Autonomic Communications in Software-Driven Networks

Zhongliang Zhao, *Member, IEEE*, Eryk Schiller, *Member, IEEE*, Eirini Kalogeiton, *Student Member, IEEE*, Torsten Braun, *Member, IEEE*, Burkhard Stiller, *Member, IEEE*, Mevlut Turker Garip, *Member, IEEE*, Joshua Joy, *Student Member, IEEE*, Mario Gerla, *Fellow, IEEE*, Nabeel Akhtar, *Student Member, IEEE*, and Ibrahim Matta, *Senior Member, IEEE*

***Abstract*— Autonomic communications aim to provide the quality-of-service in networks using self-management mechanisms. It inherits many characteristics from autonomic computing, in particular, when communication systems are running as specialized applications in software-defined networking (SDN) and network function virtualization (NFV)-enabled cloud environments. This paper surveys autonomic computing and communications in the context of software-driven networks, i.e., networks based on SDN/NFV concepts. Autonomic communications create new challenges in terms of security, operations, and business support. We discuss several goals, research challenges, and development issues on self-management mechanisms and architectures in software-driven networks. This paper covers multiple perspectives of autonomic communications in software-driven networks, such as automatic testing, integration, and deployment of network functions. We also focus on self-management and optimization, which make use of machine learning techniques.**

***Index Terms*— Autonomic communications, autonomic computing, software-defined networking (SDN), network function virtualization (NFV), self-management, self-optimization, testing, autonomic security, operation and business support system (OSS/BSS).**

I. Introduction

AUTONOMIC Communications can solve the management problem arising in dynamic and large scale networks, in which manual administration becomes difficult due to a significant number of heterogeneous components as well as continually changing network scenarios and conditions. The emerging technologies of Software-Defined Networking (SDN) and Network Function Virtualization (NFV) resulting in software-driven networks offer a unique solution for Autonomic Communications for such networks.

This paper presents and discusses challenges of Autonomic Communications related to networks and services based on Software-Defined Networking (SDN) and Network Function Virtualization (NFV). We firstly discuss the differences between Autonomic Computing and Autonomic Communications, and their latest use cases. Then we focus on network self-management as well as on-going research projects and standardization activities on different perspectives of self-management of networks. Following this discussion, we further elaborate issues of automatic service testing, integration, and deployment in the context of virtualized network functions. Another important purpose of Autonomic Communications is to automate network performance optimization procedures by analyzing real-time network data. Therefore, we discuss the importance of applying machine learning approaches to implement network self-optimization. For end-users, a well-designed Business Support System (BSS) is important. Therefore, the technical details of Autonomic Communications has to be complemented with some economic considerations. We cover this by discussing the requirements that must be met for a BSS system in Autonomic Communication systems. Last but not the least, secure communication must be guaranteed for any autonomic operations. We discuss challenges to deliver secure autonomic communication system.

Software-driven networks provide the fundamentals for Autonomic Communications systems. SDN and NFV can support a variety of networks and services such as mobile/cellular networks, Internet of Things, vehicular networks etc. In case of dynamic scenarios and large-scale networks, self-management schemes based on Autonomic Communications schemes are helpful to avoid manual operations. Section II describes the differences between Autonomic Computing and Autonomic Communications as well as discusses the use cases of Autonomic Communications.

NFV allows service providers to improve service innovation and deployment agility, running Virtual Network Functions (VNFs) in Virtual Machines (VMs) deployed over data-center infrastructures and replacing Network Functions (NFs) using dedicated hardware appliances. Softwarization of networking creates a large number of distinct stand-alone software modules that have to be managed to provide appropriate Quality-of-Experience (QoE) to end-users. This large and dynamic number of heterogeneous components causes a management problem, since a human operator is unable to manually manage a large variety of interconnected systems. Therefore, automatic management techniques are required to manage current ecosystems of mobile, vehicular, and Internet-of-Things (IoT) ecosystems (cf. Section III).

NFV requires a much higher degree of automation than physical network function infrastructures since additional

Manuscript received April 1, 2017; revised September 12, 2017; accepted September 25, 2017. Date of publication October 9, 2017; date of current version December 1, 2017. *(Corresponding author: Zhongliang Zhao.)*

Z. Zhao, E. Schiller, E. Kalogeiton, and T. Braun are with the University of Bern, Bern CH-3012, Switzerland (e-mail: zhao@inf.unibe.ch).

B. Stiller is with the University of Zürich, CH-8050 Zürich, Switzerland.

M. T. Garip, J. Joy, and M. Gerla are with the University of California Los Angeles, Los Angeles US-90095, CA USA.

N. Akhtar and I. Matta are with Boston University, Boston US-02215, MA USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2017.2760354

management tasks are needed, e.g., for orchestration, deployment, scaling etc. NFV automation should be entirely software-driven, i.e. the entire network function orchestration procedure must assemble the underlying NFV infrastructures, VNF service chains, and virtualized test devices. Given the large size of potential service chains, manual testing cannot keep pace with time-to-market requirements due to the increased flexibility and agility of the NFV service design. Therefore, test automation and service orchestration play essential roles for efficient NFV deployments, cf. Section IV.

Autonomic communication systems should be able to optimize communication performance and to detect optimal behaviors for overall performance improvement in a fully autonomous way, without any human intervention. This requires collaborating hardware and software systems to maximize resource utilization and to meet end users' needs. As discussed in Section V, Machine Learning and Big Data analysis work as ideal enablers to trigger self-optimization processes.

In general, the commercial dimension of a technology is added in terms of Business Support Systems (BSS), once an operational deployment is foreseen and supported by the respective Operation Support System (OSS). Thus, the BSS serves as an enabler for service providers in the networking domain to collect customer preferences, to deal with customer handling, and to utilize this knowledge for introducing new services. In the context of Autonomic Communications, BSSes have not been defined yet. In turn, Section VI proposes a first analysis of relevant functionality in need and proposes new management parameters to be applied.

Autonomic communication systems must adapt to real-time threats in order to provide continuous and self-managed security protection. NFV security defense chains (e.g., stateful firewalls, deep packet inspection, anti-virus proxies) enable reconfigurable security capabilities that can be dynamically adjusted and deployed across the network to protect against incoming malicious network flows. In addition to protecting the network, care must be taken to protect the security chains themselves against persistent malicious actors. Techniques such as moving target defense, formal verification, intrusion detection, and federated authentication and authorization are needed to maintain security integrity of autonomic communications systems. We describe more details in Section VII.

II. Autonomic Communications

The term *Autonomic Communications* has been heavily influenced by the *Autonomic Computing* paradigm [1]. Therefore, we first introduce Autonomic Computing principles before defining the term Autonomic Communications. Then we discuss related work in that field including Cognitive Networks as well as use cases of Autonomic Communications.

A. Autonomic Computing

An Autonomic Computing system aims to manage itself according to the goals defined by a system administrator. Autonomic Computing is much about self-management relieving the system administrator from detailed and manual operation activities. Autonomic Computing systems should adapt themselves autonomically to various dynamic contexts and conditions. Among those are varying system load, new soft- or hardware components, soft- or hardware failures, etc. Self-management has four aspects, namely self-configuration of components and systems; self-optimization by tuning parameters to improve performance; self-healing to detect soft- or hardware problems, to analyze and repair these; and self-protection to automatically protect and defend against (un)intended attacks.

An Autonomic Computing element consists of at least one managed element and an autonomic manager [1] that monitors the state and context of the managed element, analyzes the system state and context, identifies actions based on previous knowledge on how actions performed under similar conditions, and executes commands on the managed element. Machine learning has been considered as promising approach to support learning in Autonomic Computing.

B. Autonomic Communications

Autonomic Communications applies such Autonomic Computing principles to communication systems and networks, e.g., to provide end-to-end Quality-of-Service using self-management [2]. Similarities between Autonomic Computing and Autonomic Communications disappear if communication is considered as a particular service provided by underlying computing and communication infrastructures. The trend of virtualizing network functions and running them in cloud-based systems further makes the differences between Autonomic Computing and Autonomic Communications disappear. While Autonomic Computing focuses on application software, Autonomic Communications aims at running communication software and services autonomously. Resource management not only includes computing and storage resources as in Autonomic Computing, but also network resources such as fiber capacities or wireless frequency spectrum.

Autonomic Communications has emerged a decade ago. Dobson *et al.* [3] define Autonomic Communications as a way "to improve the ability of network and services to cope with unpredicted change, including changes in topology, load, task, the physical and logical characteristics of the networks that can be accessed". Similarly as in [2] an autonomic control loop is proposed to monitor and collect context and status information from the network, analyze these, decide about actions, and execute those (Collect-Analyze-Decide-Act). Some approaches for Autonomic Communications have been inspired by natural adaptive systems and their ability to self-organize their activities, e.g., bacterial and insect colonies [4].

C. Cognitive Radio and Networks

The work on cognitive radio and networks is also related to Autonomic Communications. The term Cognitive Radio [5] intended to describe intelligent radios that can autonomously make decisions using gathered information about the radio environment and can learn/plan based on their past experience. Wireless network nodes change their radio transmission or reception parameters to communicate efficiently and avoid interference with (un)licensed radio spectrum. The proposed cognition cycle in [6] is based on Observe-Orient-Plan-

Decide-Act-Learn, i.e. the cognitive radio system continually observes the environment, orients itself, creates plans, makes decisions, and then acts. Machine learning is proposed to derive decisions from previous actions and their observed impact. The cognitive radio concept was extended to higher protocol layers resulting in cognitive networks [7]. A cognitive network can perceive current network conditions, and then plan, decide, and act on those conditions. The network can learn from these adaptations and use them to make future decisions. A knowledge plane [8] was proposed to build and maintain high-level models of what the network is supposed to do, to provide services and advise other network elements.

