Towards a Unified Architecture for Resilience, Survivability and Autonomic Fault-Management for Self-managing Networks

Nikolay Tcholtchev¹, Monika Grajzer², and Bruno Vidalenc³

¹ Fraunhofer FOKUS Institute for Open Communication Systems, Berlin, Germany

² Telcordia Poland, Poznan, Poland

³ Alcatel-Lucent France, Paris, France

Nikolay.Tcholtchev@fokus.fraunhofer.de, mgrajzer@telcordia.com,
Bruno.Vidalenc@alcatel-lucent.com

Abstract. The emergence of self-managing networks can be seen as an enabler for increased dependability, reliability and robustness of the network layer. All these features are significant for the services and applications relying on the network infrastructure. This paper explores the links between traditional Fault-Management functions belonging to the management plane and the fundamental network functions for Resilience and Survivability embedded inside the protocol modules of a node/device. This results in an architectural framework that allows nodes/devices to implement the converging aspects of Fault-Management (now becoming autonomic), Resilience and Survivability in a self-managing network. The components and adaptation mechanisms of the proposed framework will make the network layer more robust and application/service aware. Thus, the dependability, reliability, and adaptability of the upper layer services and applications are expected to increase.

Keywords: Resilience, Survivability, Autonomic Fault-Management, Risk Assessment, GANA, self-managing networks.

1 Introduction

The ability of a network to introduce self-healing mechanisms and actions in order to preserve both - network and service layer functionality in the presence of faulty conditions, failures and malfunctioning is of paramount importance. Introducing autonomicity into the self-healing processes enables a network to implement novel functionalities specifying network behaviour under abnormal conditions while reducing the involvement of the network operation personnel. This becomes crucial especially in extreme networking environments, e.g. after a large scale disaster damaging the network infrastructure.

Resilience is arguably considered a necessary feature of communication system functional entities which enables the self-healing of a network facing challenging conditions. A diversity of approaches in this field is implemented in different network layers, protocols and technologies. Resilient mechanisms can be found in protection and restoration schemes as in (G)MPLS and SONET/SDH, or can be intrinsically

implemented in the protocol modules, e.g. ICMP and OSPF - with regard to the automated reaction to link failures. Early initiatives, aimed at bringing the variety of methods under a common umbrella, have followed the path of creating a framework of protocols for self-healing resilient networks [2]. Works such as [3] [4] review the Resilience of routing within an Autonomous System (AS) and across multiple ASs, and capture the overall picture of the applied mechanisms, problems, and general proactive and reactive approaches. More generic views on Resilience in telecommunications and especially IP networks are provided through the concept of Multi-layer Resilience [5] [6].

The Fault-Management functions of the management plane are strongly related to the Resilience features of the network components. The evolvement of Fault-Management towards Autonomic Fault-Management [7] requires that the processes of Fault-Detection, Fault-Isolation and Fault-Removal must be automated and combined with each other. Until recently, the research in this area has mostly concentrated on techniques for Fault-Detection and Fault-Isolation which, to the authors' best knowledge, have not been merged with Fault-Removal. Consequently, in current practices the Fault-Management processes are mainly performed manually. To address this issue, architectures targeting the idea of Autonomic Fault-Management, such as UniFAFF [7] and Omega [8], are on the rise.

Our goal is to combine some of the research results related to Autonomic Fault-Management with the diversity of the network resilience mechanisms towards a unified architectural framework. The challenge in designing an integrated architecture for Resilience, Survivability and Autonomic Fault-Management is to accommodate the plethora of existing intrinsic multi-layer resilient mechanisms into a framework that targets the goal of overseeing the network and reacting to incidents and alarms, without interfering with the inherent self-healing mechanisms of functional entities (i.e. protocols). This is the point where the converging aspects of Resilience, Survivability and Fault-Management need to be merged. In this paper we present our work in progress on a unified architectural framework that puts across design principles and defines components from the Resilience and Autonomic Fault-Management perspective. The emerging architecture is expected to increase the dependability, reliability and robustness of the network layer and to provide better conditions to the services and applications running on top of it. We believe that such an architecture, incorporating the properties of Autonomic Fault-Management, Resilience and Survivability, is a highly beneficial and promising solution that we introduce to the domain of Autonomic Networking.

