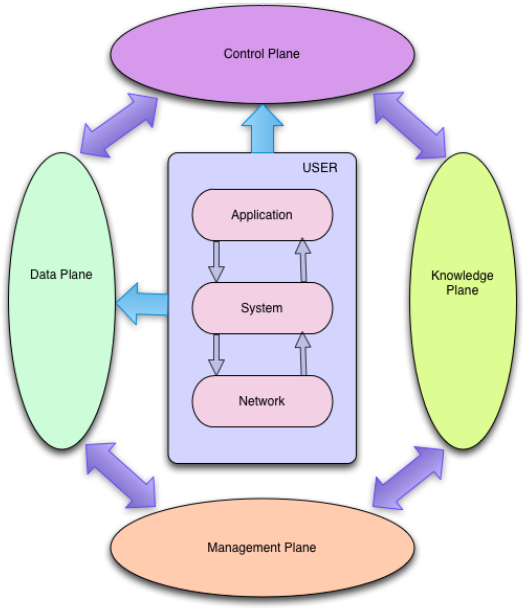

Fig. 1. Conceptual representation of Multi-Level Resilience

(Sectio II). Firstly, the main design consideration was to determine the levels of information that such a framework should process in order to compose meaningful metadata for the user. Secondly, the design process had to specify the exact type of analysis to be anticipated by each plane and further associate the activities of the control, data, management and knowledge planes in an optimal manner.

As evidenced by Fig. 1, the most important functional block within our design is the adequate representation of a user of an infrastructure. Thus, we aim at describing a user based on three levels of observation; the application/service, network and system levels. Our design argues that measurements of features related with any of these three levels is feasible under the assumption that a given resilience architecture that follows our framework will be deployed under standardized monitoring and measurement methods (e.g. SNMP, NetFlow, syslog etc.).

A. Self-awareness & self-defence

All gathered information from all the three different levels will be pre-processed at the management plane and further analysed on the knowledge plane. Fig 2 provides a visual example of how initially a client is meta-represented by the three different levels within the knowledge plane. The exemplar profiling case of Fig 2 illustrates the scenario of a ramp-up behaviour in all the three levels of observation due to the byte consumption caused by a particular application/service.

Nevertheless, given the initial user (or group of users) profiling, the knowledge plane will enforce a dedicated component within its internal structure to perform a statistical characterisation of a user's activities within the observational timeframe. Apart from developing a user-specific profile, this statistical characterisation will also be aggregated for all the

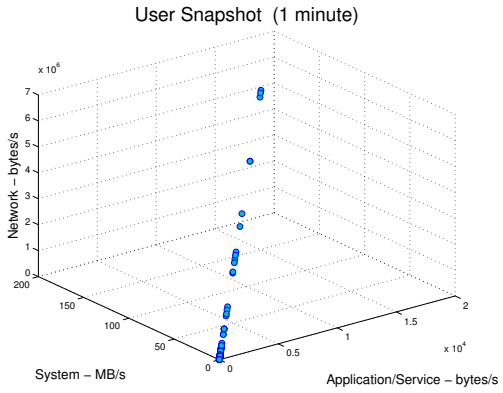

Fig. 2. An example of representing a single user based on the three levels of system, network and application/service.

users and further correlated with the monitoring and measurement components of the control and the data plane. Hence, an overall characterisation of the environment is achieved and self-awareness is ensured.

Naturally, the overall characterisation complies with a particular mathematical model which is in a position to determine the levels of normality, thus detecting abnormal characteristics in real-time. Under the scenario of detecting an anomalous pattern, components within the management plane will be in charge of informing the knowledge plane. Consequently, the knowledge plane will update the overall statistical characterization of its profiling on a set of users and further trigger fine-grained analysis in order to diagnose the exact cause of the anomaly. Given the outcome of this fine-grained analysis, the knowledge plane will inform the management plane in order to trigger remediation techniques and dimension the environment resources accordingly. Thus, the property of self-defence is accommodated.