D. Use Cases for Autonomic Communications

While early work on Autonomic Communications focused more on traditional fixed Internet based networks, other more dynamic types of networks emerged with even stronger demand for Autonomic Communications due to even higher degree of dynamics. In particular, SDN and NFV enable higher dynamics of networks and services due to dynamic instantiation and configuration of network functions and services in virtualized (fog/cloud) computing environments. As example, the Service Cloud concept [9] has been designed to provide rapid deployment of services, e.g., transcoders. Depending on the amount of traffic and the number of users, network functions and services have to be instantiated or stopped, which also requires dynamic adaptation of computing, storage, and communication resources.

We further see emerging use cases of Autonomic Communications in mobile application scenarios such as the Internet-of-Things (IoT), e.g., in smart homes and smart cities applications, with varying radio conditions and large dynamic numbers (due to node failures and duty cycling) of interconnected sensors and actuators. IoT provides self-management techniques, since smart objects are adjusting to different situations, e.g. organization in ad-hoc networks for exchanging information and performing coordinated tasks, even when the topology is dynamic [10]. Autonomic Communications and self-management capabilities should be supported by any smart object, e.g. router box, and should be used to optimize any decision and task that the network will perform, e.g. scheduling of packets [11].

Another use case in highly dynamic application scenarios includes Vehicular Ad-Hoc Networks (VANETs) with varying numbers of interconnected vehicles, radio conditions, speeds, directions, etc. The autonomy of VANETs is achieved by utilizing Vehicle-to-Vehicle, Vehicle-to-Infrastructure or Infrastructure-to-Infrastructure communications to exchange necessary information between vehicles and/or infrastructure [12]. A vehicle is self-managed, by controlling its resources and its routing tables. It makes its own decisions (forwarding, requesting, routing, storing information) based on the configuration that exists in the VANET [13]. An SDN architecture brings autonomy in VANETs, since SDN provides adaptation, flexibility and programmability with every network change (topology, channel, etc.), without interfering with other networks. With the configuration of the SDN control plane, the network could not only adjust to network changes, but also to emergency situations (e.g. an accident) [14]. In addition, SDN can be combined with machine learning methods, to learn the behavior of the network and decide what is best, considering previous patterns/behaviors. Also, machine learning can be used to predict behaviors (based on past information) and send necessary messages related to this information (e.g. send defense mechanisms as a reaction to a security breach). The network can optimize the resources, since it learns from multiple sources and can apply the changes to adapt its learning process. For instance, it can devote more resources (i.e. bandwidth) to distribute emergency signals to all nodes, instead of assigning such resources to a multimedia stream.

The mobile broadband network use-case is frequently used in subsequent sections of this paper to demonstrate various techniques of Autonomic Communications. LTE (3rd Generation Partnership Project (3GPP) Long Term Evolution) consisting of the Core Network (CN) and Radio Access Network (RAN) is a flagship example of the mobile broadband network. In LTE, the RAN is provided by the evolved NodeB (eNB) NF, while other NFs such as Serving Gateways (SGW), Packet Gateways (PGWs), Mobility Management Entities (MME), or Home Subscriber Servers (HSSs), build the CN. It is demonstrated that a mobile broadband network could be provided as a software-based VNF chain [15], [16] interconnected through SDN [17]. While the large number of heterogeneous modules (i.e., VNFs) as well as continually changing wireless network scenarios and conditions make management in mobile networks a tedious task, Autonomic Communications may significantly simplify this problem. Some basic notions of network intelligence with respect to cell auto-configuration, self-optimization, self-healing, energy optimization, etc., have been already introduced by 3GPP in LTE Rel. 8 [18]. However, softwarization and virtualization of NFs open up new possibilities for Autonomic Communication that have to be simultaneously explored in both CN and RAN.

Concluding, all those dynamic network environments create challenges for network as well as service management and demand for Autonomic Communications. Three classes of factors are responsible for those challenges [19]:

- Heterogeneity of hardware and dynamic software entities: Network devices span from small sensors and actors to high-end servers. NFV allows the dynamic creation of network function and service entities.
- Dynamics of networks: Network topologies change dependent on radio conditions, mobility of users and vehicles, varying load over a day, link failures, etc.
- Decentralization and control: Decentralization of networks makes it hard to deploy centralized forms of management and control over these devices.

III. SELF-MANAGEMENT OF NETWORKS

SDN and NFV offer a solution for service providers to achieve greater service deployment agility and provide better user experience in many application scenarios. NFV allows for dynamic deployment of VNFs, while SDN distributes traffic among the VNFs providing a dynamic VNF function chain.

TABLE I
SDN AND NFV RELEVANT STANDARDS DEVELOPING ACTIVITIES

Organization Type	Organization Name	Mission	Main Efforts
Industry Initiative	Open Networking Foundation (ONF) [25]	Industry consortium dedicated to the promotion and adoption of SDN through open standard development.	OpenFlow
SDO	Internet Engineering Task Force (IETF) [26]	The Internet technical standards body. Produces RFCs and Internet standards.	Service function chaining
SDO	Open Grid Forum (OGF) [27]	Community of vendors, developers and users. Standardization activities in grid computing.	Open Common Cloud Interface (OCCI) for IaaS clouds.
SDO	European Telecommunications Standards Institute (ETSI) [28]	EU-sponsored SDO that produces globally applicable standards for information and communications technologies.	NFV MANO architecture
Industry Initiative	OpenDaylight [29]	Collaborative project under the auspices of the Linux Foundation.	OpenDaylight
SDO	International Telecommunication Union, Telecommunication Standardization Sector (ITU-T) [30]	United Nations-affiliated agency that produces recommendations with a view to standardizing telecommunications on a worldwide basis.	SDN functional requirements and architecture
Industry Initiative	Open Platform for NFV (OPNFV) [31]	Open source project focused on accelerating the evolution of NFV.	NFV infrastructure
SDO	The 3rd Generation Partnership Project (3GPP) [32]	3GPPP virtualization network function management in mobile broadband networks.	Network function management and optimization

Therefore, the primary challenge in validating SDN/NFV-enabled network deployment is the autonomic components management that would simplify network self-management to a great extent. As an example, when a VNF fails (e.g., an MME in CN), a new VNF copy can be instantiated on the fly. The traffic can be immediately redirected toward the new component by appropriately configuring network equipment (e.g., switches) from the SDN controller. Consequently, the infrastructure can address several problems such as signaling storms and flash crowds, at short time-scales [20]. In this section, we focus on self-management issues in mobile broadband networks based on SDN/NFV, since these are intrinsically highly dynamic and difficult to manage through a human intervention. We chose a few distinct examples in this domain to discuss how SDN/NFV could help mobile broadband network operators to improve their services.

A. Mobile Network Operators

M(V)NOs (Mobile (Virtual) Network Operators) tend to replace currently available proprietary components in classical mobile broadband networks to reduce capital and operational expenditures (CAPEX and OPEX). Typical NFs such as firewalls, switches, routers, or CN functions of the telecommunications operator such as MMEs, HSSs, etc., may be provided on-demand as software components. As an example, the EU FP7 MCN project [16], [21] established one of the first fully cloud-based MNO systems, extending the cloud computing concept to support on-demand and elastic provisioning of mobile broadband network services (e.g., virtualized RAN, virtualized CN, etc.). There will be a growing demand for high complexity networks consisting of a large number of interconnected components (VNFs) [17], [22] in the control plane (C-plane), user plane (U-plane), and management plane (M-plane) that require efficient network management.

B. Standardization

Unlike some specific communication protocols, such as Bluetooth or Wi-Fi, there is no single standardization body responsible for developing open standards in SDN/NFV-related activities. This means that there is a large open community including Standards Developing Organizations (SDOs), industrial consortia, and open development initiatives creating standards for future SDN/NFV-based industry products. Table I provides a few SDOs and other organizations involved in this effort and the main outcomes achieved so far. In the remaining part of this subsection, we will elaborate on the most innovative examples of future network management.

The European Telecommunications Standards Institute (ETSI) has provided an NFV Management and Orchestration framework (NFV-MANO) [23]. NFV-MANO includes details about the roles of the NFV Orchestrator (NFVO), VNF Managers (VNFM) and Virtualized Infrastructure Managers (VIM), as shown in Figure 2. A VIM is responsible for managing and controlling the NFV infrastructure (NFVI): compute, storage, and network resources. VNFM talks directly with VIM and NFVO, and is responsible for instantiation of VNFs, scaling of VNFs, updating and upgrading VNFs, and termination of VNFs. NFVO provides resource orchestrations and network service orchestrations. NFVO ensures that adequate compute, storage, and network resources are available to provide the desired network service. It coordinates directly with VIM and NFVM to achieve this. The aim of the ETSI-MANO framework is to provide initial requirements for autonomous NFV systems, and to define NFV MANO interfaces that can be used for communications between different components (i.e. NFVO, VNFM and VIM), as well as integration with traditional network management systems. This allows NFV MANO to manage functions running on virtualization environments as well as those running on legacy network hardware. However, the scope of the ETSI NFV MANO framework is limited as it does not provide detailed definitions of required interfaces and details on the control and management of legacy equipment. It also does not answer questions such as VNFI requirements specific to different types of virtual functions, which VNFs should run in VMs and which ones in containers, as well as operational requirements specific to NFs deployed

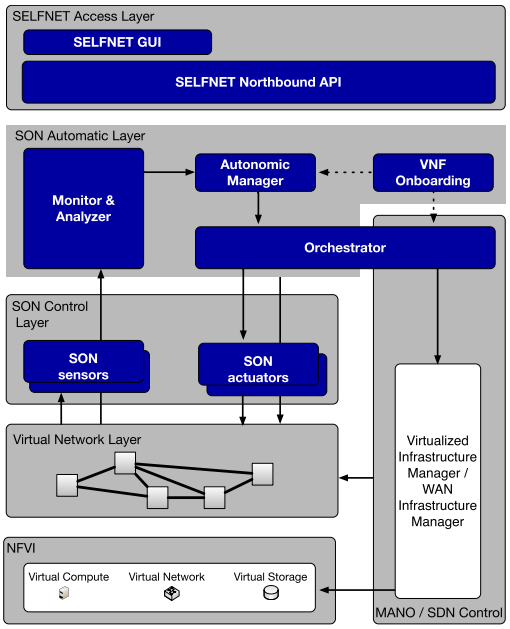


Fig. 1. Architecture of the H2020 5GPPP Selfnet Project.

in data-center environments. ETSI is planning to address many of these questions in the future. ETSI Mobile Edge Computing (MEC) ISG [24] provides enhancements (using the experience previously gathered by ETSI NFV ISG) for managing the virtualized infrastructure at the network edge.

