INVITED PAPER

Self-Organization and Resilience for Networked Systems: Design Principles and Open Research Issues

This article reviews and proposes research on the twin fields of self-organization and resilience for networked systems.

By Simon Dobson, *Senior Member IEEE*, David Hutchison, Andreas Mauthe, Alberto Schaeffer-Filho, *Member IEEE*, Paul Smith, and James P. G. Sterbenz

ABSTRACT | Networked systems form the backbone of modern society, underpinning critical infrastructures such as electricity, water, transport and commerce, and other essential services (e.g., information, entertainment, and social networks). It is almost inconceivable to contemplate a future without even more dependence on them. Indeed, any unavailability of such critical systems is—even for short periods—a rather bleak prospect. However, due to their increasing size and complexity, they also require some means of autonomic formation and self-organization. This paper identifies the design principles and open research issues in the twin fields of self-organization and resilience for networked systems. In combination, they offer the prospect of combating threats and allowing essential services that run on networked systems to continue operating satisfactorily. This will be achieved, on the one hand, through the (self-)adaptation of networked systems and, on the other hand, through structural and operational resilience techniques to ensure that they can detect, defend against, and ultimately withstand challenges.

KEYWORDS | Autonomic communications; network resilience; programmable networks; self-organization; system resilience

I. INTRODUCTION

Today's world has become strongly dependent on networked computer systems, more through evolution and opportunism than through foresight and planning. The rapid adoption of Internet technologies has been nothing short of astounding [1]—a process that was accelerated by the advent of the World Wide Web, building on the ubiquitous spread of the Internet's network infrastructure. The resulting networked systems are now so common that many, especially in the younger generation, forget (or do not know) what life was like before their advent. Prominent examples of networked systems include utility networks (e.g., Smart Grid), industrial control systems (ICSs), the emerging Internet of Things (IoT), Industry 4.0, Cloud Computing, 5G, and Smart Cities. In these systems, the architecture is essentially that of a set of distributed services operating over a communications network, where the services are characterized by the nature of the application or enterprise (whether this is IoT, ICS, 5G, etc.).

As the ubiquity of networked systems has grown, so have their speed and complexity, and their necessity for many social processes—to the point that human management is inadequate to the task of keeping the systems working. This has driven the imperative toward self-organizing systems—and also self-managing, self-protecting, and other so-called "self-*" properties [2]—that allow the

Manuscript received July 30, 2018; revised January 11, 2019; accepted January 11, 2019. Date of publication February 20, 2019; date of current version March 25, 2019. *(Corresponding author: Andreas Mauthe.)*

S. Dobson is with the School of Computer Science, University of St Andrews, St Andrews KY169AJ, U.K. (e-mail: simon.dobson@andrews.ac.uk).

D. Hutchison is with the School of Computing and Communications, Lancaster University, Lancaster LA14YW, U.K. (e-mail: d.hutchison@lancaster.ac.uk).

A. Mauthe is with the Institut für Wirtschafts- und Verwaltungsinformatik, Universität Koblenz, 55118 Koblenz, Germany, and also with the Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, 64289 Darmstadt, Germany (e-mail: mauthe@uni-koblenz.de; andreas.mauthe@maki.tu-darmstadt.de).

A. Schaeffer-Filho is with the Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre 90040-060, Brazil (e-mail: alberto@inf.ufrgs.br).

P. Smith is with the Center for Digital Safety and Security, AIT Austrian Institute of Technology, 2444 Vienna, Austria (e-mail: paul.smith@ait.ac.at).

J. P. G. Sterbenz is with the Department of Electrical Engineering and Computer Science, The University of Kansas, Lawrence, KS 66045 USA (e-mail: jpgs@ittc.ku.edu).

Digital Object Identifier 10.1109/JPROC.2019.2894512

network (in the broadest sense) to adapt its own organization and behavior in pursuit of service-level goals. Self-organization can operate in pursuit of many different goals which may themselves be stated at a number of levels. One may seek to improve (or maintain) performance in the face of changing network conditions, or to integrate new devices or access points, or to include new service variants. However, from a user's perspective, these technical issues can be subsumed under a goal of resilience: the system continues to work according to the user's expectations regardless of changes that may themselves be hidden anyway [3], [4].

It is these twin entwined concepts—self-organization in pursuit of resilience—that are our topics in this paper. We define self-organization as the techniques a system may use to change its detailed structure and behavior in response to external stimuli (extrinsic challenges) or changes in requirements (intrinsic challenges), in order to maintain service levels which may themselves be modified as part of the adaptive process. The ability to change service levels is a crucial part of this: some challenges are simply insurmountable and lead inevitably to user-visible degradation. We define resilience as the ability of the system to avoid this extremity and continue to provide acceptable service. The goal of this paper is to identify the design principles and open research issues in the combined fields of self-organization and resilience for networked systems.

The challenges to self-organization and resilience are enormous. Critical services can be subject to natural disasters, third-party failures such as power outages, configuration, and other failures. The rise of cyberattacks adds a directed dimension to these challenges, and the range of recent attacks (including Stuxnet [5], the Mirai botnet [6], and WannaCry [7]) has led to a burgeoning cybersecurity industry that supports a huge scientific and engineering effort to prevent attacks, develop mechanisms for ameliorating their effects, and provide forensic support for later investigation [8]. The critical nature of many networked systems and their increasingly intimate effects on the livelihoods (and indeed lives) of an increasing number of people is leading users to mandate specific levels of resilience for certain applications [9], [10].

The remainder of this paper is organized as follows. Section II presents the necessary background on the independent research areas of self-organization and systems resilience. Section III discusses the interplay between self-organization and resilience in the design of network systems architectures. Section IV outlines a number of open research challenges covering technical aspects such as the role of intent-based networking (IBN) and network function virtualization, and wider considerations such as those relating to the human element and the role of people in guaranteeing systems resilience. Finally, Section V presents our concluding remarks.

II. BACKGROUND

Resilient self-organized networked systems research draws on several domains. We organize our brief background review around two themes: the modeling and provision of self-organization within networks and the additional technology used to promote resilience.

A. Self-Organization

Self-organizing systems have long been a subject of research interest in computer science. The earliest example is possibly due to Dijkstra [11], who studied self-stabilizing systems that returned to a predictable state after a perturbation. The name of this paper—"Self-stabilizing systems in spite of distributed control" [our emphasis]—foreshadowed the difficulties facing anyone attempting to construct such systems.

The core challenge of self-stabilization, and of self-organization in general, is that it is a global property: no single component can determine whether or not the system as a whole is in an acceptable state, especially in the presence of component or communication failure. Similar issues occur throughout science: one example is the way in which murmurations of starlings form and evolve as the integral of a large number of individually local decision processes, with the structure (flock) being stable even as components (birds) move or dropout. It has proven possible [12] to develop surprisingly simple algorithmic descriptions of these processes, and there is now a significant class of biologically inspired approaches to self-organizing systems.

Self-organization is increasingly important as network complexity increases. The clearest examples come from domains in which there are frequent, spontaneous changes in device population, topology, services, and so on. Mobile *ad hoc* networks (MANETs) were once limited to tactical military systems but now are essential in enabling IoT systems and wireless sensor networks. They are also found in home mesh networks and vehicular networks (VANETs). We repeatedly see that, in many cases, niche techniques have mainstreamed, leading to a more complex but more independent and self-organizing overall architecture. The key observation is that management is treated as a local (or at least nonglobal) task that is the responsibility of the nodes themselves, rather than being imposed from outside.

An important phenomenon for our current purposes concerns the behavior of networks under attack. These are often studied within the framework of percolation theory: given a network and an attack that remove some proportion of nodes and/or edges, is there still a "giant component" that keeps a large fraction of nodes connected? The simplest of such attacks removes nodes or edges at random, and might more accurately be described as modeling typical failures; more structured attacks target the nodes with particular properties, such as the high-degree hubs or the low-degree bridges between otherwise sparsely connected components. Internet core routers [13] exhibit a powerlaw topology, giving the network a degree of "natural" or topological resilience to failure which nonetheless does not necessarily extend to resilience against informed and targeted attacks.

Topological techniques can be regarded as giving "spatial" resilience, in the sense that it is available statically whenever a challenge arises. This is perhaps the most stable form of resilience, since it depends only on the system's overall features. It is also possible to apply more dynamic techniques, either by adapting the network (perhaps activating additional links) or by adapting its behavior (perhaps providing new service instances or changed protocols or security stances): a more "temporal" form of resilience that can use less resources under normal circumstances and provide a more flexible array of responses [14].

Within the systems community, the most influential intellectual currents have undoubtedly been provided by research into autonomic computing defined by Kephart and Chess [15] as "computing systems that can manage themselves given high-level objectives from administrators." Autonomic computing research has explored systems that are self-managing, self-configuring, self-optimizing, and possessing a range of other self-* properties. The agenda has been structured around two parallel strands, one aimed at creating the proper *ab initio* architectures for self-* behaviors, and another aiming at adding self-* management behaviors to collections of existing services. Alongside autonomic computing has been a parallel effort in autonomic communications [2], [16] looking explicitly at adding self-* properties to networks. For both computing and communications, much of the work has been centered around a closed-loop control architecture—variously referred to as Monitor–Analyze–Plan–Execute over shared Knowledge (MAPE-K) or Collect–Analyze–Decide–Act—that generalize standard control-theoretic approaches by using various other mathematical formalisms, often adding flexibility at the expense of decidability.

In autonomic networks the underlying idea is that the network structures can form in an autonomous fashion by enabling network nodes to parameterize and operate independently using sensing and environmental awareness, and adaptation capabilities within the nodes and of protocols. Essentially, autonomous networks and their components should require little or no direct intervention during set-up as well as runtime. They learn and adapt to changes in the environment while providing a stable, reliable, and secure communication infrastructure.

The autonomic approach arose in response to the concerns of systems developers as well as network operators, and it is explicitly targeted at adding self-* properties to existing systems rather than forcing the development of new systems purely to address self-* questions—legacy software is, after all, just software that has worked well for a long time. Although this may limit the functionality that can be developed, the ability to add management functions (for example, by mining server logs to trigger reconfigurations) enormously reduces the startup cost of self-organization.