B. Self-management & self-optimization

Remediation of challenges is resulted after a coordinated act by the management and knowledge planes. In particular, the management plane is the actual co-ordinator at the onset of an event. Thus, the decision regarding the type of anomaly produced by the knowledge plane is sent over to the management plane. Subsequently, the management plane initiates a policies component that holds all the rules regarding the optimisation and management procedures that need to be taken within the ICT environment.

According to the rules provided by the policies component, immediate policies regarding traffic engineering actions (e.g. refined routing decisions, blocking) will be triggered and further received by the control plane. Eventually, the internal mechanisms of the control plane will re-configure the associated settings on the hardware (e.g. routers, sensors) and subsequently update the forwarding schemes within the data plane. Given all the remediation operations described, the properties of self-management and self-optimization for the ICT environment are empowered by our generic multi-level resilience framework.

IV. Multi-Level Resilience Framework : In practise

A. Case Study 1: Multi-level Resilience over the Cloud for Malware Detection

Under the abstractions provided by our multi-level resilience framework (Section III) it was feasible to derive a prototype resilience architecture explicitly for the identification and detection of malware over cloud environments [8]. The overall architecture of our approach can be seen in Figure 3. For simplicity only three nodes are shown and the network connections between nodes are omitted. Each node has a hypervisor, a host Virtual Machine (VM) and a number of guest VMs. Within the host VM of each node there is a dedicated Cloud Resilience Manager (CRM) which comprises one part of the wider detection system. The software components within the CRM are the Network Analysis Engine (NAE), the System Analysis Engine (SAE), the System Resilience Engine (SRE) and the Coordination and Organisation Engine (COE).

The CRM on each node performs local malware detection based on the information obtained from its node's VMs and its local network view; this is handled by the SAE and NAE components respectively. The SRE component is in charge of protection and remediation actions based on the output from the analysis engines (i.e. NAE and SAE). Such actions include destroying an infected VM or blocking information destined to a vulnerable VM. Finally, the COE component coordinates and disseminates information between other instances and, in parallel, controls the components within its own node. In addition to system level resilience, the detection system is capable of gathering and analysing data at the network level through the deployment of Network Monitors each containing a network CRM as shown in Figure 1. This monitoring system is directly connected to each Ingress/Egress Router as shown, and can gather features from all traffic passing through it. Nevertheless, the case study of this section is explicitly addressing the operations of the NAE and SAE engines using the example of characterising and detecting the Kelihos malware [9]. In general, the NAE and SAE engines enable the autonomic properties of *self-awareness*, *self-management* and *self-defence* since they could synergistically provide an overall characterisation on a system, network and application/service level alongside an indication regarding the existence of a malware.

1) **NAE & SAE Description:***:* Both engines are composed by several scripts that automate all the processes referring to the pre-processing of either network or system data. The integration of the aforementioned scripts is in practise empowering the aspect of *self-management* within the overall architecture. Their pre-processing outputs are subsequently fed as the primitive input for the analysis algorithms which are also integrated and they employ Principal Component Analysis (PCA) [1] and statistical timeseries on the jointly gathered

[1]Due to the fact that we did not change the implementation of the traditional PCA algorithm we do not present the mathematical formulation of PCA. For the definition of the PCA algorithm we refer the reader to [10].