C. Self-Organizing Network Architectures

The H2020 5GPPP Selfnet Project [33], [34] makes use of the underlying management components (c.f., Sec. III-B) and delivers an SDN/NFV-based architecture for self-management of *virtual networks*. The architecture enriches the typical SDN/MANO management with three additional layers, i.e., *Self-Organizing Network (SON) Control Layer*, *SON Autonomic Layer*, and *SON Access Layer* (c.f., Figure 1). *SON Control* is a layer responsible for gathering information about the current system state/health through a set of applications called *SON Sensors*. At this layer, network intelligence is provided through *SON Actuators* enforcing appropriate actions when required. The SON Control Layer is associated with the control and data planes of the system. The main part of network intelligence is implemented by the *SON Autonomic Layer*: *Monitor & Analyzer*, which collects information about the state/health of the system from sensors and provides recommendations to the *Autonomic Manager* that executes necessary management actions through the *Orchestrator* and the set of aforementioned *SON Actuators*. Finally, the *Selfnet Access Layer* provides a GUI for administrators to set up high-level goals of Autonomic Communications.

The self-healing of a VNF function chain is an example of the Selfnet operation in practice [35]. Component monitoring collects detailed information about the health status of VMs. The monitoring agent registers with dedicated services as an observer to gather operational data. When the failure in a module is discovered, the Autonomic Manager autonomously replaces the failed component, e.g., swaps a master component with a backup instance and initializes the chain recovery functions that assure chain consistency (e.g., SDN-based function chaining). Table II reviews projects on Autonomic Communications benefiting from SDN, NFV, and cloud concepts.

IV. Automatic Testing, Integration, and Deployment of Network Functions

With more sophisticated design of VNFs and complex service integration, the traditional way of performing network function testing manually can not meet the requirements for testing and integrating large-scale and continuously-updated network services. These new requirements are the foundations for a paradigm shift in the way network functions are tested, integrated, and deployed. Essential network operation tasks should be fully automated, which include automatic network function testing, integration, and deployment.

A. Automatic Network Function Testing

With more complex design of NFV service chain, the manual testing of different VNF components cannot keep pace with time-to-market requirements due to the increased flexibility and agility of NFV service design. Therefore, automatic network function testing must be supported to automate the testing of individual VNF modules without human intervention. To do this, a network test automation platform including special testing programs has to be designed to test NFV implementations automatically. This means when network services are designed in SDN/NFV-enabled networks, an accompanying testing service should also be designed specifically for that service, which is responsible for validating the functionality and performance of that service. Moreover, a testing module that is responsible for integrating multiple VNFs should also be ready to ensure smooth service integration.

To support automated NFV function testing, a comprehensive automation platform is required. This platform should handle all the issues of NFV function testing, including physical resource management, service provisioning and testing, and testing automation. The following capabilities should be included in a SDN/NFV testing platform [43]:

- A centralized system for managing both SDN/NFV and legacy network components.
- Integration of all types of network infrastructures, including legacy network components, SDN components(SDN switches and controllers) and VNF components.
- A visual work-flow to support automated testing.

Scalability testing as an issue to be addressed in addition to function testing aims at evaluating the maximum number of control plane sessions that could be maintained in parallel by the system. For instance, in the network routing scenario, the maximum number of routers per session and the number of routes that are recorded in the routing tables are the proper metrics of scalability testing. In VNF deployments, VNFs should be able to support auto-scaling operations such that physical network resources could be managed dynamically in response to real-time network conditions and end-user requirements. The purposes of this auto-scaling testing have to

TABLE II
EXAMPLE RESEARCH/INDUSTRY PROJECTS WORKING ON NFV-ENABLED NETWORK SELF-MANAGEMENT

Project Name	Project Scope and Contributions
EU FP7 SEMAFOUR [36]	The SEMAFOUR project was the first approach to design a SON managing and operating heterogeneous mobile broadband networks.
EU H2020 5GPPP Selfnet [33]	The SelfNet project targets to a next generation system with self-organizing capabilities. The project focuses on the network management providing SON building on top of SDN, NFV, and cloud.
EU H2020 5GPPP 5G-XHaul [37]	5G-XHaul focuses on the development of the cognitive control plane able to predict spatio-temporal traffic patterns to adequately configure SDN/NFV CN components.
EU H2020 5GPPP CogNet [38]	The CogNet project aims to develop and make use of the machine learning algorithms to provide network resiliency and anomaly detection such as intrusion and fraud.
HP OpenNFV [39]	Implementation of HP NFV Reference Architecture that is aligned with ETSI architecture
Huawei NFV Open Lab [40]	Focus on the setting of an environment to ensure that NFV solutions and carrier grade infrastructure are compatible with emerging NFV standards
Cisco Open Network Environment [41]	Implementations for some of the functional blocks of ETSI MANO framework
Intel Open Network Platform [42]	Focus on the ecosystem made up of several initiatives to advance open solutions for NFV and SDN

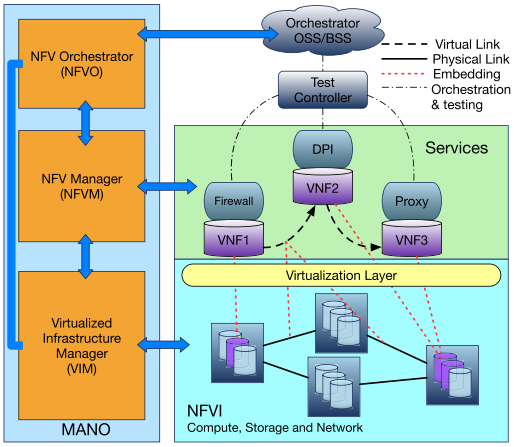


Fig. 2. ETSI NFV testing in the cloudified environment.

make sure that: (1) VNFs are able to perform the auto-scaling operations in a proper way to meet the varying network traffic; (2) physical network resources are controlled in an efficient way; (3) the overall system performance meets Service-Level Agreements (SLAs) under any traffic conditions. Function testing could be performed by running tests with different traffic rates, and scalability testing could be performed by adapting resources that are allocated in the test [43].

B. Automatic Network Function Integration

Efficient network function integration is another key factor for successful network deployment. Integration testing assesses end-to-end service provisioning across multiple network services/functions from different entities. Due to the large number of network functions involved, traditional integration testing is a manual, time consuming, and cost intensive task. Therefore, an approach to automate integration testing is important to automate service deployment and provisioning.

To support automated network function integration, ETSI has proposed a generic solution [44], which includes all the relevant network components for automated integration testing. An example of an ETSI NFV testing scenario is illustrated in Fig. 2. The idea is to automate user interactions when data traffic is detected at all standardized interfaces connecting equipment under test. As shown in Fig. 2 the "Test Controller" is responsible for handling all user interaction experiences and it will generate a test report after each integration testing for further analysis. In addition to the proposed testing architecture, ETSI also defines standard procedures to perform automated integration testing. The procedures include the detailed specification and regulations of executable test cases independent of any programming language that is used in integration testings.

C. Automatic Network Function Deployment and Adaptation

VNFs can be deployed, instantiated, and disposed on-demand using cloud application life cycle management techniques such as orchestration frameworks. However, VNFs need further configuration at runtime to fulfill customer-specific service demands while considering the available virtual resources. This means that automatic network function provisioning and dynamic adaptation capability is required, which considers not only the service requirements and characteristics but also the real-time status of the network infrastructures. This leads to the dynamic decision on the location selection of the deployed services considering not only the service requirements, and also the key indicators of the network health (anomalies, performance). The NETCONF/YANG data model [45] is an example that can be used to support automatic service dynamic adaptation.

As a summary, in the context of SDN/NFV-enabled autonomic communication systems, autonomic network function testing, integration, and deployment should follow the following principles: (1) functional validation of VNFs interaction with isolated NFV functional blocks; (2) user and control plane performance validation, which includes the assessment of capacity management, e.g., during VNF scale-out to ensure that performance levels adhere to SLAs; (3) validation of reliability and availability of VNFs, NFVI, and services during workload migration. Compared to the traditional interoperability testing process, where the execution of interoperability tests requires some degree of manual control by test operators, the service testing and integration in autonomic communication systems will be fully automated. This means that a virtualized testing function will be needed, which provides

testing scripts that are responsible for the testing purposes. This enables testing both valid and invalid behaviors, but requires writing test scripts and running them in a conformance testing environment.