B. Resilience

Resilience should now be considered a vital property of systems and networks. In [17], resilience is defined as a concept associated with telecommunications systems and supporting resources and defines their ability to resist the loss of capacity due to failures or foreseen overload. The goal is to optimize the availability and quality of service of a system and enable it to return to a previous normal condition after a challenge subsided. The emphasis in [18] is on fault management and recovery methods, and how Quality of Resilience can be achieved through resilience differentiated services. We define resilience as the ability of a system or network to maintain acceptable levels of operation in the face of challenges, including malicious attacks, operational overload, misconfiguration, and equipment failures. Hence, resilience management encompasses the traditional fault, configuration, accounting, performance, and security functionalities [19] and comprises structural as well as wider context related considerations [18].

Concerns about the dependability and resilience of computer systems date back to the earliest days of computing. Within the networking context they, for instance, motivated the early Advanced Research Projects Agency Network (ARPANET) design, resulting in the decision to use connectionless paths in order to recover more easily from a failed router [20]. However, it was recognized that basic fault tolerance is insufficient in the case of correlated failures (e.g., due to an attack), and thus, network survivability became an important discipline for modern networks [21] (along with architectural strategies to achieve survivability [21], [22]). Although fault tolerance only requires redundancy in components and paths, survivability requires diversity [4], in order for the redundant part not to share the same fate as the failed component. Other factors are active management and protection elements that allow detecting the onset of challenges and combat them through appropriate defense and mitigation action. Challenge taxonomies [23] and also a resilience ontology [17] are required to capture threats and challenges in appropriate threat models on which resilient system and network design can be based.

A more formalized and comprehensive view on network resilience is presented in [4]. Cholda *et al.* [18] present a detailed survey on resilience differentiation on the Internet that also provides a detailed discussion on resilience assessment frameworks. In essence, these discussions capture the relationship between resilience related concepts and disciplines, i.e., challenge tolerance (including fault tolerance, survivability, disruption tolerance, and traffic tolerance), trustworthiness (including dependability, performability, and security), robustness, and complexity. Furthermore, a set of principles grouped as prerequisites, enablers, tradeoffs, and behaviors are defined (Fig. 1).

Among the key resilience enablers are redundancy, diversity, and connectivity and association [24]. Redundancy refers to adding additional resources to provide

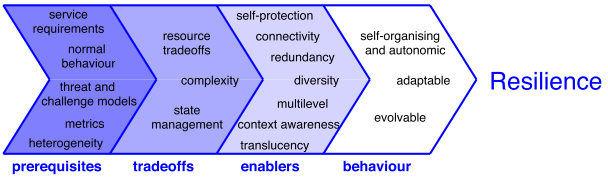


Fig. 1. ***Relationship between prerequisites, enablers, tradeoffs, and behaviors of resilience.***

fault tolerance. This principle can be employed across all system levels, including hardware redundancy (i.e., additional hardware which is added to system or system components to improve availability even in the case of failure), path redundancy (i.e., the availability of multiple alternative network paths between source and destination), and application redundancy (i.e., multiple application instances that can carry out the same task). Diversity (in addition to redundancy) provides survivability in cases where redundant components share the same fate. This includes geographically diverse paths to survive large-scale disasters, and the avoidance of "monocultures" in hardware and software, e.g., to improve resistance to zero-day attacks. Connectivity and association refer to disruption tolerance in challenged communication environments due to intermittent and episodic connectivity, e.g., for wireless links, mobility, unpredictable delay, and energy-constrained nodes. A key aspect is to be able to communicate even when stable end-to-end paths are not available.

Resilience in networks and systems has different viewpoints. In [18], structural and guaranteed resilience differentiation are distinguished. The former is concerned with structural arrangements (specifically related to the recovery of different connections) whereas the latter provides guarantees on the level of resilience. We distinguish between Structural Resilience, which expresses the resilience of the network and system infrastructures (i.e., the structural arrangements) and the assessment of the resilience level they offer (e.g., [24]); and Operational Resilience that specifies the level of active resilience management capabilities within a system or network that allow to actively defend, detect, and mitigate against threats.

An early application of adaptation for structural resilience in the context of optical network restoration is a self-healing network [25], in which a distributed algorithm restores a cut fiber in an optical mesh network. Dynamic routing as is typical on the Internet adapts to link and router failures; the ability to provide and exploit geographic diversity across redundant paths enables resilience to correlated failures from large-scale disasters [26].

Well-defined resilience targets can be used to provide more clear and tangible expressions of the desired resilience status [3]. The goal, on the one hand, is to establish if and to what extent structural resilience targets are met, i.e., if the levels of redundancy and diversity throughout the system are sufficient, and also if connectivity can be upheld. On the other hand, in the case of Operational Resilience, a challenge analysis in conjunction with a resilience estimator helps to determine if specified resilience targets are being met. If this is not the case appropriate resilience mechanisms have to be invoked to counter or adapt to challenges in order to maintain a high level of resilience.

III. SELF-ORGANIZATION AND RESILIENCE IN NETWORKED SYSTEMS ARCHITECTURES

In this section, we take a closer look at the need for resilience and self-organization and how these can be realized in networked systems architectures. We first look at the requirements of key application use cases. Critical systems form a major application domain in this context. Then, we present resilience principles and building blocks, and how self-organization and functional adaptation are leading us toward networked systems resilience.

A. Critical Systems and Requirements

Many critical infrastructures are undergoing a process of digitalization, for reasons such as improved efficiency, resilience, and the support of new services. This is, for instance, the case for energy systems that are being transformed into Smart Grids to enable the integration of renewable energy sources and stimulate the development of green energy services [27]. The digitalization depends heavily on communication networks, e.g., to enable telecontrol of field devices. A wide range of networking technologies is being considered for application in the Smart Grid, including LoRa to connect low-powered distributed sensors and programmable networks to support application-level requirements in the network.

This increased connectivity introduces new interdependencies between networks and systems, and the potential for cascading failures [28]. Previous research has sought to understand the properties of these interdependent networks, for example, to determine the risks associated with the propagation of malicious software across networks, using percolation theory [29]. With an understanding of the risks, e.g., knowledge of Shared Risk Link Groups [30] and critical systems that facilitate cascading failures, infrastructures can be designed to be structurally more resilient to failures and cyberattacks.

Unfortunately, digitalization introduces new cybersecurity vulnerabilities that can be exploited by increasingly sophisticated threat actors [31]. Lately, there have been high-profile incidents that have shown the potential consequences of introducing digital components connected via wide-area communication networks to critical infrastructures. In the energy sector, perhaps the most notable recent example of a cybersecurity incident occurred in Ukraine in December 2015, resulting in a power blackout [32].

IoT promises efficiencies and economic benefits by connecting masses of physical devices, including sensors and actuators, to wide-area networks, in particular to the

public Internet. One of the major applications of future 5G networks is providing highly reliable connectivity to the IoT. In many cases, the cybersecurity maturity of these newly connected devices is low. The consequences of this insecurity can be seen in the case of the Mirai IoT botnet, which caused the largest recorded Distributed Denial of Service (DDoS) attack on the Internet in 2016 [6].

To address these challenges, there are a number of defensive measures that can be put in place. Nevertheless, given the increasing sophistication of threat actors and the level of exposure to critical systems, a responsive capability is essential.

It is important that an incident, such as a cyberattack, is detected as early as possible. In addition to detecting a challenge, it is important to be able to determine the root cause of an incident so that effective mitigation actions can be taken. This requires situational awareness (SA) [33], which can be obtained through the use of various monitoring and detection systems. Moreover, contextual information, regarding a system's environment, can be used to support SA. For example, weather data can be used to provide insights into the case of poor performance of wireless communication networks.

Having detected an incident, it has to be mitigated. Currently, this is usually a human-resource-intensive task, requiring specialist operator support. However, as critical infrastructures increase in scale and complexity, and rely more on interconnected digital and virtualized systems, automated approaches will become necessary. In this context, the aim of mitigation is twofold: 1) to contain the effects of an incident, such as a cyber-attack or fault and 2) to eradicate the root cause of the problem. An example containment action is a scheme to push-back network traffic that is associated with a DDoS attack.

In contrast to containment, eradication is concerned with determining the root cause of an incident and removing it. For cybersecurity, this typically entails identifying affected systems and removing malicious software. For other root causes, this may involve changing equipment or system designs. For some challenges, eradication may for various reasons not be possible. Examples include resource (financial) limitations, operational availability requirements (eradication can involve taking systems offline), and inherent design limitations (e.g., one cannot completely eradicate DoS attacks on the Internet). In this case, the aim is to recover the system to a normal operational state and to disengage any mitigation actions once a challenge has abated.

B. Resilience and Its Building Blocks

1) Resilience Strategy: Resilience is a familiar notion in many walks of life—it is broadly considered as the ability to 'bounce back' in or after difficult situations. Any strategy for resilience will have a familiar set of steps that are basically active (anticipation and preparedness) and reactive (detection and mitigation). The resilience strategy [4] described in this section provides guidance on the design and operation of a resilient network (or networked system), applying principles such as redundancy, diversity, and connectivity and association. It captures the main components of the resilience process within an online and an off-line control loop expressed through D^2R^2 + DR as shown in Fig. 2.

