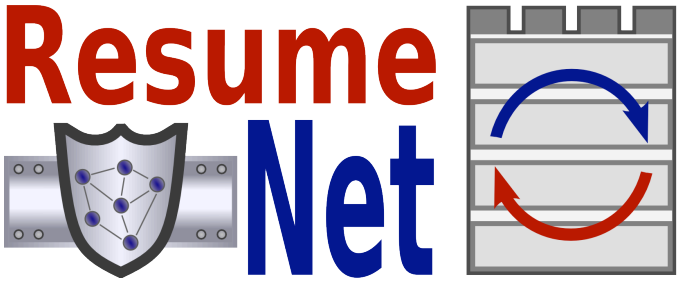
Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation



Deliverable number	1.5c
Deliverable name	Final strategy document for resilient networking
WP number	1
Delivery date	31/08/2011
Date of Preparation	16/9/2011
Editor	Paul Smith and David Hutchison (ULANC)
Contributor(s)	P. Smith (ULANC), D. Hutchison (ULANC), J. P.G. Sterbenz (KU), M. Schöller (NEC), A. Fessi (TUM), C. Doerr (TUDelft) and C. Lac (FT)
Internal reviewer	P. Gunningberg (UU) and M. Karaliopoulos

Summary

A continuous thread running through the ResumeNet project has been the development of a framework for resilient networking. The framework reflects what the project has learned about how to design and implement network and service resilience, and as such takes input from all the technical work packages. What has emerged is a systematic approach to network and service resilience, whose core component is a resilience control loop – the central element of our resilience framework. We propose our framework as a consistent approach that identifies how multilevel resilience mechanisms should be deployed. This is based on an understanding of resilience metrics and probable high impact challenges. Resilience mechanisms are managed using a loosely coupled policy-driven management architecture. Our framework improves on the state of the art through this coherent approach.

In this deliverable, we present our resilience framework, which includes implementation guidelines, processes, and toolsets that can be used to underpin the design of resilient networks and services with resilience mechanisms that function at various protocol levels. As one might expect, the deliverable (and the framework described therein) highlights research carried out in the project; throughout we point the reader to key deliverables that provide greater detail of our research outcomes. The elements of our framework that form the key contribution of our research, include:

A multilevel resilience metrics framework Being able to specify and measure desired levels of resilience is of critical importance, and is understood to be an area in which there is little consensus on how to approach it. We have developed a multilevel resilience metrics framework, summarised in Section 2, that can be used to understand and describe the resilience of networks and services, and the relationship metrics from different levels of the protocol stack have *e.g.*, whether they exhibit correlated or orthogonal behaviour. Accompanying the framework is a set of tools, such as simulation models and software libraries for examining metrics [ScH$^+$11, DMH10], that can be used to evaluate a given network topology in the presence of various challenges.

Processes for understanding challenges Deployed resilience mechanisms should be targeted at addressing the most probable high-impact challenges the network may face. In the context of network resilience, the challenges that could occur transcend those normally considered in other thematic areas, such as information security, fault tolerance and disruption tolerant networks. Without considering this broad spectrum of challenges, mechanisms could be inappropriately deployed. To manage this problem, we have developed a risk assessment process that can be used to identify high-impact challenges [SSH11]. This process builds on an informal categorisation of the forms of challenges that one must consider to ensure network resilience [SS09]. Our challenge categorisation and risk assessment process is described in Section 3.

A resilience management architecture The management of multilevel resilience mechanisms that potentially interact across different administrative domains can be complicated. Furthermore, the operation of resilience mechanisms should in many cases be done in real-time with potentially limited human intervention; incorrect operation could have significant negative consequences. To tackle these issues, we have developed a loosely coupled network management architecture, outlined here in Section 4, which makes use of policies to specify multi-stage resilience strategies – configurations of mechanisms that address a given challenge set [DSS$^+$10]. By using policies, strategies can be care-

fully crafted and evaluated, using a policy-driven network simulator we have developed, without the need to take resilience mechanisms off-line [YFSF+11].

A set of resilience mechanisms We have developed a number of resilience mechanisms that can be applied to a wide range of challenges. They span a number of stages of the $D^2R^2 + DR$ strategy and function at the network and service level. In particular, we have produced mechanisms to address malicious behaviour in networks, such as monetary-less cooperation incentives to mitigate selfish nodes in wireless mesh networks [PGLK10], game-theoretic approaches to protection against malware propagation [OOVM09], and an anomaly detection approach to detect and traceback attacks on encrypted protocols [FTV+10]. Furthermore, our mechanisms can be applied at different levels of the protocol stack in light of node and link failure, and include novel approaches to multipath routing in multi-hop wireless networks [LSZB10] and algorithms for creating resilient large-scale overlay networks [GHK11]. In Section 5, we highlight the novel aspects of the mechanisms developed in the project, and their likely deployment timescales.

An enemy of network resilience is complexity; using multilevel resilience mechanisms that share information and perform cross-layer control has the potential to increase complexity and produce undesirable emergent behaviours. To address this problem, we have developed a cross-layer framework, which uses a formalism to evaluate the optimal layer to place resilience functionality, thus reducing replicated functionality at different layers [BBF+10]. We introduce the formalism in Section 5.

Approaches to challenge detection Our understanding of the purpose of the *detect* stage of the $D^2R^2 + DR$ strategy has evolved over the lifetime of the project. We understand that its primary goal is to build situational awareness to inform decision making regarding remediation and recovery. How we have applied theories of situational awareness to the detect stage is discussed in Section 6. To identify challenges, we propose an incremental multi-stage approach that enables rapid remediation to reduce the likelihood of challenges causing catastrophic failure [YFSF+11]. Subsequently, remediation can be refined using improved challenge identification mechanisms. To support this multi-stage approach, we have developed an challenge identification architecture, which can be implemented using model-driven fault localisation techniques [SFM+10].

Ensuring resilience is a venture that should be tackled at multiple levels of the protocol stack in diverse topological (and geographical) locations. This involves information sharing across protocol layers, to build situational perception. We have investigated what multilevel metrics should be measured for resilience, and which tools should be used to collect and distribute this information. Our initial findings on this matter are presented in Section 6.

Aspects of the framework are readily applicable, whereas other elements represent our longer-term vision of how to realise network resilience. For example, the toolsets that are part of the multilevel metrics framework can be applied immediately to gain an understanding of the resilience of networks and services to various challenges. Furthermore, some of the resilience mechanisms we have developed, particularly those that operate at the service level, can be used to address challenges in the near-term future. Our longer-term vision for ensuring network resilience is embedded in our resilience management architecture and challenge detection approaches. These are arguably more disruptive from a (business) processes and technical implementation perspective, and further research is required in some cases to confirm their applicability. In Section 7, we discuss important areas for future research.

Contents

List of Terms

Below is a non-exhaustive and informal list of terms introduced and used in the deliverable, which is intended to assist the reader.

Challenge: an action, event or other phenomenon that can result in a network being unable to provide an acceptable level of service. Challenge categories are presented in Section 3.

Challenge analysis: the process of identifying the nature of a challenge, such as its root cause and origins, topologically, for example. This could be done using machine learning techniques, for example.

Challenge models: descriptions of challenges and associated symptoms that can be used to inform challenge analysis.

Consultant: an architectural component that can be used to expedite, inform or refine the process of decision making by a resilience manager.

Context information: a form of information that can be used to support decision making, which is *external* to the network and associated services, *i.e.*, is not typically gathered using network monitoring techniques. Weather information is an example of context.

D^2R^2 + DR strategy: a high-level conceptual strategy that can be used to understand how to engineer network resilience; two control loops are defined: a real-time loop and a non real-time learning loop.

Diagnose: the process of identifying shortcomings in network and service operation that could be improved via refinement.

Defend: the intended initial phase of the D^2R^2 + DR strategy that is used to harden the network and associated services to challenges.

Detect: the process of identifying a challenge using measurement and challenge analysis techniques, and determining whether a desired resilience target is being met using a resilience estimator.

Information source: an architectural component that can be used to inform challenge analysis; typically, information sources include monitoring tools and those that provide context information.

Managed entity: a hardware or software component that is controlled by the a resilience manager, typically a resilience mechanism.

Monitoring: the process of taking measurements of networks and services, in order to determine their state (in relation to a resilience target).

Network level (resilience): intended to describe or encompass approaches to ensuring resilience using mechanisms at Layer 1 (Physical) to 3 (Network) of the OSI network model.

Policy: an abstract or implementable expression of some behaviour the network should perform, which is not embodied in the mechanisms that realise the behaviour. Typically two forms of implementable policy exist: authorisation and obligation policies; see Section 4 for a description of these.

Refine: the process of improving network and service behaviour – the real-time component of the D^2R^2 + DR strategy – based on outcomes from the Diagnose phase.

Recover: the process of disengaging remediation mechanisms (see Remediate) when a challenge has abated. This is necessary as it is expected that remediation will result in a sub-optimal network state.

Remediate: the process of taking mitigative action to maintain acceptable levels of service, or provide a graceful degradation.

Resilience estimator: an architectural and conceptual component that is intended to use measurement information to determine whether resilience targets are being met.

Resilience manager: an architectural and conceptual component that is intended to make management decisions that ensure the resilience of a network and supported services.

Resilience mechanism: a hardware or software mechanism that is intended to defend against or remedy a challenge. This can include functionality associated with existing network elements, such as routers and servers, or additional mechanisms, such as firewalls and redundant links.

Resilience target: The target behaviour a network and services should provide in the presence of challenges. We envisage this is expressed in a Service Level Agreement (SLA) using various multilevel resilience metrics.

Resilience strategy: a strategy, described using a set of policies, that can be used to address a given set of challenges, which may span various stages of the D^2R^2 + DR strategy.

Service level (resilience): intended to describe or encompass approaches to ensuring resilience using mechanisms at Layer 4 (Transport) and upward of the OSI network model.

Situational awareness: models that can be used to understand the process of developing awareness of the nature of ongoing challenges and their effect on a network. Increased situational awareness can lead to improved decision making, *e.g.*, by a resilience manager.

For a more comprehensive set of definitions underlying the D^2R^2 + DR strategy, please see [SHc+10].

1 Introduction

Data communication networks are serving all kinds of human activities. Whether used for professional or leisure purposes, for safety-critical applications or e-commerce, the Internet in particular has become an integral part of our everyday lives, affecting the way societies operate. However, the Internet was not intended to serve all these roles and, as such, is vulnerable to a wide range of challenges. Malicious attacks, software and hardware faults, human mistakes (*e.g.*, software and hardware misconfigurations), and large-scale natural disasters threaten its normal operation.

Resilience, the ability of a network to defend against and maintain an acceptable level of service in the presence of such challenges [SHc+10], is viewed today, more than ever before, as a major requirement and design objective. These concerns are reflected, among others, in the Cyber Storm III exercise, carried out in the United States in September 2010, and the "cyber stress tests" conducted in Europe by the European Network and Information Security Agency (ENISA) in November 2010 [Eur11]; both aimed precisely at assessing the resilience of the Internet, this "critical infrastructure used by citizens, governments, and businesses".

Resilience evidently cuts through several thematic areas, such as information and network security, fault-tolerance, software dependability, and network survivability. A significant body of research has been carried out around these themes, typically focusing on specific mechanisms for resilience and subsets of the challenge space. We refer the reader to Sterbenz *et al.* [SHc+10] for a discussion on the relation of various resilience disciplines, and to a survey by Cholda *et al.* [CMH+07] on research work for network resilience.

However, despite these various efforts, under certain challenge conditions the Internet is less resilient that we would like it to be. There are many causes for this lack of resilience, some of the more prominent reasons include:

- networks and services are *complicated* to configure and manage, and they occasionally display undesirable emergent behaviours as a consequence of their *complexity* [Cro10];
- network resilience, in a similar manner to security, is not a core business concern, and as a consequence the cost of ensuring resilience – both capital and operational costs – can be marginalised;
- from an engineering perspective, *opacity between networking layers* can lead to inappropriate behaviour being exhibited by protocol instances because of a lack of information about the nature of a challenge;
- within the public Internet there are *low barriers to malicious behaviour* and *problems of attributing malicious behaviour to actors* [SB11] that make the orchestration of various forms of attack relatively straightforward and almost consequence free;
- and a lack of a well understood ways to *specify desired levels of network resilience*, for example in Service Level Agreements (SLAs), and mechanisms to effectively *measure and analyse* the performance of networks with respect to these requirements [Eur10]

A significant shortcoming of existing research and deployed systems is the lack of a systematic view of the resilience problem, *i.e.*, a view of how to engineer networks that are resilient to challenges that transcend those considered by a single thematic area. A non-systematic approach to understanding resilience targets and challenges, *e.g.*, one that does not cover

thematic areas, leads to an impoverished view of resilience objectives, potentially resulting in ill-suited solutions. Additionally, a patchwork of resilience mechanisms that are incoherently devised and deployed can result in undesirable behaviour and increased management complexity under challenge conditions, encumbering the overall network management task [ENI09]. We argue that resilience should be a critical and integral property of networks. Our work advances the state of the art by adopting a systematic approach to resilience, which takes into account the wide-variety of challenges that may occur. At the core of our approach is a coherent resilience framework, which includes implementation guidelines, processes, and toolsets that can be used to underpin the design of resilience mechanisms at various levels in the network.

1.1 Resilience Control Loop

Our resilience framework builds on work by Sterbenz *et al.* [SHc+10], whereby a number of resilience principles are defined, including a resilience strategy, called D^2R^2 + DR: Defend, Detect, Remediate, Recover, and Diagnose and Refine (Figure 1). The strategy describes a real-time control loop to allow dynamic adaptation of networks in response to challenges, and a non-real time control loop that aims to improve the design of the network, including the real-time loop operation, reflecting on past operational experience.

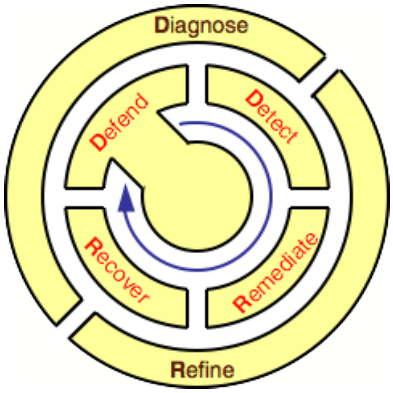


Figure 1: The D^2R^2 + DR resilience strategy [SHc+10]

The framework represents our systematic approach to the engineering of network resilience. At its core is a control loop comprising a number of conceptual components that realise the real-time aspect of the D^2R^2 + DR strategy, and consequently implement network resilience. Based on the resilience control loop, other necessary elements of our framework are derived, namely a resilience metrics framework, an approach to understanding high-impact challenges that a network may face, a policy-driven resilience management architecture, an incremental approach to challenge analysis that aims to build situational awareness, and multilevel information sharing and control mechanisms. The remainder of this section describes the resilience control loop, then motivates the need for these framework elements.

Based on the real-time component of the D^2R^2 + DR strategy, we have developed a *Resilience Control Loop*, depicted in Figure 2, in which a controller modulates the input to a system under control in order to steer the system and its output towards a desired reference value. The control loop forms the basis of our systematic approach to network resilience – it defines necessary components for network resilience from which the elements of our framework are derived. Its operation can be described using the following list; items correspond to the numbers shown in Figure 2:

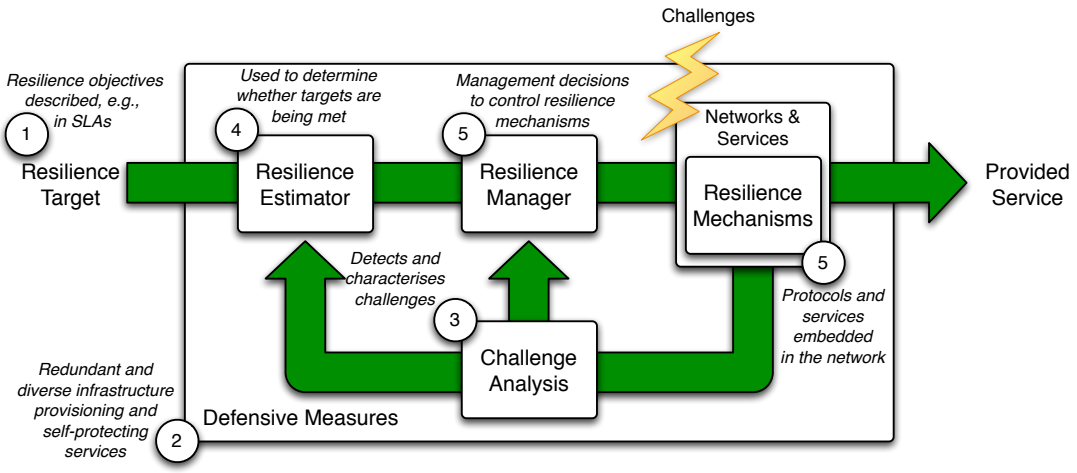


Figure 2: The Resilience Control Loop: derived from the real-time component of the $D^2R^2 +$ DR resilience strategy.

1. The reference value we aim to achieve is expressed in terms of a *Resilience Target*, which is described using resilience metrics. The resilience target reflects the requirements of end users, network operators, and service providers.

2. *Defensive Measures* need to be put *proactively* in place to alleviate the impact of *Challenges* on the network, and maintain its ability to realise the resilience target. A process for identifying the challenges that should be considered in this defence step of the strategy, *e.g.*, those happening more frequently and have high-impact, is necessary.

3. Despite the defensive measures, some challenges may cause the service delivered to users to deviate from the resilience target. These challenges could include unforeseen attacks or mis-configurations. *Challenge Analysis* components detect and characterise them using a variety of information sources.

4. Based on output from challenge analysis and the state of the network, a *Resilience Estimator* determines whether the resilience target is being met. This measure is based on resilience metrics, and is influenced by the effectiveness of defence and remediation mechanisms to respond to challenges.

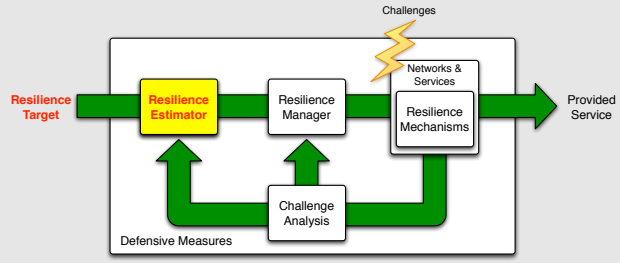
5. Output from the resilience estimator and challenge analysis is fed to a *Resilience Manager*. It is then its responsibility to control *Resilience Mechanisms* embedded in the network and service infrastructure, to preserve the target service provision level or ensure its graceful degradation. This adaptation is directed using *Resilience Knowledge*, not shown in Figure 2, such as policies and challenge models. We anticipate a cost of remediation in terms of a potentially unavoidable degradation in Quality of Service (QoS), which should not be incurred if the challenge abates. Consequently, the network should aim to recover to normal operation after a challenge has ceased.

The purpose of the background loop in the $D^2R^2 +$DR strategy is to improve the operation of the resilience control loop, such that it meets an idealised system operation. This improvement could be in response to market forces, leading to new resilience targets, new challenges, or sub-optimal performance. The *Diagnose* phase identifies areas for improvement, including defence, that are enacted through *Refinement*. In reality, and for the foreseeable future, we anticipate this outer-loop to be realised with human intervention.

1.2 Elements of a Resilience Framework

The resilience control loop motivates the need for the remaining elements of our framework: Evidently, to specify a resilience target and to be able to measure if it is being met, resilience metrics are required. Section 2 describes our metrics framework and a set of tools that can be used to evaluate the resilience of a network. In order to engineer networks and services that are resilient, it is necessary to understand the nature of the challenges they may face; we have produced a categorisation of challenges for network resilience that spans a number of thematic areas, and developed a risk assessment process that can be used to identify the probable high-impact challenges that may befall a given deployment. These two elements are discussed in Section 3. The resilience control loop embodies an automatic approach to detecting and mitigating challenges, which will require configuration and management. As mentioned earlier, this can be complicated, potentially leading to failures. We propose a policy-driven approach to resilience management, described in Section 4, which builds on a loosely coupled management architecture. Network and service adaption will involve the configuration of core functional network an service elements, *e.g.*, routers and applications, and additional resilience mechanisms that augment the network infrastructure. In Section 5, we summarise the resilience mechanisms that have been developed on the project, indicating the challenges they can be applied to and timescales in which they could be applied. A central part of the resilience control loop is challenge analysis – the primary objective of this stage is to build *situational awareness* in order to support decision making about how to remediate a challenge. We describe our incremental approach to challenge analysis, which is used to build situational awareness, in Section 6.

2 Resilience Metrics



In relation to our resilience control loop, in this section we address how to define a *resilience target* using various metrics. This is critical to many aspects of our framework, especially the resilience estimator, discussed in Section 4.

Developing a framework and toolsets for defining metrics and measuring network resilience is arguably one of the most important fundamental problems the project has addressed. Appropriate metrics are the basis for informed decision making. For example, with regard to understanding risk and the potential impact a challenge may have on a network and its services, understanding whether some capital investment for resilience is worthwhile (in terms of an improvement of resilience), and during a network's operation being able to determine whether a resilience target, expressed as a set of metrics described in an SLA, are being met – in all these cases, an understanding of resilience requirements, expressed in terms of metrics, is essential.

Understanding the importance of this problem, a survey was conducted by the European Network and Information Security Agency (ENISA) about the challenges and recommendations for resilience metrics [Eur10]. A set of challenges were identified, including a lack of standard practices, and knowledge and awareness of resilience metrics. In particular, one of the key challenges identified in the survey was:

"**The lack of a standardised framework**, even for the most basic resilience measurements. There are not that many frameworks available and none of them are globally accepted" [Eur10]

Correspondingly, recommendations included stimulating investment, facilitating and encouraging sharing of information and good practices, and the "...**development of automated tools** to help the deployment of resilience measurement (mainly data collection and data analysis)" [Eur10]. Activities conducted as part of the ResumeNet project to develop the resilience framework are targeted at addressing these two key challenges and recommendations, *i.e.*, a lack of standardised framework and automated toolsets for resilience metrics. Here we summarise our efforts with regard to these two aspects; further details are described by Doerr *et al.* in [DHH+11].

2.1 Resilience Metrics Framework

Ideally, we would like to express the resilience of a network using a single value, $\Re$, in the interval $[0, 1]$, but this is not easy to do because of the number of parameters that contribute to and measure resilience, and due to the multilevel aspects in which each level of resilience (*e.g.*, resilient topology) is the foundation for the next level up (*e.g.*, resilient routing). We have therefore developed a metrics framework, in which we model resilience as a two-dimensional state space. The vertical axis $\mathbb{P}$ is a measure of the service provided when the operational state $\mathbb{N}$ is challenged, as shown in Figure 3. Resilience is then modelled as the trajectory through

the state space as the network goes from delivering acceptable service under normal operations S_0 to degraded service S_c. Remediation improves service to S_r and recovery returns to the normal state S_0. We can measure resilience at a particular service level as the area under this trajectory $\mathbb{R}$.

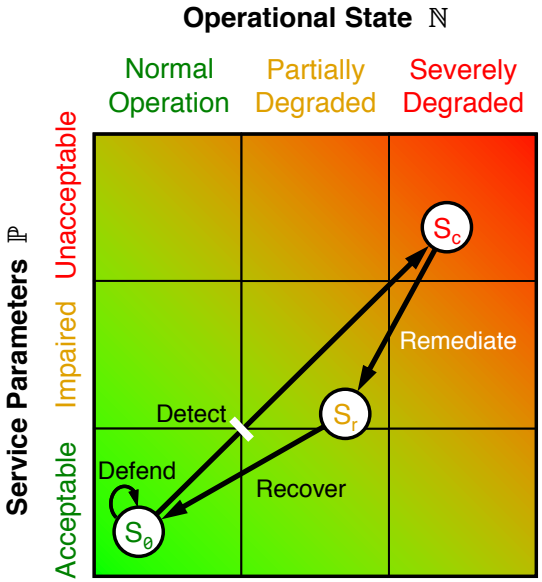


Figure 3: Resilience state space

In order to optimise the resilience of a network, it should be addressed at all levels, in the sense that each layer does the best it can, given practical constraints. These constraints are often based on the cost of resilience. Therefore, resilience must be analysed at each layer individually as well as for the network as a whole. For this purpose, the metrics framework supports multilevel resilience evaluation. Formally, resilience $\mathbb{R}_{ij}$ is defined at the boundary B_{ij} between any two adjacent layers L_i, L_j. Based on the formulation discussed earlier, let there be a set of k *operational metrics* $\mathbb{N} = \{N_1, N_2, \ldots, N_k\}$ that characterize the state of the network below the boundary B_{ij},. Similarly, let there be a set of l *service parameters* $\mathbb{P} = \{P_1, P_2, \ldots, P_l\}$ that characterize the service from layer i to layer j. Resilience $\mathbb{R}_{ij}$ at the boundary B_{ij} is then evaluated as the transition of the network through this state space. The goal is to derive the $\mathbb{R}_{ij}$ as a function of $\mathbb{N}$ and $\mathbb{P}$. In the simplest case $\mathbb{R}_{ij}$ is the area under the curve obtained by plotting $\mathbb{P}$ vs. $\mathbb{N}$ on a multivariate piecewise axis. In the multilevel analysis, as shown in Figure 4, the service parameters at the boundary B_{ij} become the operation metrics at boundary $B_{i+1,j+1}$. In other words, the service provided by a given layer becomes the operational state of the layer above, which has a new set of service parameters characterizing its service to the layer above.

This state space approach provides a way of representing and reasoning about multilevel resilience. One of the uses of the state space concept is to represent resilience classes, which offer a possible simplification for network and service providers when they wish to describe resilience in a Service Level Agreement.

A key aspect of our metrics framework is the notion of a *metric envelope*. For a given metric m, or the combination $\Re$ as shown in Figure 5, we map the trajectory of the best, average and worst case of the metric's behaviour in response to the increasing intensity of a challenge (greater values of k). For example, k could relate to the number of links that are

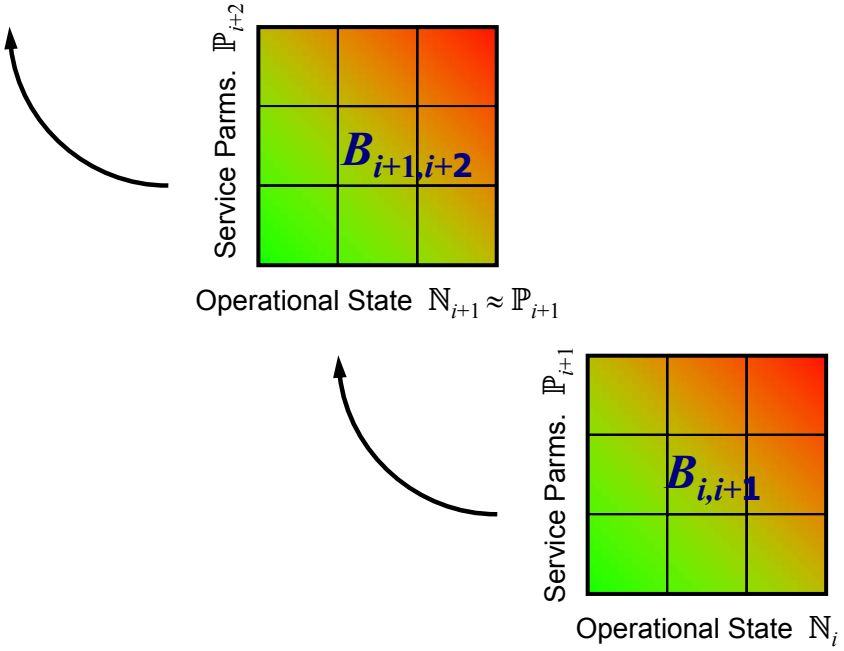


Figure 4: Resilience across multiple levels

removed for a given network topology, *e.g.*, caused by a targeted attack or an earthquake. To derive metric envelopes, using infinitely small time intervals, we dissect each challenge into atomic perturbations (k) that occur sequentially, so that between time t and $t+1$ only one component has failed and the difference between m_t and m_{t+1} exactly captures the impact of the component (link, node or node function) that failed at time t. To obtain a metric envelope for a given set of perturbations that occur at time t, depending on the metric under consideration and how it is evaluated, we exhaustively derive metric values for every combination of perturbations that could occur at a given time, *e.g.*, all the possible permutations of k link failures at t. In this way, for example, we can understand the impact a targeted attack could have, rather than random failure.