V. Self-Optimization Based on Machine Learning and Big Data

A tremendous amount of heterogeneous data is generated by the VNFs of software-driven networks, demanding for Big Data analytics. While lacking a formal definition, Big Data applications practically share three features: (1) large volume of size, (2) high velocity of generation, and (3) high degree of variety. To get insights from these data to optimize network performance, advanced machine learning (ML) technologies could be applied to learn the characteristics of data traffic such that network operations can be optimized.

In particular, monitoring of mobile broadband networks can generate such a huge amount of data at large scale. Certain network providers collect flow information, some even record information of individual packets. Optimization of network performance can be based on such collected data. However, it is rather difficult to derive how variations of monitored parameters and in particular how their combinations affect network performance. Especially, when monitored parameters are changing quickly, as for example in mobile broadband networks due to dynamic channel conditions and user mobility, performance optimization becomes tricky. Even though ML algorithms can bring huge benefits to many application scenarios, there are certain issues to be addressed before applying them to solve Big Data problems. This is because ML algorithms optimize system performance by learning patterns from historical data. However, the training/learning phase is difficult for Big Data applications due to the fact that iterative operations in ML algorithms are hard to be parallelized. When the application dataset is of huge size, this problem is more severe and can significantly reduce the performance of ML algorithms. Therefore, an effective and scalable algorithm to train large scale data models in parallel using advanced ML algorithms is of vital importance. In the following subsections we discuss examples where ML algorithms are used to optimize performance in mobile broadband and fixed networks.

A. Mobile Broadband Networks

Mobile broadband networks are characterized by dynamically changing radio conditions, large user sets, and user mobility. Information about radio signal quality, application performance, or user mobility and location can be monitored and recorded. Such data can then be stored and analyzed by ML algorithms. ML algorithms can be applied to stored monitored data to identify conditions and actions that led to performance improvements or even degradation. There are several related works where ML has been proposed to extract and predict certain information from monitored data.

ETSI has just created the Industry Specification Group of "Experience Network Intelligence", which aims at using Machine Learning/Artificial Intelligence techniques and context-aware policies to enable mobile network operators to adjust offered services based on changes in user needs, environmental conditions and business goals [46]. Jiang *et al.* [47] propose ML for 5G end systems to learn about the best spectral bands to be used to simultaneously achieve low energy consumption and robust communication, in particular in Multiple-Input-Multiple-Output (MIMO) systems. Supervised, unsupervised, and reinforcement learning algorithms can be applied. Kaup *et al.* [48] performed a comprehensive measurement of various performance parameters in cellular networks such as throughput, round-trip-times, signal strength etc. Such measurement data has been complemented by time of measurement and GPS location of the mobile device. Certain correlations between throughput and RTT/signal strength were observed in 3G and 4G networks. It was observed that the association of mobile devices to points of presence (PoPs) has significant impact on performance. ML algorithms using classification trees were used to predict PoPs with an accuracy of 56%.

Mobility prediction of users can be exploited to optimize performance in mobile networks. Based on the predicted location or cell of a user, handover decisions can be performed, e.g., avoiding service disruptions in case of too late handovers. Sousa *et al.* [49] proposed to use mobility prediction to support a priori population of Information-Centric-Networking (ICN) caches. Mobility prediction can be based on observing geographic GPS coordinates only, but better prediction accuracy could be observed if the type and semantics of a visited location is considered [50].

For performance optimization but also for security purposes, it might be beneficial to classify traffic flows, even when they are encrypted and protocol identifiers are not accessible by monitoring tools. [51] describes a system that classifies flows (transmitted over wireless networks) based on a large pre-recorded data set using several ML techniques such as KNN (K Nearest Neighbor), Gaussian Mixture Model, Tree Adaptive Parallel Vector Quantizer, Binary Classification Trees, etc. Success rates of 80-90% could be achieved.

B. Fixed Networks

ML algorithms can also be used in fixed networks, e.g., for optimizing routing decisions [52]. Reinforcement learning has been used to discover efficient routing policies in dynamically changing networks without knowing network topologies and traffic patterns. Another ML application is the detection and localization of network faults based on monitoring and observing events such as alarms, alerts, and specific metrics. The approach in [53] is based on creating so-called network signatures including fault type, time information, and event probabilities. The challenge is to find matching signatures in order to find topologically and temporally relevant events and to find reasons for network faults. The Generic Root Cause Analysis (G-RCA) platform [54] aims to support service quality management in large IP networks. G-RCA abstracts the RCA process into signature identification for symptom and diagnostic events, temporal and spatial event correlation, and reasoning and inference logic. It uses data trending and statistical methods to find temporal and spatial event correlations.

ML can also be used for recommendations of wavelength assignments in an optical network [55].

Other works focus rather on application or service level. Genetic algorithms in a cloud computing environment predict future computing load and find an appropriate balance between fulfilling service level agreements as well as energy consumption [56]. Trend-Aware Video Caching [57] can decide about which videos to store in content-delivery network nodes close to users. Based on a set of parameters the proposed system based on KNN-like learning methods predicts short-term popularity trends for videos in order to be prepared for storing most popular videos. Trend-Aware Video Caching decides whether and where to cache video content and which cache replacement strategies to employ.

VI. Autonomic Operation and Business Support Systems

When the technical perspectives of Autonomic Communications and operations in software-driven networks are handled, the respective Operational Support System (OSS) and Business Support System (BSS) should ensure that the technical operations are complemented with economic perspectives.

A. Definitions and Requirements of Operational and Business Support Systems

Autonomic operation of NFV/SDN networks requires the optimization of management operations of dedicated systems and network elements in the sense that any minor and major decisions can be taken locally by the module in operation. Such approach determines the technical perspective of Autonomic Communications and operations, which is guided typically by respective management policies. Thus, the respective Operational Support System (OSS) contains distributed functionality to monitor, analyze, control, and manage at the technical level a network operated by a telecommunications service provider. However, this needs to be extended beyond the technology perspective, since an operation may be considered only successful, if it can be viably operated commercially in a business environment. Thus, the technical perspective of Autonomic Communications has to be complemented with the economic perspective, which determines a possible approach to design BSSs for autonomic operations. Since various definitions of a BSS exist, the following based on [58] was updated here for the Autonomic Communications case. The BSS enables a service provider in the networking domain to

a) collect customer preferences and behaviors,
b) operate customer handling in terms of new and terminating subscriptions and contracts, and
c) utilize this knowledge gained and customer management operations performed to introduce new services, which are considered revenue-wise successful.

The BSS functionality typically includes

a) customer management, including order, change, and complaints handling,
b) customer data management, covering personal data for subscriptions as well as usage-specific data,
c) pricing and rating (telecommunication service providers term this functionality as billing and rating), and
d) a use case-specific instance of functionality for either Business-to-Consumer (B2C) or Business-to-Business (B2B) services.

A BSS determines an important, if not the most critical, component in a commercially operated network and service offering to map customer and business needs onto existing or future technology. All technology-related actions – as defined above – are handled by the OSS. Thus, Autonomic Communications applied to NFV/SDN leads to the clear demand for a suitable BSS, which takes explicit advantage of the autonomic dimension into account and which sees concrete instances of additional functionality and parameters for application-specific facets and operations, such as a management policy, e.g., driven by customer churn-based measurements and data. Furthermore, the question if a BSS by itself can operate in an autonomic manner, remains unanswered so far, since the nature of business-related decisions and policies seems to be more central than distributed. Therefore, the specific and general potential of autonomic BSSs still has to be evaluated.

Generally speaking, a BSS offers to any networking and service provider — guided by the respective technology-specific OSS at hand — a business environment. This business environment at best meets the following requirements:

a) it converges multiple technological variants in use under one business umbrella in terms of unified charging and accounting models,
b) it provides an integrated view of operational data as well as customer preferences and service usage to checkpoint, decide, and control based on mid-term optimization goals (in contrast to short-term technical parameter updates),
c) it synchronizes service-specific instances of technology components to match pricing models (typically based on OSS-provided technical data and measurements).

Therefore, the OSS-based overall system control and tailored scheduling of services match technology constraints and facilitate, in combination with a BSS, the instantiation of updated, changed, or newly created service offerings:

a) following business goals in close feedback cycles
b) satisfying customer experiences according to Quality-of-Experience (QoE) metrics in user feedback cycles.

In that setting BSSs embrace the entirety of services-related business processes, but typically are technology-specific. This is backed by the fact that nearly every networking technology developed and introduced into commercially operated networks has led to the design and installation of a respective BSS. Since many functions remain technology-independent, vendors of BSSs over the past three decades did develop and offer coherent BSS solutions for multiple technologies. However, due to investment costs and operational reasons of reliability and availability, integrated BSSs have not been introduced in all cases of larger or smaller networks. In the same line BSSs for Autonomic Communications still have to be specified in details and instantiated in turn.

B. OSS for Autonomic Communications

Besides this current situation, the specifics of a suitable OSS for Autonomic Communications, e.g., in a vehicular network environment, can be outlined from a pure technological perspective as follows (the key autonomic communication characteristics are based on [3]):

a) dealing with unpredictable changes, including changes in topology, load, tasks, physical as well as logical characteristics, mobility models, and movement patterns,
b) supporting end-to-end issues affecting programming models and service offerings, e.g., vehicle maintenance alerts between the owner and the repair shop or emergency services with position-based location detection,
c) enabling network and contextual modeling and reasoning in a manner that adapts to vehicle locations and passengers' interests or driving destinations,
d) applying (fully) decentralized algorithms, which may be operated policy-based on preferences and locally available context/location data,
e) handling distributed trust and its acquisition for new stakeholder interactions, and
f) performing maintenance tasks.