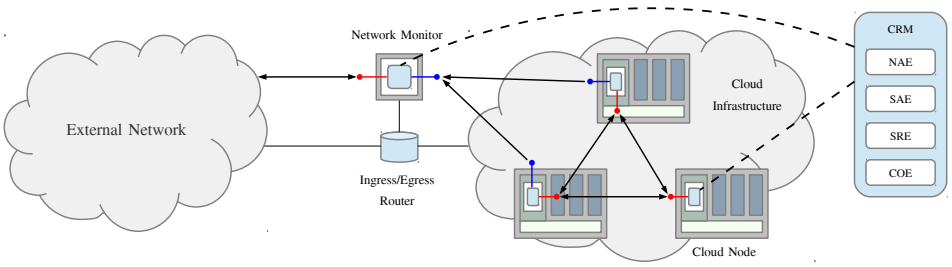

Fig. 3. Resilience Architecture over a Cloud Scenario for Malware Detection

system and network features. Hence the produced outputs are also considered as important aiding elements towards the establishment of the *self-awareness* and *self-defence* properties where the former enables the adequate characterisation of the three levels of *network*, *system* and *application/service* and the latter achieves malware detection. Undoubtedly, the outcomes for both SAE and NAE, in conjunction with the communication allowed by the COE establish a robust architecture that complies with the generic requirements stated in Section II.

2) **Experimentation Setup***:* The main objective behind this experimental setup was to illustrate a robust characterization and malware detection strategy as part of the overall resilience architecture presented earlier. In particular, we aimed at assessing the fundamental property of VM/service "live" migration as initiated in todays' cloud environments and further investigate on how malware can be detected in such a scenario within a controlled experimental testbed. The testbed consists of three physical nodes where the two run a number of VMs that behave as HTTP servers. Each physical node runs the Kernel-based Virtual Machine (KVM[2]) virtualization infrastructure and the Quick EMUlator (QEMU[3]) for hardware emulation whereas the migration functionality is enabled with the use of libvirt[4]. The third physical node is used for emulating background traffic that is mainly composed by a variety of random client requests to the VM's HTTP servers which were achieved with custom scripts.

The experiment lasted for 20 minutes and the Kelihos malware strain, *Trojan.Kelihos-5*, was injected on the 9^{th} minute in one of the HTTP servers whereas on the 10^{th} minute the infected VM was migrated to the second "clean" physical host as manually commanded by the management host. Throughout the whole experiment there was the consistent aggreagted monitoring of system-related features (e.g. counts of processes) and network packets for all VMs on both physical hosts from the hypervisor level using custom monitoring scripts embedded in the NAE and SAE. Finally, the aggregated system and network features where subsequently fed to our implemented PCA algorithm in order to firstly chartacterize the joint dataset and further pinpoint possible anomalous characteristics.

[2]Kernel-based Virtual Machine: http://www.linux-kvm.org/
[3]QUEMU:http://www.quemu.org/
[4]The virtualization API: http://libvirt.org/libvirt2

3) **Results***:* As evidenced by Fig. 4, each joint dataset is divided into 3-second bins, and each bin is converted into a feature vector per each VM node. The combined feature vector was submitted to PCA to obtain the *k-subspace* which corresponds to the normal behaviour of the traffic, and spans from a principal component pc_1, through pc_k, whereas the remaining subspace with the less significant principal components (i.e, pc_{k+1} through pc_m) maps the anomalous behaviour with respect to the variance of the dataset. Subsequently, we compute a distance metric that describes the magnitude of the projection of the original data points into the anomalous subspace to quantify their malicious behaviour which we use to produce the anomaly score graph (ASG) in Fig. 4. In practise, this plot that is generated by the NAE is a time-series representation which summarizes the anomalous score of each bin in the trace and thus indicates the level of how anomalous is each tested timebin with respect to the other measurement bins. Overall, the PCA performed extremely well and was able to show a sharp increase on the ASG plot as demonstrated by Fig. 4 as soon as the Kelihos malware was injected ($\approx 160^{th}$ bin in Fig. 4). Moreover, the ASG plot also shows that the PCA algorithm could also identify anomalous activity after the VM migration performed right after the 200^{th} time bin.

The outcomes of the PCA analysis that was achieved via the proposed resilience arhitecture has demonstrated the benefits of complying with some basic resilience requirements derived by the generic resilience framework presented earlier. Via the proposed architecture and the selected case study we have shown that by considering sources of information from different communication layers (i.e. system, network and application), the properties of *self-awareness*, *sef-management* and *self-defence* were met in the explicit case of cloud environments.