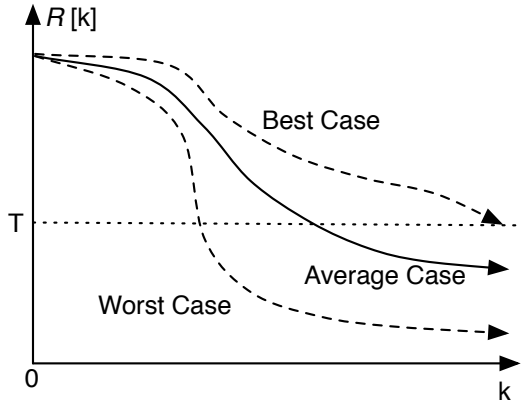


Figure 5: The metric envelope concept, reprinted from [DHH+11]

One of the aims of this approach is to plot envelopes for metrics at multiple layers of the protocol stack, in order to understand their relationship (*e.g.*, is their behaviour correlated

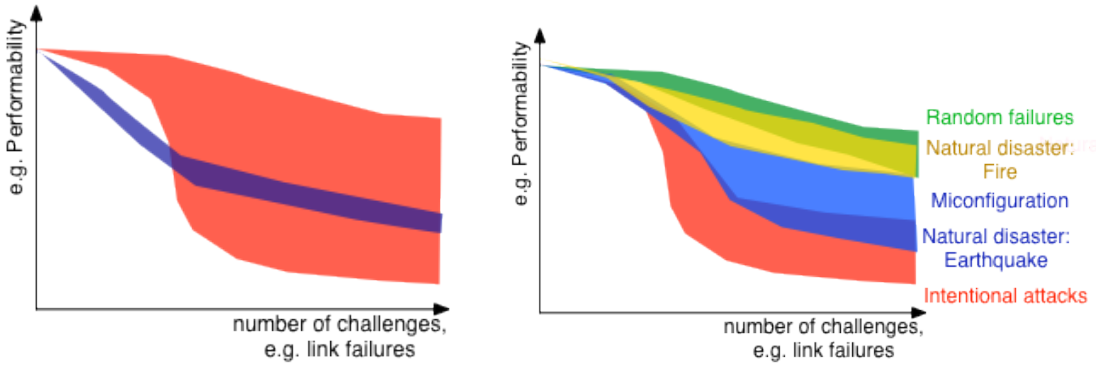
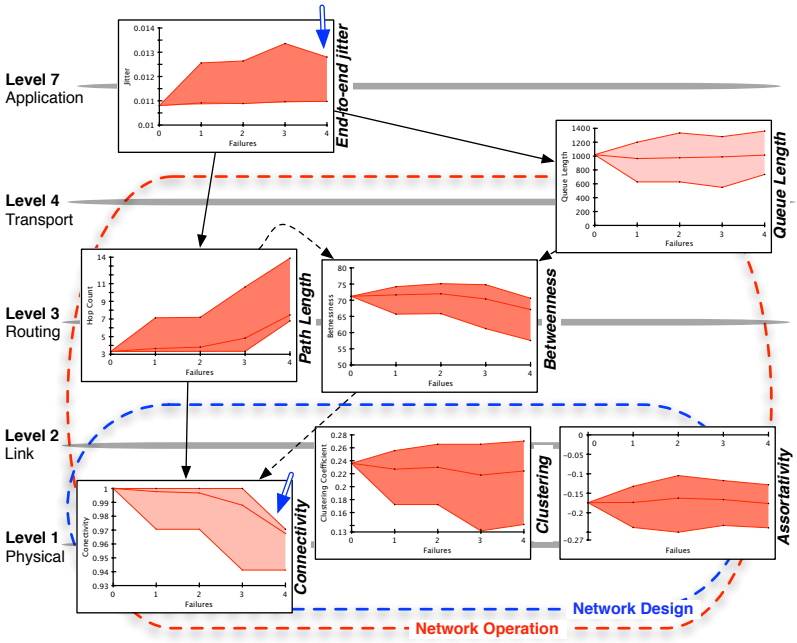


Figure 6: Applications of metric envelopes: (a) comparing different network deployments; the blue and red envelopes depict different network's response to a challenge; and (b) the behaviour of a network in response to different challenges.

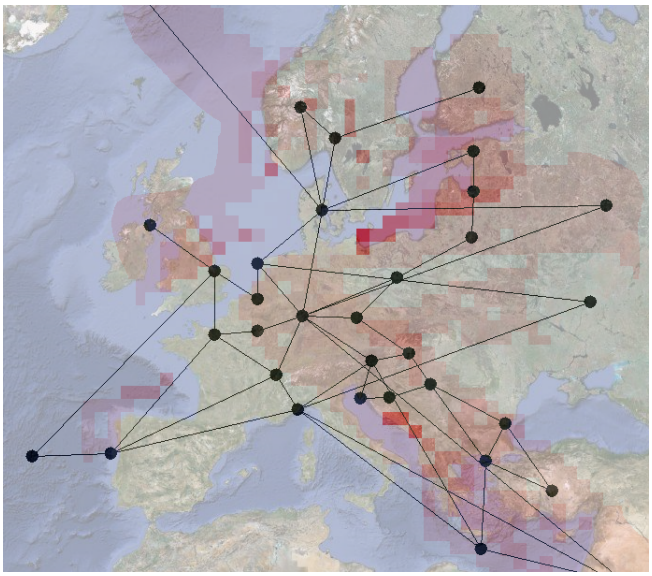
or otherwise) and determine suitable metrics that should contribute to the $\Re$ value (and are consequently key metrics for understanding the resilience of a network). Understanding the relationship between metrics at various layers of the protocol stack is explored by Doerr and Hernandez in [DMH10]. Further uses of the metric envelope concept are shown in Figure 6. For example, they could be used to compare the resilience characteristics of two network configurations, in response to a challenge. Figure 6(a) depicts possible (performability) metric envelopes associated with two network deployments – the blue and red envelopes – given challenges, *e.g.*, a targeted attack or random outages, that cause link failures. It can be seen in the best case, the red network performs better than the blue. However, the size of the metric envelope for the blue network is much smaller. Given these characteristics, one could suggest the blue network is suitable for supporting critical services, *e.g.*, in a military setting in which deterministic behaviour is desirable, and the red network would suit a best-effort context, in which high variance of performability can be tolerated in return for a higher best case offering. Figure 6(b) depicts a set of hypothetical metric envelopes showing the effect of different challenges. With this information, it should be possible to determine the impact a potential challenge has on a given network deployment. We use this form of information as part of our risk assessment process, discussed in Section 3, to determine the high-impact challenges a network may face. With this knowledge, appropriate resilience mechanisms can be selected to be protect a network against challenges. The metric envelopes can be generated by the Graph Explorer tool, developed as part of the project, which we discuss next.

2.2 Resilience Metrics Tools

We have developed a number of tools for evaluating network resilience. For example, we use MATLAB or ns-3 simulation models to measure the service at each level and plot the results under various challenges and attacks, as in Figure 3, where each axis is an objective function of the relevant parameters [ScH+11]. Furthermore, we have developed the *Graph Explorer* tool [DMH10] that takes as input a network topology and associated traffic matrix, a description of challenges, and a set of metrics to be evaluated. The result of the analysis is a series of plots that show the *metric envelope* values $(\mathrm{m_i}(\min), \mathrm{m_i}(\max))$ for each specified metric $\mathrm{m_i}$, and topology maps indicating the resilience across network regions.



(a)



(b)

Figure 7: Example output from the Graph Explorer, developed in the ResumeNet project: (a) plots showing the relationship between metrics at various layers in response to link failures on the GÉANT2 topology; and (b) a heat map showing vulnerable regions of the topology with respect to a given set of metrics. Reprinted from [DMH10]

Figure 7 shows an example of the resilience of the European academic network GÉANT2 to link failures. The set of plots in Figure 7(a) show metric envelopes at different protocol levels – the aim is to understand how jitter responds in comparison with metrics at other levels, such as queue length and connectivity. Surprisingly, jitter is not clearly related to queue length and a monotonic increase in path length does not yield a similar increase in queue length for all scenarios of link failures. In fact, the fourth link failure disconnects a region of the

network; whereas up to three failures, the heavy use of a certain path resulted in increasing queue lengths and jitter. The partition increases path length, because route lengths are set to infinity, and decreases connectivity, which is accompanied by a reduction in jitter, shown with the blue arrows in Figure 7(a). The topology map in Figure 7(b) highlights the vulnerability of regions of GÉANT2 with a heat map, which can be used by network planners to indicate where invest should be made to improve the resilience of their network.

2.3 Summary

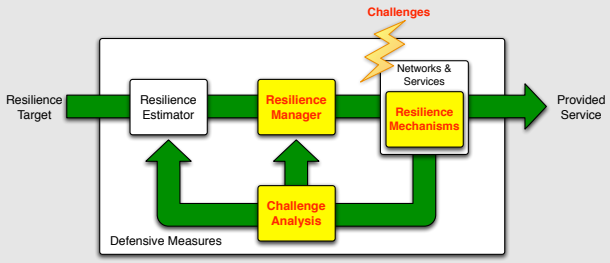
Our framework for resilience metrics – the multi-level two-dimensional state space and the use of metric envelopes – can be used to understand the resilience of networks to a broad range of challenges, such as mis-configurations, faults and attacks. We propose our metrics framework as an important step toward addressing the challenge, identified by ENISA, regarding a lack of a standardised approaches to measuring and evaluating the resilience of networks. The ability to evaluate a given network's resilience to a specific challenge is limited by the capability of the tools to create complex challenge scenarios – this is an area for further work, in which our effort should be focused on modelling pertinent high-impact challenges.

Specifying resilience requirements using multilevel metrics and making decisions related to the $D^2R^2 + DR$ strategy based on multilevel measurements is likely to be complicated. The use of *resilience classes* may be able to help reduce the complexity involved – if we can decide on a small number of classes that each represents a significant cluster of services and the responses that each would require whenever a challenge occurs. This approach would reduce the number of inter-relationships between services that have to be taken into account when specifying, decomposing or computing the $\Re$-value. Previous work has been done in this area. Autenrieth and Kirstädter propose an architecture for resilience differentiated QoS, using four classes of resilience that map onto different network level resilience schemes – protection, restoration, rerouting and preemption [AK02]. Furthermore, schemes have been proposed for differentiated Quality of Protection in optical Wavelength-Division Multiplexing (WDM) networks [SS04a, KG05]. However, they largely assume challenges to operation are random node and link failures, and they take a bottom-up, *i.e.*, mechanism-centric, approach to defining classes, rather than the top-down – service-centric – approach we propose. This may result in unsuitable resilience classes that do not reflect common service requirements, and are not realisable given the nature of the challenges a network may face. We therefore propose future work should try to identify a suitable set of resilience classes that reflect clusters of service requirements, in order to improve the likelihood of the adoption of our metrics framework by practitioners.

Key deliverable for resilience metrics

C. Doerr, J. M. Hernandez, R. Holz, D. Hutchison, A. Smeureanu, P. Smith, James P. G. Sterbenz, and P. Van Mieghem. Defining metrics for resilient networking (Final). ResumeNet Project Deliverable, September 2011

3 Understanding Challenges



Understanding challenges is central to the resilience control loop. It directs the choice of *resilience mechanisms* that should be deployed and the nature of their *management*. Furthermore, *challenge analysis* should be targeted at identifying the likely high impact challenges.

Knowledge about the nature of challenges that may affect a network deployment is important to ensure resilience. With this knowledge, the correct resilience mechanisms can be employed for likely challenges, improving the probability of ensuring acceptable service in their presence. To this end, the ResumeNet project has contributed two items that can be used to *understand challenges*: *(1)* an informal categorisation of the challenges that affect network deployments, with notable examples shown, which can be used by a resilience engineer to reason about the types of challenge that exist; and *(2)* a risk assessment process that can be used to determine the probable high-impact challenges their network may face. We discuss these two items in this section.

3.1 Challenge Categories

Learning from past failures is essential to build better networks in the future. To systematically group challenges we propose seven categories: *(1)* component faults; *(2)* hardware destruction; *(3)* communication environment related; *(4)* human mistakes *(5)* malicious attacks; *(6)* unusual but legitimate demand for service; and *(7)* failure of a provider service. The first four challenge categories are a consequence of being part of the real world, *e.g.*, deterioration, destruction, and physical channel characteristics. Challenge categories five and six originate in networked cyberspace, *e.g.*, cyberspace attacks and non-foreseeable system interaction. The last challenge, which might be also seen as overlapping with all previous six categories, results from the composition of networks from subsystems, where failures of a subsystem influences others and the network as a whole. These seven challenge categories are presented in detail, accompanied by examples.

Component Faults Internal errors occur during 'normal' operation of the system independent of outside events. They can be caused by software bugs or the deterioration of hardware, for example. In short, these are errors that are brought about by faults in components of the system, are of random nature, and are unintentional.

Bugs in soft- or hardware in communication systems are unavoidable. For example, a bug of Cisco's IOS causes a BGP session reset if the length of the AS path exceeds 255 after AS-path prepending [Pep09]. This bug was triggered by a bug from another router vendor not handling configuration parameters correctly, especially missing bound checks [Zmi09]. The result was a tenfold increase in planetary routing instability for an hour and an increase of affected prefixes from 0.45% to 4.76%.

Hardware destruction This category summarizes all challenges where destruction of hardware causes errors or failures. These can be either due to natural causes (*e.g.*, tsunamis, earthquakes or hurricanes) or man-made (*e.g.*, terrorist attacks, fires or cable-cuts). Challenges of this class can be intentional or unintentional, as well as targeted or random.

The Hinsdale Office Fire [Tow88] is an example where destruction of hardware caused significant outages of communication services. Despite having deployed redundant systems for fail-over, a system failure could not be prevented as both the primary and secondary system were physically co-located in the same office building which was destroyed by a fire.

Communication Environment related All challenges that are inherent in the communication environment due to:

- weak, asymmetric, and episodic connectivity of wireless channels
- high-mobility of nodes and subnetworks
- unpredictably long delay paths either due to length (*e.g.*, satellite) or as a result of episodic connectivity

are gathered in the communication environment category. Challenges in this category are of random nature and are unintentional.

An (extreme) example of this category is an ad hoc airborne network made up from supersonic jet aircrafts [JPS08]. The high mobility of the jets and drones challenges end-to-end communication since the network nodes are within communication range for a short time-period only. Special designed network and transport protocols are required to establish and maintain communication during aircraft operation. Similar challenges are presented by more conventional ad hoc networks; delay-tolerant, opportunistic, networks have largely emerged as a response to these challenges.

Human Mistakes Human mistakes describe non-malicious errors that are made by people interacting with the system, such as device misconfigurations or policy breaches. These can become more pernicious if the parties involved try to cover up their mistakes. Challenges in this category are unintentional.

Configuration mistakes in firewalls, automatic configuration systems (ACS) or end hosts often lead to degraded network service or prevent any communication. But misconfiguration can also have a larger impact. One example constitute the erroneous BGP advertisements of Pakistan Telecom's upstream providers, which resulted in a hijack of YouTube's Web presence [NCC08].

Malicious Attacks Malicious attacks from intelligent adversaries pose a threat to system performance and form a group of challenges to networked systems. These challenges are targeted and intentional in nature.

Cyberspace attacks can be specifically targeted at critical points of the communication system, e.g. Denial of Service attacks (DoS attacks), or be a resource exhaustion attack against a victim including collateral damage to the communication infrastructure, *e.g.*, Distributed Denial of Service attacks (DDoS) attacks.

Unusual but Legitimate Demand for Service A non-malicious request for service that is greater (or different along some other dimension) than what is provisioned for, for example, flash crowd events. Challenges in this category are unintentional.

Neither the PSTN nor the Internet were designed to cope with the amount of traffic experienced after the 9/11 attacks. Both communication systems experienced severe local degradation of service [LeF01].

Failure of a Provider Service Due to the composition of complex system from multiple services any aforementioned challenge can cause cascade effects. The failure of a provider service must be treated as challenge to the consumer services, which depend on the correct behaviour of the provider service. Service dependencies can be vertical, *e.g.*, using a lower layer service, as well as horizontal, *e.g.*, client-server based or peer service. Interoperability faults fall into this category, too. This category also captures failures of provider services due to an unidentifiable challenge to this provider service.

Packet loss on a wireless link reduces the effective bandwidth of IP over this link and a degradation of the transport service in case of TCP.

3.2 A Risk Management Process for Network Resilience

We now describe the risk assessment process that can be followed to determine the probable high-impact challenges a network will face. The process is outlined in Figure 8. For each step in the risk assessment process, we outline how the output can be used to inform the design of resilient networks and services. The description presented here is an abridged version of that presented by Schöller *et al.* in [SSH11], wherein an example scenario is worked through for a wireless mesh network deployment [IBPR08].

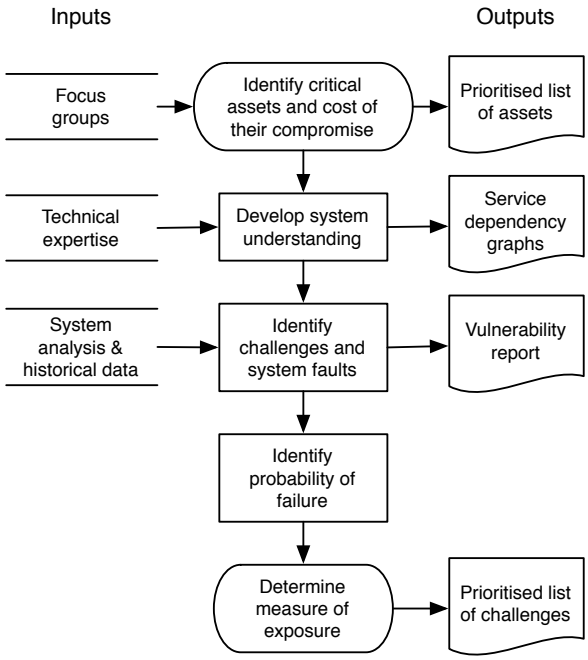


Figure 8: A risk assessment process for resilience, including example inputs and outputs

Step 1: Identify critical assets and costs of their compromise

The first step is to identify the *critical* assets of the stakeholders associated with a network. Assets can be *physical*, *logical*, or *informational*. What is considered a critical asset is con-

text specific, where the context is defined by the requirements of the organisation using the network, and its application. Physical assets can be damaged or manipulated; logical assets, implemented as services, can be attacked or used in a way that was not designed for; and informational assets can be disclosed, suppressed or manipulated. Example stakeholders include, customers, network providers and governments. Assets may be compromised in a number of ways. We build on our resilience metrics framework, discussed in Section 2, to evaluate the compromise of service-based assets. Recall that when a service is not challenged, it is considered in an *acceptable* state. Challenges can lead to a service becoming *impaired* or *unacceptable* – illustrating the non-binary loss of assets in network resilience. The trajectory a service takes between these states depends on the mechanisms in place to resist a challenge and how severe it is. The level of degradation, *e.g.*, impaired and unacceptable, and the duration the asset is in that state will have different costs – in most deployments, extended periods of impairment, due to a DDoS attack, will have a higher cost than short disruptions, due to a switch-over to a backup configuration. Measuring the cost of a compromise can be done qualitatively (*e.g.*, mild, moderate, severe, or catastrophic) or quantitatively (*e.g.*, potential financial loss or cost of replacement of equipment). For example, a DDoS attack could cause a degradation of service on an ISP's network, leading to SLAs being broken, resulting in reimbursements to customers. The same incident could lead to a loss of reputation for the ISP, a qualitative cost. At this step in the process, we consider the different modes of compromise associated with an asset, and their impact on the stakeholders; a prioritised list of assets results.

Step 1: Output The assets identified feed into Step 2 of our process, wherein we understand their implementation; the asset compromise values feed into Step 5 to calculate a measure of exposure. From an implementation perspective, the compromise values can be used to define *protection priorities* for the services realising assets. In case of insufficient resources to protect all assets from a variety of simultaneously ongoing challenges, the network should strive to protect those with the highest value in the face of failure. Resilience management strategies that honour protection priorities can be described in policies, as part of a general approach to resilience that capitalises on policy-based management frameworks [SSFA+10].

Step 2: Develop network understanding

Modern network engineering approaches decompose the provisioning of assets into multiple sub-systems and services. Naturally, these are re-used in the design and implementation of multiple assets. This inherently implies that multiple assets can be (partly) degraded if a common sub-system (vertical service dependency) or peer service (horizontal service dependency) is affected by a challenge. In this phase, networking and systems experts develop an understanding of the used sub-systems and services, as well as their interdependencies, leading to a service dependency graph, such as the example shown in Figure 9.

Step 2: Output The services realising the various assets will be examined in Step 3 for faults and vulnerabilities. Moreover, the resulting service dependency graph can be used to build generic remediation strategies that strive to ensure acceptable service provisioning in the face of unknown challenges and system faults – *i.e.*, those not identified via this process. These generic strategies are not optimised for a specific challenge, but change the network configuration in a way that challenged parts are isolated from unaffected parts, *e.g.*, using system and network virtualisation techniques, or whole sub-systems are exchanged with alternative components,

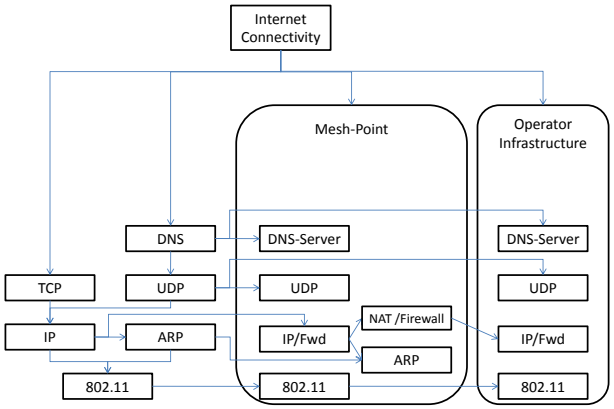


Figure 9: An example Internet connectivity service dependency graph for a wireless mesh network [SSH11]

e.g., provided by a network composition framework as proposed by Kappler *et al.* [KMP+05]. Another way to use dependency graphs has been introduced by Katzela *et al.* [KS95], who utilise such graphs for fault localisation.

Step 3: Identify challenges and system faults

The set of challenges that could affect a network is large. Here, the aim is to identify those challenges that can trigger the faults in services and sub-systems, which will most likely occur in the given deployment scenario. Faults with respect to resilience are wide-ranging and include design faults, inappropriate business-processes, and vulnerabilities to attacks. In addition, they may also include inappropriate use of defensive and remediation mechanisms.

Challenges trigger faults, causing erroneous behaviour of a service and, if not isolated, for it to fail. This causal relationship is depicted in Figure 10. Therefore, a specific service is threatened by challenges targeting the service itself, and, in addition, challenges causing service failures of dependent service instances. We use this reasoning to manage the wide-range of faults that may exist in a network, and focus on identifying those in the services (and their dependents) that could be triggered by the most likely challenges that have been identified.

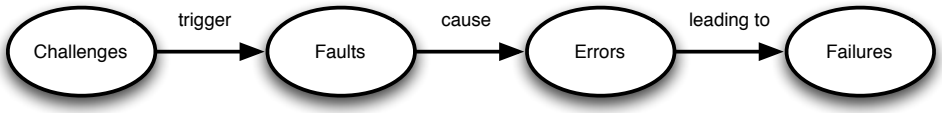


Figure 10: The *potential* causal relationship between challenges, faults, errors and failures.

System analysis and historical data can be used to derive the likely challenges and associated faults. Example approaches to system analysis include fault-tree analysis [Ves87] and event-tree analysis [Coo07], or threat-modelling techniques, such as STRIDE [HLOS06]. Security advisories provided by CERT[1] or SANS[2] can be used to identify past and on-going threats, for example. To the best of our knowledge, there are no advisory services that cover the range of challenges considered for network resilience. We see this as a key shortcoming to address, and suggest that multi-national organisations, such as the European Network and Information

[1] http://www.cert.org
[2] http://www.sans.org

Security Agency (ENISA)[3], could play an important role leading the development of such an advisory.Do

Step 3: Output The identified probabilities of challenge occurrence will be used in Step 5 to calculate the measure of exposure. In addition, the identified challenge-fault pairs can be documented in a *vulnerability report*. This report will help a network engineer choose the best defensive or remediation mechanisms to protect a network from the high-impact challenges.

Step 4: Determine the likelihood of service degradation and its impact

This phase is used to determine the likelihood of a challenge leading to a failure, which is influenced by the nature of a challenge and properties of a network. Such properties include known faults, the mechanisms that are in place to mitigate challenges, and the dependencies of a service on others. Adding mechanisms to defend against the high impact challenges or to remediate their impact on the system after this risk assessment process decreases the likelihood of a failure occurring.

To acquire justifiable probability values, analytical models and data gathered from similar deployments can be used. Analytical approaches, like the Graph Explorer described in Section 2, can provide probability density functions that describe how challenges affect various service metrics. From the anticipated perturbations of these metrics, probability values for the level of service degradation can be derived. However, following such an analytical approach might be infeasible for some challenges. For these, probability values need to be estimated during the design phase. As pointed out before, refining these values during network operation by correlating service failures with sensor data collected from the deployment is necessary in this case to avoid over- and underestimation of the impact of these challenges.

[3] http://www.enisa.europa.eu/

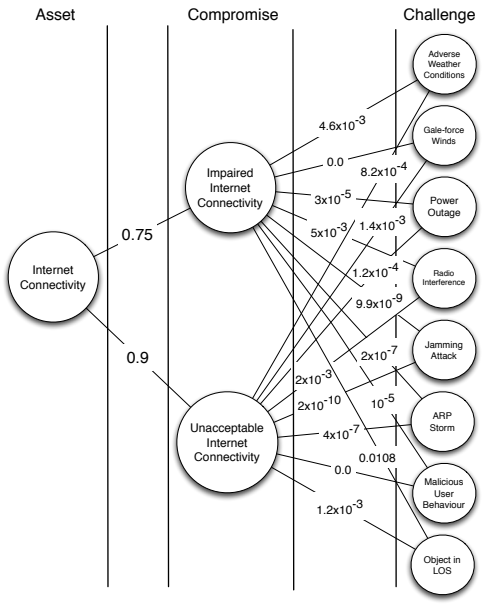


Figure 11: An example exposure graph for the wireless mesh network use-case study [SSH11]

Step 4: Output The probability of a challenge leading to a service failure will directly feed into Step 5 of this process. Together with the output of Step 1 and Step 3 the measure of exposure can be calculated.