The rest of this paper is organized as follows: Section 2 presents the GANA reference model for Autonomic Networking, which we employ for the establishment of our architecture. Section 3 conducts a requirements analysis and presents the fundamental components of our framework. Section 4 presents the overall set of components belonging to the framework and elucidates the way they inter-work. Finally, section 5 concludes the paper, and gives an outlook on the future research.

2 The GANA Architecture in Brief

In this section, we present the *GANA (Generic Autonomic Network Architecture)* reference model [1] for autonomic network engineering, which is exploited to define

the architecture for Resilience, Survivability, and Autonomic Fault-Management in self-managing networks. In order to manage node and network functionalities, the GANA specifies a *control loop* by introducing two basic concepts – a *Managed Entity (ME)* and a *Decision Element (DE)*. A DE is conceptually an autonomic element responsible for managing a set of MEs (e.g. protocol modules), and realizes a control loop based on its continuous learning cycle. Through a specified control loop a DE is able to exercise specific algorithmic schemes and/or policies on its ME(s), while the ME(s) provide feedback to the DE by sharing information related to their current state. Information exchange can be realized over a logical link using the concept of repositories and notification/callback mechanisms.

Taking into account that control loops can be implemented at different levels of abstraction – e.g. on functions or node level, GANA provides the *Hierarchical Control Loops (HCLs)* framework consisting of four levels (Figure 1) at which DEs and associated control loops can be introduced, i.e. *Protocol Level* (e.g. a control loop inside a single protocol), *Functions Level* (in-built management of basic networking functions such as Forwarding), *Node Level* and *Network Level*. Thereby DEs, i.e. control loops, on a higher level manage DEs on a lower level (an example is given in Figure 1). For more information we refer the reader to [1].

Fig. 1. An overview of a GANA control loop and an example of its placement in an autonomic node/network

3 Components of the Targeted Architectural Framework

The diverse factors that should be taken into account while designing the unified architecture for Resilience, Survivability and Autonomic Fault-Management are listed below. We believe that this set reflects the plethora of issues, brought up in literature, related to Autonomic Fault-Management, Resilience and Survivability:

Requirement #1: The proposed architecture must take into account the intrinsic resilient mechanisms implemented inside the network components.

Requirement #2: The architecture needs to be application/service aware and thus provide facilities for implementing the concept of *Survivability Requirements (SR)*. SR will provide the possibility for applications to express their needs for obtaining vital information regarding incidents/alarms in the network and should enable the application layer to react accordingly and survive in the presence of faulty conditions.

Requirement #3: The proposed framework must consider the dynamic dependencies among protocols and services in the network.

Requirement #4: The framework must consider the causality relations (fault-propagation models) regarding root causes (faults) and their propagation as one or many errors until eventually resulting in observable symptoms (failures and alarms).

Requirement #5: In order to realize *Proactive Resilience*, the architectural framework should provide facilities for estimating the risk that various components and functional entities are likely to fail in the near future. Such a *Risk Assessment* can be employed to avoid the use of components which are likely to fail and to proactively sustain (e.g. by proactive rerouting) the traffic in the case of challenging conditions.

Requirement #6: The architecture needs to ensure that the actions undertaken by different autonomic entities do not contradict and the decision making process effectively contributes to the overall fitness of the network. Hence, a component is required that synchronizes the actions issued by diverse autonomic entities in an appropriate way.

Requirement #7: The effectiveness of the undertaken management actions must be verified.

Requirement #8: The autonomic entities of the architectural framework must be given exclusive access to all functional entities inside a node, in order to manage them with respect to incidents and alarms.

Based on these requirements and the control loop structures of GANA, a number of components and inter-working mechanisms are defined, which eventually result in a unified architecture for Resilience, Survivability, and Autonomic Fault-Management. First, we present the two major components that make the core of the proposed architecture – the *Resilience and Survivability DE (RS_DE)* and the *Fault-Management DE (FM_DE)*

Fig. 2. The sharing of responsibilities among the RS_DE and the FM_DE

DE (FM_DE). The basic separation of their operational domains is illustrated in Figure 2. The RS_DE is responsible for an immediate reaction to failures and faults aimed at sustaining an acceptable QoS level by employing diverse fault-tolerant mechanisms, i.e. it performs Fault-Masking. In parallel, the FM_DE is responsible for automatically repairing the damaged network infrastructure by diagnosing the underlying root cause (fault) and removing it, i.e. the FM_DE performs Fault-Isolation and Fault-Removal. Additionally, the RS_DE implements proactive mechanisms based on the estimated fitness of the network components.