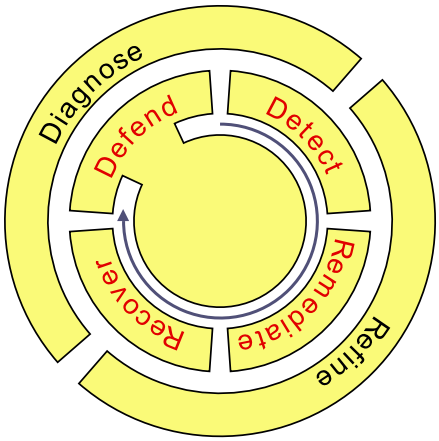


Fig. 2. *D^2R^2 + DR conceptual resilience model.*

The first part, i.e., D^2R^2 specifies a real-time control loop defend, detect, remediate, recover. The center of the circle denotes structural defenses such as geodiverse redundant paths. The first part of the control loop consists of active operational defenses, for example, filtering known attack signatures. Should a challenge penetrate the system detection has to identify and flag the threat. Remediation mechanisms are then applied to ensure the best possible service during an ongoing adverse event (such as a traffic attack) or after an event that has destroyed parts of the infrastructure (such as a large-scale disaster). For example, traffic may be rerouted around failed infrastructure areas. Finally, recovery returns the network to normal operations once the challenge has ended and failed infrastructure elements have been repaired. This control loop operates in across all components and in all network protocols.

The outer DR diagnose and refinement loop represents a nonreal-time control loop through which the resilience related infrastructure aspects, as well as the operational strategies, are analyzed and improved, reflecting the longer term experiences and developments within the environment.

The D^2R^2 + DR conceptual resilience model provides design guidance, but is not an architectural blueprint. Within this model, different aspects as part of overall resilience architecture have been addressed. It has also been shown that it can address resilience in a cloud networked system [34].

The model considers the resilience properties as introduced in [4]. It is important to note that this resilience model is based on the concept of autonomic components that have large degrees of self-organization. These autonomic components are not only adaptable

to the environment but can also evolve. Collaboration between them largely happens through information exchange and the adoption of joint policies and rule sets.

This model can be applied to individual, autonomic resilience instances, but can also be extended to apply to the networked system level, where there is a set of interconnected autonomic managers within one administrative domain.

A process for building resilient computer networks and the set of steps involved has also been derived by other research groups and standards bodies. A more or less common understanding has emerged of the life cycle of resilience, including the D^2R^2 + DR model, and a National Institute of Standards and Technology (NIST) version that can be summarized as follows: identify; protect; detect; respond; and recover [35]. More recent research [3] has clarified the need for a number of sub-steps. For example, related to the D^2R^2 + DR model, risk assessment needs to be involved in defend, instrumentation of the system under inspection in detect, and the need to move toward an enhanced system state (taking account of the challenge and its adverse effects on the system) in recover. For the outer loop of the D^2R^2 + DR approach, the diagnose and refine phases will probably require human intervention [3], unless it can be made autonomic.

2) Interactions and Policies: Policy-based management (PBM) has been proposed as one of the means for instantiating appropriate resilience mechanisms and for defining their behavior within the D^2R^2 + DR control loop [36]. Policies can be defined as rules governing the choices in the behavior of a system [37]. Policies provide a useful abstraction to encode choices that must be made when a range of mechanisms can be used to realize a particular phase of D^2R^2 + DR, or when tradeoffs must be considered in the decision-making process (e.g., to deploy a more lightweight detection mechanism that has lower accuracy, or a more accurate but heavyweight one). Furthermore, as more information about an ongoing challenge is gathered, policies can modify the configuration of resilience mechanisms deployed in the network, e.g., by applying a more targeted remediation strategy.

Previous work [38] has assessed how PBM could be used to realize resilience strategies to combat different types of challenges and network anomalies. Fig. 3 shows the interaction between a number of mechanisms that cooperate through policies to contain a volume-based DDoS attack. Initially, it is assumed that a link monitor component (configured via policies) is able to tell simply whether the incoming traffic rate on a particular link is too high. In this case, a preventive rate limiting of the link is triggered. Furthermore, an anomaly detection component is configured, also via policies, to identify the destination IP address of the victim. As a result, policies start shaping only traffic destined to the victim's IP address—legitimate traffic not destined for the victim, which previously was blocked, is now allowed to go through. A final step, also configured

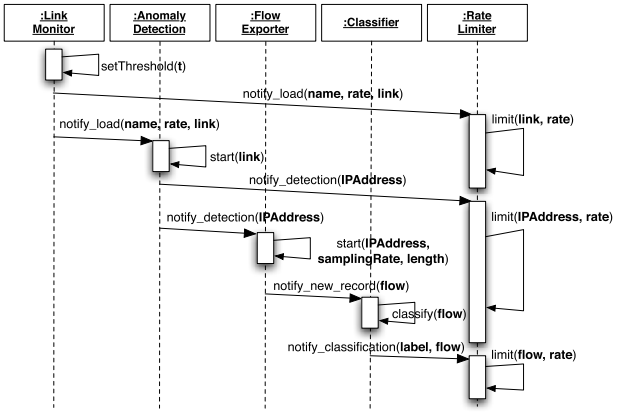


Fig. 3. ***Policy-based DDoS resilience strategy [39].***

via policies, activates a traffic classification component, which more precisely identifies the malicious flows; rate limiting is confined just to the attack flows, whereas legitimate traffic can continue. By having the interactions between the mechanisms implemented with policies, one can easily change the strategy by adding or removing the corresponding policies. Similar resilience strategies have been applied to other challenges, such as to the detection and mitigation of worm propagations [39].

We argue that policy-based frameworks can be seen as enablers of autonomic computing and self-management. The self-managed cell (SMC) framework [40] provides an infrastructure for engineering autonomous systems. An SMC is the building block component, which relies on the use of a policy-driven control-loop to determine the kind of management actions that should be performed in response to changes in the SMC context.

3) Autonomic, Self-Organized Adaptation: Another alternative is to return to Dijkstra's original position on self-stabilizing systems and develop algorithms that are inherently self-stabilizing or self-restoring in the presence of localized failures. A good example of this can be seen in the work of Zambonelli, Viroli, Beal, Pianini, and their collaborators on building computational fields and other distributed spatial structures [41], [42]. Neighborhood interactions can be used to create gradient structures to route data and perform "in-passing" computations. Such approaches allow building a range of systems that self-stabilize and reconfigure as the underlying network is challenged, for example, by node failure or mobility. The techniques can also provide a substrate for exploring location- or topology-dependent processes that are specified as global functions and then decomposed onto the spatial structure (cf. [43]).

Clearly, it is not possible to damage such structures arbitrarily and expect them to retain their functionality, at the very least, for instance, network partitions destroy convergence. However, accepting these caveats, the advantage of such "aggregate" programming approaches [44] is that they allow low-level mechanisms to be treated

as building blocks that can be reasoned over as to the correctness and robustness of the aggregate under stress.

4) Structural Resilience Through Self-Adaptation: Achieving system resilience through the structural design of its component parts is not only appealing but is a rather obvious idea. This has been around for a very long time in high-availability systems such as in aerospace where techniques of redundancy, hot standby components, and spatial diversity are widely adopted.

Triple modular redundancy (TMR), for example, was used from early days to improve computer reliability [45]. This method was also applied in space shuttles where the concept of diversity was realized using different software implementations for modular elements in order to avoid the same software bug affecting all operations. Literature in this area is abundant, for example, Shooman's work [46] on reliability of computer systems and networks: fault tolerance, analysis, and design.

Digital system design is mainly about structure in both theory and practice. There are various abstract representations of systems including finite state machines (FSMs), which have inputs, outputs, internal state, and a specific function that the machine is intended to carry out. This also encompassed the idea of a hierarchy of systems and subsystems, where an FSM can equally well represent an entire system, one of its subsystems, or even one of its smallest elements.

As described above, the Internet's original design provides some degree of resilience. However, the designed topology of a network (e.g., full mesh, star or some other structure) crucially impacts on the resilience of the system with respect to multiple or even single failures of nodes or links. If there is a critical point of failure (such as the central node in a star topology), the entire system may stop working if this component fails. It is possible to compare the resilience of different network topologies as done by Jabbar [47], who proposes a framework to quantify network resilience and survivability.

Modern practice suggests carrying out a risk assessment before choosing the appropriate structure to meet the desired resilience. Of course, there will typically be costs involved in such a decision; more redundancy implies more expense. However, such a "static" approach may not work well in practice, and it may be appropriate to consider some form of self-adaptation that would enable a system to reconfigure itself (respectively its structure) during operation, especially in the face of structural challenges. TMR is a specific example of a "dynamic" approach, i.e., if one (of the three) components goes wrong, its output will be different from the other two, which will outvote and remove it.

Internet routing and domain naming systems can also reconfigure themselves when a failure is observed in one of the elements. In this case, reconfiguration time may be long, and one of the design criteria will be to minimize this time.

5) Situational Awareness: In order to make adaptations to a running system, for example, in response to the need for additional resources to be deployed, or because of an identified challenge or threat, it may be appropriate to use situational information. This is particularly important in the context of Operational Resilience where, in the detection phase, a whole range of data and information has to be considered to ensure an appropriate response.

SA has long been used in military systems as an essential part of understanding the environment in which the system is operating. Another term for this sort of activity is context awareness. The need for this became particularly apparent during research on network resilience.[1] During the detect phase in the D^2R^2 + DR model, the focus is on network traffic in order to identify anomalies, which then leads into the remediation phase with the intention to bring the system back to normal (or at least toward normal). However, anomalous behavior can be caused by a large variety of challenges where the effects of several of these could look the same. Thus, the choice of remediation method (typically some form of traffic engineering) may end up being based on the symptoms of a challenge rather than knowing from where the challenge originated or indeed its nature. In order to ensure the appropriate form of remediation a root cause analysis is required. Unfortunately, this is something that typically takes time and is more suited to an offline analysis. Since one of the ultimate goals of the strategy is to support real-time operation and adaptation, the use of context information providing insights about the external conditions (i.e., external to the network itself) that have contributed to the challenge that has led to the anomaly is required. The context depends on the application or service that the system is underpinning. Thus, a source of useful information can be as varied and different as weather reports, environmental conditions as measured by sensors, any relevant newsfeeds (whether local, regional or broader), or indeed information from relevant social networks.