Step 5: Determine a measure of exposure

In Step 2 of this process the cost of a particular mode of compromise for an asset will have been identified. Based on this, we would like to compute a numeric value of exposure – a measure to assess the impact a challenge has on network assets. We determine a measure of exposure using the Equation 1; $\mathrm{challenge\_prob}$ is the probability that a challenge will occur (from Step 4), $\mathrm{compromise\_prob}$ is the probability that a compromise will occur to an asset, which is based on the likelihood of a failure (from Step 5), and impact is the cost associated with an asset being compromised, from Step 1. This information can be depicted as an exposure graph, shown in Figure 11, wherein the values shown on the left relate to the measure of impact and those on the right show the product of $\mathrm{challenge\_prob}$ and $\mathrm{compromise\_prob}$. It is clear that using this strategy to determine a measure of risk will yield similar values for high-probability, low-impact events and low-probability, high-impact events. Depending on which scenarios the network engineer would like to account for, exposure measures could be filtered out.

$$\mathrm{exposure} = (\mathrm{challenge\_prob} \times \mathrm{compromise\_prob}) \times \mathrm{impact} \tag{1}$$

Step 5: Output The last stage of our process outputs a list of challenge-fault pairs ordered by their exposure. The network engineer will select and implement defensive or remediation mechanisms for the high impact challenges to maximise network resilience.

3.3 Summary

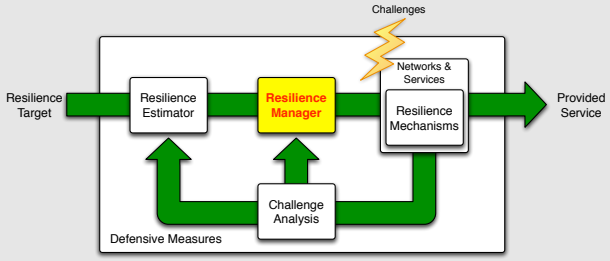
It is of vital importance to understand the challenges that are most likely to have the highest impact to a network deployment. Without this information, appropriate defensive measures cannot be put in place. Furthermore, knowing the likely high-impact challenges is useful for challenge analysis and remediation, too. In this section, we have described a set of challenge classes that can be used to consider the forms of challenge that may affect a network's resilience. This knowledge can be applied when using the risk assessment process we have developed. Future work in this area should investigate determining challenge occurrence probabilities and quantifying their impact. The use of appropriate modelling and simulation approaches can be used to help determine meaningful values for these important inputs to the risk assessment process.

Key deliverable for understanding challenges

M. Schöller and P. Smith. Understanding challenges and their impact on network resilience. ResumeNet Project Deliverable, October 2009

C. Doerr, J. M. Hernandez, R. Holz, D. Hutchison, A. Smeureanu, P. Smith, James P. G. Sterbenz, and P. Van Mieghem. Defining metrics for resilient networking (Final). ResumeNet Project Deliverable, September 2011

4 Resilience Management



In this section, we discuss the *resilience manager*, which is used to steer the *provided service*, using *resilience mechanisms* embedded in the network, toward the *resilience target*. It does this using information from *challenge analysis* and the *resilience estimator*; the latter is described here.

One of the issues addressed by the ResumeNet project relates to the management of networks, services and mechanisms for resilience. From the outset, the project proposed to build on existing work on policy-based network management frameworks in order to orchestrate resilience mechanisms that realise a *resilience strategy* – a configuration of mechanisms, which address a challenge. Consequently, we conducted a survey of policy-based management frameworks and their suitability for resilience; we summarise our findings here. In addition, the project has developed a *resilience management* architecture that can be used to realise our resilience control loop, shown in Figure 2. Similarly, the main components of the architecture that relate to decision making are summarised. Finally, a realisation of the architecture using the Or-BAC policy framework is briefly discussed.

4.1 Policy-based Resilience Management

Challenges to the operation of a network can occur rapidly and with little warning, requiring a fast response in order to maintain acceptable service levels. This dynamic behaviour is embodied in the $D^2R^2 + DR$ strategy adopted by the ResumeNet project – namely the online detection of challenges, and real-time remediation and recovery. In many cases, this adaptation needs to be undertaken in real-time by the network itself, *e.g.*, in response to a DDoS attack or device failure. However, the invocation of remediation mechanisms may require the intervention of a suitably authorised human operator. In some cases, to mitigate a challenge, complex multi-phase strategies are required, which combine various monitoring and detection mechanisms that influence the behaviour of remediation mechanisms [YFSF+11]. Moreover, down-time of resilience mechanisms in order to (re-)configure such strategies is undesirable, given the unpredictable nature of challenges – down-time leaves the network exposed.

To address these issues, the ResumeNet project advocates the use of policy-based network management techniques for the configuration of resilience strategies. These techniques allow descriptions of real-time adaptation strategies, including points when human intervention is required, which are separate from the implementation of the mechanisms that realise the strategy. This separation allows changes to be made to strategies without the need to take resilience mechanisms off-line [D+01]. In short, two forms of policy are supported: *authorisation* (or *access control*) and *obligation* policies. Authorisation policies describe prohibition and permission rules for actors, *e.g.*, resilience mechanisms, associated with the network. Obliga-

tion policies describe what actions should be taken by actors in response to events, *e.g.*, the detection of a challenge, given certain conditions exist, such as the time of day. Combinations of these policies are used to define a resilience strategy.

In Deliverable D1.3b [SSSF+11], we document our assessment of the suitability of three significant policy-based management frameworks for realising network resilience strategies: XACML, Ponder2 and Or-BAC. Ponder2 [D+01] is predominately a policy framework that implements obligation policies, which can support access control rules for the actors (resilience mechanisms) it controls. In contrast, XACML [Lis10] and Or-BAC [KBM+03] are access control policy frameworks to which obligation policy support is being added. We suggest that there is not a single policy framework that could be used for all network deployments and applied to a comprehensive set of resilience problems, and that while many of our requirements for resilience are met by the frameworks, none meet them all. Our main findings regarding each policy framework are summarised in Table 1.

Table 1: Summary of findings of policy framework analysis presented in Deliverable D1.3b [SSSF+11]

Framework	Summary
XACML	Access-control based framework that makes use of XML to describe policies. Obligation policy support currently being developed. Some conflict resolution support built in. XOML, the obligation support component of XACML, lacks support for an *obligation subject* which would simplify integration significantly.
Ponder2	Intended to be a lightweight so it can execute on resource constrained devices. Consequently, lacks some of the more sophisticated support functionality, *e.g.*, conflict resolution. Introduces the concept of hierarchical domains of control, which can make the process of policy definition more tractable. Main focus is on obligation policy support.
Or-BAC	Largely targeted at realising security policies by defining various contexts, such as nominal and threat contexts, which are invoked when challenges are detected. Enables the expression of an abstract global policy view, which is made concrete, *i.e.*, decision and enforcement points are configured, at run-time.

In addition to the findings regarding specific policy-based management frameworks, we have identified a number of items for research in the area of policy-based management and its application to network resilience. (Some of these concerns have been investigated in other contexts, also.)

Perhaps one of the most challenging tasks is to take a high-level resilience target, expressed using various metrics as discussed in Section 2, and determine a network configuration that will realise the target. From a technical perspective, this will have two aspects: selection of resilience mechanisms to deploy, *e.g.*, defensive measures, and a configuration that organises them, *e.g.*, into a multi-stage resilience strategy as discussed earlier. A way to approach the latter problem is to recursively elaborate on goals by iteratively transforming high-level goals into more concrete ones – this technique has been used to determine implementable policies that realise Quality of Service (QoS) requirements [BLMR04, BLR+06]. For resilience, this task is more difficult as we must generalise these concepts to address not only network-related QoS requirements, but co-ordinate multi-level resilience aspects of a network, both at the *network* and *service* levels.

Another key problem, which requires future work, relates to identifying potentially conflicting behaviours that are expressed in policies. We envisage an abundance of policies controlling various aspects of a network that address a number of concerns, including resilience issues. Conflicting behaviours may be expressed between those policies related to resilience and other issues, such as organisational security requirements. Furthermore, considering resilience issues alone, we foresee two forms of conflicting behaviour occurring: *(1) vertically* between network and service-level resilience mechanisms; and *(2) horizontally* across stages of the D^2R^2 + DR strategy [SSSF+11]. Existing work has categorised the different forms of conflicting behaviour that could occur [AGLL05], but lacks domain-specific knowledge, which is required to identify that a particular network-level mechanism, *e.g.*, traffic shaping, could inhibit the operation of service-level mechanism, *e.g.*, virtual machine replication. Developing a systematic approach to identifying this domain-specific knowledge is an area for future work.

As a starting point for evaluating resilience strategies – and identifying the domain-specific resilience knowledge just mentioned – the project has developed a policy-based resilience simulator, which couples a network simulator (OMNeT++ [OMN]) with the Ponder2 policy management framework [SFSM11]. In short, resilience mechanisms are implemented as modules in the OMNeT++ simulator; they generate events, *e.g.*, the detection of an anomaly, that are sent to Ponder2, triggering policies and causing actions to be invoked back in the simulator, *e.g.*, traffic shaping. This simulation environment was used in our work to evaluate our multi-stage approach to challenge analysis and remediation [YFSF+11, SFM+10], whereby a number of resilience mechanisms were implemented, which triggered policies that orchestrated a multi-stage approach to detect and remediate high-traffic volume network challenges, such as a DDoS attack.

4.2 Resilience Management Architecture

The ResumeNet project has developed a resilience management architecture, depicted in Figure 12. This architecture can be used as a basis for implementing the resilience control loop shown in Figure 2, which can be used to realise dynamic adaptation of a network and associated resilience mechanisms. Here, we briefly summarise its components; for a detailed description of the architecture, see Deliverable D2.3b [DSS+10].

The central component of the architecture is the *Network Resilience Manager*, which makes use of a policy-based decision engine, realised using one of the policy management frameworks discussed earlier, to activate management actions on the *Managed Entities* embedded in the network. Managed entities can include resilience mechanisms at both the network and service level, as well as functionality exposed by existing network and service elements, such as routers and servers. To assist decision making by the resilience manager, we propose to use *Consultants* that, for example, pro-actively collect state about the network and resilience mechanisms. An example consultant developed by the project is the Rope Ladder Routing (RLR) mechanism [LSZB10] – a multi-path forwarding strategy that builds a primary and secondary path between endpoints with "rungs" between paths. Information derived by the RLR mechanism can be used when device failure is detected to inform path selection. The resilience estimator, which is used to determine whether a resilience target is being met (depicted in Figure 2), is also implemented as part of the resilience manager.

Ongoing work is investigating a design for the resilience estimator, which is shown in Figure 13. As discussed in Section 2, a resilience target describes the behaviour of a set of metrics $\{\mathrm{m}_0 \ldots \mathrm{m}_{\mathrm{n}-1}\}$ at various levels of the protocol stack, *e.g.*, from average node degree

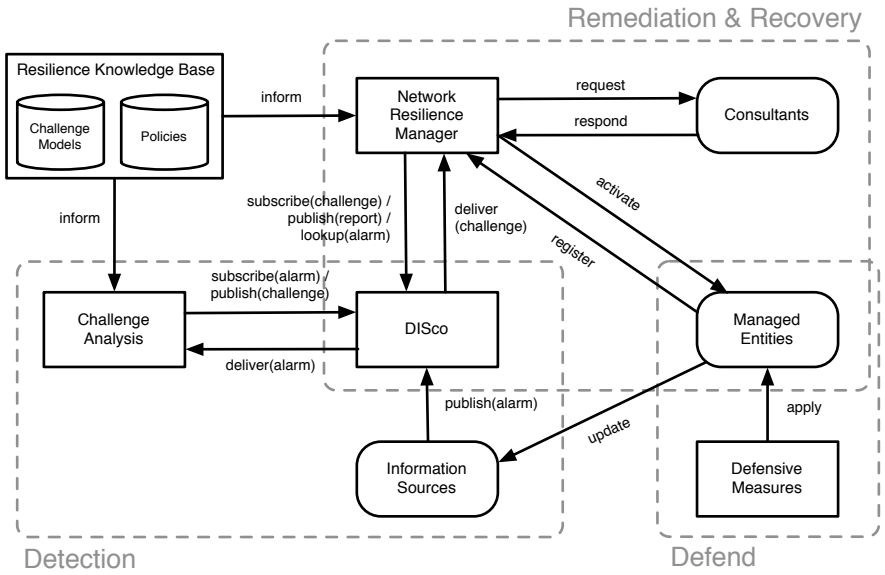


Figure 12: A dynamic adaptation architecture that realises the resilience control loop.

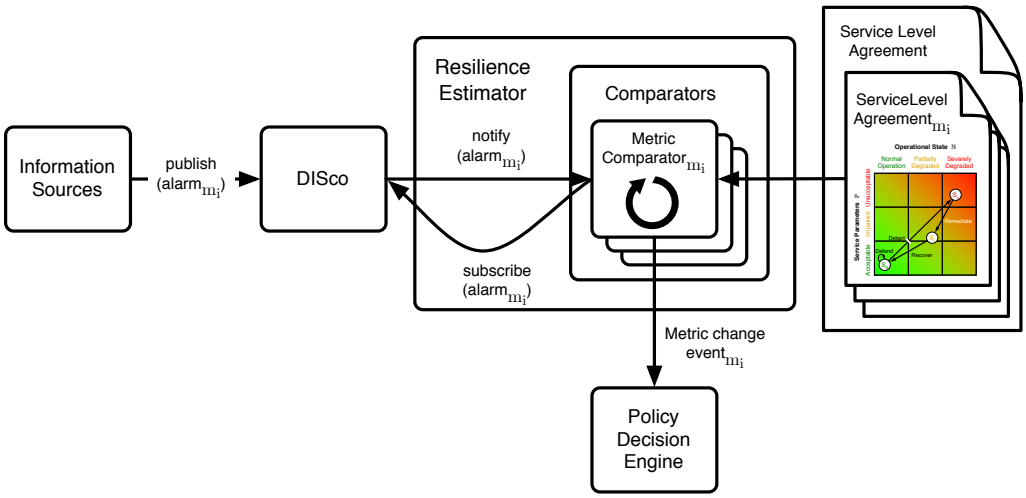


Figure 13: The resilience estimator, which forms part of the network resilience manager.

to end-to-end delay, that constitute a desired $\Re$ value. In reality, this information is expressed in a Service Level Agreement (SLA). For each metric m_i, we propose to have a corresponding *Metric Comparator* that takes measurement information collected from distributed monitoring mechanisms – *Information Sources* in Figure 12 – and the behaviour described in the SLA for m_i, and determines whether a *Metric Change Event* should be generated. A metric comparator can use transitions between service states – acceptable, impaired and unacceptable – from our metrics framework to generate metric change events. Furthermore, temporal and spatial constraints could be described, *e.g.*, generate an event after one minute of degraded performance for a given subnetwork. We envisage the combination of information derived from *Challenge Analysis* (described in Section 6) about the existence and nature of challenges, and the resilience estimator will drive the selection of policies by the resilience manager, which configures the managed entities in the network.

4.3 Architecture Deployment

We envisage the management architecture proposed here, which realises the resilience control loop, could be deployed in a number of ways. For example, an ISP could realise the architecture within its own domain *e.g.*, with a centralised resilience manager that controls its network elements (managed entities). Furthermore, a similar deployment style could be adopted on a university or corporate network, ensuring resilience for the local area networks on the campus. For other types of network deployments, for example, in an opportunistic or mobile ad-hoc network, a selection of the management architecture functionality could be realised on a single node, potentially calling on the capabilities of other devices to implement some resilience functionality. At the service level, applications operate on an end-to-end basis, and their implementation can span multiple administrative domains, in which case functionality needs to be distributed across these multiple domains.

To facilitate this multi-domain interaction *e.g.*, between nodes in a MANET or components of a service that span multiple administrative domains, we propose to build on the concept of *resilience patterns* [SF09] – reusable descriptions of the configuration of resilience mechanisms that address a well-known challenge. Resilience patterns define the relationship mechanisms have, *e.g.*, peer-to-peer or client-server, the policies they should exchange and the interfaces they expose. Explicitly defining the nature of the relationships between components of the management architecture in this way could expedite multi-domain interactions. This is part of our ongoing work; an initial proposal regarding the application of resilience patterns to the configuration of components of the management architecture is discussed by Schaeffer-Filho *et al.* in [SFSM+12].

4.4 Resilience Management using Or-BAC

Within the project we have explored approaches to realising aspects of the resilience management architecture, using contextual security policies (*i.e.*, an access control policy-based approach) that makes use of the Or-BAC formalism. The use of contextual security policies for network resilience requires the adaptation of policy contexts to dynamic system operation. Challenges modify the currently held policy contexts, causing new security policies to be triggered, as specified in the Or-BAC formalism, described in Deliverable D1.3b [SSSF+11]. Figure 14 presents a summary of the approach we have taken.

This architecture proposes two main modules, which are the Policy Instantiation Engine (PIE) and the Policy Decision Point (PDP), which collectively implement the network resilience manager functionality. The PIE implements an Or-BAC access policy, and dynamically adapts this policy according to the information retrieved from Intrusion Detection Message Exchange Format (IDMEF) alert messages [DCF07]. These message originate from the Distributed Store for Challenges and their Outcome (DISco), which is described in ResumeNet Deliverable D2.2b [SFM+10]. The PDP retrieves the newly triggered policy rules and explores techniques to deploy the new policy, based on the capabilities of the Policy Enforcement Points, i.e., resilience mechanisms embedded in the network, such as firewalls and servers.

This approach is being applied to the experiments being conducted within the project on ensuring the resilience of a publish-subscribe platform, which supports an Internet of Things application [RPFL10]. The changing environment of the publish-subscribe platform, such as events from various sources and frequent accesses by new customers, requires a dynamic security policy. Furthermore, this need is motivated by the diversity of threats facing the platform. The response to these threats can be achieved by applying a security policy using contextual

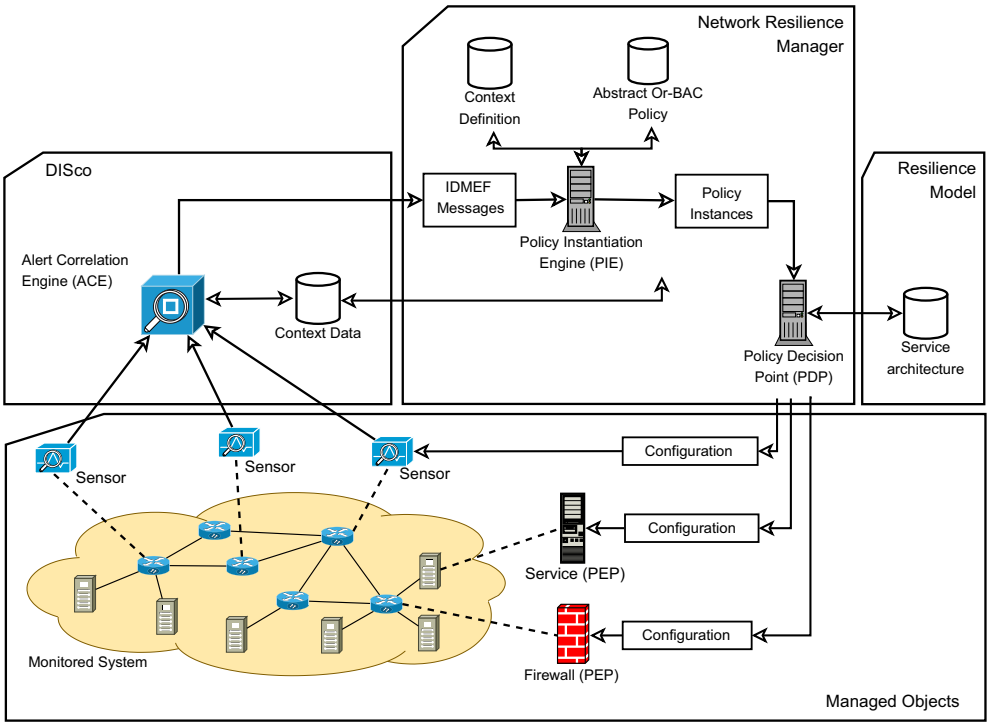


Figure 14: Access Policies Adaptation Framework [DSS+10]

rules, allowing both to specify the system's normal behaviour and its nominal behaviour in the presence of threats. The process for defining and invoking dynamic behaviour based on contextual policies includes several steps: *(1)* the specification of the nominal behaviour of the system, i.e., the normal security policy applied to the platform in the absence of any external risk; *(2)* the specification of the policy in reaction to threats, i.e., safety rules related to threat contexts; *(3)* definition of the mapping functions connecting the threat contexts to certain attributes of IDMEF alert messages; *(4)* deployment of the dynamic security policy, preceded by a translation phase of this policy into a set of configurations that apply to the various checkpoints (routers) of the platform; and *(5)* disabling threat contexts after a latency phase depending on the level of risk that activated this context. This process is discussed in further detail in ResumeNet deliverable D3.2 [DDK+10].

4.5 Summary

In this section, we have summarised the project's efforts toward the management of networks for resilience. Implicitly, one of the underlying assumptions of our work is that dynamic real-time adaptation is necessary, and can be realised through the use of policies and our proposed management architecture. However, to be deployed successfully, such management techniques must work alongside or augment existing approaches that are, for example, used by network operators – approaches that typically heavily involve human operators, following well-defined incident management processes. Identifying how our proposed management architecture could augment existing approaches is an area for future work.

As a starting point, one might consider that well-understood and proven resilience strategies – technical solutions that successfully mitigate a challenge – would be captured in policies and invoked dynamically upon detection of a challenge. These could reflect current incident

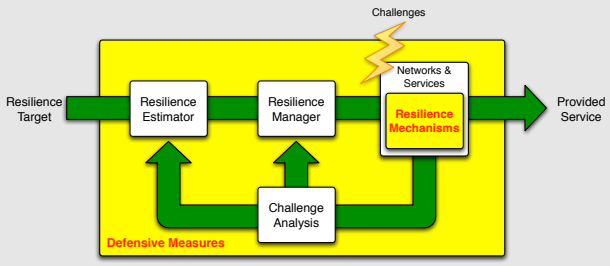
management processes, with "hooks" into the strategy specified where humans are required for decision making, *e.g.*, when invoking remediation mechanisms. A difficulty lies in determining how effective a resilience strategy is. Our policy-driven resilience simulator could be used to give an indication of this. Successful strategies that have been shown to be effective, either via simulation or deployment, could be encoded into resilience patterns, thus increasing their chance of (further) real-world application.

Key resilience management deliverables

C. Doerr, M. Schöller, P. Smith, N. Kheir, J. Lessmann, and C. Rohner. Remediation, recovery, and measurement framework. ResumeNet Project Deliverable, September 2010

M. Schöller, P. Smith, A. Schaeffer-Filho, N. Kheir, and S. Natouri. Policies for resilience (Final). ResumeNet Project Deliverable, August 2011

5 Multilevel Resilience Mechanisms



In this section, we discuss various multilevel *resilience mechanisms*, which include *defensive measures* and techniques for remediating challenges. Although not highlighted in the diagram, approaches that contribute to *challenge analysis* are also discussed.

Information about high-impact challenges derived from the risk assessment process defined in Section 3 can be used to identify the nature of the resilience mechanisms that are required for a network deployment. Since challenges may vary broadly from topology-level link failures to application-level malware, measures against anticipated high-impact challenges need to be applied at different levels and phases: in the network topology design phase, and within protocols; across a network domain, as well as at individual nodes.

A selection of resilience mechanisms developed on the ResumeNet project is summarised in this section using a series of *fact sheets*. For each of the mechanisms developed in the project, we briefly **describe** its purpose; the **innovative** aspects of the mechanism; suggest a **deployment** timescale, *i.e.*, to what extent it is readily usable; and the classes of **challenge** it can be applied to (see Section 3). Initially, we present *network* resilience mechanisms, we then continue by showing *service* resilience mechanisms.

From the project outset, we have understood multilevel aspects to be an important for network resilience. A great deal of research is required to understand the potential benefits of multilevel information sharing and control by resilience mechanisms, which has been shown in a number of contexts [BBF+10], versus the additional complexity it introduces to network design. To this end, we have worked on a cross-layer framework that can be used to understand the optimal layers to introduce resilience functionality. In this section, we briefly introduce the formalism that forms the basis of the cross-layer framework.

5.1 Network Resilience Mechanisms

The following is a set of network-level mechanisms developed in ResumeNet.

Name	Survivable Network Design (SND) [Gou10]
Description	During network planning, SND is used to optimise network operations, such as routing and transport, in the presence of high-impact challenges.
Challenges	The basic mechanism [Gou10] can be applied to consider the resilience of a network's design to different classes of challenge, *e.g.*, identified during a risk assessment process. The novel aspect introduced by the project is its application to *malicious attacks*, *i.e.*, selfish behaviour.
Innovation	Expansion of the methodology to derive a cooperation-friendly routing scheme for wireless mesh networks (WMNs) to mitigate selfish nodes, accounting for radio interference [PGLK10]
Deployment	Near-term

Name	Game-theoretical node protection [OOVM09]
Description	Malware continues to have a pernicious effect on resilience. To help address this problem, we use game theory dynamics to devise node protection schemes against malware propagation
Challenges	This mechanism is intended to improve the resilience of networks to *malicious attacks*.
Innovation	The achieved *node* protection is shown to heavily depend on the underlying *topology*, with direct implications for the network design process.
Deployment	Near-term

Name	Rope-ladder routing [LSZB10]
Description	A multi-path forwarding structure which combines link and node protection in a way that the loss gap and QoS penalty, *e.g.*, delay, during fail-over is minimised
Challenges	This mechanism can be used to improve the resilience of multi-hop wireless (mesh) networks to a range of challenge types, including *component faults*, *hardware destruction*, *failure of a provider service*, *e.g.*, a routing daemon, and *communication environment related* issues.
Innovation	Existing work largely focuses on overall packet loss measures. This is inadequate for real-time traffic like voice flows, for which minimising consecutive packet loss in the time between a failure occurring and switching to a backup path is crucial.
Deployment	Medium-term

Name	Path diversity mechanism [RJS09]
Description	A mechanism that can be used to select multiple paths between a given ingress and egress node pair using a quantified diversity measure to achieve maximum flow reliability.
Challenges	In a similar manner to rope-ladder routing, this mechanism can be used to improve the resilience of networks to a range of challenge types, including *component faults*, *hardware destruction*, *failure of a provider service* and localised *communication environment related* issues.
Innovation	Takes into account higher level requirements for low-latency or maximal reliability in selecting appropriate paths.
Deployment	Medium-term

Name	QoS^2: Integrating QoS with Quality of Security [THAB10]
Description	A Quality of Protection (QoP) framework that tunes between security requirements and Quality of Service (QoS) using a multi-attribute decision making model.
Challenges	The QoS^2 framework aims to improve the performance of networks in response to various forms of *malicious attack*.
Innovation	Makes autonomic decisions about the level of security required given the current threat level. Unlike previous work that performs this trade-off, does not introduce new vulnerabilities.
Deployment	Medium-term

Name	Detection and TRAceBack (DTRAB) mechanism [FTV$^+$10]
Description	An anomaly detection approach that makes use of strategically placed Monitoring Stubs (MSs) to detect attacks on "encrypted protocols" and trace the origin of the attack.
Challenges	DTRAB is intended to detect *malicious attacks* to secure cryptographic protocols.
Innovation	Many existing intrusion detection systems that are deployed use payload data to detect malicious behaviour. However, for protocols that encrypt packet payload this is not a viable solution. DTRAB uses a novel detection approach that identifies deviations from the normal behaviour of protocols.
Deployment	Medium-term

5.2 Service Resilience Mechanisms

Below are service-level mechanisms that have been developed in ResumeNet.