B. Case Study 2: Multi-level Resilience in Access Networks for Characterization & Detection of Systematic Downloads

The increasing Internet use has unavoidably brought issues raised with respect to the excessive usage of several application-layer services. In particular, systematic downloads performed at services offered by academic websites (e.g. IEEEXplore) have become commonplace and consequently lead academic institutional campus networks being blacklisted [11]. Thus, given the properties of our generic framework presented earlier (Section III) we have derived formulations in order

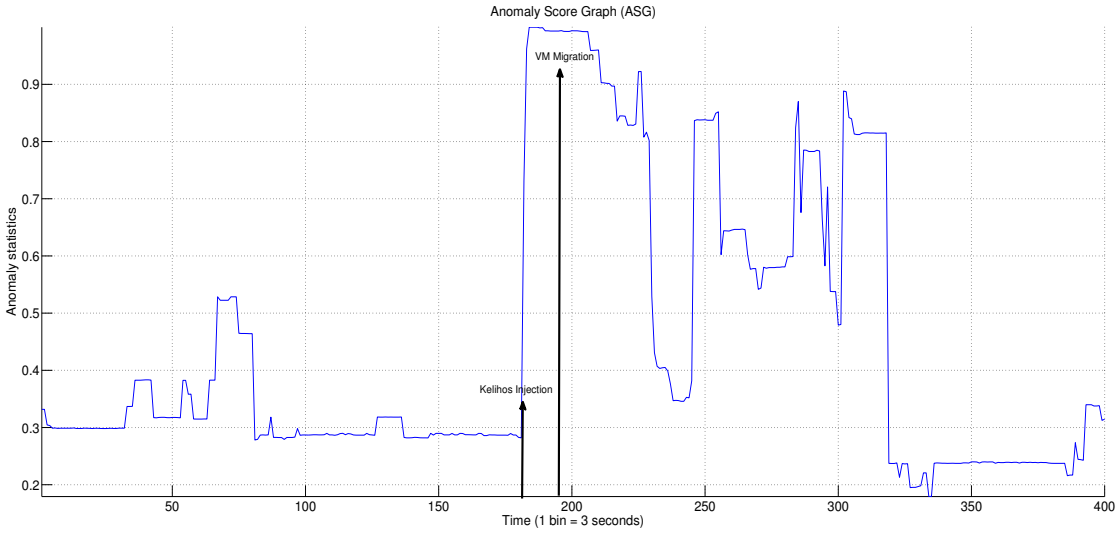

Fig. 4. Output of the resulting PCA-based anomaly detection as jointly performed on network and system data that were monitored by the NAE and SAE.

to adequately detect systematic downloads using time series models. As already being well known techniques timeseries analysis have been used to model Internet traffic and DOS attacks [12], [13], [14].

In this work, the number of requests per second made to a specific publisher as obtained from the proxy logs captured a a proxy server on the Indian Institute of Technology Madras campus is used to model the time series. In order to build robust models, the data is obtained for 106 such publishers where each one is represented by a different timeseries since license agreements with each is likely to be different. Hence our modelling approach complies with the principles of our generic framework (section III) since it considers application-specific requirements gathered from the application layer and aim to provide network-based knowledge with respect to user utilisation on a particular service. We following describe the specifics of our modelling scheme by first starting with the model identification.

1) **Model Identification***:* The type and order of our model is obtained by first considering the autocorrelation function (ACF) of the time series. Similar to the work in [14], it was observed that the ACF of the differenced timeseries was stationary and could be modeled as an Auto Regressive (AR) process

$$\Delta(x_n) \quad = y_n = \quad x_n - x_{n-1} \tag{1}$$

$$y_n \quad = \quad \sum_{k=1}^{p} a_k y_{n-k} + e_n \tag{2}$$

where e_n is the prediction error assumed to be a generated by a white noise process. In parallel, the order of the model is estimated again from the ACF of the differenced series as shown in Figure 6. The differencing operation aims to address the non-anomalous structural breaks disclosed within network data where average statistics can vary depending upon the downloading time (prime versus nonprime time).