SA provides a similar sort of information base, although it was not originally intended for use in the specific case of resilience, but rather to provide and maintain an outlook for whatever use would be appropriate. One area of possible interest would be building and maintaining an SA city database that can be queried by various applications—including ones that try to keep an eye on resilience, e.g., as proposed in standards for cities).[2] SA can also be used to enhance the D^2R^2 + DR model with a predictive element based on a history of previous events. Such prediction in resilient systems operation might prove useful if expensive to provide, and it would require machine learning. This is one of the not yet fully explored elements of the DR outer loop of the resilience model.

The engineering of realistic resilience assurance engines in networks and networked systems has barely begun

[1]ResumeNet project: http://www.resumenet.eu
[2]ISO standards: https://www.iso.org/news/ref2305

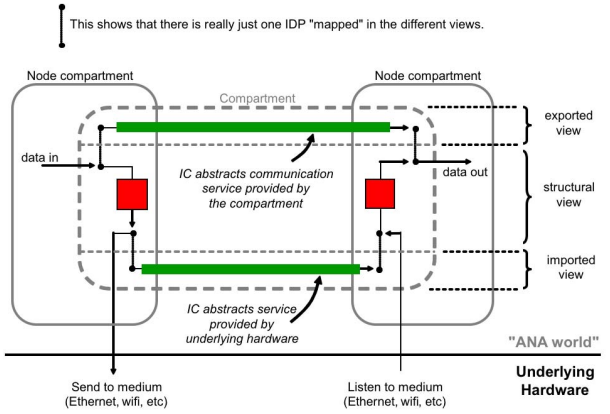


Fig. 4. ***Fundamental ANA abstractions [49].***

and requires use cases and experiments such as identified in [48].

C. Toward Networked System Resilience: Self-Organization and Functional Adaptation

1) ANA and Autonomic Networking: The autonomic networking architecture (ANA) project was a European funded project that researched the principles underpinning autonomic networking and proposed a generic architectural framework that enables autonomic communication at all levels [16]. Central to the ANA framework are abstractions that model network operations, rather than defining protocols or protocol functions [49]. These are namely compartments (i.e., "regions" of the network that are homogeneous in terms of their function, spatial and/or temporal extent), information channels—IC (i.e., defining available communication services, e.g., unicast, multicast, broadcast and anycast, and their properties, e.g., reliability, QoS, etc.), functional blocks—FB (i.e., protocol entities generating, consuming, processing and forwarding information) and information dispatch points—IDP (i.e., the interfaces through which FBs can be accessed and communicate with). Fig. 4 shows the fundamental ANA abstractions [49].

The concept of functional composition (FC) [50] as used in ANA is an essential element for a flexible network subsystem that enables autonomic behavior since it provides the "machinery" for adaptation. FC is not only used to provide different services and various service qualities but is also needed to customize and adapt the communication structure. Hence, FC provides the building blocks for the realization of self-organization, self-optimization, and self-healing properties within an autonomic network architecture. A framework that enables dynamic composition of functionality to leverage autonomic behavior has been developed and evaluated in ANA through a prototype userspace implementation [50]. The adaptation supported by FC can influence behavioral and structural adaptation. The former allows the immediate reaction to changes in the environment as well as evolutionary adaptation whereas the latter is more in support of long-term adaptation.

In order to provide network resilience within ANA, a monitoring architecture was developed providing monitoring and environmental awareness through a service used by all FB that need knowledge about the network state [51]. The monitoring service makes full use of the ANA abstractions, and the ANA monitoring architecture allows the dynamic placement of monitoring components as well as changing the monitoring functionality during runtime (e.g., to adapt to the monitoring requirements of new threats).

Using monitoring in conjunction with the ANA abstractions and FC allows for fully integrated Operational Resilience by building anomaly detection on top of the monitoring framework, and remediation and recovery functionality using FC within a Resilience Compartment. Structural Resilience can be implemented as distributed functionality that is invoked during network formation and adaptation, which may be required in order to meet defined resilience targets. Hence, ANA laid some of the groundwork for resilient networking through the use of autonomic network principles with self-management, self-organization, and self-adaptation as core concepts.

2) Programmable Networks: The limitations of inflexible rigid network protocols and operation were recognized in the late 1980s in the Internet and public-switched telephone network (PSTN). Very different architectures emerged, which, however, form the foundation of modern software-defined networking (SDN) and network function virtualization (NFV).

Active networks began with the premise of considering what could be done if Moore's-law enabled processing were exploited to provide additional functionality over the traditional simple IP forwarding [52], [53]. A separate direction occurred in the PSTN, motivated by telephony switch vendors and service providers wishing the ability to provision new services (initially such as teleconferencing and reverse billing) without modifying the switch hardware. This was integrated into the SS7 signaling network defined as intelligent network (IN) capability sets [54]. IN functionality provided control-plane programmability, including the ability to alter the behavior of the call state machine.

The reemergence of programmable networking in the 2010s, generically referred to as SDN, promoted the broad adoption of programmability of the control plane of computer networks. OpenFlow is the current widespread implementation of SDN, providing programmability of the control plane with a standard interface [55], and standards now developed by the Open Network Foundation. OpenFlow specifies a (logically) centralized controller that can be used to program network functionality by installing entries into switch flow tables that specify rules on the packet actions that can be taken. The logically

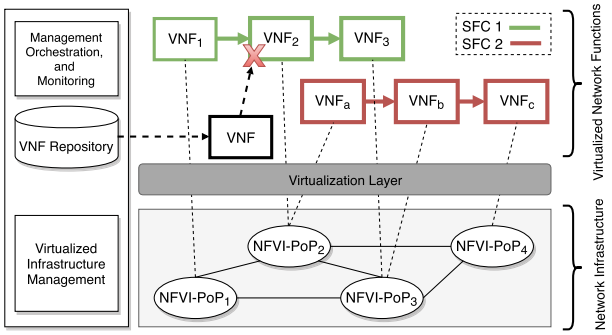


Fig. 5. *VNF service function chaining and systems resilience.*

centralized controller must be physically distributed to provide resilience and scalability for large networks. OpenFlow does not inherently support control plane resilience, though research is in progress [56].

More recently, protocol-oblivious forwarding [57] and P4 (Programing Protocol-Independent Packet Processors) [58] have been introduced to add rich data-plane programmability into modern SDN routers and networks. These efforts bring further opportunities to explore programmability in the construction of a more resilient data plane.

3) Network Function Virtualisation: NFV is another potentially fruitful direction for realizing resilient systems. The European Telecommunications Standards Institute has been developing a set of proposals for exploiting virtualization technologies to build telecommunications systems that have a range of improved properties, including flexibility but also resilience [59]. Future telecommunications systems will, in engineering terms, consist of some key physical network functions (PNF) that cannot or should not be virtualized, while everything else will be composed of virtual network functions (VNFs) that run on commodity hardware. These VNFs (together with the relevant PNFs) will be composed to form a network service and/or application. This is facilitated through an orchestrator using some sort of user intent statement in conjunction with appropriate policies to instruct how the VNFs (and PNFs) will be chained together [60]. This is work in progress and the two crucial aspects are: 1) the construction of the chain to be suitably resilient (resilience by design) and 2) to monitor (and control) the resulting system so that it can cope with the various challenges that will inevitably come its way (e.g., apply the D^2R^2 + DR model to the system).

Fig. 5 illustrates the composition of virtualized network functions into service function chains (SFCs). Individual VNFs are mapped to NFV infrastructure points of presence (NFVI-PoPs). Furthermore, two SFCs demonstrate the composition of VNFs to provide a specific service. The role of the NFV architecture for the support of system resilience is twofold, i.e., allowing 1) the migration and placement of virtual functions into alternative NFVI-PoPs in case of hardware failure and 2) to replace a failed VNF in an SFC (e.g., VNF_2 in Fig. 5) by another VNF dynamically downloaded from a VNF repository.

Interest in autonomic operation goes back quite some years, including notably the as-yet-unfulfilled aspiration of autonomic network (and services) management [61], [62]. Telecommunications operators wish to reduce the very large cost burden of operations support system/business support system, and have been interested in ways of achieving this for decades. NFV is a potentially important driving force in this context.

IV. OPEN CHALLENGES AND RESEARCH DIRECTIONS

Numerous research challenges still exist within the resilient networked systems space. On the one hand, they are related to technical aspects within the fields of networking, resilience, and self-organization. On the other hand, the systems aspect, the human factors, and the role of people within the overall system context are crucial and have long been neglected. In this section, we review these factors in order to form a more comprehensive view on the directions research in this field should take. The following sections discuss each research area in detail, while Table 1 highlights the key points and summarizes the impact of each area on systems resilience, either positively (+) or negatively (−).

A. Intent-Based Networking

Specifying complex resilience configurations may demand the composition of a range of detection and mitigation mechanisms. Even though policy frameworks can be used to specify management policies to control the operation of these resilience mechanisms, such low-level policies still have to be manually specified by a human administrator. Complex scenarios that for instance require the cooperation of a large number of resilience mechanisms make deriving concrete policies by hand intractable. Alternatively, it would be desirable to (semi-)automatically derive strategies of implementable policy configurations from high-level specifications and requirements with minimum human involvement. Techniques for refining high-level goals into concrete policy specifications have been investigated for several years [63]–[66]. These typically employ some form of reasoning or decomposition to recursively transform high-level goals into more concrete ones.