Name	Cooperative SIP (CoSIP) [FNKC07]
Description	An extension of the Session Initiation Protocol (SIP), whereby end-points are organised into a peer-to-peer (P2P) network. The P2P network stores location information and is used when the SIP server infrastructure is unavailable.
Challenges	CoSIP is intended to improve the resilience of a SIP infrastructure to a range of challenges, including *component faults*, *hardware destruction*, *human mistakes*, and the *failure of a provider service*, *e.g.*, the DNS.
Innovation	Optimal setting of the number of replica nodes in the P2P network for given service reliability levels, inline with an enhanced trace-driven reliability model.
Deployment	Near-term

Name	Virtual service migration [FAHF10]
Description	Enables redundancy and spatial diversity by relocating service instances on-the-fly, such that a continuous acceptable service can be provided to its users.
Challenges	Wide-area virtual service migration could improve the resilience of services to a range of challenge classes, in addition to the *hardware failures*, which it is typically applied to. *Components faults* caused by software bugs may not be mitigated by virtualisation techniques alone; implementation diversity is required in this case.
Innovation	Existing approaches are tailored toward resilience to hardware failures within data centres. The derivation of service migration strategies from migration primitives, providing resilience against a variety of challenges.
Deployment	Medium-term

Name	Perco Pastry [GHK11]
Description	A mechanism that creates redundant overlay network paths in case of failures on a primary path. Additional overlays paths are sought using increasingly intensive search strategies depending on the severity of connectivity issues. The protocol is implemented as an extension to the Pastry [RD01] Distributed Hash Table (DHT) algorithm.
Challenges	Perco Pastry can be applied in the presence of a number of different classes, including *hardware faults*, *e.g.*, of various systems along an end-to-end path. Also, nodes that are conducting a *malicious attack*, *e.g.*, by not forwarding traffic, could be circumvented using Perco Pastry.
Innovation	Existing approaches, such as Resilient Overlay Networks (RON) [ABKM01] have been limited in scale. Perco Pastry is intended to scale to large overlay networks. Furthermore, it adjusts how aggressively it tries to find an end-to-end path based on the severity of connectivity issues.
Deployment	Near-term

5.3 Minimising Multilevel Resilience Complexity

One of the problems of applying multilevel resilience mechanisms, which share information and control across protocol layer boundaries, is that they increase the complexity of networks, which can potentially lead to unexpected and undesirable emergent behaviours. To help address this issue, we are developing a *cross-layer framework*, based on a formalism, that can be used to evaluate the tradeoff of increased complexity compared to an increase in resilience. At the core of this is determining the optimal layer to introduce some resilience functionality, *e.g.*, error correction or path diversity.

In the framework, we model *dials* to provide information to higher layers, and *knobs* to influence the operation of lower layers. These cross-layer knobs and dials will enable the selection and tuning of specific resilience mechanisms that are appropriate for a current network state. For a given scenario, it can be apparent what cross-layer knobs and dials are needed. With this understanding it becomes an iterative process to reduce the complexity of the cross-layer control loops, which will involve re-evaluating the layer at which it is most appropriate to implement a mechanism. For example, consider a path diversification mechanism: the optimal use of paths requires information from both the routing and transport layers, and a path diversification algorithm itself may operate in either of the layers. By evaluating the cross-layer controls needed, we can determine which layer the functionality should reside in to minimise the complexity of the knobs and dials needed. We formalise the representation of knobs and dials as follows.

The set of all knobs

$$\mathbb{K} = \mathbf{K} \cup \mathbf{k}$$

is the union of out-of-band **K** and in-band **k**. The set of all dials

$$\mathbb{D} = \mathbf{D} \cup \mathbf{d}$$

is the union of out-of-band **D** and in-band **d**. Knobs and dials are defined on the boundary between layers L_i and L_j where i and j are either numbers, e.g., {1, 1.5, 2, 2.5, 3, 4, 7, 8} or layer designators, e.g., {HBH, NET, APP}. An individual knob or dial between layers L_i and L_j is then $K_{i\rightarrow j}(\mathrm{desc})$ where 'desc' is a descriptor, e.g. Bit Error Rate (BER). Therefore, the set of all out-of-band knobs and dials between layers i and j

$$\mathbf{K} = \cup_{\forall K \in \mathbf{K}} K_{i\rightarrow j}$$

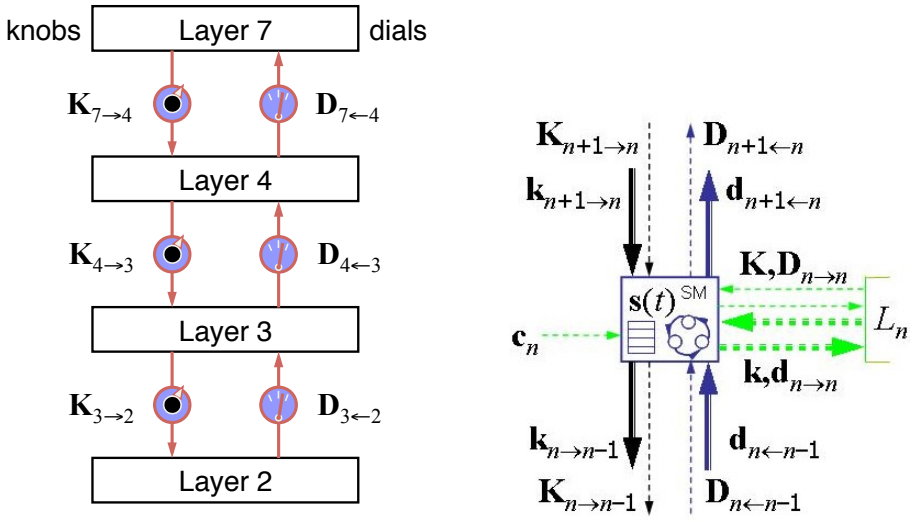
represents a vertical relationship when $i \neq j$ and represents a horizontal relationship when $i = j$.

To carry this further, we can fully represent a layer n protocol instance at time t in terms of its knobs and dials, as well as its *state* $s(t)$ and *context* $c_n(t)$. For L_n, we can define state with respect to time as:

$$\mathrm{s}(t+1) = f(\mathbb{K}_{n+1\rightarrow n}, \mathbb{D}_{n\leftarrow n-1}, \mathrm{s}(t), \mathrm{c}_n)$$

where f is a function specific to the internal algorithm of that particular protocol.

Figures 15(a) and 15(b) illustrate these relationships, which we have applied to mechanisms such as explicit cross-layer support for error control [BBF+10]. In this work, we aimed to evaluate which combination of error control mechanism – Forward Error Correction (FEC) or Automatic Repeat reQuest (ARQ) – operating at either (or a combination of) the hop-by-hop (HBH) or end-to-end (E2E) layer provides optimal performance given certain service



(a) Multilayer overview (b) Detail

Figure 15: Cross-layer formalism

requirements, *e.g.*, regarding timeliness and acceptable loss rates. Future work will aim to generalise the use of the cross-layer framework to understand how it can be applied to other cross-layer concerns, *e.g.*, the optimality of path diversity provided at the overlay (application) layer provided by Perco Pastry [GHK11] versus the physical layer *e.g.*, using the Survivable Network Design (SND) tool [Gou10].

5.4 Summary

In this section, we have summarised a number of novel resilience mechanisms that have been developed on the project. They are intended to operate both at the network and service level, and span a number of the stages of the $D^2R^2 + DR$ strategy. Future work should investigate how mechanisms at the network and service levels could interact synergistically to improve resilience. Multilevel resilience introduces complexities that make understanding emergent behaviours that may be undesirable potentially less tractable, compared to strictly layered networks. To help address this issue and minimise complexity, we are developing a multilevel framework, based on a formalism for cross-layer sharing and control, which can be used to understand the optimal layers to place resilience functionality.

Key deliverables related to resilience mechanisms

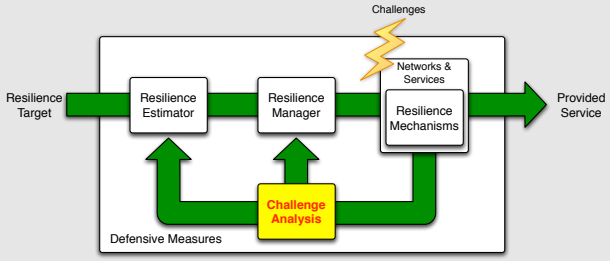
C. Doerr, E. Gourdin, G. Popa, J. Omic, J. P.G. Sterbenz, J. P. Rohrer, T. Taleb, and P. Van Mieghem. Second draft on defensive measures for resilient networks. ResumeNet Project Deliverable, August 2010

A. Fischer, Y. Al-Hazmi, and A. Fessi. P2P Overlays and Virtualization for Service Resilience. ResumeNet Project Deliverable, August 2010

S. Günther, R. Holz, and N. Kammenhuber. Overlay-based end-to-end connectivity. ResumeNet Project Deliverable, March 2011

R. Bruncak, N. Bohra, A. Fischer, S. Simpson, J. P. Rohrer, J. P.G. Sterbenz, D. Hutchison, and H. De Meer. Cross-layer optimization and multilevel resilience. ResumeNet Project Deliverable, August 2010

6 Challenge Detection



Challenge analysis is used to determine the nature of the challenges that are impacting a network and supported services. This is in contrast to the *resilience estimator*, whose primary purpose is to determine if the *resilience target* is being met. Together they inform *management* decisions.

The *detect* stage of the D^2R^2+DR strategy involves many aspects. Throughout the project, our understanding of the purpose and means of realising this stage has evolved considerably. In particular, we have found theories of *situational awareness* [End95, Bas00] to be informative when considering the purpose of the detect stage. In short, situational awareness is used to inform decision making and action taking, *i.e.*, remediation and recovery, either by a human operator or automatically using the resilience management architecture discussed in Section 4. Perhaps the most prominent theory of situational awareness, proposed by Endsley, describes three levels of awareness [End95]:

1. *perception* of elements in the current situation;
2. *comprehension* of the current situation;
3. and the *projection* of future status.

As one advances through these levels, decision making capabilities are improved. Our work on the project has largely focused on mechanisms and architectures for perception and comprehension. We propose two key sources of information for situational perception for network resilience: *(1) multilevel network measurement information*, using monitoring tools such as X-Trace [FPK+07], and *(2) context information*, which is external to the network, such as weather data [JRO+09]. These two forms of information – network and context – are used as inputs to various techniques that are used to build situational comprehension.

To assist decision making for resilience, we see three main outputs of comprehension: *(1)* detection of the presence of a challenge, *e.g.*, provided by anomaly and intrusion detection systems; *(2)* identification of the characteristics of the challenge, *e.g.*, provided by classification [NA08] and data fusion [TSB+06] techniques; and *(3)* the impact a challenge is having on the network and associated services, which is provided by the resilience estimator (Figure 13). This view of the purpose of the detect stage in relation to situational awareness is summarised in Figure 16.

While we have not focused on projection as part of our activities, we acknowledge it plays a critical role in correct decision making. We envisage outcomes from our risk assessment process and threat analysis techniques would be used to inform mechanisms that anticipate the trajectory a challenge will take, and its impact on the network. This is an important area for future research.

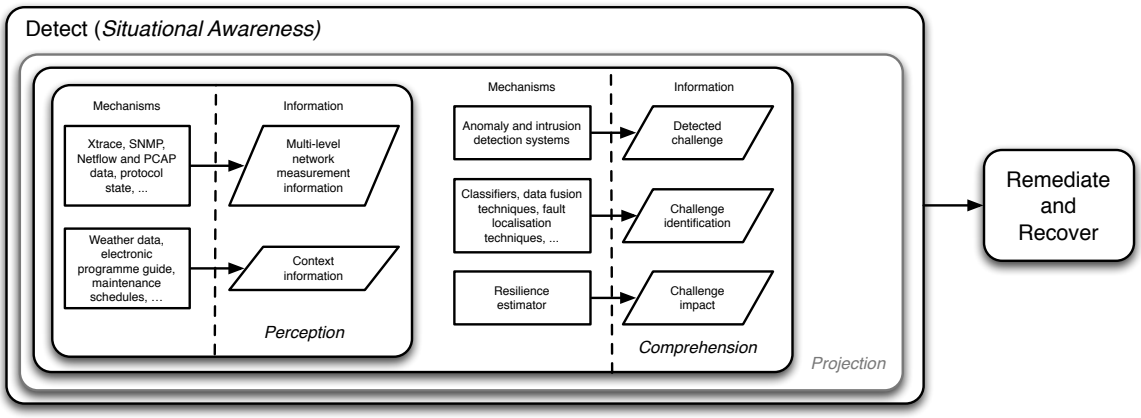


Figure 16: The *detect* stage of the D^2R^2 + DR strategy placed in the context of situational awareness [End95]

6.1 Information for Situational Perception

The two key sources of information for situational perception are multilevel network measurements and context information. Arguably, network measurements are the primary information source used to determine the nature of challenges and the effect they are having on a network and associated services, with context providing supplementary evidence to affirm or further explain phenomena observed from measurement data. Measurement information should be gathered from various levels of the protocol stack and from different geographical (topological) locations. With this in mind, two important questions arise: *(1)* what should be measured in order to inform situational awareness?; and *(2)* how should the information be gathered and communicated to the various components of our resilience architecture? We address these two questions in the following sections.

What to measure for resilience

To address the first question on *what* to measure, output from our findings on resilience metrics, discussed in Section 2, can be applied. Specifically, we can use results from the Graph Explorer about the relationship multilevel metrics have to help make decisions. Recall the metric envelopes shown in Figure 7(a) that show the behaviour of metrics at different levels of the protocol stack for the GÉANT2 example. The Graph Explorer indicates to what extent the behaviour of metrics at different levels is correlated. In Figure 7(a), correlated metrics are shown with a blue arrow – application-level end-to-end jitter and physical-level connectivity are, in this case. This understanding of the relationship between metrics at different levels can be used to identify what to measure in order to develop situational awareness.

Let us consider a perhaps simplistic example application of the use of the Graph Explorer for determining what to measure. The aim is to measure a minimum set of metrics that indicate some perturbation has occurred with respect to a set of SLAs, based on the assumption measuring correlated metrics that exhibit overlapping behaviour is not informative. Using our risk assessment process (see Section 3), a set of probable high-impact challenges can be determined $\mathrm{C} = \{\mathrm{c}_0 \ldots \mathrm{c}_{\mathrm{n}-1}\}$. Furthermore, a resilience target, expressed as a series of Service Level Agreements (SLAs), *e.g.*, for different customers and services, will have been defined $\mathrm{S} = \{\mathrm{s}_0 \ldots \mathrm{s}_{\mathrm{n}-1}\}$, with each SLA s_i describing the desired behaviour of a set of metrics

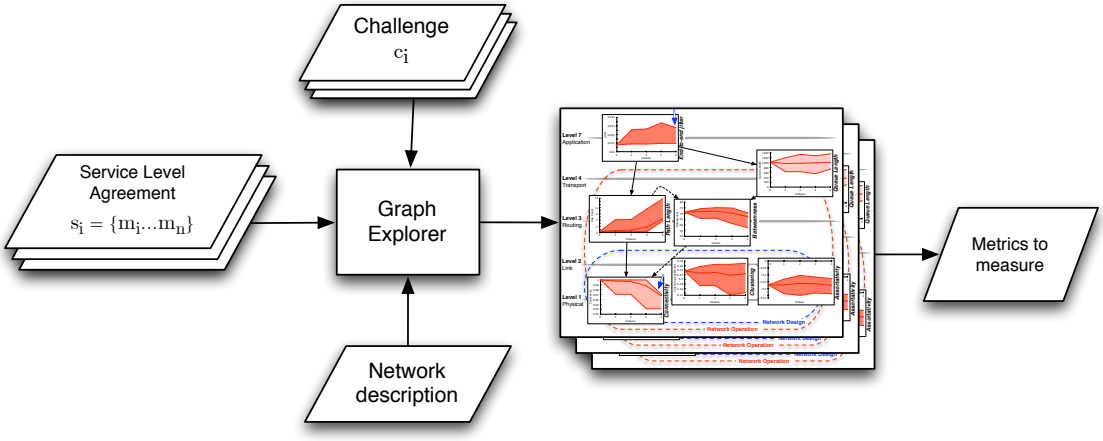


Figure 17: An approach that uses the Graph Explorer tool to determining what multilevel metrics should be measured to inform situational awareness

$s_i = \{m_0 \dots m_{n-1}\}$. Also, a model of the network N to be evaluated will have been developed for use in the the Graph Explorer[4]. The tuple, or *scenario*, (c_i, S, N) can be given to the Graph Explorer $\forall c_i \in C$, producing a set of metric envelope sets for each scenario $E = \{e_0 \dots e_{|C|-1}\}$, where $|E| = |C|$ and e_i is the set of envelopes for the metrics defined in S (see Equation 2). Taking output from the GraphExplorer about which metrics are correlated, we can produce a set of metric envelope sets $E' = \{e'_o \dots e'_{|C|-1}\}$ that contains only the correlated metrics from each scenario. To get a minimum set of metrics to measure E'', we can take the intersection of the correlated sets of metrics from each scenario (Equation 3); exceptionally, we choose a single metric to add to E'' under the conditions shown in Equation 4, and select a metric from e'_i and e'_{i+1} given the condition shown in Equation 5 holds. (The choice of metric to include could relate to the probability a perturbation in a metric is indicative of a particular challenge.) An overview of this approach to determining what to measure is presented in Figure 17.

$$|e_i| = \sum_{s_i \in S} |s_i| \tag{2}$$

$$E'' = \bigcap_{e'_i \in E'} e'_i \tag{3}$$

$$|e'_i \bigcap e'_{i+1}| > 1 \tag{4}$$

$$e'_i \bigcap e'_{i+1} = \{\emptyset\} \tag{5}$$

This simple example shows how an approach can be derived to computationally determine what multilevel metrics should be measured for network resilience, based on an aspect of our metrics framework. While the metrics framework can provide support to answer this question, a great deal of further work is required to answer it satisfactorily in reality. For example, it is not clear which metrics should form part of the initial SLA set S; effort developing resilience classes, described in Section 2, could help. Furthermore, understanding what to measure

[4]Depending on the metrics to be evaluated, the Graph Explorer can use a combination of structural analysis, simulation or testbed emulation.

may practically be heavily influenced by determining what *can* we measure, given problems associated with organisational boundaries that inhibit information sharing, privacy issues, and the extent networks and services are instrumented for measurement, for example. We do not have carte blanche when deciding what to measure.

As discussed earlier, context information can provide important supplementary evidence to affirm awareness derived from multilevel network measurement information. There has been a great deal of work showing various functional and performance improvements through the use of context information for mobile (ad-hoc) networks [ECD+01, KK02, CEM03]. With regard to resilience, earlier work has demonstrated how the use of weather information, an example of context, improves the resilience of millimetre wave wireless mesh networks, which perform poorly in heavy rain [JRO+09]. In short, information regarding the trajectory of weather fronts is used to make pre-emptive routing decisions to avoid potentially affected regions of the mesh. Another use of context could include using information regarding news events to rationalise an unusually high-volume of network traffic that is causing problems. Anecdotal evidence suggests this form of information is used frequently in Network Operation Centres (NOCs) by human operators. Therefore, we propose a key area for further work is to investigate the systematic use of context information for resilience.

Measuring and sharing multilevel measurement information

Network measurements should be taken at multiple levels of the protocol stack, and in diverse geographical and topological locations – typically these are related. We have explored various monitoring tools that can be used to conduct these measurements, our results are discussed in Deliverable D1.4 [BBF+10]. In summary, various forms of monitoring tools should be used, which either operate in a *centralised* [CFSD90] or *distributed* [SWKA00, SPS+01] fashion, collecting *logging*, *tracing* and *profiling* information. Monitoring tools can be further classified as being either *node* or *task*-centric; the latter providing causal measurement data associated with a task, *e.g.*, a session initiation request. Because of the complicated and distributed nature of networks and services, we argue that "causal, end-to-end (cross domain) task-centric monitoring must be an integral part of network services to enable resilience" [BBF+10]. As a promising candidate for this, we have been exploring the use of X-Trace framework for multilevel resilience monitoring [FPK+07], which reports the message and protocol header flow between supported network layers, nodes and applications. Supported applications and devices generate metadata, which is sent in-band with the actual data communication of the application. When X-Trace enabled implementations [Fis10, Pet09] receive metadata-enriched packets, they generate reports on the packet's state and any actions taken. These reports are sent out-of-band to either a centralised reporting server or a requesting party. The distinguishing feature of X-Trace is that it records the causal relations among these events in a deterministic fashion, across threads, software layers, different machines, and potentially different network layers and administrative domains. Furthermore, it groups events into tasks, which are sets of causally related events with a definite start.

As part of our studies on multilevel resilience, we investigated the suitability of various architectures for multilevel information sharing and control [BBF+10]. These included approaches that allowed direct interactions between protocol levels [WR03], and others that use a shared database architecture [DZK06, CSN02]. Bearing in mind the resilience principle [SHc+10] regarding trading off complexity with a gain in functionality, we propose that a shared database approach to multilevel information sharing to be more suitable for resilience. Within the project, to disseminate multilevel information gathered by monitoring tools, such

as X-Trace, and enable its persistent storage, *e.g.*, for use as part of the *diagnose* and *refine* stages of the $D^2R^2 + DR$ strategy, we have developed a distributed publish-subscribe system called DISco (DIstributed Store for Challenge and their Outcomes) [SFM+10]. The publish-subscribe approach enables components of our architecture (Figure 12) to be loosely coupled, improving resilience to component failure. Information stored in DISco includes monitoring data, and actions performed to detect and remediate challenges. Information sources may report more data than we can afford or wish to relay on the network, particularly *during* challenge occurrences. DISco is able to aggregate information from multiple sources to tackle this problem. To assist the two phases of the outer loop, DISco employs a distributed peer-to-peer storage system for longer-term persistence of data, which is aware of available storage capacity and demand.

6.2 Challenge Identification Architecture

Building on information for situational perception, we propose an incremental approach to challenge identification, one of the outputs of situational comprehension, whereby an evolving understanding of the nature of a challenge is developed. The aim is to enable early remediation to protect infrastructure and services from potential collapse, using imperfect information, and subsequently use more-specific remediation activities as a more detailed picture is constructed, *i.e.*, as comprehension improves. In addition, this approach acknowledges the varying computational overhead, timescales, and (potentially limited) accuracy of current detection approaches [FFK+10]. Our work is similar in nature to that proposed by Gamer [Gam09]. However, we introduce greater flexibility and reusability of identification strategies by using policies to orchestrate the identification process.

Figure 18 identifies the main architectural components of our incremental challenge identification approach. *Challenge models* that describe the challenges identified in the risk assessment process (shown in Section 3) inform the operation of an *identification engine*. Within the project, we have studied the use of the Chronicle Recognition System (CRS) [MD03] and Incremental Hypothesis Updating (IHU) technique [SS04b] to fulfil these two roles; we summarise them later in this section. Via the publish-subscribe system *DISco*, the identification engine subscribes to alarms from a number of *information sources*, such as those discussed earlier for perception (*i.e.*, that supply monitoring and context information) and comprehension (*e.g.*, anomaly detection and classification systems). Based on the models used by the identification engine, information sources are used to develop an evolving picture of the nature of challenges. When a challenge has been identified, or an hypothesis about the nature of an ongoing challenge is reached, the identification engine publishes challenge information to DISco. Using policies, the *resilience manager* configures the various information sources, *e.g.*, invokes new detection and classification mechanisms, when a challenge event has been published by the identification engine. Using this architecture, we envisage relatively lightweight mechanisms being initially on that can be used to detect broader classes of challenge. When the initial mechanisms indicate the onset of a challenge, further mechanisms are invoked to gain a better understanding of its nature.

As an example of how the architecture can be applied, consider a high traffic volume challenge, such as a Distributed Denial of Service (DDoS) attack or a flash crowd event. Initially, simple link monitoring mechanisms could generate an alarm if the volume of traffic goes over a given threshold. The link monitor is an information source in Figure 18, that publishes alarms the identification engine component subscribes to. On detection of high traffic volumes, the link could be rate limited, *e.g.*, by randomly dropping packets, to protect

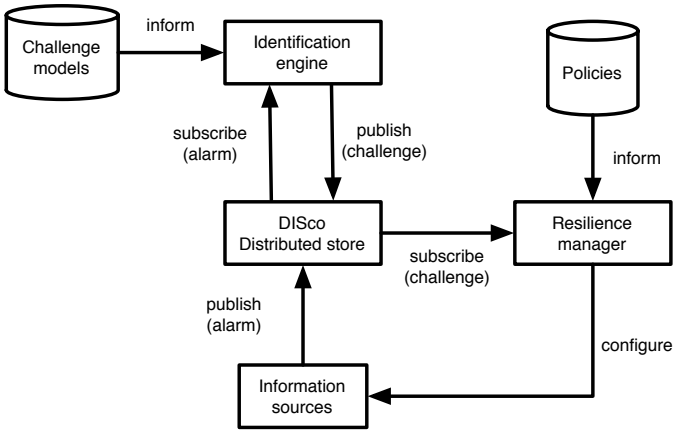
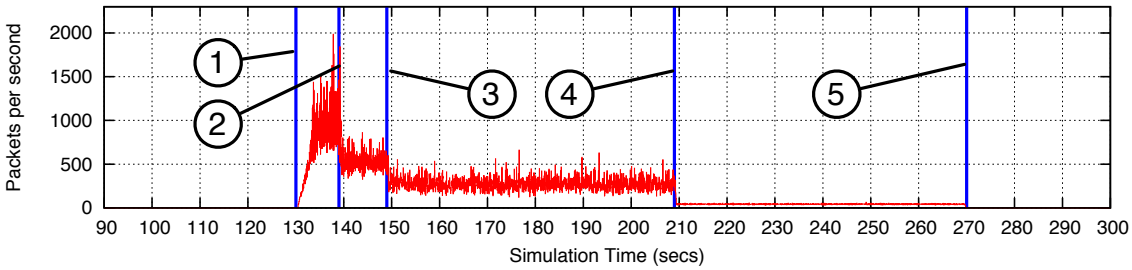


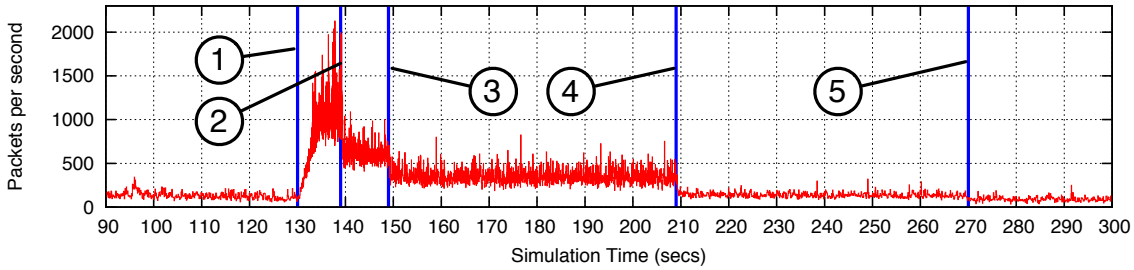
Figure 18: The ResumeNet challenge identification architecture

the challenge target or router resources. An anomaly detection system could then be invoked to determine the target of the challenge, leading to more specific rate limiting of traffic to this destination. Finally, a more expensive and none real-time traffic classifier could be used to understand the nature of the flows causing the challenge, *e.g.*, resulting in flows being blocked if they are deemed to be malicious, or re-directed or no longer subjected to rate-limiting if they are seen to be benign. The specific multi-stage configuration of detection mechanisms (*i.e.*, how the identification engine and resilience manager coordinate information sources) will be dependent on the network deployment context. For example, in some deployments, *e.g.*, on a relatively resource abundant ISP network, all the detection mechanisms described in the example could be always-on. (Although, they will yield results at different timescales, which leads to multi-stage remediation, as described above.) However, for example, in a resource-scarce wireless mesh network or sensor network deployment, the necessary computational, bandwidth and power resources may not be available.