2) **Emulation and detection of Systematic downloads***:* By virtue of privacy constraints, access to data corresponding to the investigated systematic downloads was not feasible.

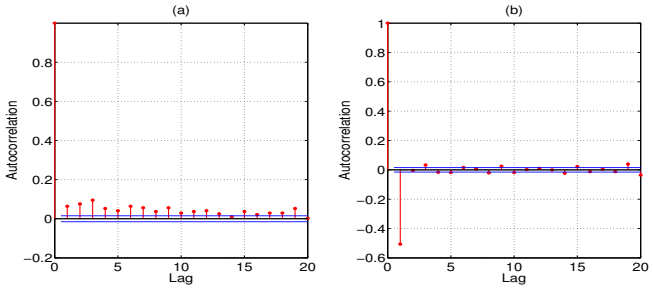

Fig. 5. Autocorrelation function of differenced data before (left) and after differencing (right) in order to estimate the order of the model regarding a publisher(s) timeseries.

Therefore we have used the setup given in Fig. 7 in order to generate anomalous data. The traces of normal data as obtained from the proxy logs were first generated and subsequently interspersed with systematic download. The emulation setup consisted of varying each of the following parameters:

1) Number of files downloaded
2) Files downloaded at random time intervals
3) Different sizes of files downloaded
4) Random ordering of files of different sizes

After obtaining the emulation logs we formed the timeseries by computing the number of requests at different polling intervals varying from 5-30 seconds in 5 second bins. The time-dependent AR process was modelled with a framelength of 3 minutes and we further computed the AR process roots. We have identified a specific root which characterised the anomaly and was obtained by using ground truth data (i.e. the location of the region when systematic downloads are in progress), and feature selection (albeit root selection) was performed using the algorithm suggested in [15]. The AR

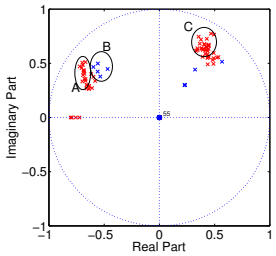

Fig. 6. Roots of the time-varying AR model during periods of both systematic downloads (clusters A,B) and normal downloads (cluster C)

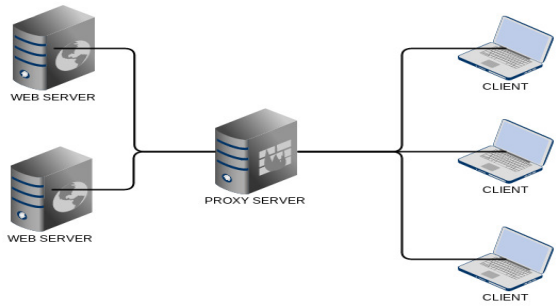

Fig. 7. Emulation Architecture for Achieving a Systematic Downloads Scenario.

process roots are shown in Figure 5 where different colours are associated with the roots corresponding to that of systematic download and normal traffic. In general, it is evidenced that the roots associated to that of systematic downloads (marked with A and C) are closer to the unit circle and belong to a different cluster compared to that of a normal download (marked with B).

Overall, this exemplar scenario has considered service-level requirements in order to become self-aware regarding the utilisation of a given networked environment. Moreover, the proposed mechanisms are currently employed as intelligent functional elements within the network management software of this particular campus network and empower the aspect of *self-management* and further aid the properties of *self-optimization* and *self-defence* of the overall network management process.

V. Conclusions

Today's networked ICT environments are increasingly challenged by misuse and security issues. These manifest themselves at different levels of the system architecture and are so far dealt with independently at the different levels. This paper introduces a multi-level resilience framework in which the resilience activities (such as anomaly detection) are co-ordinated in order to provide better and early defence and awareness regarding threats and challenges. This paper illustrates that architectures that comply with the requirements derived from the generic multi-level resilience framework can adequately relate several types of information with respect to application, system and network-specific characteristics. Further, we show how mechanisms that confront particular challenges at the different levels at which they are likely to manifest themselves can be co-ordinated and hence produce a better result. This is demonstrated through case studies focusing on the central aspects of analysis and aggregation of heterogeneous types of information. These case studies look at the three proposed levels and how these relate to the generic framework. We argue that our suggested framework sets new horizons towards the development of autonomic and intelligent mechanisms within the area of network management and will ensure a robust scheme for adaptive resilient and secure solutions.