3.1 Fault-Management Decision Element

The autonomic entity realizing Fault-Isolation and Fault-Removal during the (long-term) operation of the network is the Fault-Management DE (FM_DE). In a GANA autonomic node/device, the FM_DE is required to have exclusive access to all other entities (MEs and DEs) inside the node. This requirement can be realized only if the FM_DE belongs to the highest GANA control loop layer within a device, which is the node level. Since the node level is conceptually considered to contain only the Node_Main_DE, the FM_DE is naturally specified as a sub-DE of the Node_Main_DE (Figure 1).

Figure 3 depicts the constellation of entities that are involved in realizing Autonomic Fault-Management inside the network devices. Autonomic Fault-Management is triggered whenever symptoms/anomalies indicating the presence of faults in the network are detected. The corresponding process of Incident-Detection is realized by specially instrumented Monitoring Components/Functions inside the autonomic nodes/devices (Figure 2) or by some other MEs or DEs (as illustrated in Figure 3) which have the capabilities to share alarm/incident information with the components of the proposed architecture. Upon detection of particular symptoms (alarms/failures) the Monitoring Functions (or the aforementioned MEs/DEs) report their observations to a set of common fault/error/failure/alarm repositories which in turn notify the FM_DE about the newly arrived information. Consequently, the FM_DE performs Fault-Isolation by employing the features of its embedded *Fault-Diagnosis /Localization/Isolation block (FDLI functions)*. The FDLI functions aim at finding the root cause for the observed symptoms based on fault propagation models represented by a *Causality Model* and stored in the *Causality Model Repository (CMR)* inside an autonomic node. Once the Fault-Isolation is successfully completed, the FM_DE removes the isolated root cause (fault) by employing the *Fault-Removal Functions (FRF)* block. The FRF functions perform actions such as “reloading a functional entity”, “rebooting a node/device” or “reconfiguring a functional entity”. After executing the intended Fault-Removal actions, the FM_DE ensures that the taken decision was indeed the right one and the source of the problem has been successfully eliminated. This is realized by the *Fault Removal Assessment Functions (FRAF)* part of the FM_DE. The assessment can be realized either by employing some probing and network debugging methods, or by using the risk assessment services provided by the RS_DE, as pointed out in the next sections. However, before executing a Fault-Removal function, the FM_DE ensures that its tentative action is indeed contributing to the overall fitness of the network and is not contradicting with other actions intended by the RS_DE. Therefore the FM_DE and the RS_DE refer to the *Action Synchronization Functions (ASF)*, embedded in the FM_DE, before

performing an action on the MEs inside a node. The ASF block is responsible for ensuring the stability of the FM_DE and RS_DE control loops running in parallel. Moreover, in the course of Fault-Removal, the FM_DE must be aware which entities are affected by the undertaken actions, in order to remove the fault in a stable way without introducing additional problems to the network. Thus, knowledge about the dependencies among nodes/devices, protocol modules, and services is required, which is provided by the *Dependability Models Repository* of an autonomic node.

Fig. 3. The entities involved in Autonomic Fault-Management

3.2 Resilience and Survivability Decision Element

To support generic Resilience and Survivability functions within an autonomic node, the Resilience and Survivability DE is proposed as a core part of the presented architecture. The relations between RS_DE and other architectural components involved in resilience and survivability functionality are depicted in Figure 4. Following the same line of arguments as in the case of the FM_DE, we introduce the RS_DE at the node level inside the GANA reference model (Figure 1).

The operation of the RS_DE is based on information from the FM_DE and from other diverse functional entities that supply information to the fault/error/failure/alarm repositories of an autonomic node. The Monitoring Components or some alarm/incident information sharing DEs and MEs, as depicted in section 3.1, supported by the node repositories, provide data related to the directly detected faults, errors and failures or generated alarms. The direct connection with the FM_DE ensures that the RS_DE has the up-to-date knowledge about isolated faults and the Fault-Removal process.