More recently, IBN has emerged as a promising approach that can help operators and users in the specification and translation of high-level goals. Intents can be used to define what (as opposed to how) the network should do. Intent-driven networking enables the network to be configured according to the operators' intentions, without requiring them to specify low-level technical policies. Both industry [67], [68] and academia [69], [70] have devoted efforts to IBN. Although an intent can be seen as an abstract, high-level policy used to operate a network, it also relies on the use of cognitive/autonomic functions

Table 1 Key Findings Observed About the Open Research Challenges on Systems Resilience

Research area	Key aspects and potential impact on theory and practice of systems resilience
Intent-based networking	(+) minimize effort in the specification of resilience requirements; (+) operator needs to specify *what* should be achieved as opposed to *how* it should be achieved; (−) need to ensure that the produced resilience configurations are consistent; (−) difficult to deal with conflicting requirements.
Network function virtualization	(+) virtualization technology allows for resilience reconfigurations that can be enacted much faster; (+) reduced capital expenditure (CAPEX) and operational expenditure (OPEX) since additional resilience functions can be executed on commodity servers, as opposed to dedicated hardware; (−) need to rely on some form of dynamic verification to avoid producing inconsistent/incompatible VNF chainings, or deployments in suboptimal locations.
Programmable self-organisation	(+) better abstraction and composition of mechanisms for self-* and resilience; (+) improved reasoning about system capabilities and behaviours; (−) over-abstraction may hide important details and prevent some calculations; (−) remains difficult to specify the appropriate envelopes of correct behaviour.
Cyber-physical systems resilience	(+) interaction between cyber and physical domains can improve critical infrastructures and service resilience; (+) novel modelling approaches can measure the resilience of cyber-physical systems and improve their design; (−) consequences of mitigation systems interacting in the cyber and physical domain are not well-understood; (−) engineering approaches considering safety and security aspects of cyber-physical systems are immature.
Resilient systems design/operation; people and their roles	(+) resilience techniques in design and operation increase overall system resilience; (+) people develop and provide coping strategies when computing or communication systems fail; (−) resilience by design and in operation each require careful attention to many procedural details; (−) user behavior is unpredictable and may be a major source of problems.

to map high-level requests to low-level configurations and on programmable networks (e.g., SDN, NFV) to facilitate resource deployment and configuration. Intents could specify high-level security and resilience objectives using a controlled natural language as in [71]. To realize an intent, it is needed to: 1) determine the set of both physical and virtualized resources required to fulfill the high-level goal and 2) instantiate and configure the low-level information of the physical or virtual components. In our opinion, the technology is mature enough to address these challenges. More specifically, using cognitive reasoning and learning can address the former, and network control/data plane programmability the latter.

B. Resilience Research and Network Function Virtualisation

Recent interest and activity in NFV have renewed the prospect of automated and even autonomic management and control. The difference between these is that autonomic operation implies some form of (machine) learning. This is not necessarily novel and has been proposed under various circumstances before. However, the technology and engineered artifacts that are at our disposal nowadays keep on improving. NFV, enabled by highly capable virtualization technology, comes along at a time when it is possible to construct highly flexible systems in software. These will run on fast, low-cost, commodity hardware with an abundance of memory. Thus, it should be more easily feasible to build resilience-by-design systems with appropriate numbers and location of redundant components according to the system's resilience specification, compared to previous times when systems were based on more specialized, less flexible, and costly hardware. Operational resilience based on NFV should also be more feasible because of the possibility of lightweight monitoring software that can be used to observe and detect anomalies and to automatically (if not autonomically) remediate while updating the system's resilience knowledge base. Included in this process is the replacement of failed VNF modules (see Fig. 5) with the possibility of minimal disruption to normal service. Maintenance of a suitable QoS and Quality of Experience (QoE) for the system's users is the ultimate aim, though this remains very difficult to achieve. Current research is targeted at building VNF-based implementations and experimenting with them using relevant use cases in which a range of challenges is introduced to the system under test (e.g., the NG-CDI project).[3] There are several research questions including: 1) what granularity of VNFs should be used (should they be large, i.e., realizing a whole service, very basic, i.e., a microservice, or somewhere in between); 2) should the structure (and granularity) of VNFs be classified and standardized in order to assist the service construction (chaining) process; and 3) and how will the complexity of typically large-scale, real-world systems be handled at the design stage, not least when they involved human users and operators (see also Section IV-E).

C. Programmable Self-Organization and Resilience

Self-organizing systems have enormous potential for reducing the need for human intervention in both the operation and final function of networks—and this is both a strength and a weakness. Clearly, the ability of networks and the services running over them to self-organize initially—and in response to changes in their environment, function, load, and so forth—is highly desirable and potentially makes systems far more robust, responsive, and resilient than they would be if requiring human management. Equally clearly, removing humans from the loop reduces oversight and the ability to intervene when autonomic mechanisms fail—and at the current state of the art, it is more a case of when, and not if, such interventions will be required.

One weakness of many current approaches is that it is difficult to specify the desired behaviors (to exhibit and to avoid) with sufficient precision and breadth for long-term autonomic deployment. In most other branches of

[3]http://www.ng-cdi.org

software engineering such *a priori* specifications have been abandoned, recognizing that they often do not deliver regardless of the care or technology. This being the case the questions are: 1) How can we describe the "envelopes of behavior" within which systems should adapt? 2) What constitutes "correct" behavior when "correct" is itself a complex function of environment and user expectation?

A possible way to address these issues is to move toward a more "self-organizing by design" programming model. The advantage is that the "plumbing" functions of self-organization and adaptation are moved out of the application domain and into the infrastructure. This has numerous advantages, including: allowing programmers to focus on service-level code, unbundled from the mechanics of self-organization; providing richer programming abstractions for developing such service logic; permitting higher-level reasoning about behavior and correctness; and supporting compositionality and other features needed for scaling-out services.

Whether there exists a set of abstractions that will support these features over a broad class of application domains remains an open question and almost every aspect remains an active subject of research. It is not clear, for example, whether self-* properties can be added incrementally to existing service architectures and ecosystems, or whether new services must be developed *ex nihilo*, with all the costs and difficulties this would imply: clearly incremental deployment would be both more cost-effective and more likely to preserve existing correct behavior.

Self-organization by design is not the same as resilience by design. Resilience is one self-* property among many, needing to be specified precisely and then integrated with (and/or traded off against) others. One might approach this problem by trying to find a collection of inherently resilient programming abstractions, for instance, aggregate programming provides a practical implementation of a theory of resilient computational fields [41], [72] (see Section III-B3 and several other similar examples). However, it is possible that the very act of abstracting one aspect of self-management, while simplifying construction and reasoning in another aspect of a system, complicates those processes throughout. Computational fields make few if any guarantees about timely adaptation and mask the mechanisms that might allow one to estimate key properties (e.g., stabilization time after disruption). This may be a tradeoff worth making for some applications, but not for others.

It is also worth observing that many (although not all) rich programming models come at a further cost of reduced performance and increased resource utilization. In networks, the resource being consumed is often bandwidth, and the costs can be prohibitive for some applications. Wireless sensor networks, for example, would benefit hugely from self-stabilizing and resilient communications provided generically, but typically cannot support the extra communications this entails, given that the extra messages also consume power. One is, therefore, often reduced to manual optimization and convoluted design even when a better solution is, in some sense, readily available.

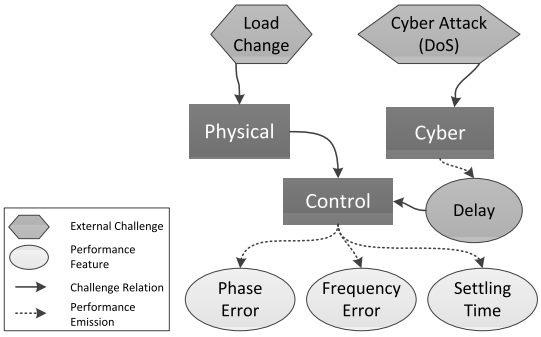


Fig. 6. ***Challenges and performance measures with respect to the cyber-physical system domains for a power systems example. A DoS attack in the cyber domain, along with a load change, can result in phase and frequency errors, and poor settling times in the control domain [75].***

D. Cyber–Physical Systems Resilience

Societally critical systems are in the process of being digitalized. In many cases, they are cyber-physical systems (and systems of systems) wherein a combination of cyber components (including communication networks) and control algorithms are used to manage a physical process or system [73]. Examples of cyber-physical systems include autonomous vehicles, industrial plants, and electricity distribution systems.

Previous research has investigated the effects that perturbations in the cyber domain, e.g., caused by unreliable communication networks or cyberattacks can have on an algorithm's ability to control a physical system (e.g., instabilities can occur) [74]. Therefore, it is important to understand and measure the relationships between the cyber, control, and physical domains to derive self-organizing system designs that are resilient. A resilience metric framework has been proposed that can be used to explore these interdependencies and examine the resilience of a cyber-physical system [75]. This involves modeling the relationships between the domains and acknowledges that reduced performance in one domain, e.g., increased network delay that is caused by a Denial of Service (DoS) attack, can result in a challenge to another—the ability for a system to be controlled. This concept is exemplified in Fig. 6, which shows an example of a power system. The measurement framework can provide useful insights at design time, e.g., by modeling a system's performance. However, the modeling process is—from an application perspective—rather complicated, and the future research needs to be carried out to determine the benefits of such an approach to others that do not aim to model these interdependencies [76].

Resilient control schemes have been developed to improve the robustness of control systems to

various challenges, including faults and cyberattacks. In general, control systems are designed to be resilient to missing or erroneous measurements from field devices that are, for instance, caused by component failures. State estimation algorithms are typically used to infer a system's state in these circumstances [77].

Recent research has investigated how ICS can be made resilient to cyberattacks [78]. In this paper, the authors present a scheme that combines an anomaly detection system (using measurements taken from a physical process) with a resilient control scheme to mitigate the effects of data manipulation attacks to a heating system that uses a water boiler. The aim of the research is to examine whether resilient control decisions could be improved, i.e., be more appropriate, with knowledge of the presence of a cyberattack. Similar schemes have been proposed for use in electricity substations [79].

This avenue of research has the potential to improve the resilience of cyberp-hysical systems, enabling automatic adaptation of a system to different forms of challenges, including cyberattacks. However, there are several concerns that need to be addressed. For instance, the resilient control schemes make the system less controllable (e.g., by introducing a delay before telecontrol commands are applied). This behavior is desirable, as it can make the system more stable if it is being manipulated by an attacker. Unfortunately, such schemes could also reduce the controllability of a system because of false positives from detection systems or as a consequence of the schemes themselves being targeted by an attacker. This is highly problematic for safety-critical systems. Therefore, research is needed into when such schemes can be applied and systematic design guides, e.g., reference architectures, that support their application.