To demonstrate the feasibility of the policy-driven incremental approach to challenge identification, we simulated the high traffic volume challenge example using the policy-driven network simulator discussed in Section 4. Figure 19 shows results from these simulations, described in more detail by Yu *et al.* in [YFSF+11]. Fig. 19(a) shows the onset of a DDoS attack on the ingress link of an ISP at approximately 130 seconds (1). The raising of an alarm by a link monitor is seen at 139 seconds (2), whereby a sustained traffic load in excess of a threshold defined in policies has been reached. Shortly thereafter, the effects of the initial rate limiting of the ingress link by a rate limiter remedy can be observed. The limiter is configured by policies to discard 70% of all incoming traffic in order to protect downstream servers and infrastructure. At 149 seconds (3), an intrusion detection system identifies the destination IP address of the victim. This is achieved in this case by examining the destination address of each incoming packet, and raising an event when one destination accounts for 60% of all packets. The rate limiter is now reconfigured to drop 70% of the traffic destined for the victim only. Some legitimate traffic that is not destined for the victim, which previously was blocked, is now not filtered. Also, a classifier is initiated at (3) and flow exporting from the ingress router is started. The policy used to trigger these actions is shown in Figure 20. Hereafter, the classifier, receiving flow records from the flow exporter, attempts to identify the specific attack flows. At 209 seconds (4), rate limiting is confined just to the attack flow and legitimate traffic to the Web service can continue. After 270 seconds (5), all the malicious traffic is blocked, shown in Fig. 19(a), and the remaining traffic, shown in Fig. 19(b), pertains to



(a) DDoS traffic



(b) DDoS and benign traffic

Figure 19: Initial results from simulations that implement an incremental detection and remediation approach to a resource starvation attack [YFSF+11]. Numbered labels pertain to points in our multi-stage approach.

normal background traffic.

```
on IntrusionDetectionMO.detection(IPAddress,EAccessLink)
    do
    {
        RateLimiterMO.limit(IPAddress, 70%);

        FlowExporterMO.setTimeOut(EAccessLink,60s);
        FlowExporterMO.setSampleRate(EAccessLink,75%);
    }
```

Figure 20: Policy configuring a rate limiter and flow exporter, in response to a detection event generated by an intrusion detection system

As discussed earlier, the identification engine (using challenge models) and the resilience manager (using policies) configure the various information sources, such as those presented in this example, that incrementally build understanding about the nature of a challenge. We have investigated the use of the Incremental Hypothesis Updating (IHU) [SS04b] technique and the Chronicle Recognition System (CRS) [MD03] as ways of realising the challenge models and the identification engine. We now summarise these two approaches.

6.3 Incremental Hypothesis Updating (IHU)

Incremental Hypothesis Updating is a probabilistic fault localisation technique that can be used to identify a set of faults that are the cause of symptoms observed by various monitoring functionalities, e.g., as a consequence of SNMP traps. Faults f and their symptoms s are modelled as a symptom-fault map, using a bipartite directed graph. One set of nodes in

the map identifies the faults to be modelled, which are connected to the symptoms that are observed if the fault manifests. Faults in the graph are annotated with the probability of a fault independently occurring, and edges are weighted with the probability that a fault will cause a symptom. An example symptom-fault map is shown in Figure 21. On receipt of a newly observed symptom, the IHU technique immediately yields a set of the most likely hypotheses explaining the observed symptoms to-date. This information is published to DISco and enables the selection of remediation mechanisms. We refer the reader to [SS04b] for a description of the IHU algorithm.

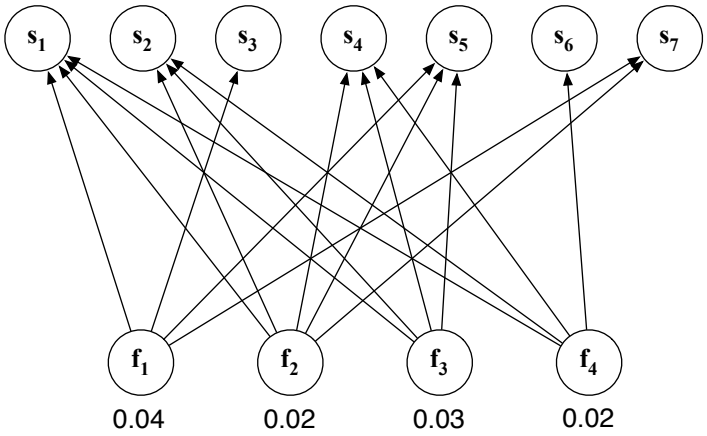


Figure 21: An example symptom-fault map used as part of the Incremental Hypothesis Updating (IHU) fault-localisation approach. Example probabilities of faults (challenges) occurring are shown. For clarity, the probability of a symptom pertaining to a challenge are not shown. See Tables 2 and 3 for a description of the symptoms and challenges, respectively.

Symptom	Description	Cost (c)
s_1	High utilisation of links	0.1
s_2	Sharp increase in per-client request rate	0.5
s_3	Steady per-client request rate	0.5
s_4	Large number of connections from previously unseen ASes	0.7
s_5	Abnormal number of TCP packets	0.6
s_6	Abnormal number of UDP packets	0.6
s_7	Abnormal page request distribution	0.7

Table 2: Symptoms associated with the challenges shown in Table 3 and Figure 21. They are derived from observations described by [JKR02].

Challenge	Description
f_1	Flash crowd event
f_2	TCP SYN attack
f_3	HTTP flood attack
f_4	UDP jumbo packet attack

Table 3: Challenge that could cause the symptoms shown in Table 2, as depicted in Figure 21.

In relation to the challenge identification architecture, depicted in Figure 18, the symptom-fault map corresponds to the challenge model. Information sources generate symptoms that

are used to trigger hypotheses being generated by the IHU algorithm, which realises the identification engine component. Symptoms in the symptom-fault map are extended by annotating them with properties describing the mechanisms that are used to generate them. This includes the cost c (example values are shown in Table 2), which may relate to computational or bandwidth overhead, accuracy a, and timeliness t of the mechanism. These values are used to select the mechanisms to invoke, given a hypotheses set, policies regarding the use of available resources and the (measured) computational and network resources.

6.4 Chronicle Recognition System (CRS)

The Chronicle Recognition System (CRS), developed by France Telecom, is an event correlation tool based on symptom-to-fault knowledge represented in the form of chronicles. A chronicle is a set of events, interlinked by time constraints [CD00]. An example chronicle is shown in Figure 22, in which events are partially sequenced and the time interval between A2 and A5 must be in the range of one to three minutes. Events are represented by their name, state/transition (from state 1 to state 2) and time/interval of occurrence. Time information enables the sequencing of events and the ability to specify time spans between their occurrence. In the context of the challenge identification architecture, we use chronicles to represent models of challenges.

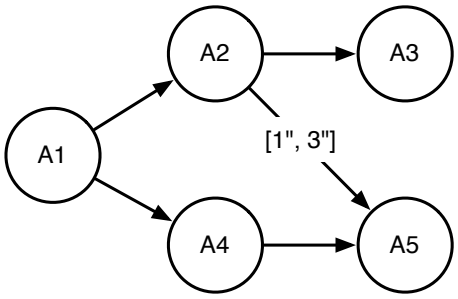


Figure 22: An example chronicle showing sequencing of events and timing constraints [CD00]

A system to recognise chronicles, which realises the identification engine of our architecture, is implemented using a framework called Time-stamped Event Stream System (TESS) [DM09]. TESS detects in real-time any subset of alarms of the input stream that matches the set of constrained alarms of a chronicle model (*i.e.*, according to all time constraints). A matching of some alarms with an subset of a chronicle is known as a partial instance of the chronicle model. When a complete match is found, the chronicle is recognised and any associated action is performed by the system. In the context of the challenge identification architecture, partial or full recognition of a chronicle results in a challenge event being published, which may lead to the reconfiguration of information sources by the resilience manager.

6.5 Challenge Analysis Techniques

Research conducted as part of the ECODE project [ECO] aims to address the complexities of on-line detection and classification issues discussed, and relates to our multi-stage challenge identification approach. Specifically, within the project, they are developing an automated classification approach, based on a two-step process: *(1)* detecting anomalies using a relatively lightweight algorithm that requires easily derived traffic information, and identifying flows associated with the anomaly; and *(2)* use the flow information from the detection phase to

derive metrics that can be used for signature-based classification. Specifically, they aim to automatically classify network traffic anomalies, such as DDoS attacks and network scans. We refer the reader to [BKL+09] for a more detailed description of the approach adopted by the ECODE project, including implementation and experimentation details. This activity relates to our multi-stage approach to challenge identification, whereby the algorithm used for step one of the ECODE approach could be initially enabled, and the classification steps could be subsequently invoked on detection of anomaly. This configuration could be expressed using policies, as we discuss next.

6.6 Summary

To inform decision making activities as part of the remediation and recovery stage of the D^2R^2 + DR strategy, it is necessary to build situational awareness, which consists of three levels: perception, comprehension and projection. The two sources of information required for perception for resilience are multilevel network measurements and context information. A key question is to understand what information to measure to build situational awareness. Output from the Graph Explorer tool about the relationship multilevel metrics have to each other can be used to complement the processes used to make this decision. Ultimately, determining what to measure is a complex problem, involving many factors including what can be measured and the nature of SLAs that are in place. In specific scenarios and through anecdotal evidence, context information has been shown to play an important supplementary role in developing situational awareness. Further work is required on the systematic use of context information for resilience.

There are many monitoring approaches that could be used for network resilience; we suggest that causal task-centric monitoring tools, such as X-Trace, are important for determining the root cause of challenges for multi-domain networks and services. To distribute and store multilevel information we have developed DISco – a distributed publish-subscribe system – that provides loose coupling of resilience mechanisms and a shared database-style approach to information sharing: both properties that are understood to be beneficial for resilience.

To enable comprehension we have developed a challenge analysis architecture that can be used to realise an incremental multi-stage approach to challenge identification. This incremental approach can enable rapid remediation based on imperfect information to protect infrastructure and services from collapse; this remediation can be refined as challenge identification is elaborated with more elaborate detection and classification techniques. We have briefly introduced two approaches to realising this incremental approach, namely Incremental Hypothesis Updating (IHU) and the Chronicle Recognition System (CRS). Both of these approaches use challenge models to build comprehension about the nature of an ongoing challenge. Arguably, chronicles is a more complete challenge model, describing causal and temporal relationships of events that identify a challenge. Significant further work needs to be done to develop these two models.

Key challenge detection deliverables

B. Delosme, C. Dousson, N. Kheir, C. Lac, and P. Smith. Service surveillance and detection of challenging situation (interim). ResumeNet Project Deliverable, July 2010

P. Smith, M. Fry, S. Martin, L. Chiarello, M. Fischer, C. Rohner, G. Popa, H. De Meer, A. Fischer, and N. Bohra. New challenge detection approaches. ResumeNet Project Deliverable, September 2010

R. Bruncak, N. Bohra, A. Fischer, S. Simpson, J. P. Rohrer, J. P.G. Sterbenz, D. Hutchison, and H. De Meer. Cross-layer optimization and multilevel resilience. ResumeNet Project Deliverable, August 2010

7 Conclusions and Future Work

In this deliverable, we have presented our framework for resilient networking – the product of reflections on the research outcomes of the technical work packages from the ResumeNet project. It provides a schematic for the systematic engineering of resilient networks and services. A resilience control loop, based on the D^2R^2+DR strategy, is the focal point of the framework, from which the other elements are derived. *The elements of the framework and the way we have chosen to realise them form the key contributions of our work*; we summarise these here, along with items for important future work:

Multilevel metrics framework Being able to specify and measure desired levels of resilience is of critical importance, and is understood to be an area in which there is little consensus on how to approach it. We have developed a multilevel resilience metrics framework that can be used to understand and describe the resilience of networks and services, and the relationship metrics from different levels of the protocol stack have *e.g.*, whether they exhibit correlated or orthogonal behaviour. Accompanying the framework is a set of tools, such as simulation models and libraries for examining metrics, that can be used to evaluate a given network topology to various challenges. Our framework presents a significant step towards being able to specify desired resilience targets by giving a framework to describe metric behaviours, and then evaluate a network's performance with respect to them. Our metrics framework has begun to be of interest to ENISA and network operators.

Future work: The Internet, and most networks in operation today, support a number of services with different resilience requirements. Specification of resilience metrics for a wide range of services and using them to inform decision making as part of the D^2R^2+ DR strategy is likely to be complicated. To enable wider adoption of our framework a simplification is required to address this potential complexity. The use of *resilience classes* may be able to help reduce the complexity involved – if we can decide on a small number of classes that each represents a significant cluster of services and the responses that each would require whenever a challenge occurs. This approach would reduce the number of inter-relationships between services that have to be taken into account when specifying, decomposing or computing resilience.

Processes for understanding challenges Deployed resilience mechanisms should be targeted at addressing the most probable high-impact challenges the network may face. In the context of network resilience, the challenges that could occur transcend those normally considered in other thematic areas, such as information security, fault tolerance and disruption tolerant networks. Without considering this broad spectrum of challenges, mechanisms could be inappropriately deployed. To manage this problem, we have developed a risk assessment process that can be used to identify high-impact challenges. This process builds on an informal categorisation of the forms of challenges that one must consider to ensure network resilience.

Future work: Perhaps the most difficult aspect when trying to determine how a network may be challenged is determining the probabilities a challenge will occur and then understanding what impact they will have. A great deal of work is required in this area in order to obtain realistic values for these. Regarding the former issue, we suggest

that a resilience advisory service, that operates in a similar way to information security advisories, would be a great help. Extensions to our resilience metrics tools, described earlier, could help us to understand the impact of challenges, along with results from wide-scale tests, such as the one carried out by ENISA in 2010 [Eur11].

Resilience management architecture The management of multilevel resilience mechanisms that potentially interact across different administrative domains can be complicated. Furthermore, the operation of resilience mechanisms should in many cases be done in real-time with potentially limited human intervention; incorrect operation could have significant negative consequences. To tackle these issues, we have developed a loosely coupled network management architecture, which makes use of policies to specify multi-stage resilience strategies – configurations of mechanisms that address a given challenge set. By using policies, strategies can be carefully crafted and evaluated, using a policy-driven network simulator we have developed, without the need to take resilience mechanisms off-line. To accompany our architecture, we have performed a critical evaluation of existing policy-driven management frameworks to determine their applicability for resilience. We found they have many useful functionalities, but do not meet all our requirements and a single framework cannot be applied in all deployment contexts.

Future work: We have evaluated policy-driven resilience strategies for the detection and mitigation of high traffic volume challenges, such as DDoS attacks targeted toward an ISP infrastructure [YFSF+11]. To understand the generality and applicability of our approach, we would like to apply it to a wider range of challenges, *e.g.*, human mistakes, in different network architectures, such as opportunistic networks. Furthermore, an implementation of aspects currently realised in simulation alone would be desirable, for example, using OpenFlow [MAB+08] to implement enhanced resilience functionality in network devices, which are controlled by policies. Furthermore, we would like to investigate approaches to transforming known successful resilience strategies into general *resilience patterns* that can be applied to well-understood challenges. These patterns could be distributed in a similar manner to detection signatures for intrusion detection systems, and configured specifically at deployment time.

Resilience mechanisms We have developed a number of resilience mechanisms that can be applied to a wide range of challenges. They span a number of stages of the D^2R^2+DR strategy and function at the network and service level. In particular, we have produced mechanisms to address malicious behaviour in networks, such as monetary-less cooperation incentives to mitigate selfish nodes in wireless mesh networks [PGLK10], game-theoretic approaches to protection against malware propagation [OOVM09], and an anomaly detection approach to detect and traceback attacks on encrypted protocols [FTV+10]. Furthermore, our mechanisms can be applied at different levels of the protocol stack in light of node and link failure, and include novel approaches to multi-path routing in multi-hop wireless networks [LSZB10] and algorithms for creating resilient large-scale overlay networks [GHK11]. In Section 5, we have highlighted the novel aspects of the mechanisms developed and their likely deployment timescales. An enemy of network resilience is complexity; and using multilevel network control has the potential to increase complexity and produce undesirable emergent behaviours. To address this problem, we have developed a cross-layer framework, which uses a formalism to evaluate the optimal layer to place resilience functionality, thus reducing replicated functionality at different layers.

Future work: A number of the resilience mechanisms could employed in a synergistic way between the network and service level. For example, the Perco Pastry mechanism that identifies redundant overlay connections in light of failures could be used in conjunction with the rope-ladder routing and Survivable Network Design (SND) mechanisms, which operate at the network level. Investigating how this could be done using our management architecture, based on information from the resilience metrics framework, could be an area for future work. The use of our cross-layer framework has focused on network and end-to-end resilience issues, future work should investigate how it could be applied to protocols implemented at the application layer.

Challenge detection Our understanding of the purpose of the *detect* stage of the $D^2R^2 +$ DR strategy has improved over the lifetime of the project. We understand its primary goal is to build situational awareness to inform decision making regarding remediation and recovery. A model of situational awareness has proved very useful for identifying the various aspects of detection and what their outputs should be. In short, we suggest *situational perception* should be based on multilevel network measurements and context information, which informs mechanisms for *situational comprehension* that aim to *detect* the presence of a challenge, *identify* (or characterise) it, and determine its *impact* on a network and services.

We have investigated what multilevel metrics should be measured for resilience, and which tools should be used to collect and distribute this information. We propose that output from our multilevel metrics framework can be used to support solving the difficult problem of determining what to measure. Furthermore, we suggest that causal end-to-end task-centric monitoring tools, such as X-Trace [FPK+07] are important for determining the root cause of challenges to complex multi-domain network services. To enable information sharing between components of our resilience architecture, we have developed a distributed publish-subscribe system that has persistent storage capabilities, called DISco.

To identify challenges, we propose an incremental multi-stage approach that enables rapid remediation to reduce the likelihood of challenges causing catastrophic failure. Subsequently, remediation can be refined using improved challenge identification mechanisms. To support this multi-stage approach, we have developed an challenge identification architecture, which can be implemented using model-driven fault localisation techniques, such as Incremental Hypothesis Updating (IHU) or the Chronicle Recognition System (CRS).

Future work: The third level of situational awareness is *situational projection*, *i.e.*, anticipating how a challenge's behaviour will unfold and impact a network and services. We have not investigated this important aspect of informing decision making for resilience; this an interesting and rich area for future research. Perhaps one of the most difficult aspects of using model-driven fault-localisation techniques, such as IHU and the CRS, is developing the models that characterise challenges. Developing tool support for this is an important area for future research, and could involve using machine learning techniques to derive them from observed operational behaviour, and taking output from our various toolsets to evaluate resilience metrics in order to identify pertinent symptoms that characterise a challenge.

Finally, our work on the project (and the framework) is ongoing in two main ways, which are not reflected in this deliverable. Our experimental evaluation of project results in a number

of case studies will deliver its findings at the end of the project, including reflections on the applicability of the metrics framework to opportunistic networks. We have not focused in this deliverable in any great detail on the outer learning loop of the $D^2R^2 + DR$ strategy – similarly, we anticipate results regarding this aspect at the end of the project. These findings will be documented in Deliverable D6.5, the final project deliverable, which will summarise our overall project results.

Appendix

In work package one, we have published two articles since the previous version of this deliverable, which relate to our resilience framework. They are appended to this deliverable.

P. Smith, D. Hutchison, J. P.G. Sterbenz, M. Schöller, A. Fessi, M. Karaliopoulos, C. Lac, and B. Plattner. Network resilience: a systematic approach. *IEEE Communications Magazine*, 49(7):88–97, July 2011

This publication summarises our resilience framework, and introduces its application to one of the experimentation scenarios that are being carried out.

M. Schöller, P. Smith, and D. Hutchison. Assessing Risk for Network Resilience. In *3rd International Workshop on Reliable Networks Design and Modeling, Budapest (RNDM 2011)*, Budapest, Hungary, October 2011

This paper presents our risk assessment process, and applies it to a wireless mesh network scenario.

References

[ABKM01] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles (SOSP)*, pages 131–145, October 2001.

[AGLL05] Dakshi Agrawal, James Giles, Kang-Won Lee, and Jorge Lobo. Policy ratification. In *6th IEEE Int. Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 223–232, Stockholm, Sweden, June 2005.

[AK02] A. Autenrieth and A. Kirstadter. Engineering end-to-end IP resilience using resilience-differentiated QoS. *IEEE Communications Magazine*, 40(1):50–57, January 2002.

[Bas00] Tim Bass. Intrusion detection systems and multisensor data fusion. *Commununications of the ACM*, 43:99–105, April 2000.

[BBF+10] R. Bruncak, N. Bohra, A. Fischer, S. Simpson, J. P. Rohrer, J. P.G. Sterbenz, D. Hutchison, and H. De Meer. Cross-layer optimization and multilevel resilience. ResumeNet Project Deliverable, August 2010.

[BKL+09] Chadi Barakat, Amir Krifa, Yann Labit, Imed Lassoued, Johan Mazel, Philippe Owezarski, and Kavé Salamatian. Implementation of adaptive traffic sampling and management, path performance monitoring and cooperative intrusion and attack-/anomaly detection techniques. Deliverable 3.2, The ECODE Project, October 2009.

[BLMR04] Arosha K. Bandara, Emil C. Lupu, Jonathan Moffett, and Alessandra Russo. A Goal-based Approach to Policy Refinement. In *POLICY '04: Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*, page 229, Washington, DC, USA, 2004. IEEE Computer Society.

[BLR+06] Arosha K. Bandara, Emil Lupu, Alessandra Russo, Naranker Dulay, Morris Sloman, Paris Flegkas, Marinos Charalambides, and George Pavlou. Policy Refinement for IP Differentiated Services Quality of Service Management. *IEEE Transactions on Network and Service Management*, 3(2), 2006.

[CD00] Marie-Odile Cordier and Christophe Dousson. Alarm driven monitoring based on chronicles. In *4th Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCESS)*, pages 286–291, Budapest, Hungary, June 2000.

[CEM03] L. Capra, W. Emmerich, and C. Mascolo. CARISMA: Context-Aware Reflective Middleware System for Mobile Applications. *IEEE Transactions on Software Engineering*, 29(10):929–945, October 2003.

[CFSD90] J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin. Simple Network Management Protocol (SNMP). RFC 1157 (Historic), May 1990.

[CMH+07] P. Cholda, A. Mykkeltveit, B.E. Helvik, O.J. Wittner, and A. Jajszczyk. A survey of resilience differentiation frameworks in communication networks. *IEEE Communications Surveys Tutorials*, 9(4):32–55, 2007.

[Coo07] M. J. Cooper. *Event Tree Analysis*. Brunel Technical Press, 2007.

[Cro10] J. Crowcroft. Internet Failures: an Emergent Sea of Complex Systems and Critical Design Errors? *Computer Journal*, 53(10):1752–1757, January 2010.

[CSN02] K. Chen, S. H. Shah, and K. Nahrstedt. Cross-layer Design for Data Accessibility in Mobile Adhoc Networks. *Wireless Personal Communications*, 21:49–75, 2002.

[D+01] Nicodemos Damianou et al. The Ponder policy specification language. In *Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks (Policy)*, pages 18–39, Bristol, U.K., January 2001. IEEE Computer Society.

[DCF07] H. Debar, D. Curry, and B. Feinstein. The Intrusion Detection Message Exchange Format (IDMEF). RFC 4765 (Experimental), March 2007.

[DDK+10] B. Delosme, C. Dousson, N. Kheir, C. Lac, and P. Smith. Service surveillance and detection of challenging situation (interim). ResumeNet Project Deliverable, July 2010.

[DGP+10] C. Doerr, E. Gourdin, G. Popa, J. Omic, J. P.G. Sterbenz, J. P. Rohrer, T. Taleb, and P. Van Mieghem. Second draft on defensive measures for resilient networks. ResumeNet Project Deliverable, August 2010.

[DHH+11] C. Doerr, J. M. Hernandez, R. Holz, D. Hutchison, A. Smeureanu, P. Smith, James P. G. Sterbenz, and P. Van Mieghem. Defining metrics for resilient networking (Final). ResumeNet Project Deliverable, September 2011.

[DM09] C. Dousson and P. Le Maigat. Controlling the Event Streams to Become More Autonomous. In *5th International Conference on Autonomic and Autonomous Systems (ICAS)*, pages 100–105, Valencia, Spain, April 2009.

[DMH10] C. Doerr and J. Martin-Hernandez. A Computational Approach to Multi-level Analysis of Network Resilience. In *Proceedings of the 2010 Third International Conference on Dependability*, DEPEND '10, pages 125–132, Washington, DC, USA, 2010. IEEE Computer Society.

[DSS+10] C. Doerr, M. Schöller, P. Smith, N. Kheir, J. Lessmann, and C. Rohner. Remediation, recovery, and measurement framework. ResumeNet Project Deliverable, September 2010.

[DZK06] Karim M El Defrawy, Magda S. El Zarki, and Mohamed M. Khairy. Proposal for a cross-layer coordination framework for next generation wireless systems. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing (IWCMC)*, pages 141–146, Vancouver, British Columbia, Canada, July 2006.

[ECD+01] C. Efstratiou, K. Cheverst, N. Davies, A. Friday, and L. Yr. Architectural Requirements for the Effective Support of Adaptive Mobile Applications. In *Second International Conference on Mobile Data Management (MDM)*, Hong Kong, January 2001.

[ECO] The EU-funded ECODE Project. http://www.ecode-project.eu/.

[End95] Mica R. Endsley. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors Journal*, 37(1):32–64, March 1995.

[ENI09] ENISA Virtual Working Group on Network Providers' Resilience Measures. Network resilience and security: Challenges and measures. Technical Report v1.0, European Network and Information Security Agency (ENISA), December 2009.

[Eur10] European Network and Information Security Agency (ENISA). Measurement Frameworks and Metrics for Resilient Networks and Services: Challenges and Recommendations. White paper, 2010.