Having information from the FM_DE and dedicated repositories, the RS_DE triggers diverse reactive fault-masking mechanisms depending on the situation in the network. This process is controlled by the *Fault Masking Functions* (FMF) module

Fig. 4. The entities involved in Resilience and Survivability functionality

realizing *Reactive Resilience* functionalities. The role of RS_DE is to trigger specific fault masking functionalities that would be best applicable as a reaction to the current incident(s). However specific fault masking algorithms/procedures/behaviours are not implemented within RS_DE, but their functionality is specific to the underlying lower level MEs and DEs (e.g. protocols, mechanisms, etc.). After being informed about an incident or a fault, the RS_DE synchronizes, i.e. negotiates, its tentative action with the FM_DE. Based on the negotiation result, the RS_DE discards or performs the Fault-Masking action on the corresponding MEs allowing them to execute their embedded fault masking behavior. Thus, the RS_DE acts in a fault-tolerant manner and orchestrates resilient mechanisms of the selected node entities (MEs of this DE) to ensure acceptable QoS and sustain network functionality. In order to select appropriate solutions and to manage the resilient behaviour of a node, the RS_DE must be aware of the resilient mechanisms implemented intrinsically inside the MEs which are under its control (protocol modules and other lower-level DEs). This knowledge is stored and retrieved from a dedicated repository – *Multi-Layer Resilience Properties Repository*. Moreover, before imposing its own decisions, the RS_DE is required to give the low level MEs a chance to recover based on their own intrinsic resilient mechanisms. However, managerial responsibilities belong to the RS_DE and thus it has the right to intercept the intrinsic actions undertaken by the functional entities if they are contradicting.

While the FMF module of the RS_DE takes reactive actions upon detected incidents, the *Risk Assessment Module* (RAM) tries to predict the occurrence of certain incidents in the near future. As a result, the RS_DE is able to proactively steer the node entities such that the node and/or network can prepare for the problems that are likely to manifest in the future. As a consequence, it is possible to avoid future fault activations. The RAM module enables the FM_DE and the RS_DE to shift from basic, reactive only schemes, to new proactive ones where prediction allows the framework to anticipate failures as well as dynamically improve the Causality Model at the same time.

4 The Overall Architecture for Autonomic Fault-Management, Resilience and Survivability for Self-managing Networks

All the modules of the architecture, as well as their relationships are presented in Figure 5. As it can be observed, the FM_DE and the RS_DE, along with their embedded components, play a central role inside a device. The MEs and DEs inside a node are managed by means of instantiated GANA control loops and exchange knowledge with the core DEs of the architecture over a set of faults/errors/failures/alarms repositories.

Autonomic Node:

Fig. 5. The unified node architecture for Resilience, Survivability and Autonomic Fault-Management for self-managing networks

There are several points where the FM_DE and the RS_DE interact with each other. The fault/error/failure/alarm repositories are one of the links between both DEs. Secondly, the RAM module inside the RS_DE exposes its knowledge to the CMR repository, such that the adjustment of parameters inside the Causality Model is based on changes in the likelihood for some incidents or node/device failures. Moreover, the information from the RAM can be used to assess whether a Fault-Removal action undertaken by the FM_DE was successful or not. That is, by monitoring the probability (risk) for devices to fail, the *Fault Removal Assessment Functions (FRAF)* can verify the effectiveness of the Fault-Removal process. Another interaction between the RS_DE and the FM_DE takes place during the synchronization of management actions, which is realized by the ASF module of the FM_DE. Besides the above mentioned mechanisms and components, we argue that the sharing and dissemination of incidents/alarms between the functional entities of a node/device and

the network as a whole is vital for implementing Autonomic Fault-Management, Resilience and Survivability. Therefore, similar to [7] we propose a component called the *Incident Information Dissemination Engine*. This is also the component which is suitable for accommodating the notion of *Survivability Requirements (SR)* as presented in section 3. Hence, besides the general benefits for the applications due to the increased robustness of the network layer, the *Incident Information Dissemination Engine* would enable applications and services to increase their adaptability according to the current faults/errors/failures/alarms events across the network.