E. Resilient Systems Design and Operation

As has been discussed, resilient systems and applications must continue to offer an acceptable level of service, whatever the nature of the challenge or hazard. Applications that (will) need to be resilient include ICS, IoT networks that support health care, and future networks that support autonomous vehicles in the future Smart Cities. There are many other cases where communication systems provide the underpinning for critical infrastructures or services. In all of these humans form, an integral part of the overall system alongside the core technical elements. This applies to all stages of the system design and implementation, and usually also in their operation. Many issues still need to be investigated and solved in order to build successful and cost-effective resilient systems.

1) Architecture and Realization: The architecture and the realization of resilience are far from being mature, despite prior work that sheds understanding of the principles of resilience [4], as previously described in this paper.

In principle questions remain, notably how to specify resilience in a way that systems can be engineered to have the right properties. How resilient services can be composed of a service-level agreement (SLA) that describes the desired level of resilience is still unsolved. Further questions are, for example: 1) what granularity of resilience is implied by the specification; is the entity under consideration a service, a subsystem, or the entire system? 2) what classes of resilience are to be indicated in the SLA; should it be (near to) 100 %, should it be the best effort, or something in between? 3) how would the SLA be monitored and, more significantly, how would it be enforced? Yet, another issue is what would be the consequence of violating the SLA for the service provider, e.g., would this be in financial or legal terms, or both?

While techniques have been developed to analyze and quantify the network resilience in terms of topological robustness to attacks and disasters, as well as a static state-space analysis for user service level versus network operational state, much more work needs to be done. For example, in the Collaborative Research Center MAKI (Multi-Mechanisms Adaptation for the Future Internet) adaptive monitoring in a mobile environment has been investigated [80]. A monitoring service that executes transitions between distinct monitoring mechanisms has been developed to adapt to dynamic network conditions [81]. This shows how the mechanisms and services that resilience builds on can also adjust depending on the prevailing conditions, in addition to self-organization and adaptation in response to resilience challenges. Analysis of MANETs, in which the topology is dynamically changing, shows that future topologies are generally unpredictable [82]. In this case, the self-organizing nature includes self-repair as the network automatically repairs failed nodes. However, a temporal analysis that considers the tradeoff between severity and duration is needed (short–severe vs. long–mild) [83]. Other issues are related to large complex heterogeneous networks that challenge both graph-theoretic analysis and simulation-based modeling For example, a Smart City environment incorporating smart home, smart building, vehicular, and 5G networking is extraordinarily complex and challenges known analysis techniques [84].

2) People and Their Roles: A fundamental issue is the involvement of people in the operation of critical infrastructures. People, their organizational roles and responsibilities, and their behavior, are crucial elements in these systems.

It is crucial to take a full and proper account of the various, distinct roles of people in systems design and operation. This issue has been studied in related disciplines such as computer supported cooperative work and human–computer interaction, involving alongside computer scientists. Also, the roles of people in systems have been previously investigated in principle and practice by Checkland [85]. His work recognized the important but extremely difficult-to-model role of humans within

organizations and the systems of which they are an intrinsic part. Also, the work in [86] has added significantly to the theory and practice of resilient systems.

Despite the considerable work that has been carried out in this area in the past, it is worth revisiting the issue of people. This is especially so since the goals have changed with the new focus on designing and building natively resilient systems, in spite of the inevitable presence of people at many if not all stages of their design and operation. There is no doubt that the presence of people, with all their frailties, can be a major source of problems. However, individuals and groups can also be a source of strength, not least when it comes to coping strategies and applying their intelligence in situations where machines have failed or unanticipated problems have arisen.

People may be categorized as owners, policy makers, designers, implementers, operators, or users. These roles reflect the viewpoint from which the person sees the system in question. Of these, the designers and implementers are the only ones who can influence the "internals" of the system; all others will essentially see the system as a "black box" with its inputs and outputs.

One of the issues that arise is whether or to what extent people can be omitted or merely sidelined in the interactions they have with the system. For design and implementation, this is extremely difficult as people have to be involved in gathering requirements and are (still) the origins of designs, although programming can to some extent be automated. The question of automation at the operator or user levels is perhaps the one most visited in the past. However, it is one that may still arouse the most controversy, not least over safety concerns, as well as fears around the loss of employment. Further research also needs to deal with the influence and impact of humans as system owners, operators, and policy makers.

From this arises the fundamental research question: whether the behavior of humans in their interactions with systems can be properly modeled. Can, for instance, people be simply represented as components of systems with appropriate properties and risks assigned to them? Should they be represented in some sort of statistical way, or can their roles (at least in some circumstances) be constrained in such a way that their behavior is more deterministic—perhaps by the application of rules and checks [87], [88].

V. CONCLUSION

Resilience needs to be an inherent property of networked systems in order to maintain a good service while withstanding attacks or dealing with other adverse events. Increasingly, networked systems are deployed in mobile and highly dynamic environments (e.g., in a smart city, collaborative autonomous driving, or digital health context). Furthermore, they control more and more critical infrastructure elements. Within these use cases, different system parts have a high degree of autonomicity; thus, self-organization is the most appropriate way of allowing systems to form and reform. In resilience research it was acknowledged early on that self-* properties can help ensure that a satisfactory level of service can be maintained and that challenges can be dealt with appropriately. This is, for instance, reflected in the D^2R^2 model and the policy-based resilience research that uses self-management and self-organization to adapt to challenges. We conclude that resilience cannot properly be provided as an add-on but that networked systems require resilience by design, making use of autonomic principles throughout the entire resilience process.

In this paper, we have argued that self-organization is essential within elements of the resilience ecosystem, such as an adaptive monitoring infrastructure. Thus, mechanisms and services that resilience relies on can also adjust depending on the prevailing conditions, in addition to self-organization and adaptation in response to resilience challenges. However, self-organization poses challenges in itself since the behavior of system components can be less predictable, which makes detection and mitigation more difficult.

An aspect we have highlighted that requires urgent attention is the behavior of humans who interact with networked systems, including system operators employed within the relevant organizations. Humans are participants within the system context, and their behavior has to be taken into account when designing resilient solutions. The modeling of people, and potentially their replacement by self-organizing system elements, are crucial aspects in ensuring the resilience of future networked systems.

Finally, apart from the specific technical aspects addressed in this paper, it is essential to consider the overall system and its context, i.e., systems thinking is required from the design phase onward in order to ensure coordinated resilience across all layers and elements in networked systems. ■

REFERENCES

[1] B. M. Leiner *et al.*, "A brief history of the internet," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 22–31, Oct. 2009. [Online]. Available: http://doi.acm.org/10.1145/1629607.1629613

[2] S. Dobson *et al.*, "A survey of autonomic communications," *ACM Trans. Auto. Adapt. Syst.*, vol. 1, no. 2, pp. 223–259, Dec. 2006, doi: 10.1145/1186778.1186782.

[3] P. Smith *et al.*, "Network resilience: A systematic approach," *IEEE Commun. Mag.*, vol. 49, no. 7, pp. 88–97, Jul. 2011.

[4] J. P. G. Sterbenz *et al.*, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, pp. 1245–1265, Jun. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610000824

[5] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May 2011.

[6] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. 26th USENIX Secur. Symp. (USENIX Security)*, Vancouver, BC, Canada: USENIX Association, 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[7] Q. Chen and R. A. Bridges, "Automated behavioral analysis of malware: A case study of WannaCry ransomware," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 454–460.

[8] J. J. Santanna, R. de O. Schmidt, D. Tuncer, J. de Vries, L. Z. Granville, and A. Pras, "Booter blacklist: Unveiling DDoS-for-hire websites," in *Proc. 12th Int. Conf. Netw. Service Manage. (CNSM)*, Oct./Nov. 2016, pp. 144–152.

[9] J. Fiksel, "Sustainability and resilience: Toward a systems approach," *IEEE Eng. Manag. Rev.*, vol. 35, no. 3, p. 5, 3rd Quart., 2007.

[10] G. Dondossola, G. Deconinck, F. D. Giandomenico, S. Donatelli, M. Kaâniche, and P. Veríssimo, "Critical utility infrastructural resilience," *CoRR*, 2012. [Online]. Available: http://arxiv.org/abs/1211.5736

[11] E. W. Dijkstra, "Self-stabilizing systems in spite of distributed control," *Commun. ACM*, vol. 17, no. 11, pp. 643–644, Nov. 1974, doi: 10.1145/361179.361202.

[12] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Trans. Autom. Control*, vol. 51, no. 3, pp. 401–420, Mar. 2006, doi: 10.1109/TAC.2005.864190.

[13] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 1999, pp. 251–262. [Online]. Available: http://doi.acm.org/10.1145/316188.316229

[14] M. Médard and S. S. Lumetta, "Network reliability and fault tolerance," in *Encyclopaedia of Telecommunications*, J. Proakis, Ed. Hoboken, NJ, USA: Wiley, 2003, doi: 10.1002/0471219282.eot281.

[15] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, vol. 36, no. 1, pp. 41–52, Jan. 2003.

[16] G. Bouabene, C. Jelger, C. Tschudin, S. Schmid, A. Keller, and M. May, "The autonomic network architecture (ANA)," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 1, pp. 4–14, Jan. 2010.

[17] P. Vlacheas, V. Stavroulaki, P. Demestichas, S. Cadzow, S. Gorniak, and D. Ikonomou, "Ontology and taxonomies of resilience, version 1," in *Proc. Eur. Netw. Inf. Secur. Agency (ENISA) Rep.*, 2011, pp. 1–59.

[18] P. Cholda, A. Mykkeltveit, B. E. Helvik, O. J. Wittner, and A. Jajszczyk, "A survey of resilience differentiation frameworks in communication networks," *IEEE Commun. Surveys Tuts.*, vol. 9, no. 4, pp. 32–55, 4th Quart., 2007.

[19] A. Clemm, *Network Management Fundamentals*. Hoboken, NJ, USA: Cisco Press, 2006.