[Eur11] European Network and Information Security Agency (ENISA). Cyber Europe 2010 – Evaluation Report. White paper, April 2011.

[FAHF10] A. Fischer, Y. Al-Hazmi, and A. Fessi. P2P Overlays and Virtualization for Service Resilience. ResumeNet Project Deliverable, August 2010.

[FFK+10] M. Fry, M. Fischer, M. Karaliopoulos, P. Smith, and D. Hutchison. Challenge Identification for Network Resilience. In *6th Euro-NF Conference on Network Generation Internet (NGI 2010)*, pages 1–8, Paris, France, June 2010.

[Fis10] Matthias Fischaleck. Discovery of malicious nodes in p2p-overlay-networks using the X-Trace framework. Diploma thesis, Faculty of Informatics and Mathematics, Computer Networks and Computer Communication, University of Passau, Germany, 2010.

[FNKC07] Ali Fessi, Heiko Niedermayer, Holger Kinkelin, and Georg Carle. A cooperative SIP infrastructure for highly reliable telecommunication services. In *ACM Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, pages 29–38, New York, NY, USA, July 2007.

[FPK+07] R. Fonseca, G. Porter, R. H. Katz, S. Shenke, and I. Stoica. X-trace: A pervasive network tracing framework. In *4th USENIX Symposium on Networked Systems Design & Implementation*, pages 271–284, Santa Clara, CA, USA, June 2007.

[FTV+10] Z. Fadlullah, T. Taleb, A. Vasilakos, M. Guizani, and N. Kato. DTRAB: Combating Against Attacks on Encrypted Protocols through Traffic-Feature Analysis. *IEEE/ACM Transactions on Networking*, 18(4):1234–1247, August 2010.

[Gam09] T. Gamer. Anomaly-based identification of large-scale attacks. In *28th IEEE conference on Global telecommunications (GLOBECOM)*, pages 6638–6643, Honolulu, Hawaii, USA, November-December 2009.

[GHK11] S. Günther, R. Holz, and N. Kammenhuber. Overlay-based end-to-end connectivity. ResumeNet Project Deliverable, March 2011.

[Gou10] E. Gourdin. A mixed-integer model for the sparsest cut problem. In *International Symposium on Combinatorial Optimization*, pages 111–118, Hammamet, Tunisia, March 2010.

[HLOS06] Shawn Hernan, Scott Lambert, Tomasz Ostwald, and Adam Shostack. Uncover Security Design Flaws Using The STRIDE Approach. *MSDN Magazine*, November 2006.

[IBPR08] J. Ishmael, S. Bury, D. P. Pezaros, and N J.P. Race. Deploying Rural Community Wireless Mesh Networks. *IEEE Internet Computing Magazine*, 12(4):22–29, July–August 2008.

[JKR02] Jaeyeon Jung, Balachander Krishnamurthy, and Michael Rabinovich. Flash crowds and denial of service attacks: characterization and implications for CDNs and web sites. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 293–304, New York, NY, USA, 2002. ACM.

[JPS08] Abdul Jabbar, Erik Perrins, and James P.G. Sterbenz. A cross-layered protocol architecture for highly-dynamic multihop airborne telemetry networks. In *International Telemetering Conference (ITC) 2008*, San Diego, CA, October 2008.

[JRO+09] A. Jabbar, J. P. Rohrer, A. Oberthaler, E. K. Çetinkaya, V .S. Frost, and J. P.G. Sterbenz. Performance comparison of weather disruption-tolerant cross-layer routing algorithms. In *28th IEEE Conference on Computer Communications (INFOCOM)*, pages 1143–1151, Rio de Janeiro, Brazil, April 2009.

[KBM+03] Anas Abou El Kalam, Salem Benferhat, Alexandre Miège, Rania El Baida, Frédéric Cuppens, Claire Saurel, Philippe Balbiani, Yves Deswarte, and Gilles Trouessin. Organization Based Access Control. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks (Policy)*, Lake Como, Italy, June 2003.

[KG05] A. Kodian and W.D. Grover. Multiple-quality of protection classes including dual-failure survivable services in p-cycle networks. In *2nd International Conference on Broadband Networks (BroadNets)*, pages 231–240, Boston, MA, USA, October 2005.

[KK02] M. Khedr and A. Karmouch. Acan – Ad hoc Context Aware Network. In *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 12–15, Winnipeg, Manitoba, Canada, May 2002.

[KMP+05] Cornelia Kappler, Paulo Mendes, Christian Prehofer, Petteri Pöyhönen, and Di Zhou. A framework for self-organized network composition. In Michael Smirnov, editor, *Autonomic Communication*, volume 3457 of *Lecture Notes in Computer Science*, pages 261–266. Springer Berlin / Heidelberg, 2005.

[KS95] I. Katzela and M. Schwartz. Schemes for fault identification in communication networks. *IEEE/ACM Transactions on Networking*, 3(6):753 –764, December 1995.

[LeF01] William LeFebvre. CNN.com: facing a world crisis, 2001. http://www.tcsa.org/lisa2001/cnn.txt.

[Lis10] Mario Lischka. Dynamic Obligation Specification and Negotiation. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 155–162, Osaka, Japan, April 2010.

[LSZB10] J. Lessman, M. Schöller, F. Zdarsky, and A. Banchs. Rope ladder routing: position-based multipath routing for wireless mesh networks. In *Proc. 2nd IEEE WoWMoM Workshop on Hot Topics in Mesh Networking*, pages 1–6, Montreal, QC, Canada, June 2010.

[MAB+08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: enabling innovation in campus networks. *SIGCOMM Computer Communication Review*, 38:69–74, March 2008.

[MD03] Benjamin Morin and Hervé Debar. Correlation of intrusion symptoms: an application of chronicles. In *RAID'03 : Proceedings of the 6th International Conference on Recent Advances in Intrusion Detection*, pages 94–112, Pittsburgh, PA, USA, September 2003.

[NA08] Thuy T. T. Nguyen and Grenville J. Armitage. A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys and Tutorials*, 10(1–4):56–76, 2008.

[NCC08] RIPE NCC. YouTube Hijacking: A RIPE NCC RIS case study, 2008. http://www.ripe.net/news/study-youtube-hijacking.html.

[OMN] OMNeT++ Website. OMNeT++. http://www.omnetpp.org/. Accessed March 2010.

[OOVM09] J. Omic, A. Orda, and P. Van Mieghem. Protecting against network infections: A game theoretic perspective. In *Proc. 28th IEEE INFOCOM*, pages 1485–1493, Rio de Janeiro, Brazil, April 2009.

[Pep09] Ivan Pepelnjak. Oversized AS paths: Cisco IOS bug details, 2009. http://blog.ioshints.info/2009/02/oversized-as-paths-cisco-ios-bug.html.

[Pet09] Stefan Peters. Identifizierung von Routingfehlverhalten in IPv4 Netzwerken mit Hilfe des X-Trace Frameworks. Bachelor Thesis, 2009.

[PGLK10] Gabriel Popa, Eric Gourdin, Franck Legendre, and Merkouris Karaliopoulos. On maximizing collaboration in Wireless Mesh Networks without monetary incentives. In *8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 402–411, Avignon, France, May 2010.

[RD01] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, Heidelberg, Germany, November 2001.

[RJS09] Justin P. Rohrer, Abdul Jabbar, and James P. G. Sterbenz. Path diversification: A multipath resilience mechanism. In *Proceedings of the 7th International Workshop on the Design of Reliable Communication Networks (DRCN)*, pages 343–351, Washington, DC, USA, October 2009.

[RPFL10] C. Rohner, G. Popa, A. Fessi, and C. Lac. Interim report on experimental evaluation of resilient networking: scenarios and experimentation facilities. ResumeNet Project Deliverable, August 2010.

[SB11] P. Sommer and I. Brown. Reducing Systemic Cybersecurity Risk. White paper, January 2011.

[ScH+11] James P. G. Sterbenz, Egemen K. Çetinkaya, Mahmood A. Hameed, Abdul Jabbar, Qian Shi, and Justin P. Rohrer. Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation (invited paper). *Springer Telecommunication Systems*, 2011. accepted March 2011.

[SF09] Alberto Schaeffer-Filho. *Supporting Management Interaction and Composition of Self-Managed Cells*. PhD thesis, Imperial College London, 2009.

[SFM+10] P. Smith, M. Fry, S. Martin, L. Chiarello, M. Fischer, C. Rohner, G. Popa, H. De Meer, A. Fischer, and N. Bohra. New challenge detection approaches. ResumeNet Project Deliverable, September 2010.

[SFSM11] A. Schaeffer-Filho, P. Smith, and A. Mauthe. Policy-driven Network Simulation: a Resilience Case Study. In *26th ACM Symposium on Applied Computing (SAC 2011)*, pages 492–497, Taichung, Taiwan, March 2011.

[SFSM+12] Alberto Schaeffer-Filho, Paul Smith, Andreas Mauthe, David Hutchison, Yue Yu, and Michael Fry. A Framework for the Design and Evaluation of Network Resilience Management. In *IEEE/IFIP Network Operations and Management Symposium (under submission)*, Maui, Hawaii, USA, April 2012.

[SHc+10] J. P.G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Elsevier Computer Networks: Special Issue on Resilient and Survivable Networks*, 54(8):1243–1342, June 2010.

[SHS+11] P. Smith, D. Hutchison, J. P.G. Sterbenz, M. Schöller, A. Fessi, M. Karaliopoulos, C. Lac, and B. Plattner. Network resilience: a systematic approach. *IEEE Communications Magazine*, 49(7):88–97, July 2011.

[SPS+01] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. Hash-based IP traceback. In *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*, pages 3–14. San Diego, California, US, August 2001.

[SS04a] G. Sahin and S. Subramaniam. Providing quality-of-protection classes through control-message scheduling in DWDM mesh networks with capacity sharing. *IEEE Journal on Selected Areas in Communications*, 22(9):1846–1858, November 2004.

[SS04b] M. Steinder and A. S. Sethi. Probabilistic fault diagnosis in communication systems through incremental hypothesis updating. *Computer Networks*, 45(4):537 – 562, 2004.

[SS09] M. Schöller and P. Smith. Understanding challenges and their impact on network resilience. ResumeNet Project Deliverable, October 2009.

[SSFA+10] Paul Smith, Alberto Schaeffer-Filho, Azman Ali, Marcus Schöller, Nizar Kheir, Andreas Mauthe, and David Hutchison. Strategies for network resilience: capitalising on policies. In *Proceedings of the 4th international conference on Autonomous infrastructure, management and security*, AIMS'10, pages 118–122, Berlin, Heidelberg, 2010. Springer-Verlag.

[SSH11] M. Schöller, P. Smith, and D. Hutchison. Assessing Risk for Network Resilience. In *3rd International Workshop on Reliable Networks Design and Modeling, Budapest (RNDM 2011)*, Budapest, Hungary, October 2011.

[SSSF+11] M. Schöller, P. Smith, A. Schaeffer-Filho, N. Kheir, and S. Natouri. Policies for resilience (Final). ResumeNet Project Deliverable, August 2011.

[SWKA00] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for IP traceback. *SIGCOMM Comput. Commun. Rev.*, 30(4):295–306, 2000.

[THAB10] T. Taleb, Y. Hadjadj-Aoul, and A. Benslimane. Integrating security with QoS in next generation networks. In *IEEE Global Communication Conference (GLOBECOM)*, Miami, FL, USA, December 2010.

[Tow88] Patrick A. Townson. The great fire. Telecom Digest, vol.8 iss.76, 10 May 1988, May 1988. http://massis.lcs.mit.edu/archives/history/fire.in.chicago.5-88.

[TSB+06] George Tadda, John J. Salerno, Douglas Boulware, Michael Hinman, and Samuel Gorton. Realizing situation awareness within a cyber environment. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, Orlando, FL, USA, April 2006.

[Ves87] William E. Vesely. *Fault Tree Handbook*. Nuclear Regulatory Commission, ISBN-10: 0160055822, 1987.

[WR03] Q. Wang and M. A.A. Rgheff. Cross-layer signalling for next generation wireless systems. *IEEE Wireless Communication and Networking*, pages 1084–1089, 2003.

[YFSF+11] Y. Yu, M. Fry, A. Schaeffer-Filho, P. Smith, and D. Hutchison. An Adaptive Approach to Network Resilience: Evolving Challenge Detection and Mitigation. In *8th International Workshop on Design of Reliable Communication Networks (DRCN) (to appear)*, Krakow, Poland, October 2011.

[Zmi09] Earl Zmijewski. Longer is not always better, 2009. http://www.renesys.com/blog/2009/02/longer-is-not-better.shtml.

Network Resilience: A Systematic Approach

Paul Smith, Lancaster University
David Hutchison, Lancaster University
James P. G. Sterbenz, University of Kansas and Lancaster University
Marcus Schöller, NEC Laboratories Europe
Ali Fessi, Technische Universität München
Merkouris Karaliopoulos, NKU Athens
Chidung Lac, France Telecom (Orange Labs)
Bernhard Plattner, ETH Zurich

ABSTRACT

The cost of failures within communication networks is significant and will only increase as their reach further extends into the way our society functions. Some aspects of network resilience, such as the application of fault-tolerant systems techniques to optical switching, have been studied and applied to great effect. However, networks — and the Internet in particular — are still vulnerable to malicious attacks, human mistakes such as misconfigurations, and a range of environmental challenges. We argue that this is, in part, due to a lack of a holistic view of the resilience problem, leading to inappropriate and difficult-to-manage solutions. In this article, we present a systematic approach to building resilient networked systems. We first study fundamental elements at the framework level such as metrics, policies, and information sensing mechanisms. Their understanding drives the design of a distributed multilevel architecture that lets the network defend itself against, detect, and dynamically respond to challenges. We then use a concrete case study to show how the framework and mechanisms we have developed can be applied to enhance resilience.

INTRODUCTION

Data communication networks are serving all kinds of human activities. Whether used for professional or leisure purposes, for safety-critical applications or e-commerce, the Internet in particular has become an integral part of our everyday lives, affecting the way societies operate. However, the Internet was not intended to serve all these roles and, as such, is vulnerable to a wide range of challenges. Malicious attacks, software and hardware faults, human mistakes (e.g., software and hardware misconfigurations), and large-scale natural disasters threaten its normal operation.

Resilience, the ability of a network to defend against and maintain an acceptable level of service in the presence of such challenges, is viewed today, more than ever before, as a major requirement and design objective. These concerns are reflected in, among other ways, in the Cyber Storm III exercise carried out in the United States in September 2010, and the "cyber stress tests" conducted in Europe by the European Network and Information Security Agency (ENISA) in November 2010; both aimed precisely at assessing the resilience of the Internet, this "critical infrastructure used by citizens, governments, and businesses."

Resilience evidently cuts through several thematic areas, such as information and network security, fault tolerance, software dependability, and network survivability. A significant body of research has been carried out around these themes, typically focusing on specific mechanisms for resilience and subsets of the challenge space. We refer the reader to Sterbenz *et al.* [1] for a discussion on the relation of various resilience disciplines, and to a survey by Cholda *et al.* [2] on research work for network resilience.

A shortcoming of existing research and deployed systems is the lack of a systematic view of the resilience problem, that is, a view of how to engineer networks that are resilient to challenges that transcend those considered by a single thematic area. A non-systematic approach to understanding resilience targets and challenges (e.g., one that does not cover thematic areas) leads to an impoverished view of resilience objectives, potentially resulting in ill suited solutions. Additionally, a patchwork of resilience mechanisms that are incoherently devised and deployed can result in undesirable behavior and increased management complexity under chal-

0163-6804/11/$25.00 © 2011 IEEE

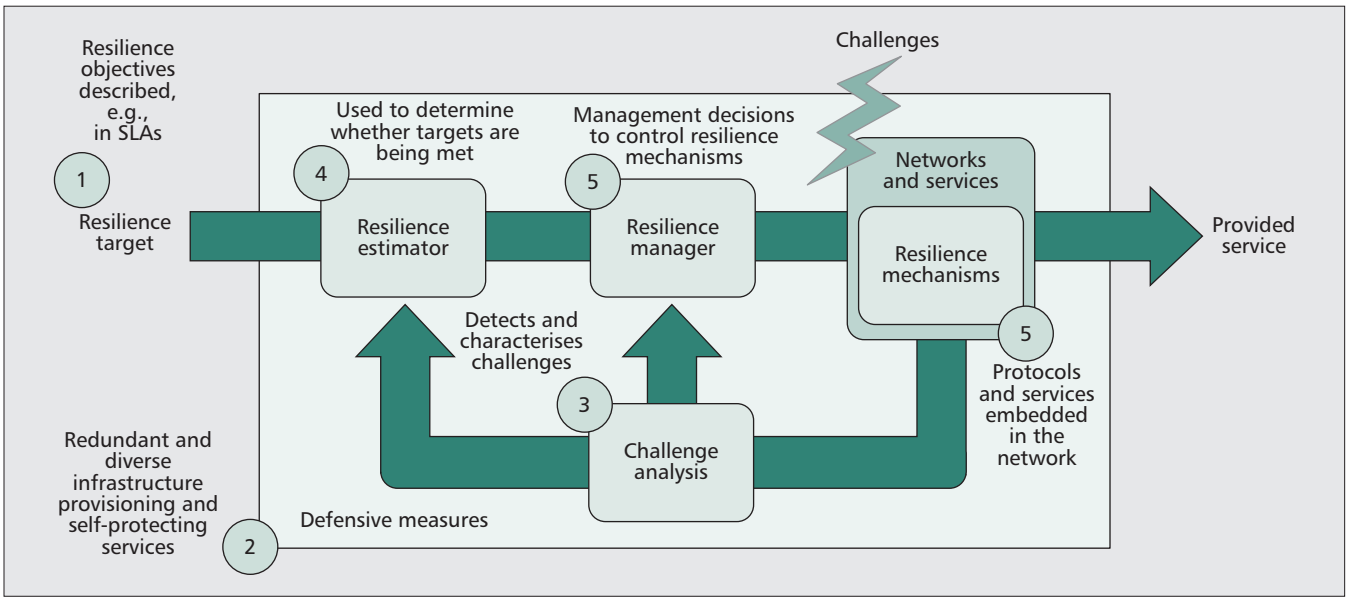


Figure 1. *The resilience control loop: derived from the real-time component of the D²R² + DR resilience strategy.*

lenge conditions, encumbering the overall network management task [3].

The EU-funded *ResumeNet* project argues for resilience as a critical and integral property of networks. It advances the state of the art by adopting a systematic approach to resilience, which takes into account the wide-variety of challenges that may occur. At the core of our approach is a coherent resilience framework, which includes implementation guidelines, processes, and toolsets that can be used to underpin the design of resilience mechanisms at various levels in the network. In this article, we first describe our framework, which forms the basis of a systematic approach to resilience. Central to the framework is a control loop, which defines necessary conceptual components to ensure network resilience. The other elements — a risk assessment process, metrics definitions, policy-based network management, and information sensing mechanisms — emerge from the control loop as necessary elements to realize our systematic approach. We show how these elements drive the design of a novel architecture and mechanisms for resilience. Finally, we illustrate these mechanisms in a concrete case study being explored in ResumeNet: a future Internet smart environments application.

Framework for Resilience

Our resilience framework builds on work by Sterbenz *et al.* [1], whereby a number of resilience principles are defined, including a resilience strategy, called D²R² + DR: Defend, Detect, Remediate, Recover, and Diagnose and Refine. The strategy describes a real-time control loop to allow dynamic adaptation of networks in response to challenges, and a non-real time control loop that aims to improve the design of the network, including the real-time loop operation, reflecting on past operational experience.

The framework represents our systematic approach to the engineering of network resilience. At its core is a control loop comprising a number of conceptual components that realize the real-time aspect of the D²R² + DR strategy, and consequently implement network resilience. Based on the resilience control loop, other necessary elements of our framework are derived, namely resilience metrics, understanding challenges and risks, a distributed information store, and policy-based management. The remainder of this section describes the resilience control loop, then motivates the need for these framework elements.

Resilience Control Loop

Based on the real-time component of the D²R² + DR strategy, we have developed a *resilience control loop*, depicted in Fig. 1, in which a controller modulates the input to a system under control in order to steer the system and its output towards a desired reference value. The control loop forms the basis of our systematic approach to network resilience — it defines necessary components for network resilience from which the elements of our framework, discussed in this section, are derived. Its operation can be described using the following list; items correspond to the numbers shown in Fig. 1:

1. The reference value we aim to achieve is expressed in terms of a *resilience target*, which is described using resilience metrics. The resilience target reflects the requirements of end users, network operators, and service providers.

2. *Defensive measures* need to be put in place *proactively* to alleviate the impact of *challenges* on the network, and maintain its ability to realize the resilience target. A process for identifying the challenges that should be considered in this defense step of the strategy (e.g., those happening more frequently and having high impact) is necessary.

3. Despite the defensive measures, some challenges may cause the service delivered to users to deviate from the resilience target. These challenges could include unforeseen attacks or mis-

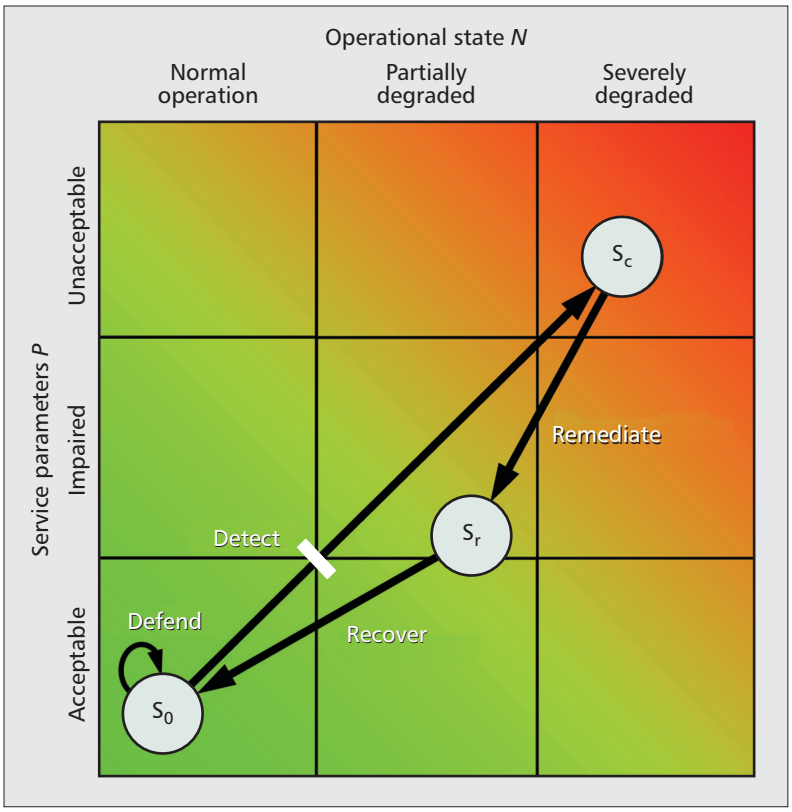


Figure 2. *Resilience state space*

configurations. *Challenge analysis* components detect and characterize them using a variety of information sources.

4. Based on output from challenge analysis and the state of the network, a *resilience estimator* determines whether the resilience target is being met. This measure is based on resilience metrics, and is influenced by the effectiveness of defense and remediation mechanisms to respond to challenges.

5. Output from the resilience estimator and challenge analysis is fed to a *resilience manager*. It is then its responsibility to control *resilience mechanisms* embedded in the network and service infrastructure, to preserve the target service provision level or ensure its graceful degradation. This adaptation is directed using *resilience knowledge*, not shown in Fig. 1, such as policies and challenge models. We anticipate a cost of remediation in terms of a potentially unavoidable degradation in quality of service (QoS), which should not be incurred if the challenge abates. Consequently, the network should aim to recover to normal operation after a challenge has ceased.

The purpose of the background loop in the D^2R^2 + DR strategy is to improve the operation of the resilience control loop such that it meets an idealized system operation. This improvement could be in response to market forces, leading to new resilience targets, new challenges, or suboptimal performance. The *diagnose* phase identifies areas for improvement, including defense, that are enacted through *refinement*. In reality, and for the foreseeable future, we anticipate this outer loop to be realized with human intervention.

Resilience Metrics

Defining a resilience target requires appropriate metrics. Ideally, we would like to express the resilience of a network using a single value, $\Re$, in the interval [0,1], but this is not a simple problem because of the number of parameters that contribute to and measure resilience, and due to the multilayer aspects in which each level of resilience (e.g., resilient topology) is the foundation for the next level up (e.g., resilient routing). We model resilience as a two-dimensional state space in which the vertical axis $\mathbb{P}$ is a measure of the service provided when the operational state $\mathbb{N}$ is challenged, as shown in Fig. 2. Resilience is then modeled as the trajectory through the state as the network goes from delivering acceptable service under normal operations S_0 to degraded service S_c. Remediation improves service to S_r and recovery returns to the normal state S_0. We can measure resilience at a particular service level as the area under this trajectory, $\mathbb{R}$.

We have developed a number of tools for evaluating network resilience. For example, we use MATLAB or ns-3 simulation models to measure the service at each level and plot the results under various challenges and attacks, as in Fig. 2, where each axis is an objective function of the relevant parameters [4]. Furthermore, we have developed the *Graph Explorer* tool [5] that takes as input a network topology and associated traffic matrix, a description of challenges, and a set of metrics to be evaluated. The result of the analysis is a series of plots that show the *metric envelope* values (m_i(min), m_i(max)) for each specified metric m_i, and topology maps indicating the resilience across network regions.

Figure 3 shows an example of the resilience of the European academic network GÉANT2 to link failures. The set of plots in Fig. 3a show metric envelopes at different protocol levels — the aim is to understand how jitter responds in comparison with metrics at other levels, such as queue length and connectivity. Surprisingly, jitter is not clearly related to queue length, and a monotonic increase in path length does not yield a similar increase in queue length for all scenarios of link failures. In fact, the fourth link failure disconnects a region of the network; whereas up to three failures, the heavy use of a certain path resulted in increasing queue lengths and jitter. The partition increases path length, because route lengths are set to infinity, and decreases connectivity, which is accompanied by a reduction in jitter, shown with the blue arrows in Fig. 3a. The topology map in Fig. 3b highlights the vulnerability of regions of GÉANT2 with a heat map, which can be used by network planners.

Our framework for resilience metrics (i.e., the multilevel two-dimensional state space and the use of metric envelopes) can be used to understand the resilience of networks to a broad range of challenges, such as misconfigurations, faults, and attacks. The ability to evaluate a given network's resilience to a specific challenge is limited by the capability of the tools to create complex challenge scenarios — this is an area for further work, in which our effort should be focused on modeling pertinent high-impact challenges.