In addition to the node intrinsic mechanisms presented above, we now elaborate on Autonomic Fault-Management, Resilience and Survivability in the network as a whole. First, a distributed collaboration of FM_DEs and RS_DEs is expected. The core architectural DEs located in different nodes would strive to have a consistent view of the knowledge related to the incidents in the network by employing the services of the *Incident Information Dissemination Engine*. Thereafter, each FM_DE in a GANA node would proceed with the Fault-Isolation, and eventually remove the isolated fault (root cause) if it is directly or closely related to the local node for which the FM_DE is responsible. Alternatively, if none of the node-level FM_DEs are able to resolve the problem, a centralized network-level FM_DE (Figure 1) would be requested to react in the best possible way or even escalate the problem to the network operation personnel. The same procedure is applicable to the RS_DE, which in parallel to the FM_DE, would try to mask the activated faults and would ensure an acceptable QoS in the network while the faults are being removed.

5 Conclusions and Future Work

As self-managing networks are emerging, the aspects of Resilience, Survivability and Fault-Management, now becoming autonomic, are converging and require a unified architectural framework enabling them to inter-work harmoniously in an autonomic fashion. Our work aims at developing such an architecture by identifying the design principles and the major components that nodes/devices inside an autonomic network must be equipped with. In order to introduce the reader to our vision, we presented briefly the GANA, which is the reference model for autonomic network engineering that we instantiate. Furthermore, we conducted a requirements analysis of the factors that need to be addressed by the unified architecture for Resilience, Survivability and Autonomic Fault-Management for self-managing networks. Taking into account the diverse requirements, we defined the major decision making components, as well as additional entities that support the processes inside our architectural framework. The proposed framework brings a number of advantages to the services and applications running on top of the network layer. Besides the expected improvement in the robustness, dependability, and reliability of the network, our architecture provides components for application/service awareness inside the network layer. Such components enable the cross-layer information sharing and provide the means for services and applications to express survivability requirements with respect to incidents and alarms. This opportunity would enable the adaptability of the application layer to the challenging conditions the network is exposed to.

At the current stage we are developing some components of the presented architecture with the aim of conducting evaluations in diverse network environments – fixed and MANETS (Mobile ad hoc Networks). Case studies related to the impact of the proposed mechanisms on the application layer are also envisioned. In the near future, we will be working towards providing a proof of concept that our proposed architecture indeed improves the performance and dependability of the network and thus helps maintaining the QoS provided to the end user.

Acknowledgment. This work has been partially supported by EC FP7 EFIPSANS project (INFSO-ICT-215549).

References

1. Chaparadza, R.: Fraunhofer FOKUS Institute for Open Communication Systems Berlin, Germany: Requirements for a Generic Autonomic Network Architecture (GANA), suitable for Standardizable Autonomic Behavior Specifications for Diverse Networking Environments, Published by the International Engineering Consortium (IEC). Annual Review of Communications 61 (December 2008)
2. Green, C.J.: Protocols for a self-healing network. In: IEEE Conference Record, Military Communications Conference, MILCOM 1995, November 6, vol. 1, pp. 252–256 (1995)
3. Larrabeiti, D., et al.: Multi-Domain Issues of Resilience. In: The proceedings of the 7th International Conference on Transparent Optical Networks, Barcelona, Catalonia, Spain, July 3-7 (2005)
4. Rai, S., Mukherjee, B., Deshpande, O.: IP resilience within an autonomous system: current approaches, challenges, and future directions. IEEE Communications Magazine 43(10), 142–149 (2005)
5. Suwala, G., Swallow, G.: SONET/SDH-like resilience for IP networks: a survey of traffic protection mechanisms. IEEE Network 18(2), 20–25 (2004)
6. Touvet, F., Harle, D.: Network Resilience in Multilayer Networks: A Critical Review and Open Issues. In: Proceedings of the First International Conference on Networking-Part 1, July 09-13, pp. 829–838 (2001)
7. Chaparadza, R.: UniFAFF: A Unified Framework for Implementing Autonomic Fault-Management and Failure-Detection for Self-Managing Networks. International Journal of Network Management (2008)
8. Baliosian, J., Sailhan, F., Devitt, A., Bosneag, A.-M.: The Omega Architecture: Towards Adaptable, Self-Managed Networks. In: Proceedings of the 1st Annual Workshop on Distributed Autonomous Network Management Systems, Dublin, Ireland (June 2006)