C. BSS for Autonomic Communications

Since the autonomic control loop Collect-Analyze-Decide-Act [3] is applicable to operations of a BSS too, (with the exception that such loops are not run at short time-scales of seconds and minutes, but operated on a daily, weekly or monthly manner), the Autonomic Communications' addition to those six characteristics as defined above and to a new BSS with respect to the economic perspective covers economically relevant parameters and business logic. Such parameters for an Autonomic Communications BSS include, among others, the following parameters described in the same order as above:

a) customer churn, customer behavior changes, changing customer price sensitivity, and updated customer service preferences,
b) the value of (i) emergency services, (ii) real-time services including streaming, and (iii) non-real time services with a minimal level of QoE guarantees, all depending on the specific customer needs,
c) economic parameters in support of reasoning and decisions, such as the value of a service itself or an economically valued outcome of a service usage,
d) typically, based on a price-related information push to devices or system components, the application of decentralized algorithms in certain policy-constrained cases optimizes based on cost effectiveness, physical contexts and communication costs, or customer's price sensitivity,
e) the valuation of trust in economic terms, which has not yet been fully developed, needs a suitable integration of efficient and attack-resilient reputation systems,
f) maintenance costs, service deployment costs, service discontinuation costs, service interruption costs, and service parametrization or tailoring costs.

Thus, the embedding of the Autonomic Communications-related OSS perspective into a newly adapted BSS reveals that local decisions to be taken will see an impact on the overall business model implemented for current service offerings, since it does constrain possible actions due to traditional business and customer requirements available in a more central entity only. For an operational approach, a decentralized policy control (in contrast again to traditional BSSs) will be able to operate locally. However, any update of data collected centrally (from within the BSS) will be essential to be distributed back to these components to adapt to key business model variations (cf. especially the first item above). As such, the Autonomic Communications model will remain unchanged (especially its local and short-term decision making), but a smaller centralized interaction cycle, operating on a much longer time-scale, to update the policy base and its associated economic values will become mandatory. Thus, the BSS for Autonomic Communications sees those major influencing facets as outlined, from which autonomic BSSs still have to be evaluated and operations on longer time-scales and associated economic values will become mandatory. However, investigations on, e.g., the frequency of such interactions as well as the level of detail and granularity of important economic and business parameters are still to be undertaken.

VII. Autonomic and Distributed Security Functions

Secure data communication should always be guaranteed for any Autonomic communications systems. In this section, we discuss design challenges and propose possible solutions for distributed and secure autonomic communication systems.

The self-managing and dynamic nature of Autonomic Communications enables timely and effective defenses against numerous incoming attacks. Relying on human intervention for a prompt defense against the attacks being performed on the complex network architectures such as SDN/NFV architectures in mobile broadband networks as well as in vehicular and IoT networks is infeasible. Distributed auto-defense systems in Autonomic Communications (e.g., NFV security chains) constantly monitor the system parameters for possible ongoing attacks and determine the best course of action—hopefully without significantly hurting overall performance—while preset security configurations manage authentication, authorization, and access control. In this section, we present crucial security tasks in Autonomic Communications and existing mechanisms that perform these tasks.

A. Federated Authentication and Authorization

Autonomic networks, i.e., networks based on Autonomic Communications principles, will consist of many separate dynamic autonomous security domains. These security domains are designed to be self-maintained and self-adaptive without depending on user administration. Therefore, federated authentication and authorization (FAA) mechanisms play a crucial role in autonomic networks. Their main objective is to seamlessly facilitate secure inter-domain data access to users.

The concept of federation in authenticating and authorizing users enables secure data access in highly dynamic Autonomic Communications, where network nodes might frequently switch from one security domain to another. In order to achieve this, the existing FAA mechanisms consist of two main entities: *Identity Providers* (IdPs) and *Service Providers* (SPs). IdPs are distributed systems that are responsible for authenticating users for access to SPs. One IdP is capable of granting access to multiple SPs—from different institutions and security domains—with just one authentication as long as these SPs are part of the same federation.

When a user attempts to access an SP, the SP redirects the user to one of the IdPs of the federation for authentication. After the correct credentials have been provided by the user, the IdP issues an identity for the user and this identity is sent back to the requesting SP for authorization, and then to the user for future requests. This identity is called *federated identity*, which is formatted and exchanged according to the Security Assertion Markup Language (SAML) standard [59]. This mechanism is also called Single Sign-On mechanism [60]. The federated identity issued to the user is formatted as XML, XML-signed by the issuing IdP, and sent to all the SPs (not only the SP being accessed) participating in the federation. The exchanged XML packets contain *assertions* that include information about the IdP, identifying attributes (roles, privileges, etc.) of the authenticated user and authorization decisions based on these attributes. The user then can access all the SPs from more than one enterprise/institution in the same federation with just one authentication step, even though authorization decisions might be different for each SP. Liberty Alliance [61], WS-Federation [62], OpenID [63], and Shibboleth [64] adopt this mechanism.

Authorization in federations is performed in a completely distributed manner [65]–[68]. After a federated identity is assigned by an IdP and sent to all SPs in a federation, each SP grants permissions based on the roles that the authenticated user has for that SP, which are specified in the assertions of the user's federated identity. This is called Role-Based Access Control (RBAC). RBAC models address the issues associated with frequent security policy updates that might occur in large federations in autonomic networks. Whenever an SP decides to update a security policy, it can just change the permissions of the corresponding roles, rather than iterating over all user profiles in the SP that might get affected by this update and determining the most appropriate permissions for each profile according to the new policy.

B. Intrusion Detection

An intrusion detection system (IDS) in the field of communications refers to a monitoring tool that analyzes incoming/outgoing network traffic based on configured heuristics and security policies to detect any malicious activity. Based on these pre-configured settings, IDSs build a model of what is regarded as normal traffic and apply machine learning techniques to identify anomalies, which are often network attacks. After the detection, an IDS either just alerts the network administrator or takes an action to prevent the ongoing attack in addition to the alert. Vanerio and Casas [69] evaluate different Machine Learning (ML) techniques and ensemble learning, i.e. combinations of ML techniques, for anomaly detection. Ensemble learning shows superior results.

In the current Internet, IDSs are usually deployed at the network edge (on user machines or access networks) analyzing both incoming and outgoing traffic to identify malicious behavior. However, IDSs could be much more effective when deployed on the backbone routers in the network core: First, an IDS at the network edge can protect only the individual machines behind the gateway where the IDS is deployed, or the nodes being attacked by these machines. On the other hand, IDSs deployed on the backbone routers can protect the network as a whole. Second, detecting the intrusion at the edge is already too late; for example, a DDoS attack will still consume the victim's resources and affect the legitimate traffic if it gets detected at the edge. The main reason that it is infeasible to deploy IDSs in the network core is that the backbone routers in the current Internet architecture are not designed to be flexible or programmable. SDN routers, on the other hand, make the deployment of new systems such as an IDS much easier. Therefore, SDN also enables IDSs to be able to build better traffic models due to the sampling of larger traffic volumes.

Even at the edge of the network, [70] shows that an SDN-based IDS performs better than the IDS deployed by the Internet Service Provider (ISP) in detecting intrusion for home networks. Even more effective and accurate defense could be achieved by pushing the deployment of IDSs further to the network core as aforementioned, an example of which is demonstrated in [71], making it possible to prevent attacks close to their sources. For example, during a DDoS attack, anomalously large traffic flows from individual machines could be filtered out before they merge and reach dangerous volumes on the way to the victim. Alternatively, [72] proposes NICE, an even more proactive IDS, which analyzes the vulnerabilities in the end nodes—instead of analyzing the network traffic—to identify and blacklist possible malicious nodes even before they can participate in a DDoS attack.

There are already numerous IDSs designed for the current Internet architecture. Despite the disadvantage of not being deployable on the Internet's backbone routers due to the aforementioned reasons, some are still very powerful and should be taken advantage of. CloudWatcher [73] is an SDN monitoring tool that allows external IDS software—possibly one of these powerful legacy IDSs—to be plugged-in to its traffic analyzer to detect intrusions. As a result, any legacy IDS can then turn into an SDN-based distributed IDS deployed in the network core. Other effective IDSs are empowered by the capabilities that SDN brings to networks [74]–[77].

The implementation of an SDN-based IDS as a controller application is the most desirable design choice due to its simplicity. On the other hand, during the network traffic analysis to detect intrusions, every IDS uses packet-level information to be fed into their detection mechanisms to identify anomalies. However, as a controller application, an IDS will not be able to access this packet-level information. Therefore, [78] proposes FleXam, a flexible sampling extension to the controller plane

so that these packet-level details can be available to the IDSs for detecting intrusions.

Every IDS bases its decisions on its model of what is regarded as legitimate traffic and the security policies that are set by network administrators. However, topologies and policies are frequently updated due to the dynamic nature of SDN. These frequent changes can often aggravate the accuracy of IDSs since their existing models and policies might not be up-to-date any longer in these cases. FLOWGUARD [79] addresses the issue of designing IDSs robust to highly dynamic SDN-enabled networks. Further issues associated with SDN-based IDSs are discussed in [80].