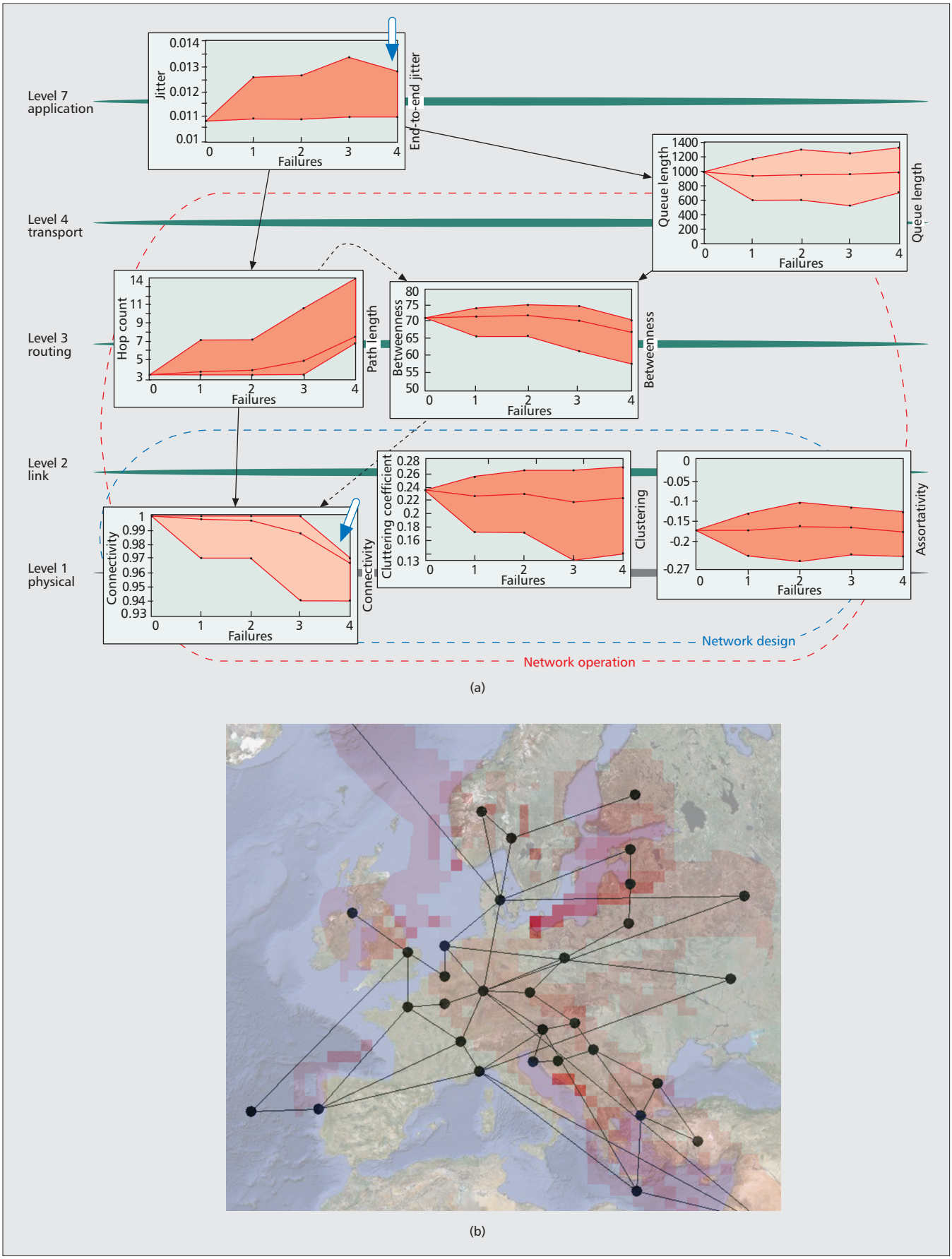


Figure 3. *Example output from the Graph Explorer, developed in the ResumeNet project: a) plots showing the relationship between metrics at various layers in response to link failures on the GÉANT2 topology; b) a heat map showing vulnerable regions of the topology with respect to a given set of metrics. Reprinted from [5].*

We advocate the use of a policy-based management framework to define the behavior of real-time loop instantiations. Consequently, the implementation of resilience mechanisms can be decoupled from the resilience management strategies, which are expressed in policies.

Understanding Challenges and Risks

Engineering resilience has a monetary cost. To maximize the effectiveness of the resources committed to resilience, a good understanding of the challenges a network may face is mandatory. We have developed a structured risk assessment approach that identifies and ranks challenges in line with their probability of occurrence and their impact on network operation (i.e., how disruptive they are to the provision of its services). The approach should be carried out at the stage of network design when proactive defensive measures are deployed, and repeated regularly over time as part of the process of network improvements.

Central to determining the impact of a challenge is to identify the critical services the network provides and the cost of their disruption: a measure of *impact*. Various approaches can be used to identify the critical services, such as discussion groups involving the network's stakeholders. Networked systems are implemented via a set of dependent subsystems and services (e.g., web and Session Initiation Protocol [SIP] services rely on Domain Name Service [DNS]). To identify whether challenges will cause a degradation of a service, it is necessary to explicate these dependencies.

The next phase is to identify the occurrence probabilities of challenges (*challenge_prob*). Some challenges will be unique to a network's context (e.g., because of the services it provides), while others will not. In relation to these challenges, shortcomings of the system (e.g., in terms of faults) should be identified. The aim is to determine the probability that a challenge will lead to a failure (*fail_prob*). We can use tools, such as our Graph Explorer, analytical modeling, and previous experience (e.g., in advisories) to help identify these probabilities. Given this information, a measure of *exposure* can be derived using the following equation:

$$exposure = (challenge\_prob \times fail\_prob) \times impact$$

With the measures of exposure at hand, resilience resources can be targeted at the challenges that are likely to have the highest impact.

Information Sources and Sharing for Resilience

For the most part, network management decisions are made based on information obtained from monitoring systems in the network (e.g., via Simple Network Management Protocol [SNMP]). However, to be able to make autonomic decisions about the nature of a wide range of challenges and how to respond to them — a necessary property of resilient networks — a broader range of information needs to be used. In addition to traditional network monitoring information, *context information*, which is sometimes "external" to the system can be used. Earlier work has demonstrated how the use of weather information, an example of context, improves the resilience of millimeter-wave wireless mesh networks, which perform poorly in heavy rain [4]. Also, in addition to *node-centric* monitoring tools, such as NetFlow and SNMP, task-centric tools can be used to determine the root cause of failures. For example, X-trace [6] is a promising task-centric monitoring approach that can be used to associate network and service state (e.g., router queue lengths and DNS records) with service requests (e.g., retrieving a web page). This *multilevel* information can then be used to determine the root causes of failures.

We are developing a Distributed Store for Challenges and their Outcome (DISco), which uses a publish-subscribe messaging pattern to disseminate information between subsystems that realize the real-time loop. Such information includes actions performed to detect and remediate challenges. Information sources may report more data than we can afford or wish to relay on the network, particularly *during* challenge occurrences. DISco is able to aggregate information from multiple sources to tackle this problem. Decoupling information sources from components that use them allows adaptation of challenge analysis components without needing to modify information sources. To assist the two phases of the outer loop, DISco employs a distributed peer-to-peer storage system for longer-term persistence of data, which is aware of available storage capacity and demand.

Policies for Resilience

We advocate the use of a policy-based management framework to define the behavior of real-time loop instantiations. Consequently, the implementation of *resilience mechanisms* can be decoupled from the *resilience management* strategies, which are expressed in policies. This has two immediate benefits: the nature of challenges changes over time — management strategies can be adapted accordingly without the need for network down-time; and policies allow network operators to clearly express when they would like to intervene in the network's operation (e.g., when a remediation action needs to be invoked).

Research outcomes from the policy-based management field can help address the complexities of resilience management [7]. A difficult task is deriving implementable policies from high-level resilience requirements, say, expressed in service level agreements (SLAs). With appropriate modifications, techniques for *policy refinement* can be used to build tools to automate aspects of this process. Policy-based learning, which relies on the use of logical rules for knowledge representation and reasoning, is being exploited to assist with the improvement stages of our strategy. Techniques for *policy ratification* are currently used to ensure that invocation of different resilience strategy sets does not yield undesirable conflicting behavior. Conflicts can occur horizontally between components that realize the resilience control loop, and vertically across protocol levels. For example, a mechanism that replicates a service using virtualization techniques at the service level could conflict with a mechanism that is rate-limiting traffic at the network level. Example policies of this sort are shown in Fig. 4. So that these forms of conflict can be detected, Agrawal *et al.* [8] provide a theoretical foundation for conflict resolution that needs to be extended with domain-specific knowledge, for example, regarding the nature of resilience mechanisms.

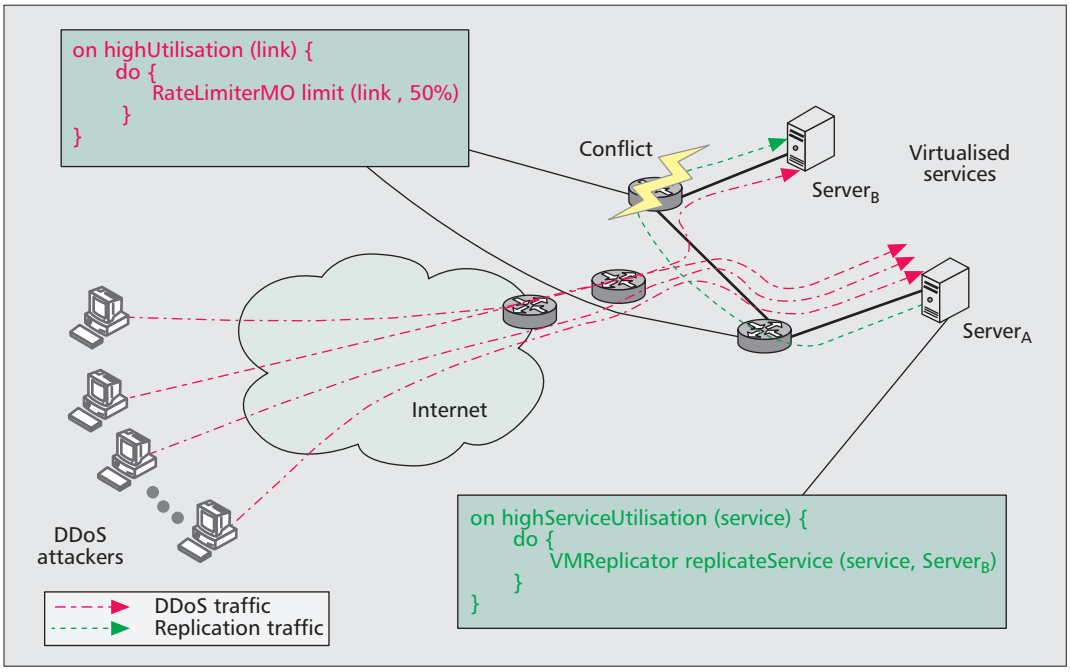


Figure 4. *Potentially conflicting policies at the service level (the replication of a service) and the network level (rate-limiting traffic) that could be triggered by the same challenge, such as a distributed denial of service (DDoS) attack. Rate limiting traffic could cause the replication to fail.*

Since challenges may vary broadly from topology-level link failures to application-level malware, defensive measures against anticipated high-impact challenges need to be applied at different levels and locations.

Defense and Dynamic Adaptation Architecture

In this section, we describe a set of defensive mechanisms and an architecture that realize our systematic approach to resilience, described earlier. The architecture, shown in Fig. 5, consists of several subsystems implementing the various tasks of the communication system as well as the challenge detection components and adaptation capabilities. The behavior of all these subsystems is directed by the *resilience manager* using policies, which are held in a *resilience knowledge base*. Central to this architecture is DISco, which acts as a publish-subscribe and persistent storage system, containing information regarding ongoing detection and remediation activities. From an implementation perspective, based on the deployment context, we envisage components of the architecture to be distributed (e.g., in an Internet service provider [ISP] network) or functioning entirely on a single device (e.g., nodes in a delay-tolerant network).

Defensive Measures

As a first step, defensive measures need to be put in place to alleviate the impact of challenges on the network. Since challenges may vary broadly from topology-level link failures to application-level malware, defensive measures against anticipated high-impact challenges need to be applied at different levels and locations: in the network topology design phase, and within protocols; across a network domain, as well as at individual nodes. Defensive measures can either prevent a challenge from affecting the system or contain erroneous behavior within a subsystem in such a way that the delivered service still meets its specification. A selection of defensive measures developed in the ResumeNet project is shown in Table 1.

Detection Subsystems

The second step is to detect challenges affecting the system leading to a deviation in delivered service. We propose an incremental approach to *challenge analysis*. Thereby, the understanding about the nature of a challenge evolves as more inputs become available from a variety of information sources. There are two apparent advantages of this incremental approach. First, it readily accommodates the varying computational overhead, timescales, and potentially limited accuracy of current detection approaches [9]. Second, relatively lightweight detection mechanisms that are always on can be used to promptly initiate remediation, thus providing the network with a first level of protection, while further mechanisms are invoked to better understand the challenge and improve the network response. Lightweight detection mechanisms can be driven by *local* measurements carried out in the immediate neighborhood of affected nodes.

For example, consider high-traffic volume challenges, such as a DDoS attack or a flash crowd event. Initially, always-on simple queue monitoring could generate an *alarm* if queue lengths exceed a threshold for a sustained period. This could trigger the rate limiting of links associated with high traffic volumes. More expensive traffic flow classification could then be used to identify and block malicious flows, consequently not subjecting benign flows to rate limiting. *Challenge models*, shown in Fig. 5, describe symptoms of challenges and drive the analysis process. They can be used to initially identify broad classes of challenge, and later to refine identification to more specific instances.

We are currently evaluating our generic approach to resilience through concrete study cases that cover a range of future networking paradigms: wireless mesh and delay tolerant networks, peer-to-peer voice conferencing and service provision over heterogeneous smart environments.

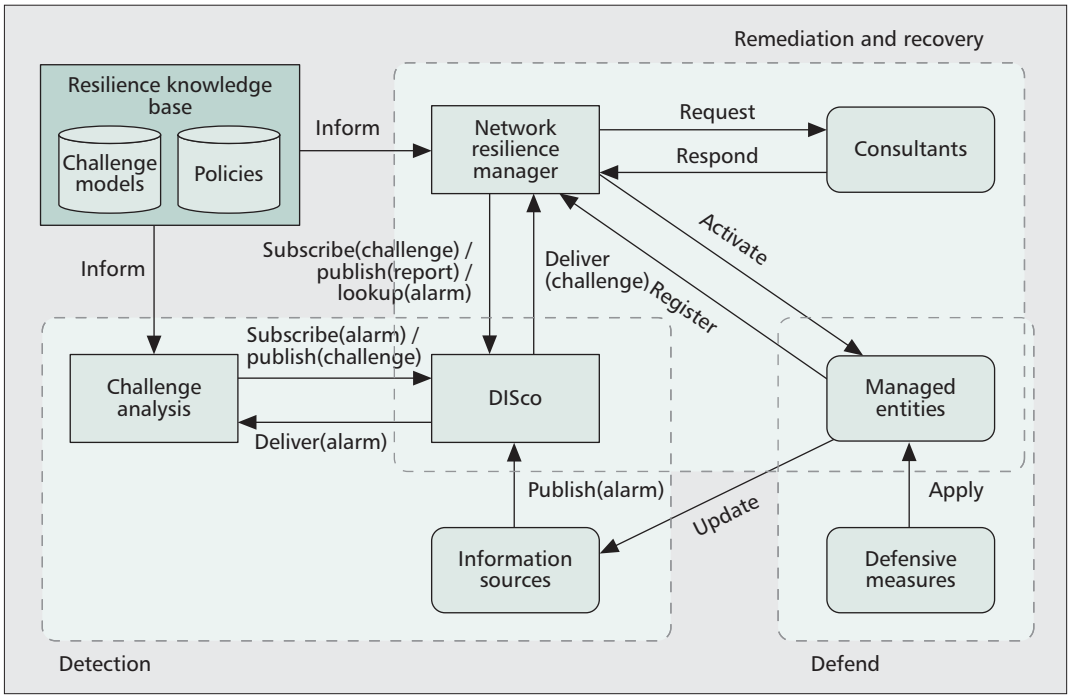


Figure 5. *A dynamic adaptation architecture that realizes the resilience control loop.*

Remediation and Recovery Subsystems

The challenge detection subsystem interfaces with the remediation and recovery subsystem, the third and final step, by issuing *alerts* to DISco using the `publish(challenge)` primitive. These alerts contain information about the challenge and its impact on the network, in terms of the metrics that are falling short of the resilience target. The *network resilience manager* takes this information as context data, and, based on *policies*, selects an adaptation strategy. In doing so, the network resilience manager realizes the resilience management functionality in Fig. 1. If further information is required by the network resilience manager that is not contained in the alert, the `lookup(alarm)` primitive can be used. Furthermore, the network resilience manager can make use of *consultants*, such as path computation elements, which can compute new topological configurations, such as new channel allocations or new forwarding structures. Resilience mechanisms are deployed by enforcing new configurations on the *managed entities* (e.g., routers and end hosts) in the network. To implement the resilience estimator, the network resilience manager assesses the success of chosen remedies. The assessment is stored in DISco to aid the diagnosis and refinement steps of the background loop. Carrying out this assessment is not straightforward since it requires spatio-temporal correlation of changes in network state, which is an issue for further work.

Resilience in Smart Environments: A Case Study

We are currently evaluating our generic approach to resilience through concrete study cases that cover a range of future networking paradigms: wireless mesh and delay-tolerant networks, peer-to-peer voice conferencing, and service provision over heterogeneous smart environments. Herein, we focus our discussion on the last study case. The widespread use of smart mobile devices, together with identifiers such as radio frequency identification (RFID), embedded in objects such as products, enables communication with, and about, these objects. The French national project Infrastructure for the Future Trade (ICOM) has developed an intra- and inter-enterprise infrastructure, depicted in Fig. 6, that allows the connection of objects with enterprise information systems and fixed or mobile terminals. This ICOM platform can be used as a foundation for a number of enterprise applications. The experimentation makes use of three different entities:

- The data acquisition site is the data source — items identified by RFID, for example, are read and their information sent to a processing centre located remotely.
- The data processing site houses different modules of the platform (e.g., data collection, aggregation, and tracking), which will forward the enriched data to the core application.
- The application provision site hosts the platform's central element — it is also where the data subscriber applications (web services, legal application, etc.) are linked.

Based on outcomes of our risk assessment approach, high-impact challenges to the platform include those that are intentional and accidental: malicious attacks that threaten the confidentiality and integrity of commercially sensitive data, DDoS attacks by extortionists, and, given the immature nature of the platform, software and hardware faults. This understanding ensures that we implement appropriate

Defensive measure	Description	Innovation
Survivable Network Design (SND) [11]	During network planning, SND optimizes network operations, such as routing and transport, in the presence of high-impact challenges.	Expansion of the methodology to derive a cooperation-friendly routing scheme for Wireless Mesh Networks (WMNs) to cope with node selfishness, explicitly accounting for radio interference constraints [12].
Game-theoretical node protection [13]	Node protection schemes are deployed against propagation of malware, which may compromise network nodes and threaten the network resilience.	The game-theoretic formulation of the problem confirms heavy dependence on the underlying *topology* and allows for optimal tuning of node protection level.
Rope-ladder routing [14]	Multi-path forwarding structure combining link and node protection in a way that the loss gap and QoS penalty, e.g., delay, during fail-over is minimized.	Better use of path diversity for support of real-time traffic, e.g., voice flows, for which burst packet loss during the path recovery time matters.
Cooperative SIP (CoSIP) [15]	An extension of the Session Initiation Protocol (SIP), whereby endpoints are organized into a peer-to-peer (P2P) network. The P2P network stores location information and is used when the SIP server infrastructure is unavailable.	Optimal setting of the number of replica nodes in the P2P network for given service reliability levels, inline with an enhanced trace-driven reliability model.
Virtual service migration	Enables redundancy and spatial diversity by relocating service instances on-the-fly, such that a continuous acceptable service can be provided to its users.	Existing approaches are tailored toward resilience to hardware failures within data centers. The derivation of service migration strategies from migration primitives, providing resilience against a variety of challenges.

Table 1. *A selection of defensive measures developed in the ResumeNet Project.*

defensive measures and dynamic adaptation strategies.

Consequently, defensive measures primarily include secure VPN connections between sites, enabling confidentiality and integrity of the data in transit. Security mechanisms, such as authentication and firewalls, are also implemented. Redundancy of infrastructure and implementation diversity of services are exploited to maintain reliability and availability in the presence of failures caused by software faults.

Incremental challenge analysis is realized using the Chronicle Recognition System (CRS), a temporal reasoning system aimed at alarm-driven automated supervision of data networks [10]. Lightweight detection mechanisms generate alarms based on metrics, such as anomalous application response times and data processing request rates. Finally, policy-based adaptation, implementing remediation and recovery, is achieved through the specification of the platform's *nominal* and *challenge context* behavior (i.e., its configuration in response to anticipated challenges). In our case study, challenge context policies describe configurations in response to alarms indicating a DDoS attack. For example, modified firewall configurations are defined to block traffic deemed to be malicious; service virtualization configurations that make use of redundant infrastructure are also specified to load balance increased resource demands. The transition between behaviors is based on alert messages, generated via challenge analysis, and outcomes from continuous threat level assessment. The case study sketched above illustrates the gain from applying our resilience strategies in a systematic approach: starting from a risk assessment, challenges are derived, allowing defense measures to be deployed. The following step is the specification of *chronicles* — temporal descriptions of challenges — for detection by the CRS, and policy-driven mechanisms to remediate and recover from unforeseen failures.

Conclusion

Given the dependence of our society on network infrastructures, and the Internet in particular, we take the position that resilience should be an integral property of future networks. In this article, we have described a systematic approach to network resilience. Aspects of our work represent a longer-term vision of resilience and necessitate more radical changes in the way network operators currently think about resilience. Further experimentation and closer engagement with operators through initiatives like ENISA, which focus on the resilience of public communication networks and services, are required before some of this research becomes standard practice. On the other hand, application-level measures, such as service virtualization, necessitate fewer changes at the network core and lend to easier implementation. Further benefits for network practitioners are anticipated through the use of tools like the Graph Explorer, which can explore correlations among metrics at various levels of network operation.

Acknowledgements

The work presented in this article is supported by the European Commission under Grant No. FP7-224619 (the ResumeNet project). The authors are grateful to the members of the ResumeNet consortium, whose research has contributed to this article, and in particular to Christian Doerr and his colleagues at TU Delft for the work presented in Fig. 3.

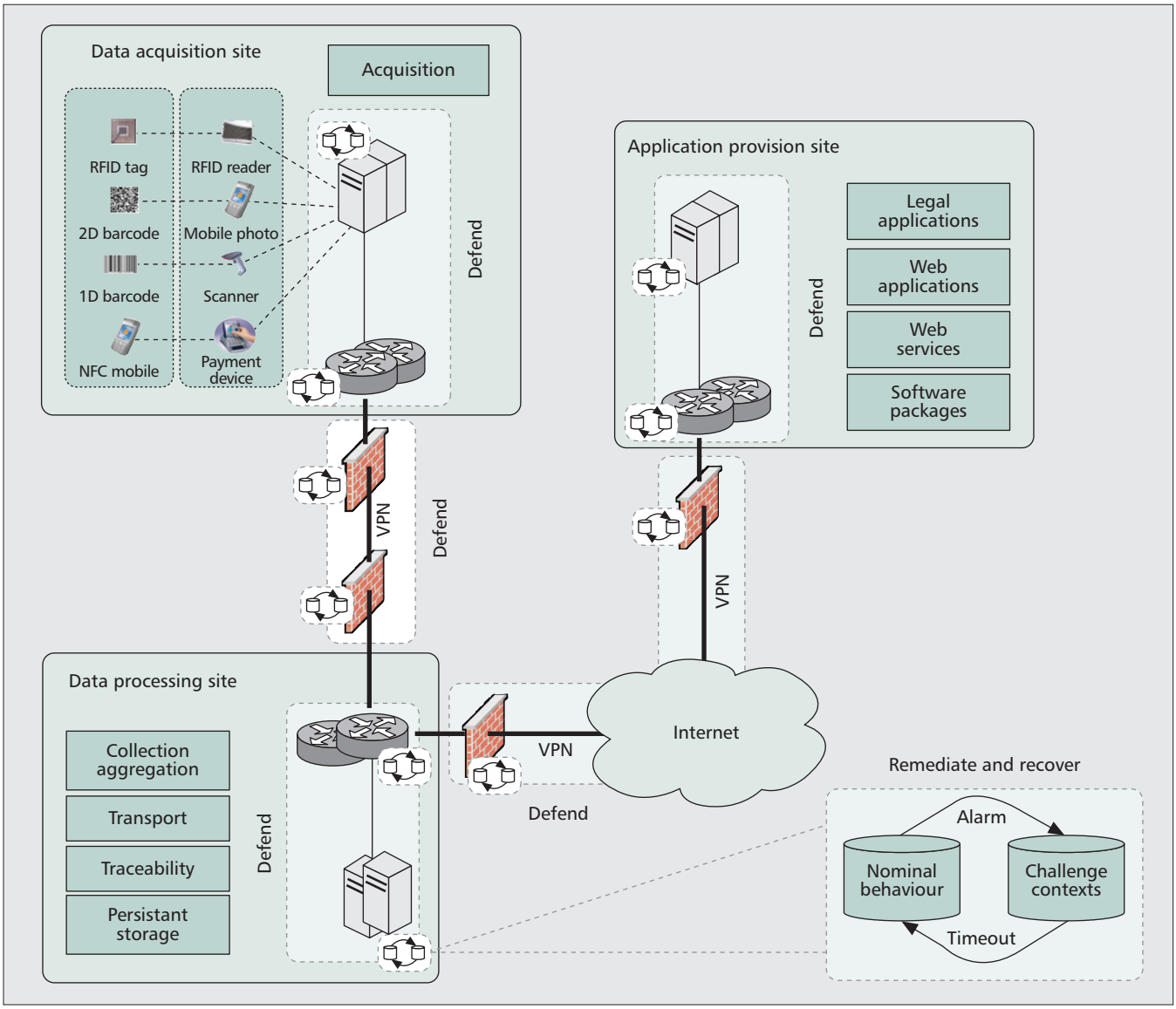


Figure 6. *The ICOM platform connecting enterprise sites that perform data processing and application provisioning with objects in a smart environment. Selected resilience mechanisms are shown that can be used to mitigate identified challenges.*

References

[1] J. P. G. Sterbenz *et al.*, "Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines," *Elsevier Computer Networks*, Special Issue on Resilient and Survivable Networks, vol. 54, no. 8, June 2010, pp. 1243–42.

[2] P. Cholda *et al.*, "A Survey of Resilience Differentiation Frameworks in Communication Networks," *IEEE Commun. Surveys & Tutorials*, vol. 9, no. 4, 2007, pp. 32–55.

[3] ENISA Virtual Working Group on Network Providers' Resilience Measures, "Network Resilience and Security: Challenges and Measures," tech. rep. v1.0, Dec. 2009.

[4] J. P. G. Sterbenz *et al.*, "Evaluation of Network Resilience, Survivability, and Disruption Tolerance: Analysis, Topology Generation, Simulation, and Experimentation (invited paper)," *Springer Telecommun. Sys.*, 2011, accepted Mar. 2011.

[5] C. Doerr and J. Martin-Hernandez, "A Computational Approach to Multi-Level Analysis of Network Resilience," *Proc. 3rd Int'l. Conf. Dependability*, Venice, Italy, July 2010.

[6] R. Fonseca *et al.*, "X-trace: A Pervasive Network Tracing Framework," *4th USENIX Symp. Networked Sys. Design & Implementation*, Santa Clara, CA, June 2007, pp. 271–84.

[7] P. Smith *et al.*, "Strategies for Network Resilience: Capitalizing on Policies," *AIMS 2010*, Zürich, Switzerland, June 2010, pp. 118–22.