[20] J. M. McQuillan and D. C. Walden, "The ARPA network design decisions," *Comput. Netw.*, vol. 1, no. 5, pp. 243–289, 1977. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0376507577900149

[21] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. A. Longstaff, and N. R. Mead, "Survivability: Protecting your critical systems," *IEEE Internet Comput.*, vol. 3, no. 6, pp. 55–63, Nov. 1999, doi: 10.1109/4236.807008.

[22] J. P. G. Sterbenz *et al.*, "Survivable mobile wireless networks: Issues, challenges, and research directions," in *Proc. 1st ACM Workshop Wireless Security (WiSE)*. New York, NY, USA: ACM, 2002, pp. 31–40. [Online]. Available: http://doi.acm.org/10.1145/570681.570685

[23] E. K. Çetinkaya and J. P. G. Sterbenz, "A taxonomy of network challenges," in *Proc. 9th Int. Conf. Design Reliable Commun. Netw. (DRCN)*, Mar. 2013, pp. 322–330.

[24] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, and M. Schöller, and P. Smith, "Redundancy, diversity, and connectivity to achieve multilevel network resilience, survivability, and disruption tolerance," *Telecommun. Syst.*, vol. 56, no. 1, pp. 17–31, May 2014, doi: 10.1007/s11235-013-9816-9.

[25] W. D. Grover, "The selfhealing network," in *Proc. IEEE GLOBECOM*, Nov. 1987, pp. 1090–1095.

[26] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. G. Sterbenz, "Analysing GeoPath diversity and improving routing performance in optical networks," *Comput. Netw.*, vol. 82, pp. 50–67, May 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128615000699

[27] C. Cecati, G. Mokryani, A. Piccolo, and P. Siano, "An overview on the smart grid concept," in *Proc. 36th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2010, pp. 3322–3327.

[28] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, pp. 1025–1028, Apr. 2010.

[29] S. Konig, S. Rass, S. Schauer, and A. Beck, "Risk propagation analysis and visualization using percolation theory," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, pp. 1–8, 2016, doi: 10.14569/IJACSA.2016.070194.

[30] D. Xu, Y. Xiong, C. Qiao, and G. Li, "Failure protection in layered networks with shared risk link groups," *IEEE Netw.*, vol. 18, no. 3, pp. 36–41, May 2004.

[31] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security Privacy*, vol. 7, no. 3, pp. 75–77, Jun. 2009.

[32] R. M. Lee, M. J. Assante, and T. Çonway, "Analysis of the Cyber Attack on the Ukrainian Power Grid," SANS ICS and E-ISAC, White Paper, Mar. 2016. [Online]. Available: https://ics.sans.org/media/E-ISAC_SANS_Ukraine_DUC_5.pdf

[33] M. R. Endsley, *Designing for Situation Awareness: An Approach to User-Centered Design*, 2nd ed. 2016. [Online]. Available: https://nls.ldls.org.uk/welcome.html?ark:/81055/vdc_100045254794.0x000001

[34] M. R. Watson, N.-U.-H. Shirazi, A. K. Marnerides, A. Mauthe, and D. Hutchison, "Towards a distributed, self-organising approach to malware detection in cloud computing," in *Self-Organizing Systems*, W. Elmenreich, F. Dressler, and V. Loreto, Eds. Berlin, Germany: Springer, 2014, pp. 182–185.

[35] NIST, "Framework for improving critical infrastructure cybersecurity, version 1.0," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., Feb. 2014.

[36] A. Schaeffer-Filho, P. Smith, A. Mauthe, and D. Hutchison, "Network resilience with reusable management patterns," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 105–115, Jul. 2014.

[37] M. Sloman and E. Lupu, "Security and management policy specification," *IEEE Netw.*, vol. 16, no. 2, pp. 10–19, Mar. 2002.

[38] P. Smith *et al.*, "Strategies for network resilience: Capitalising on policies," in *Mechanisms for Autonomous Management of Networks and Services*, B. Stiller and F. De Turck, Eds. Berlin, Germany: Springer, 2010, pp. 118–122.

[39] A. Schaeffer-Filho, P. Smith, A. Mauthe, D. Hutchison, Y. Yu, and M. Fry, "A framework for the design and evaluation of network resilience management," in *Proc. IEEE Netw. Oper. Manage. Symp.*, Apr. 2012, pp. 401–408.

[40] E. Lupu *et al.*, "AMUSE: Autonomic management of ubiquitous e-Health systems," *Concurrency Comput., Pract. Exper.*, vol. 20, no. 3, pp. 277–295, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.1194

[41] M. Viroli, F. Damiani, and J. Beal, "A calculus of computational fields," in *Advances in Service-Oriented and Cloud Computing (ESOCC)* (Communications in Computer and Information Science), vol. 393, C. Canali and M. Villari, Eds. Springer, 2013.

[42] F. Zambonelli *et al.*, "Developing pervasive multi-agent systems with nature-inspired coordination," *Pervas. Mobile Comput.*, vol. 17, pp. 236–252, Feb. 2015, doi: 10.1016/j.pmcj.2014.12.002.

[43] D. Pianini, S. Dobson, and M. Viroli, "Self-stabilising target counting in wireless sensor networks using euler integration," in *Proc. 11th IEEE Int. Conf. Self-Adapt. Self-Organizing Syst. (SASO)*, Sep. 2017, pp. 11–20, doi: 10.1109/SASO.2017.10.

[44] J. Beal, D. Pianini, and M. Viroli, "Aggregate programming for the Internet of Things," *Computer*, vol. 48, no. 9, pp. 22–30, Sep. 2015, doi: 10.1109/MC.2015.261.

[45] R. E. Lyons and W. Vanderkulk, "The use of triple-modular redundancy to improve computer reliability," *IBM J. Res. Develop.*, vol. 6, no. 2, pp. 200–209, Apr. 1962, doi: 10.1147/rd.62.0200.

[46] M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*. New York, NY, USA: Wiley, 2002.

[47] A. Jabbar, "A framework to quantify network resilience and survivability," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. Kansas, Lawrence, KS, USA, 2010.

[48] A. K. Marnerides, D. P. Pezaros, J. Jose, A. U. Mauthe, and D. Hutchison, "A situation aware information infrastructure (SAI^2) framework," in *Smart Objects and Technologies for Social Good*, O. Gaggi, P. Manzoni, C. Palazzi, A. Bujari, and J. M. Marquez-Barja, Eds. Cham, Switzerland: Springer, 2017, pp. 186–194.

[49] C. Jelger, C. Tschudin, S. Schmid, and G. Leduc, "Basic abstractions for an autonomic network architecture," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2007, pp. 1–6.

[50] M. Sifalakis, A. Louca, A. Mauthe, L. Peluso, and T. Zseby, "A functional composition framework for autonomic network architectures," in *Proc. IEEE Netw. Oper. Manage. Symp. Workshops (NOMS Workshops)*, Apr. 2008, pp. 328–334.

[51] M. May, M. Siekkinen, V. Goebel, T. Plagemann, R. Chaparadza, and L. Peluso, "Monitoring as first class citizen in an autonomic network universe," in *Proc. 2nd Bio-Inspired Models Netw., Inf. Comput. Syst.*, Dec. 2007, pp. 247–254.

[52] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research," *IEEE Commun. Mag.*, vol. 35, no. 1, pp. 80–86, Jan. 1997.

[53] K. L. Calvert, S. Bhattacharjee, E. Zegura, and J. Sterbenz, "Directions in active networks," *IEEE Commun. Mag.*, vol. 36, no. 10, pp. 72–78, Oct. 1998.

[54] J. J. Garrahan, P. A. Russo, K. Kitami, and R. Kung, "Intelligent network overview," *IEEE Commun. Mag.*, vol. 31, no. 3, pp. 30–36, Mar. 1993.

[55] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008, doi: 10.1145/1355734.1355746.

[56] A. S. da Silva, P. Smith, A. Mauthe, and A. Schaeffer-Filho, "Resilience support in software-defined networking: A survey," *Comput. Netw.*, vol. 92, pp. 189–207, Dec. 2015, doi: 10.1016/j.comnet.2015.09.012.

[57] H. Song, "Protocol-oblivious forwarding: Unleash the power of SDN through a future-proof forwarding plane," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*. New York, NY, USA: ACM, 2013, pp. 127–132, doi: 10.1145/2491185.2491190.

[58] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2656877.2656890

[59] ETSI, "Network functions virtualisation (NFV); resiliency requirements," ETSI, Sophia Antipolis, France, Tech. Rep. ETSI GS NFV-REL 001 V1.1.1, 01 2015.

[60] C. Rotsos *et al.*, "Network service orchestration standardization: A technology survey," *Comput. Standards Interfaces*, vol. 54, pp. 203–215, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548916302458

[61] B. Jennings *et al.*, "Towards autonomic management of communications networks," *IEEE Commun. Mag.*, vol. 45, no. 10, pp. 112–121, Oct. 2007.