SDN and NFV systems are also utilized for IDSs in 5G networks [81], which have more connected devices than the current Internet infrastructure. The great number of connected devices makes it harder to defend against cyberthreats in 5G networks. Selfnet project produced many SDN/NFV IDS proposals to alleviate this problem [82]. Reference [83] proposes several machine learning algorithms powered by SDN/NFV to detect and mitigate botnet attacks in 5G networks. Reference [84] detects and reacts to botnets using SDN/NFV sensors and actuators. Reference [85] uses multiple layers for botnet detection, each of which has different performance and overhead, to overcome the challenge of analyzing traffic flows in high-throughput 5G networks.

C. Moving Target Defense

One important concern facing Autonomic Communications system security is that an attacker is able to repeatedly probe and gain a deeper understanding of the attack surface and vulnerabilities of the system, as the configuration is static and not changing. System configuration (e.g., topology, network addresses, ports, application traffic patterns) leaks the system's attack surface to an adversary. Attackers exploit the asymmetry that administrators must continuously protect the (relatively static) network from all attacks, while a single determined attacker needs only to find one vulnerability to gain access to the system [86], [87]. Moving target defense (MTD) aims to reduce the attack surface by enabling a *dynamic* communication system, making it difficult for an adversary to understand the attack surface and vulnerabilities of Autonomic Communications. The goal of MTD is to significantly reduce the probability that an attacker is able to easily learn or understand the system and its associated vulnerabilities.

There are two main pieces of structured information that are protected via MTD [88], [89]. The first is network structure. Network structure can be protected and made difficult to learn via techniques such as randomization of the IP addresses and ports, opening and closing extra ports, fake hosts that do no processing, and misleading information regarding system information and version numbers. For example, the immense address space of IPv6 can be used to generate new addresses to provide "unlinkability". This randomized address rotation makes it difficult for the attacker to learn the sender and receiver node pairs. Address rotation techniques are able to perform the address rotation mid-session [90], which further complicates an attacker's understanding of network structure and flow. The second structured information to protect is application structure. Techniques such as randomizing the operating system's address space, varying the application routing between nodes, or even obfuscating the application can be used to protect NFV security chains, operating system, and software applications. For example, zero-day exploits can severely cripple Autonomic Communications systems. As these zero-day exploits rely on buffer overflow attacks, taking away the exploits' ability to guess or decipher the memory address space is vital. Techniques such as Address Space Layout Randomization (ASLR) randomizes the address space of processes (e.g., stack and heap pointers) reducing the probability that a zero-day exploit is able to successfully execute an attack [91], [92]. Instead of performing a successful attack, the zero-day attack will crash the application due to the invalid memory address. Autonomic Communications can detect the failed attack and deploy the necessary recovery mechanisms. ALSR is widely deployed in the major operating systems today and would be suited for NFV security chains.

D. Formal Verification

A majority of the systems' security flaws today are attributed to human error and oversights (e.g., configuration error, unexamined scenarios). Thus, there are large potential benefits of an autonomic network to remove the human in the loop and for the system to adhere to it's desired behavior and specifications via a self-management mechanism. The concept of model checking [93] or formal verification is important to allow the Autonomic Communications systems to validate that it is indeed dynamically reconfiguring and healing itself according to the desired behavior and specification [94]. Any deviations from the model or specifications can have the appropriate fall-back policies executed. Example formal specification languages include NASA's Autonomic System Specification Language (ASSL). ASSL is widely used for unmanned space exploration missions where the space vehicle must be independent and autonomous [95], [96]. Similar specifications and concepts can be utilized by Autonomic Communications systems.

For example, MTD will dynamically reconfigure and randomize the network and application structure. Security policies can be defined to specify the allowable application flows. These security policies are then used to verify the network flow between "users" and "application" nodes. That is, if a particular application has the permission to read a particular object, then the network allows such a flow. Network traffic that lacks application permissions are blocked. Thus, the network flows are verified to match the defined security policies. Flows can be classified into consumer/producer between end nodes, propagation of flows (routing), transformation flows (interoperability between security protocols), and finally filtering flows (firewalls) [97].

To formalize the access control model, NIST has defined the Role-Based Access Control (RBAC) model [98]. Security policies can be defined and enforced utilizing the RBAC model. In the RBAC model, the definitions are as follows: a user is the intelligent agent, role is the authority and

responsibility, permission is the authorization associated with a particular object, and object is any entity.

In addition to specifying and adhering to policies, model checking can be utilized to identify policy conflicts [93], [99]. This prevents authorization conflicts for permissible objects, which may frequently occur in systems with a large number of users or devices. Constraint violations can be verified in addition to exhaustively verifying for conflict states.

Finally, there are higher order graph calculus that are suited for distributed topologies where nodes can perform local computations and verification without relying on a global controller, such as for port graph rewriting [100]. The elimination of a global verifier controller ensures that there is no single point of failure vulnerabilities (e.g., DDOS attacks, advanced persistent attacks, hardware or software failures). Nodes are able to locally compute and validate the integrity of the network and system security policies.

VIII. Conclusions and Future Research

This paper discussed the need for Autonomic Communications in software-driven, i.e. SDN/NFV-based, networks. Autonomic Communications is characterized by self-management properties to adapt automatically to dynamic network conditions and contexts. This is increasingly important due to the increasing size and complexity of networks and many dynamic factors that make it hard for human operators to manually manage networks and services. NFV/SDN concepts have been applied in mobile broadband networks. Standardization activities address autonomic management and operation frameworks to a certain extent, but concrete mechanisms and algorithms for self-management need to be developed.

Multiple perspectives of Autonomic Communications in the context of SDN/NFV-enabled communication system, such as network self-management, automatic service testing, integration, and deployment have been discussed. Network self-optimization using machine learning algorithms is another important purpose of Autonomic Communications to make sure that network performance could be enhanced, and distributed security functions should be provided for any Autonomic Communications system. In addition to self-management, automatic testing of network functions, their integration and deployment also need to be supported in software-driven networks. Machine learning applied on Big Data sets collected by network monitoring infrastructures have great potential to further self-optimize network functions and services. Several examples of current research activities have been discussed, but there is more potential as machine learning toolsets are becoming widely available. Traditional Operation and Business Support Systems (OSS/BSS) have been rather based on centralized approaches, which somewhat contradicts the Autonomic Communications paradigm, since OSS/BSSes do not rely on outsider determining the formulation and distribution of any policy to be operated on. Thus, a BSS/OSS for Autonomic Communications following more decentralized concepts of parameter and policy integrations need to be redesigned. Finally, secure operation of network functions and services must be guaranteed. The paper also discusses approaches to support security without the need of human operators' intervention. Although a significant amount of related work on Autonomic Communications has been done during the last decade, emerging software-driven networks call for novel approaches and mechanisms. Thus, future research challenges on Autonomic Communications in software-driven networks are as follows:

- *Self-Management:* has several open challenges. Cloud-based systems quickly recover from failures by using redundant components requiring low delay in recovery procedures. Moreover, monitoring applications for SON will provide vast volumes of data putting a particular focus on efficient big-data processing. Novel highly efficient algorithms (e.g., machine learning) for SON will have to cope with system predictions to immediately optimize the system state. Moreover, the system will have to automatically recognize different reasons of malfunctioning, (e.g., failures from intrusions) and efficiently cope with such situations (e.g., compromised subsystems should be identified and automatically replaced with NFs immune to failures or attacks).
- *Autonomic Service Testing and Integration:* should be included in the network service design. For instance, in addition to network function virtualization, the service testing, integration and deployment modules could also be orchestrated and automated in virtualized network environments. Where to place this virtualized testing function and when to trigger the deployment requires future research efforts.
- *Autonomic Self-Optimization:* can be achieved by combining machine learning-based data analysis and big data parallelized processing. Therefore, the design of efficient and scalable machine learning algorithms that can run on big data platforms in a synchronized way requires further investigation. It has to be investigated which machine learning algorithms and their combinations are most suited for which optimization problem.
- *Autonomic Business Support Systems:* The specific potential of autonomic BSSes still has to be evaluated in dedicated use cases to determine priorities for mandatory or optional functionality needed (potentially prepared for management standardization inputs) in the context of inter-operable OSSes for Autonomic Communications. Additionally, the operation of those on longer time-scales, especially under the perspective of regularly updated policies from economic and service input dimensions, and their associated economic values' evaluation are needed.
- *Autonomic Security:* Autonomic Communications enable novel defense mechanisms due to flexible and programmable backbone routers. However, as networked systems evolve, new vulnerabilities and attacks arise with them. Therefore, although many effective intrusion detection systems powered by SDN/NFV have been proposed, the attack surface they consider needs to be expanded as future research. In addition, moving target defense is a constant battle between security defenders and attackers that merits more attention in future work. Finally, formal verification of both network topology and system software requires further research.

REFERENCES

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

[2] D. Lewis, D. O'Sullivan, and J. Keeney, "Towards the knowledge-driven benchmarking of autonomic communications," in *Proc. WoWMoM*, Jun. 2006, pp. 499–505.

[3] S. Dobson *et al.*, "A survey of autonomic communications," *ACM Trans. Auto. Adapt. Syst.*, vol. 1, no. 2, pp. 223–259, 2006.

[4] N. Bicocchi and F. Zambonelli, "Autonomic communication learns from nature," *IEEE Potentials*, vol. 26, no. 6, pp. 42–46, Nov. 2007.

[5] J. Mitola and G. Q. Maguire, Jr., "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Apr. 1999.

[6] J. Mitola, "Cognitive radio architecture evolution," *Proc. IEEE*, vol. 97, no. 4, pp. 626–641, Apr. 2009.