Acknowledgments

The authors would like to thank the UK EPSRC and DST India funded Indian-UK Advanced Technology Centre of Excellence (IU-ATC) research project that has kindly supported this work.

References

[1] Smith, P., Hutchison, D., Sterbenz, J., P., G., Scholler, M., Fessi, A., Karaliopoulos, M., Lac, C., Plattner, B., *Network Resilience: A Systematic Approach*, in IEEE Communications Magazine, Vol. 49, I: 7 , pp 88-97, July 2011

[2] Marnerides, A., K., Pezaros, D. , P., Hutchison, D., *Detection and Mitigation of Abnormal Traffic Behaviour in Autonomic Networked Environments*, in Proceedings of ACM SIGCOMM CoNeXT Conference, Student Workshop, 2008

[3] Marnerides, A., K., Pezaros, D. , P., Hutchison, D., *Autonomic Diagnosis of Anomalous Network Traffic*, in Proceedings of IEEE WoWMoM Conference, AOC Workshop, 2010

[4] Sterbenz, J., P., G., et. al., *Resilience and Survivability in Communication Networks: Strategies, Principles and Survey of Disciplines*, in Elsevier Computer Networks (COMNET), Special Issue on Resilient and Survivable Networks, vol. 54, no. 8, June 2010

[5] Cholda, P., et. al., *A Survey of Resilience Differentiation Frameworks in Communication Networks*, in IEEE Communications Surveys & Tutorials, vol. 9, no. 4, 2007

[6] Conti, M., Maselli, G., Giordano, S., *Cross-layering in mobile ad-hoc network design*, in IEEE Computer, 37(2):48-51,Feb 2004

[7] Winter, R., Schiller, J., *Cross-layer decision support based on global knowledge*, in IEEE Communications Magazine, 44:2-8, Jan 2006

[8] Watson, M., Shirazi, N., Marnerides, A., K., Mauthe, A., Hutchison, D., Towards a Distributed, Self-Organizing Approach to Malware Detection in Cloud Computing , in 7th IFIP International Workshop on Self-Organizing Systems, IFIP/IFISC IWSOS 2013, May 2013

[9] Garnaeva, M. "Kelihos/Hlux Botnet Returns with New Techniques." Securelist, http://www.securelist.com/en/blog/655/Kelihos_Hlux_botnet_returns_with_new_techniques.

[10] Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E. D., Taft, N., Structural analysis of network traffic flows. in ACM SIGMETRICS Perform. Eval. Rev. 32, 1, June 2004

[11] Andrew Waller, "When the Licensor Calls: Some Thoughts on Systematic Downloading," *Informed Librarian Online*, `http://hdl.handle.net/10760/10206`, August 2007.

[12] A Ramasamy, Hema A Murthy and T A Gonsalves, "Linear Prediction for Network Management," *Proc. NCC-2000*, pp.199-202, Delhi, 2000.

[13] Dinil Mon Divakaran, Hema A Murthy and Timothy A Gonsalves, "Detection of SYN Flooding Attacks using Linear Prediction Analysis," 14th IEEE International Confernce on Networks, *ICON-2006*, pp.218-223, Singapore, Sep 2006.

[14] Cyriac James and Hema A Murthy, "Time series models and its relevance to modeling TCP SYN based DOS attacks," *Proc. IEEE Euro NGI*, pp.1-8, 2011.

[15] S R Singh, Hema A Murthy and T A Gonsalves, "Feature Selection for Text Classification Based on Gini Coefficient of Inequality,"*Proc. of the Fourth International Workshop on Feature Selection in Data Mining*, pp.10:76–85, 2010.