[62] J. Strassner, "Autonomic networking: Theory and practice," in *Proc. IEEE Netw. Oper. Manage. Symp.*

(NOMS), Apr. 2008, p. 26.
[63] A. K. Bandara *et al.*, "Policy refinement for IP differentiated services quality of service management," *IEEE Trans. Netw. Service Manage.*, vol. 3, no. 2, pp. 2–13, Apr. 2006.
[64] M. S. Beigi, S. Calo, and D. Verma, "Policy transformation techniques in policy-based systems management," in *Proc. POLICY*, Jun. 2004, pp. 13–22.
[65] J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, G. Pavlou, and A. L. Lafuente, "Using linear temporal model checking for goal-oriented policy refinement frameworks," in *Proc. 6th IEEE Int. Workshop Policies Distrib. Syst. Netw. (POLICY)*, Jun. 2005, pp. 181–190.
[66] C. Brodie *et al.*, "The coalition policy management portal for policy authoring, verification, and deployment," in *Proc. POLICY*. Washington, DC, USA: IEEE Computer Society, Jun. 2008, pp. 247–249.
[67] Cisco. *Intent-Based Networking*. Accessed: May 2018. [Online]. Available: https://www.cisco.com/c/en/us/solutions/intent-based-networking.html
[68] Open Networking Foundation, "Intent NBI—Definition and principles," ONF, Tech. Rep. ONF TR-523, Oct. 2016.
[69] Y. Elkhatib, G. Coulson, and G. Tyson, "Charting an intent driven network," in *Proc. 13th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2017, pp. 1–5.
[70] W. Cerroni *et al.*, "Intent-based management and orchestration of heterogeneous openflow/IoT SDN domains," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, Jul. 2017, pp. 1–9.
[71] E. J. Scheid *et al.*, "INSpIRE: Integrated NFV-based intent refinement environment," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 186–194.
[72] M. Viroli and F. Damiani, "A calculus of self-stabilising computational fields," in *Coordination Models Languages: COORDINATION 2014* (Lecture Notes in Computer Science), vol. 8459, E. Kühn and R. Pugliese, Eds. 2014.
[73] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Design Autom. Conf. (DAC)*. New York, NY, USA: ACM, 2010, pp. 731–736. [Online]. Available: http://doi.acm.org/10.1145/1837274.1837461
[74] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Trans. Ind. Electron.*, vol. 57, no. 7, pp. 2527–2535, Jul. 2010.
[75] I. Friedberg, K. McLaughlin, and P. Smith, "A cyber-physical resilience metric for smart grids," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Apr. 2017, pp. 1–5.
[76] J.-P. Watson *et al.*, "Conceptual framework for developing resilience metrics for the electricity, oil, and gas sectors in the United States," Sandia National Lab., Albuquerque, NM, USA, Tech. Rep., 2014.
[77] H. Yih-Fang, S. Werner, H. Jing, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Process. Mag.*, vol. 29, no. 5, pp. 33–43, Sep. 2012.
[78] K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proc. IEEE*, vol. 106, no. 1, pp. 113–128, Jan. 2018.
[79] D. Mashima, P. Gunathilaka, and B. Chen, "Artificial command delaying for secure substation remote control: Design and implementation," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 471–482, Jan. 2019.
[80] N. Richerzhagen, D. Stingl, B. Richerzhagen, A. Mauthe, and R. Steinmetz, "Adaptive monitoring for mobile networks in challenging environments," in *Proc. 24th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2015, pp. 1–8.
[81] N. Richerzhagen, P. Lieser, B. Richerzhagen, B. Koldehofe, I. Stavrakakis, and R. Steinmetz, "Change as chance: Transition-enabled monitoring for dynamic networks and environments," in *Proc. 14th Wireless Demand Netw. Syst. Services Conf. (WONS)*, Feb. 2018, pp. 1–8.
[82] D. Zhang and J. P. G. Sterbenz, "Analysis of critical node attacks in mobile ad hoc networks," in *Proc. 6th Int. Workshop Reliable Netw. Design Modeling (RNDM)*, Nov. 2014, pp. 171–178.
[83] J. P. G. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, S. Qian, and J. P. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation," *Telecommun. Syst.*, vol. 52, no. 2, pp. 705–736, Feb. 2013, doi: 10.1007/s11235-011-9573-6.
[84] J. P. G. Sterbenz, "Smart city and IoT resilience, survivability, and disruption tolerance: Challenges, modelling, and a survey of research opportunities," in *Proc. 9th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Sep. 2017, pp. 1–6.
[85] P. B. Checkland, *Systems Thinking, Systems Practice*. Chichester, U.K.: Wiley, 1981.
[86] D. D. Woods and E. Hollnagel, *Resilience Engineering: Concepts and Precepts*. Aldershot, U.K.: CRC Press, 2006.
[87] J. S. Busby, D. Hutchison, M. F. Rouncefield, H. Niedermayer, and P. Smith, "Network of excellence in internet science: Social aspects in understanding Internet as critical infrastructure and implications for future networks (EINS Internet Science)," Lancaster Univ., Lancaster, U.K., Tech. Rep. D7.2.2, Jan. 2016. [Online]. Available: http://www.internet-science.eu/sites/eins/files/biblio/EINS_JRA7_D7.2.2.pdf
[88] ETH Zürich (Future Resilient Systems). *People and Operations in Resilient Systems*. Accessed: Jun. 2018. [Online]. Available: http://www.frs.ethz.ch/research/energy-and-comparative-system/people-and-operations.html

ABOUT THE AUTHORS

Simon Dobson (Senior Member, IEEE) received the B.Sc. degree in computer science from the University of Newcastle, Newcastle upon Tyne, U.K., in 1989, and the D.Phil. degree in computer science from the University of York, York, U.K., in 1993.

He was a Lead Investigator on the EUR5M EPSRC-funded program grant in the Science of Sensor Systems Software. He is currently a Professor of Computer Science at the School of Computer Science, University of St Andrews, St Andrews, U.K. He is involved in complex and sensor systems, especially on sensor analytics and the modeling of complex processes. He has authored or co-authored more than 150 internationally peer-reviewed publications, driven by leadership roles in research grants worth over EUR30M.

Dr. Dobson is a Chartered Fellow of the British Computer Society and a Senior Member of the Association for Computing Machinery (ACM). He has served, among other activities, as the Program Chair and the General Chair for the IEEE International Conference on Autonomic Computing; a member of UKCRC, the expert committee on U.K. computing research; and on the program committees of a wide range of leading international conferences and specialized workshops. He is a Chartered Engineer. He served as an Associate Editor for *ACM Transactions on Autonomous and Adaptive Systems*.

David Hutchison has helped build a strong research group in computer networks, which is well known internationally for contributions in a range of areas including Quality of Service architecture and mechanisms, multimedia caching and filtering, multicast engineering, active and programmable networking, content distribution networks, mobile IPv6 systems and applications, communications infrastructures for Grid-based systems, testbed activities, and Internet Science. He is currently a Professor of Computing at Lancaster University, Lancaster, U.K., and the Founding Director of the InfoLab 21, Lancaster. He current research interests include resilient and secure networking, especially future Internet and also the protection of critical infrastructures including industrial control systems.

He has served on the TPC of top conferences such as ACM SIGCOMM and IEEE INFOCOM. He served on Editorial Boards of the Springer *Lecture Notes in Computer Science*, *Computer Networks Journal*, and the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, and the Editor of the Wiley book series in *Computer Networks and Distributed Systems*.

Andreas Mauthe was with the DFG-SFB Maki, Technische Universität Darmstadt, Darmstadt, Germany, where he was involved in transition and adaptation of networked systems in the context of content networks (specifically Information Centric Networks). He is currently a Professor with Data and Information Security, University Koblenz, Koblenz, Germany. In the resilience domain, his research includes anomaly detection, and resilience management in systems such as Clouds, Smart Grids, Internet of Things environments, and industrial control systems. Within this domain, he has been developing novel research in areas such as multilevel anomaly detection (e.g., inventing a Cloud malware detection which has been patented), situational awareness in cyber systems, and human factors in system resilience. His current research interests include network management and security, autonomic and resilient systems, and content networking and adaptive networks.

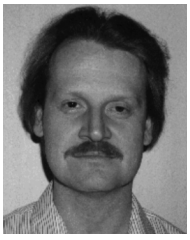
Alberto Schaeffer-Filho (Member, IEEE) received the Ph.D. degree in computer science from Imperial College London, London, U.K., in 2009.

From 2009 to 2012, he was a Research Associate at Lancaster University, Lancaster, U.K. He is currently an Associate Professor with the Federal University of Rio Grande do Sul, Porto Alegre, Brazil. His current research interests include network/service management, network virtualization and software-defined networks, policy-based management, and security and resilience of networks. He has authored or co-authored more than 50 papers in leading peer-reviewed journals and conferences related to these topics,

Dr. Schaeffer-Filho is a CNPq-Brazil Research Fellow. He served as a TPC Member for important conferences in these areas, including CNSM in 2018, IEEE/IFIP NOMS in 2018, CNSM in 2017, IFIP/IEEE IM in 2017, IEEE CCNC in 2017, and IEEE INFOCOM CNTCV Workshop in 2017. He serves as the General Chair for SBRC 2019, Co-Chair for the IEEE ICC 2018 CQRM Symposium, Demo Co-Chair for IFIP/IEEE IM 2017, and Workshop Co-Chair for ManSDN/NFV 2015.

Paul Smith received the Ph.D. degree in computing from Lancaster University, Lancaster, U.K., in 2003.

He is currently a Senior Scientist with the Center for Digital Safety and Security, AIT Austrian Institute of Technology, Vienna, Austria. He has participated in a number of international research projects in this area, and has published articles on various aspects that relate to this core interest. His current research interests include developing applied solutions to ensuring the security and resilience of critical information infrastructures, in particular, securing future digitalized energy systems; solutions focused on approaches to risk management, anomaly detection, secure architecture specification, incident response, and resilience measurement; and understanding cyber-physical aspects of security and resilience.



James P. G. Sterbenz held Senior Staff and Research Management positions with BBN Technologies, Cambridge, MA, USA, GTE Laboratories, Waltham, MA, USA, and IBM Research, Rochester, MN, USA. He has been Principle Investigator in a number of projects including the NSF FIND and GENI Programs, the EU FIRE ResumeNet Project, led the GpENI international programmable network testbed Project, and has led a U.S. DoD Project in highly mobile *ad hoc* disruption-tolerant networking. He is currently a Professor of electrical engineering and computer science and the Director of the Networks Systems Laboratory, Information and Telecommunication Technology Center, The University of Kansas, Lawrence, KS, USA; a Visiting Professor of computing with the InfoLab 21, Lancaster University, Lancaster, U.K.; a Visiting Professor with the Chinese Academy of Sciences, Beijing, China; an Adjunct Professor with The Hong Kong Polytechnic University, Hong Kong; a Visiting Guest Professor with the Communication Systems Group, ETH Zurich, Zurich, Switzerland; and the Director of the ResiliNets Research Group, Lawrence, KS, USA, and Lancaster. His current research interests include resilient, survivable, and disruption tolerant networking, future Internet architectures including the Internet of Things and 5G mobile networks in smart cities, active and programmable networks, and high-performance networking and components.