[7] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives," *IEEE Commun. Mag.*, vol. 44, no. 12, pp. 51–57, Dec. 2006.

[8] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the Internet," in *Proc. ACM SIGCOMM*, 2003, pp. 3–10.

[9] P. K. McKinley, F. A. Samimi, J. K. Shapiro, and C. Tang, "Service clouds: A distributed infrastructure for constructing autonomic communication services," in *Proc. IEEE DASC*, Sep./Oct. 2006, pp. 341–348.

[10] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of Things: Vision, applications and research challenges," *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[11] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[12] I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, "Towards software-defined VANET: Architecture and services," in *Proc. IEEE MED-HOC-NET*, Jun. 2014, pp. 103–110.

[13] J. J. Mulcahy, S. Huang, and I. Mahgoub, "Autonomic computing and VANET," in *Proc. SoutheastCon*, Apr. 2015, pp. 1–7.

[14] H. Li, M. Dong, and K. Ota, "Control plane optimization in software-defined vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7895–7904, Oct. 2016.

[15] T. Taleb *et al.*, "EASE: EPC as a service to ease mobile core network deployment over cloud," *IEEE Netw.*, vol. 29, no. 2, pp. 78–88, Mar./Apr. 2015.

[16] B. Sousa *et al.*, "Toward a fully cloudified mobile network infrastructure," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 547–563, Sep. 2016.

[17] N. Nikaein, E. Schiller, R. Favraud, and K. Katsalis, "Network store: Exploring slicing in future 5G networks," in *Proc. ACM MobiArch*, 2015, pp. 8–13.

[18] *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Telecommunication Management; Self-Organizing Networks (SON); Concepts and Requirements*, document 3GPP TS 32.500, 3GPP, 2009.

[19] R. Quitadamo and F. Zambonelli, "Autonomic communication services: A new challenge for software agents," *Auto. Agents Multi-Agent Syst.*, vol. 17, no. 3, pp. 457–475, Dec. 2008.

[20] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 44–53, Jul. 2013.

[21] *EU FP7 Mobile Cloud Networking Project*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.mobile-cloud-networking.eu/site/

[22] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 146–154, Aug. 2017.

[23] *Network Functions Virtualisation (NFV)*, ETSI, Sophia Antipolis, France, Oct. 2013.

[24] M. Patel *et al.*, "MobilEe-edge computing—Introductory technical white paper," Sep. 2014.

[25] *Open Networking Foundation (ONF) Portal*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.opennetworking.org/

[26] *The Internet Engineering Task Force (IETF) Portal*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.ietf.org/

[27] *Open Grid Forum (OGF) Portal*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.ogf.org/

[28] *European Telecommunications Standards Institute (ETSI) Portal*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.etsi.org/

[29] *OpenDaylight Portal*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.opendaylight.org/

[30] *ITU Telecommunication Standardization Sector Portal*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.itu.int/

[31] *OPNFV Portal*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.opnfv.org/

[32] *The 3rd Generation Partnership Project (3GPP) Portal*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.3gpp.org/

[33] *H2020 5GPPP Selfnet Project*, Accessed: Oct. 24, 2017. [Online]. Available: https://selfnet-5g.eu

[34] P. Neves *et al.*, "The SELFNET approach for autonomic management in an NFV/SDN networking paradigm," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 2, p. 2897479, 2016.

[35] G. Bernini, G. Landi, D. Lopez, and P. A. Gutierrez. *VNF Pool Orchestration For Automated Resiliency in Service Chains*, Accessed: Oct. 24, 2017. [Online]. Available: https://tools.ietf.org/html/draft-bernini-nfvrg-vnf-orchestration-03/

[36] *EU FP7 SEMAFOUR Project*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.fp7-semafour.eu/

[37] *H2020 5GPPP 5G-XHaul Project*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.5g-xhaul-project.eu/

[38] *H2020 5GPPP CogNet Project*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.cognet.5g-ppp.eu/

[39] *HPE OpenNFV Solution Portal*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.hpenfv.com

[40] *Huawei NFV Open Lab Portal*, Accessed: Oct. 24, 2017. [Online]. Available: http://pr.huawei.com/en/news/hw-411889-nfv.htm/

[41] *Cisco Open Network Environment Portal*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.cisco.com/c/en/us/products/software/one-software/index.html/

[42] *Intel Open Network Platform Portal*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.intel.com/onp/

[43] *Network Functions Virtualization: Testing Best Practices*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.qualitestgroup.com/

[44] *Methods for Testing and Specification (MTS); Automated Interoperability Testing; Methodology and Framework*, document ETSI EG 202 810, Mar. 2010.

[45] M. Björklund, *YANG—A Data Modeling Language for the Network Configuration Protocol (NETCONF)*, document RFC 6020, 2015.

[46] *New ETSI Group on Improving Operator Experience Using Artificial Intelligence*, Accessed: Oct. 24, 2017. [Online]. Available: http://www.etsi.org/

[47] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[48] F. Kaup, F. Michelinakis, N. Bui, J. Widmer, K. Wac, and D. Hausheer, "Assessing the implications of cellular network performance on mobile content access," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 2, pp. 168–180, Jun. 2016.

[49] B. Sousa *et al.*, "Enabling a mobility prediction-aware follow-me cloud model," in *Proc. IEEE LCN*, Nov. 2016, pp. 486–494.

[50] Z. Zhao, M. Karimzadeh, and T. Braun. (2017). *Next Place Prediction With Hybrid Features Using Ensemble Learning*. [Online]. Available: http://boris.unibe.ch/id/eprint/98674

[51] J. Kornycky, O. Abdul-Hameed, A. Kondoz, and B. C. Barber, "Radio frequency traffic classification over WLAN," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 56–68, Feb. 2017.

[52] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Proc. ACM NIPS*, 1994, pp. 671–678.

[53] T. Wang, M. Srivatsa, D. Agrawal, and L. Liu, "Spatio-temporal patterns in network events," in *Proc. ACM CONEXT*, 2010, pp. 1–12.

[54] H. Yan, L. Breslau, Z. Ge, D. Massey, D. Pei, and J. Yates, "G-RCA: A generic root cause analysis platform for service quality management in large IP networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1734–1747, Dec. 2012.

[55] C. L. Gutterman, W. Mo, S. Zhu, Y. Li, D. C. Kilper, and G. Zussman, "Neural network based wavelength assignment in optical switching," in *Proc. ACM Workshop Big-DAMA*, 2017, pp. 1–6.

[56] A.-F. Antonescu and T. Braun, "Service level agreements-driven management of distributed applications in cloud computing environments," in *Proc. IFIP/IEEE IM*, May 2015, pp. 1122–1128.

[57] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2503–2516, Dec. 2016.

[58] *Business Support System, Definition*, Accessed: Oct. 24, 2017. [Online]. Available: https://www.techopedia.com/definition/26873/business-support-system/
[59] N. Ragouzis *et al.*, "Security assertion markup language (SAML) V2.0 technical overview," OASIS Security Services (SAML) TC, Tech. Rep., Oct. 2007.
[60] T. Wason, S. Cantor, J. Hodges, J. Kemp, and P. Thompson, *Liberty ID-FF Architecture Overview*. Piscataway, NJ, USA: Liberty Alliance Project, 2005.
[61] *Liberty ID-FF Architecture Overview*, Liberty Alliance Project, 2005.
[62] C. Kaler, M. Mclntosh, M. Goodner, and A. Nadalin, *Web Services Federation Language (WS-Federation) Version 1.2*, OASIS Standard, May 2006.
[63] OpenID. (2007). *OpenID Authentication 2.0—Final*. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html
[64] M. Needleman, "The Shibboleth authentication/authorization system," *Serials Rev.*, vol. 30, no. 3, pp. 252–253, 2004.
[65] D. Chadwick, G. Zhao, S. Otenko, R. Laborde, L. Su, and T. A. Nguyen, "PERMIS: A modular authorization infrastructure," *Concurrency Comput., Pract. Exper.*, vol. 20, no. 11, pp. 1341–1357, Aug. 2008.
[66] C. Bailey, D. W. Chadwick, and R. de Lemos, "Self-adaptive authorization framework for policy based RBAC/ABAC models," in *Proc. IEEE 9th Int. Conf. Dependable, Auto. Secure Comput.*, Dec. 2011, pp. 37–44.
[67] J. Jeong, W. Yu, D. Shin, D. Shin, K. Moon, and J. Lee, "Integration of single sign-on and role-based access control profiles for grid computing," in *Proc. Asia–Pacific Web Conf.*, 2006, pp. 880–885.
[68] F. Gao and J. Tan, "Shibboleth and community authorization services: Enabling role-based grid access," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, 2011, pp. 131–140.
[69] J. Vanerio and P. Casas, "Ensemble-learning approaches for network security and anomaly detection," in *Proc. ACM Workshop Big-DAMA*, 2017, pp. 1–6.
[70] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Proc. Int. Workshop Recent Adv. Intrusion Detect.*, 2011, pp. 161–180.
[71] K. Giotis, G. Androulidakis, and V. Maglaris, "Leveraging SDN for efficient anomaly detection and mitigation on legacy networks," in *Proc. 3rd Eur. Workshop Softw. Defined Netw.*, Sep. 2014, pp. 85–90.
[72] C. J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Depend. Sec. Comput.*, vol. 10, no. 4, pp. 198–211, Jul. 2013.
[73] S. Shin and G. Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," in *Proc. IEEE ICNP*, Oct./Nov. 2012, pp. 1–6.
[74] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Comput. Netw.*, vol. 62, pp. 122–136, Apr. 2014.
[75] C. Jeong, T. Ha, J. Narantuya, H. Lim, and J. Kim, "Scalable network intrusion detection on virtual SDN environment," in *Proc. IEEE 3rd Int. Conf. Cloud Netw.*, Oct. 2014, pp. 264–265.
[76] B. Mantur, A. Desai, and K. S. Nagegowda, "Centralized control signature-based firewall and statistical-based network intrusion detection system (NIDS) in software defined networks (SDN)," in *Emerging Research in Computing, Information, Communication and Applications*. New Delhi, India: Springer, 2015, pp. 497–506.
[77] L. Zhang, G. Shou, Y. Hu, and Z. Guo, "Deployment of intrusion prevention system based on software defined networking," in *Proc. IEEE Int. Conf. Commun. Technol.*, Nov. 2013, pp. 26–31.
[78] S. Shirali-Shahreza and Y. Ganjali, "Flexam: Flexible sampling extension for monitoring and security applications in openflow," in *Proc. ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, 2013, pp. 167–168.
[79] H. Hu, W. Han, G.-J. Ahn, and Z. Zhao, "FlowGuard: Building robust firewalls for software-defined networks," in *Proc. ACM 3rd Workshop Hot Topics Softw. Defined Netw.*, 2014, pp. 97–102.
[80] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.
[81] *5G Depends on SDN & NFV*. Accessed: 2017. [Online]. Available: https://www.sdxcentral.com/articles/news/5g-depends-on-sdn-nfv/2016/01
[82] *Selfnet Publications*. Accessed: 2017. [Online]. Available: https://selfnet-5g.eu/publications
[83] M. Zago, V. M. R. Sánchez, and M. G. Pérez, "Tackling cyber threats with automatic decisions and reactions based on machine-learning techniques," in *Proc. EuCNC*, 2017, pp. 1–10.
[84] M. G. Pérez *et al.*, "Dynamic reconfiguration in 5G mobile networks to proactively detect and mitigate botnets," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 28–36, Sep./Oct. 2017.
[85] M. G. Perez *et al.*, "Keeping an eye on botnets in 5G networks: Detection and mitigation by NFV and SDN apps," in *Proc. EuCNC*, 2017, pp. 1–3.
[86] W. Peng, F. Li, and X. Zou, *Moving Target Defense for Cloud Infrastructures: Lessons from Botnets*. New York, NY, USA: Springer, 2014, pp. 35–64.
[87] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, Eds., *Moving Target Defense—Creating Asymmetric Uncertainty for Cyber Threats* (Advances in Information Security), vol. 54. New York, NY, USA: Springer-Verlag, 2011.
[88] *Moving Target Defense: Common Practices*. Accessed: Mar. 24, 2017. [Online]. Available: http://blog.morphisec.com/moving-target-defense-common-practices
[89] *Moving Target Defense vs. Moving Target Attacks: The Two Faces of Deception*. Accessed: Mar. 24, 2017. [Online]. Available: http://www.networkworld.com/article/3018881/tech-primers/moving-target-defense-vs-moving-target-attacks-the-two-faces-of-deception.html
[90] M. Dunlop, S. Groat, W. Urbanski, R. C. Marchany, and J. Tront, "MT6D: A moving target IPv6 defense," in *Proc. IEEE Military Commun. Conf. (MILCOM)*, Baltimore, MD, USA, Nov. 2011, pp. 1321–1326.
[91] *Address Space Layout Randomization*. Accessed: Mar. 28, 2017. [Online]. Available: https://en.wikipedia.org/wiki/Address_space_layout_randomization
[92] *Address space layout randomization (ASLR)*. Accessed: Mar. 28, 2017. [Online]. Available: http://searchsecurity.techtarget.com/definition/address-space-layout-randomization-ASLR
[93] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 2001.
[94] P. Cong-Vinh, "Formal and practical aspects of autonomic computing and networking: Specification, development, and verification," *Information Science Reference*. Hershey, PA, USA: Imprint of IGI Publishing, 2011.
[95] E. Vassev and M. Hinchey, "The challenge of developing autonomic systems," *Computer*, vol. 43, no. 12, pp. 93–96, Dec. 2010.
[96] E. Vassev, M. Hinchey, and A. J. Quigley, "Model checking for autonomic systems specified with ASSL," in *Proc. 1st NASA Formal Methods Symp. (NFM)*, 2009, pp. 16–25.
[97] R. Laborde, B. Nasser, F. Grasset, F. Barrère, and A. Benzekri, "A formal approach for the evaluation of network security mechanisms based on RBAC policies," *Electron. Notes Theor. Comput. Sci.*, vol. 121, pp. 117–142, Feb. 2005.
[98] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Trans. Inf. Syst. Secur.*, vol. 4, no. 3, pp. 224–274, Aug. 2001.
[99] H. Wang, H. Lv, and G. Feng, "A self-reflection model for autonomic computing systems based on p-calculus," in *Proc. 3rd Int. Conf. Netw. Syst. Secur. (NSS)*,2009, pp. 310–315.
[100] O. Andrei and H. Kirchner, *A Higher-Order Graph Calculus for Autonomic Computing*. Berlin, Germany: Springer, 2009.

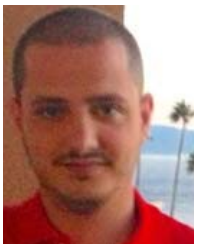
Zhongliang Zhao received the Ph.D. degree from the University of Bern in 2014. He was a Senior Researcher with the University of Bern. He has been active as multiple Work Package leaders in the EU FP7 project Mobile Cloud Networking, a Co-PI of the Sino-Swiss Science and Technology Cooperation project M3WSN. He is currently the Technical Coordinator of the Swiss National Science Foundation project SwissSenseSynergy.

Eryk Schiller received the M.Sc. diploma degree in electronics and telecommunications from University of Science and Technology, the M.Sc. diploma degree in theoretical physics from Jagiellonian University, Cracow, Poland, in 2006 and 2007, respectively, the Ph.D. degree in computer science from the University of Grenoble, France, in 2010. He was a Post-Doctoral Scholar with the University of Neuchatel, Switzerland. Since 2014, he has been Senior Researcher with the University of Bern.

Eirini Kalogeiton received the Dipl.Eng. and M.Sc. degrees in electrical and computer engineering from the Democritus University of Thrace, Xanthi, Greece, in 2014 and 2016, respectively. She is currently pursuing the Ph.D. degree with the University of Bern. Her main research interests include vehicular ad-hoc networks, named data networking networks, and software defined networking.

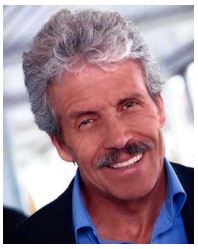
Torsten Braun received the Ph.D. degree from the University of Karlsruhe, Germany, in 1993. Since 1998, he has been a Full Professor in computer science with the University of Bern. He has been the Vice President of the SWITCH Foundation since 2011. He received the Best Paper Award from the IEEE LCN 2001, WWIC 2007, EE-LSDS 2013, IFIP WMNC 2014, ARMSCC 2014 Workshop, and the GI-KuVS Communications Software Award in 2009.

Burkhard Stiller received the Diplom-Informatiker (M.Sc.) degree in computer science and the Dr. rer.-nat. (Ph.D.) degree from the University of Karlsruhe, Germany. He held previous positions with the Computer Laboratory, University of Cambridge, U.K., the Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, and the University of Federal Armed Forces, Munich, Germany. He has been a Full Professor with the Communication Systems Group, Department of Informatics, University of Zurich, since 2004.



Mevlut Turker Garip received the B.Sc. degree in computer science from Bilkent University, Ankara, Turkey. He is currently pursuing the Ph.D. degree in computer science with the University of California Los Angeles, under the supervision of Prof. P. Reiher. He is a member of the Laboratory for Advanced Systems Research. His research field is computer and network security. His current research focuses on vehicular ad hoc network security.

Joshua Joy is currently pursuing the Ph.D. degree in computer science with the University of California Los Angeles (UCLA), under the supervision of Prof. Gerla. He recently led the deployment of CrowdZen at UCLA. CrowdZen privately computes the real-time activity levels across the UCLA campus and is used daily by thousands of students. His research interests include the Internet of Vehicles and scalable privacy within vehicular clouds.



Mario Gerla (M'75–SM'01–F'03) received the Ph.D. degree from the University of California Los Angeles (UCLA). He was part of the team that developed the ARPANET models and protocols that formed the basis of the modern INTERNET as a graduate student at UCLA in the early 70's. After a period in industry, he joined the UCLA Faculty in 1976. He is currently a Professor of computer science with UCLA, where he holds the Jon Postel Chair in networking. He serves on the IEEE TON Scientific Advisory Board and was recognized with the ACM Sigmobile Outstanding Contribution Award in 2015.

Nabeel Akhtar received the B.Sc. degree in computer science from the Lahore University of Management Sciences, Lahore, Pakistan, in 2011, and the M.Sc. degree in computer science and Engineering from Koc University, Istanbul, Turkey, in 2013. He is currently pursuing the Ph.D. degree with Boston University, under the supervision of Prof. Matta. His research interests include network virtualization, future internet architectures, software-defined networking, and edge computing.

Ibrahim Matta received the Ph.D. degree in computer science from the University of Maryland at College Park in 1995. He is currently a Professor of computer science with Boston University. His research involves network protocols, architectures, and performance evaluation. He received the NSF CAREER Award in 1997 and a patent in 2011. He also received the Best-Paper Award on his work on wireless ad hoc and sensor networks in 2008 and 2010, respectively.