[8] D. Agrawal *et al.*, "Policy Ratification," *6th IEEE Int'l. Wksp. Policies for Distrib. Sys. and Networks*, Stockholm, Sweden, June 2005, pp. 223–32.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Comp. Surveys*, vol. 41, July 2009, pp. 1–58.

[10] M.-O. Cordier and C. Dousson, "Alarm Driven Monitoring Based on Chronicles," *4th Symp. Fault Detection, Supervision and Safety for Technical Processes*, Budapest, Hungary, June 2000, pp. 286–91.

[11] E. Gourdin, "A Mixed-Integer Model for the Sparsest Cut Problem," *Int'l. Symp. Combinatorial Optimization*, Hammamet, Tunisia, Mar. 2010, pp. 111–18.

[12] G. Popa *et al.*, "On Maximizing Collaboration in Wireless Mesh Networks Without Monetary Incentives," *8th Int'l. Symp. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, May 2010, pp. 402–11.

[13] J. Omic, A. Orda, and P. Van Mieghem, "Protecting Against Network Infections: A Game Theoretic Perspective," *Proc. 28th IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1485–93.

[14] J. Lessman *et al.*, "Rope Ladder Routing: Position-Based Multipath Routing for Wireless Mesh Networks," *Proc. 2nd IEEE WoWMoM Wksp. Hot Topics in Mesh Networking*, Montreal, Canada, June 2010, pp. 1–6.

[15] A. Fessi *et al.*, "A Cooperative SIP Infrastructure for Highly Reliable Telecommunication Services," *ACM Conf. Principles, Sys. and Apps. of IP Telecommun.*, New York, NY, July 2007, pp. 29–38.

BIOGRAPHIES

PAUL SMITH is a senior research associate at Lancaster University's School of Computing and Communications. He submitted his Ph.D. thesis in the area of programmable networking resource discovery in September 2003, and graduated in 1999 with an honors degree in computer science from Lancaster. In general, he is interested in the various ways that networked (socio-technical) systems fail to provide a desired service when under duress from various challenges, such as attacks and misconfigurations, and developing approaches to improving their resilience. In particular, his work has focused on the rich set of challenges that face community-driven wireless mesh networks.

DAVID HUTCHISON is director of InfoLab21 and professor of computing at Lancaster University, and has worked in the areas of computer communications and networking for more than 25 years, recently focusing his research efforts on network resilience. He has served as member or chair of numerous TPCs (including the flagship ACM SIGCOMM and IEEE INFOCOM), and is an editor of the renowned Springer *Lecture Notes in Computer Science* and the Wiley CNDS book series.

JAMES P. G. STERBENZ is director of the ResiliNets research group at the Information & Telecommunication Technology Center and associate professor of electrical engineering and computer science at The University of Kansas, a visiting professor of computing in InfoLab21 at Lancaster University, and has held senior staff and research management positions at BBN Technologies, GTE Laboratories, and IBM Research. He received a doctorate in computer science from Washington University in St. Louis, Missouri. His research is centered on resilient, survivable, and disruption-tolerant networking for the future Internet for which he is involved in the NSF FIND and GENI programs as well as the EU FIRE program. He is principal author of the book High-Speed Networking: A Systematic Approach to High-Bandwidth Low-Latency Communication. He is a member of the ACM, IET/IEE, and IEICE.

MARCUS SCHÖLLER is a research scientist at NEC Laboratories Europe, Germany. He received a diploma in computer science from the University of Karlsruhe, Germany, in 2001 and his doctorate in engineering in 2006 on robustness and stability of programmable networks. Afterward he held a postdoc position at Lancaster University, United Kingdom, focusing his research on autonomic networks and network resilience. He is currently working on resilience for future networks, fault management in femtocell deployments, and infrastructure service virtualization. His interests also include network and system security, intrusion detection, self-organization of networks, future network architectures, and mobile networks including mesh and opportunistic networks.

ALI FESSI is a researcher at the Technische Universität München (TUM). He holds a Ph.D. from TUM and a Diplom (Master's) from the Technische Universität Kaiserslautern. His research currently focuses on the resilience of network services, such as web and SIP, using different techniques (e.g., P2P networking, virtualization, and cryptographic protocols). He is a regular reviewer of several scientific conferences and journals, such as ACM IPTComm, IEEE GLOBECOM, IFIP Networking, and *IEEE/ACM Transactions on Networking*.

MERKOURIS KARALIOPOULOS is a Marie Curie Fellow in the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece, since September 2010. He was a postdoctoral researcher in the University of North Carolina, Chapel Hill, in 2006 and a senior researcher and lecturer at ETH Zurich, Switzerland, from 2007 until 2010. His research interests lie in the general area of wireless networking, currently focusing on network resilience problems related to node selfishness and misbehavior.

CHIDUNG LAC is a senior researcher at France Telecom (Orange Labs). Besides activities linked with network architecture evolution, for which he contributes to the design of scenarios and roadmaps, his research interests are centered on network and services resilience, particularly through his involvement in European projects such as the ReSIST Network of Excellence (2006–2009) and the present STREP ResumeNet (2008–2011). He holds a Doctorat d'Etat-és-Sciences Physiques (1987) from the University of Paris XI Orsay.

BERNHARD PLATTNER is a professor of computer engineering at ETH Zurich, where he leads the communication systems research group. He has a diploma in electrical engineering and a doctoral degree in computer science from ETH Zurich. His research currently focuses on self-organizing networks, systems-oriented aspects of information security, and future Internet research. He is the author or co-author of several books and has published over 160 refereed papers in international journals and conferences.

Assessing Risk for Network Resilience

Marcus Schöller
NEC Europe Ltd., Heidelberg, Germany
Email: Marcus.Schoeller@neclab.eu

Paul Smith and David Hutchison
Lancaster University, Lancaster, UK
Email: {p.smith, dh}@comp.lancs.ac.uk

***Abstract*—Communication networks and the Internet, in particular, have become a critical infrastructure for daily life, business and governance. Various challenging conditions can render networks or parts thereof unusable, with severe consequences. Protecting a network from all possible challenges is infeasible because of monetary, hardware and software constraints. Hence, a methodology to measure the risk imposed by the various challenges threatening the system is a necessity. In this paper, we present a risk assessment process to identify the challenges with the highest potential impact to a network and its users. The result of this process is a prioritised list of challenges and associated system faults, which can guide network engineers towards the mechanisms that have to be built into the network to ensure network resilience, whilst meeting cost constraints. Furthermore, we discuss how outcomes from the intermediate steps of our risk assessment process can be used to inform network resilience design. A better understanding of these aspects and a way to determine reliable measures are open issues, and represent important new research items in the context of resilient and survivable networks.**

I. Introduction

Multi-service networks are essential for business, social life and entertainment. Sensor-actor networks provide the basis for a huge variety of deployments, e.g., industrial production, home automation, alarm systems, and environmental monitoring to name a few. Control networks build the backbone for remote operation of facilities and transportation. In short, communication networks have become critical for our business, social life, and governmental operation. This implies that failures of such networks, or parts of them, can have severe consequences.

Our work is based on the understanding that we can neither build fault-free networks, nor can we forecast all possible challenges to a network deployment. Multiple strategies have been proposed to design networks to survive different types of challenges. Examples include contributions from the ANSI/ATIS T1A1.2 Working group [1], ANSA [2], and Sterbenz *et al.*, as part of the ResiliNets initiative [3]. Central to all of them is the architectural view that challenges, as external events, trigger dormant faults of network systems' services, which manifest as errors. If these errors cannot be contained within the challenged service, they lead to a deviation of delivered service outside of acceptable bounds – a failure. The acceptable service bounds are defined in terms of dependability, security, and quality of service (QoS). The variety of challenges, which can degrade the delivered service, is large: hardware destruction, communication environment related challenges, human mistakes, cyber attacks, unusual but legitimate request for service, and failure of a service provider.

Based on this understanding, two components of a resilient network design can be derived. First, preventing challenges from affecting the system at all and second, to isolate erroneous behaviour within a service instance by building containment mechanisms. However, it is often not clear what is the suitable set of prevention and isolation mechanisms for a given network context. This is a problem given the potentially limited resources set aside for ensuring network resilience.

In this paper, we propose a risk assessment process for network resilience that aims to identify the challenge-fault pairs that are likely to have the highest impact on a network stakeholder's assets. This information can be used to make informed decisions about the nature and configuration of protection and isolation mechanisms, and increases overall network resilience within cost constraints.

The process we outline is similar to those proposed in the information security domain, but includes novel aspects that are necessary to consider network resilience matters. Unlike for information security, losing an asset is not binary in network resilience. Service disruptions, i.e., loss of connectivity or reduced bandwidth, are often acceptable within defined bounds. Based on previous work, we show how to associate costs of compromise to various states of network service.

We highlight how information generated as part of our risk assessment process, in addition to the high-impact challenge-fault pairs, can be used by network engineers to ensure resilience. For example, using information about the priority of assets can be used to guide remedy selection in resource constrained environments, such as sensor networks or during a resource starvation attack. We suggest that service dependency graphs can be used as the basis for challenge-independent remedies, which can mitigate unforeseen challenges.

The rest of the paper is structured as follows: we describe the state-of-the-art in risk management processes in Section II. Afterwards, in Section III, we introduce our risk assessment strategy for resilience. To demonstrate the applicability of our approach, we apply it to determining the high-impact challenges in a community wireless mesh network. There are a number of open issues for future investigation, including determining appropriate values, e.g., probabilities of challenge occurrence, to be used at various points in the process; we describe these in Section IV.

978-963-8111-77-7

II. Related Work

In a number of disciplines, risk management has been the focus of research for many years. The aim of these methods is to identify the most probable and high-impact attacks, so that appropriate security mechanisms can be deployed, given a set of organisational and monetary constraints. The CCTA Risk Analysis and Management Method (CRAMM)[1] was originally developed by the UK government in 1985, and is still widely used and supported by Siemens Enterprise Communications Ltd. The Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) method, proposed by Alberts *et al.* [4], is a threat-modelling process intended for use in large corporate, military, and governmental organisations. It can be used to determine the critical assets of an enterprise and the technical vulnerabilities associated with them. Based on the overarching OCTAVE method, a process for use in businesses with less than 100 employees has been created [5].

Both CRAMM and OCTAVE consist of a three-stage process: first, assets are identified and evaluated. The aim of this first phase is to identify the monetary cost to the organisation if assets become impaired (e.g., the cost to replace equipment, if it was stolen). Second, threat and vulnerability assessment considers the probabilities of these costs being incurred, soliciting both intentional and accidental threats to assets, such as attacks or human error. The result of this phase is a measure of risk. Finally, countermeasure selection and recommendation is carried out. A key facet of this countermeasure selection is comparing the level of risk measured in phase two against the potential security gain of a particular countermeasure. As a logical first step, we similarly begin by identifying the costs associated with asset impairment.

However, when considering network-related assets, such as services, it may be difficult to classify impairments in such a clear way, and consequently associate a cost. For example, it is not clear how to measure the cost associated with partial service degradation. We propose to build on previous work on evaluating resilience metrics to tackle this problem. Furthermore, to our understanding, both approaches do not provide specific guidance on how to understand the nature of the networked systems and services that realise the assets that are to be protected. Here, we explicitly aim to give suggestions on how this could be approached. Finally, we aim to show how the outcomes of the various stages of the risk assessment process can be used when developing resilient networks; no such guidance is given in the approaches we have investigated.

Vajanapoom *et al.* propose a risk-based approach to resilient network design, which includes three network topology design techniques that can be used to minimise various aspects, such as the maximum damage that could incur in the network [6]. Our work differs from this, as we propose to consider challenges that transcend those at topology level, and do not focus on prescribing solutions to identified risks. Synergies between our work is an area for further consideration.

[1] http://www.cramm.com

III. A Risk Management Process for Network Resilience

We now describe the risk assessment process that can be followed to determine the probable high-impact challenges a network will face. The process is outlined in Fig. 1. We demonstrate the applicability of the proposed approach by applying it to a community wireless mesh network (WMN) that is operational in a rural village in the north-west of England, called Wray. Ishmael *et al.* [7] have described the deployment in detail; thus we keep our introduction to it short. The Wray WMN case study is used as we have access to information about assets associated with the network and detailed information regarding its implementation.

The Wray WMN deployment: WMNs create a network infrastructure using a combination of wireless networking technology and ad-hoc routing protocols that together provide the ability to establish networks in locations with no prior groundwork. For example, the community of Wray felt that the lack of broadband Internet connectivity (due to their remote location) in their village was jeopardising local businesses, education, and the community itself. In Wray, home and enterprise computers connect wirelessly using IEEE 802.11b to the network via a mesh device, which forwards traffic to a single point in the village (the school), where back-haul is provided to the Internet. At the time of writing, mesh devices use hard-coded routes and run a number of network services, e.g., DNS, NAT, and a firewall.

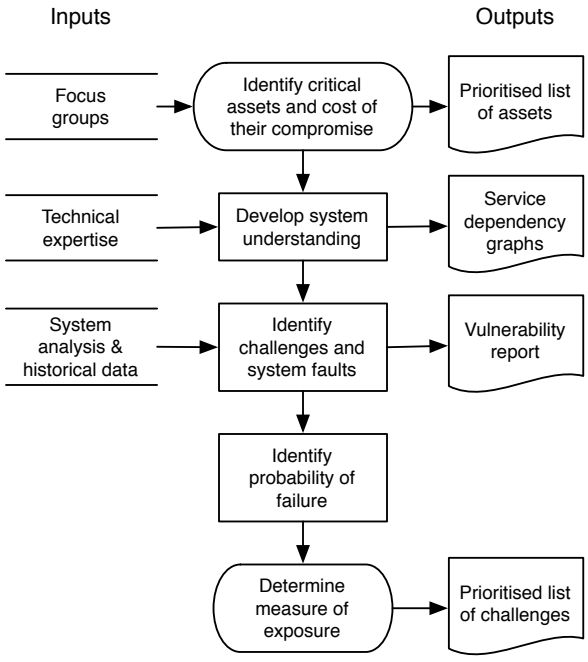


Fig. 1. A risk assessment process for resilience, including example inputs and outputs

Step 1: Identifying the critical assets the stakeholders associate with a network and the cost of their respective compromise.

Assets can be *physical*, *logical*, or *informational*. What is considered a critical asset is context specific, where the context is defined by the requirements of the organisation using the network, and its application. Physical assets can be damaged or manipulated; logical assets, implemented as services, can be attacked or used in a way that was not designed for; and informational assets can be disclosed, suppressed or manipulated. Example stakeholders include, customers, network providers and governments. Assets may be compromised in a number of ways.

A way to evaluate the compromise of service-based assets has been proposed by Sterbenz *et al.* [8], who use a multi-metric approach to derive three states in a service context. Initially, when a service is not challenged, it is considered in an *acceptable* state. Challenges can lead to a service becoming *impaired* or *unacceptable* – illustrating the non-binary loss of assets in network resilience. The trajectory a service takes between these states depends on the mechanisms in place to resist a challenge and how severe it is.

The level of degradation, e.g., impaired and unacceptable, and the duration the asset is in that state will have different costs – in most deployments, extended periods of impairment, due to a DDoS attack, will have a higher cost than short disruptions, due to a switch-over to a backup configuration. Measuring the cost of a compromise can be done qualitatively (e.g., mild, moderate, severe, or catastrophic) or quantitatively (e.g., potential financial loss or cost of replacement of equipment). For example, a DDoS attack could cause a degradation of service on an ISP's network, leading to SLAs being broken, resulting in reimbursements to customers. The same incident could lead to a loss of reputation for the ISP, a qualitative cost. At this step in the process, we consider the different modes of compromise associated with an asset, and their impact on the stakeholders; a prioritised list of assets results.

Assets of the Wray WMN: There are a number of stakeholders associated with the Wray village network, including its users (who also act as infrastructure providers within the village) and the Internet back-haul provider. To understand the nature of the assets from the user group's perspective, Bury *et al.* held an OCTAVE-style discussion group within the village [9]. They found the types of assets described by the villagers to be wide-ranging, and included such things as stored and transmitted personal information, e.g., photographs, their reputation within the community – a form of informational asset – and high quality access to the Internet, a combination of physical assets, e.g., mesh devices, logical assets, e.g., software services and protocols, and informational assets, e.g., user account data. The back-haul provider's main concerns are focused on the technical aspects of the network, including network infrastructure and deployed services – key components of ensuring SLAs are being met.

TABLE I
IMPACT DIFFERENTIATION FOR ASSET COMPROMISE

Asset	State	Impact	Value
Internet connectivity	impaired	medium-to-high	0.75
	unacceptable	high	0.9

Asset values of the Wray WMN: With an understanding of the critical assets, we need to determine the cost of the different ways in which they could be compromised. Consider a member of the Wray community who uses the WMN for business purposes, of which there a number of examples, an impaired service over short periods of time (e.g., high delay, some packet loss, etc.) is likely to have some impact on their business through loss of sales, time and competitive advantage. We therefore suggest a medium impact to this scenario. If Internet connectivity is deemed to be unacceptable for extended periods (e.g., no connectivity at all for a series of days), then the business impact will be much greater. For each of these levels of impact, we assign a value in the range $[0, 1]$, which indicates the severity of the scenario in relation to other assets and their mode of compromise. This is summarised in Table I.

As provision of Internet connectivity is a central asset for the two main stakeholders of our scenario, and many other assets depend on its correct operation, henceforth we use this asset as an example to illustrate our process.

Step 1: Output: The assets identified feed into Step 2 of our process, wherein we understand their implementation; the asset compromise values feed into Step 5 to calculate a measure of exposure. From an implementation perspective, the compromise values can be used to define *protection priorities* for the services realising assets. In case of insufficient resources to protect all assets from a variety of simultaneously ongoing challenges, the network should strive to protect those with the highest value in the face of failure. Resilience management strategies that honour protection priorities can be described in policies, as part of a general approach to resilience that capitalises on policy-based management frameworks [10].

Step 2: Develop an understanding of the network's components, their inter-dependencies, and how they contribute to the implementation of the identified assets

Modern network engineering approaches decompose the provisioning of assets into multiple sub-systems and services. Naturally, these are re-used in the design and implementation of multiple assets. This inherently implies that multiple assets can be (partly) degraded if a common sub-system (vertical service dependency) or peer service (horizontal service dependency) is affected by a challenge. In this phase, networking and systems experts develop an understanding of the used sub-systems and services, as well as their interdependencies, leading to a service dependency graph.

Service dependencies in the Wray WMN: Continuing our example, we show the translation of the "Internet connectivity"

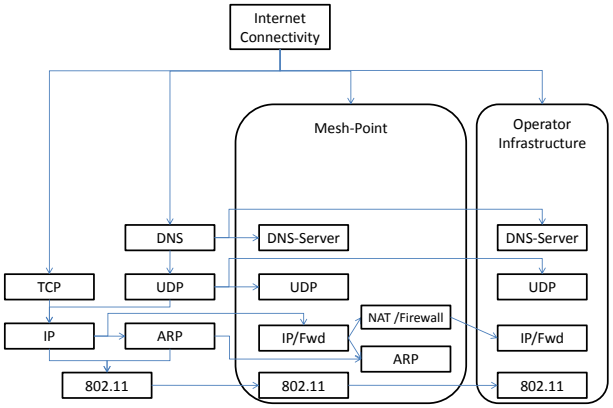


Fig. 2. Internet connectivity service dependency graph

asset into a service dependency graph. A simplified version is depicted in Fig. 2, which shows the service instances and their dependencies. An application based on the TCP/IP protocol stack requires the correct functioning of the TCP, IP, and 802.11 protocol instances on the client for the data path. For the signalling path, DNS, UDP and ARP are required to establish any communication. Note, that client authentication on the MAC and session layer, as well as client configuration, is not depicted. These are the vertical service dependencies on the client computer.

Having identified the local service dependencies, the next step is to analyse horizontal service dependencies. As routes are currently fixed in the Wray WMN, the user directly depends on the availability of the associated mesh points as well as on the correct functioning of the ARP responder located on the mesh point. Moreover, the mesh points provide network address and firewall services. Failures, such as mis-configurations can lead to (partial) loss of the Internet connectivity asset. Similarly, if the backhaul link fails for any reason Internet access is no longer available. In contrast, the primary DNS service is provided by the mesh points, with a backup instance in the operator infrastructure network. The client DNS resolver depends on the correct functioning of any one of these.

Step 2: Output: The services realising the various assets will be examined in Step 3 for faults and vulnerabilities. Moreover, the resulting service dependency graph can be used to build generic remediation strategies that strive to ensure acceptable service provisioning in the face of unknown challenges and system faults – i.e., those not identified via this process. These generic strategies are not optimised for a specific challenge, but change the network configuration in a way that challenged parts are isolated from unaffected parts, e.g., using system and network virtualisation techniques, or whole sub-systems are exchanged with alternative components, e.g., provided by a network composition framework as proposed by Kappler *et al.* [11]. Another way to use dependency graphs has been introduced by Katzela *et al.* [12], who utilise such graphs for fault localisation.

Step 3: Identify challenges, the probability of their occurrence, and faults that can potentially be triggered by them

As discussed earlier, the set of challenges that could affect a network is large. Here, the aim is to identify those challenges, and the faults in services and sub-systems that will most likely occur in the given deployment scenario. Faults with respect to resilience are wide-ranging and include design faults, inappropriate business-processes, and vulnerabilities to attacks. In addition, they may also include inappropriate use of protection and isolation mechanisms.

Challenges trigger faults, causing erroneous behaviour of a service and, if not isolated, for it to fail. Therefore, a specific service is threatened by challenges targeting the service itself, and, in addition, challenges causing service failures of dependent service instances. We use this reasoning to manage the wide-range of faults that may exist in a network, and focus on identifying those in the services (and their dependents) that could be triggered by the most likely challenges that have been identified.

System analysis and historical data can be used to derive the likely challenges and associated faults. Example approaches to system analysis include fault-tree analysis [13] and event-tree analysis [14], or threat-modelling techniques, such as STRIDE [15]. Security advisories provided by CERT[2] or SANS[3] can be used to identify past and on-going threats, for example.

To the best of our knowledge, there are no advisory services that cover the range of challenges considered for network resilience. We see this as a key shortcoming to address, and suggest that multi-national organisations, such as the European Network and Information Security Agency (ENISA)[4], could play an important role leading the development of such an advisory. As a starting point, a comprehensive taxonomy of resilience challenges is necessary.

Challenges and Faults in the Wray WMN: To understand challenges associated with a new network, the analyst has to rely on expert knowledge from similar deployments. Table II shows a non-comprehensive list of challenges identified for the Wray scenario, and gives probabilities of them occurring. The table is split into two parts: The first part shows the challenges identified before deploying the network, and the second shows two challenges that have occurred during its operation, which were not initially foreseen. Note, that the probability of a challenge occurring is independent from it leading to a failure (that probability is determined next). We give examples of how we determined the probability values shown in Table II.

- **Adverse weather condition** Data collected by weather stations in the north of England for the last two years show an average of 20 days per year with more than 10mm of rain. Our assumption is that only on these days rain is temporarily severe enough to have an impact on the wireless mesh network.

[2]http://www.cert.org
[3]http://www.sans.org
[4]http://www.enisa.europa.eu/

TABLE II
IDENTIFIED CHALLENGE, THEIR PROBABILITY OF OCCURRENCE AND JUSTIFICATIONS

Challenge	Prob.	Explanation
Adverse weather condition	0.055	Heavy rain must be expected in the north of England which can cause signal attenuation.
Gale-force winds	0.014	Very hard winds can damage outdoor antennas.
Power outages	0.00015	As an effect of thunderstorms or DIY work, mesh points get off-line temporarily.
Radio interference	0.1	Many wireless communication devices use the same frequencies as 802.11 such as Bluetooth, WiMAX, ANT, and also 802.11 based home area networks.
Jamming attack	10^{-8}	This attack does not require much technical expertise but access to locality. Social feedback loops in a rural environment, like Wray, make such attacks unlikely.
ARP storm	10^{-6}	Some file sharing activity on the network and relatively novice user-base could lead to malicious code infection which can cause ARP storms.
Malicious user behaviour	10^{-4}	Users reboot mesh points at their location to get a higher bandwidth share after reconnect.
Objects in LOS	0.012	The milk truck coming to Wray is blocking the line of sight (LOS) between two mesh nodes. The link goes down until the truck moved on.

- **Gale-force winds** The UK MET office gives an average of five to seven gale days per year.
- **Power outage** We assume that an average household switches off electricity for 10 hours per year for home improvement work. As there are eight mesh nodes deployed in Wray, we expect 80 hours of power unavailability for parts of the mesh network.
- **Object in LOS** A large milk truck parks in the line of sight of an antenna in Wray each day except Sundays. It stops for 15 to 20 minutes before moving on.

Regarding the faults in the Wray network, in relation to the signal attenuation related challenges, we have identified a lack of redundant paths, inability of wireless devices to dynamically change channels, and hard-coded routing tables as the main shortcomings.

Step 3: Output: The identified probabilities of challenge occurrence will be used in Step 5 to calculate the measure of exposure. In addition, the identified challenge-fault pairs can be documented in a *vulnerability report*. This report will help a network engineer choose the best protection or isolation mechanisms to protect a network from the high-impact challenges.

Step 4: Determine the likelihood of service degradation for each identified challenge and its impact on the stakeholder's assets

This phase is used to determine the likelihood of a challenge leading to a failure, which is influenced by the nature of a challenge and properties of a network. Such properties include known faults, the mechanisms that are in place to mitigate challenges, and the dependencies of a service on others. Adding mechanisms to defend against the high impact challenges or to remediate their impact on the system after this risk assessment process decreases the likelihood of a failure occurring.

To acquire justifiable probability values, analytical models and data gathered from similar deployments can be used. Analytical approaches, like the Graph Explorer proposed by Doerr *et al.* [16], can provide probability density functions that describe how challenges affect various service metrics. From the anticipated perturbations of these metrics, probability values for the level of service degradation can be derived. However, following such an analytical approach might be infeasible for some challenges. For these, probability values need to be estimated during the design phase. As pointed out before, refining these values during network operation by correlating service failures with sensor data collected from the deployment is necessary in this case to avoid over- and underestimation of the impact of these challenges.

Service failure probabilities in the Wray WMN: For the challenges we have identified in Table II, the probability they will cause the Internet connectivity asset to become impaired, unacceptable, or remain unaffected is presented in Table I. The boundaries for these three states are based on service availability guarantees given in an SLA. An example for this scenario could be the following: If a single challenge event impacts Internet connectivity, bringing it down for more than 20 minutes (roughly a 99.95 % availability guarantee), we consider the asset to be lost, denoted as an *unacceptable service* in Table II. If multiple challenge events occur within one month, bringing the Internet connection down for more than 20 minutes in total with no single challenge event exceeding the 20 minutes time limit alone, we consider the service to be *impaired*.

It can be seen in Table III that we expect that rain will have no effect on the wireless links in most cases. As we do not have enough measurement data to correlate link failures with rain intensity yet, we need to make estimates. Hence, we assume that in 90 % of the time the link will maintain fully operational, in 8.5 % of the time severe rain leads to

TABLE III
PROBABILITIES OF IMPACT OF CHALLENGES TO INTERNET CONNECTIVITY

Challenge	Unacceptable	Impaired	Normal
Adverse weather conditions	0.015	0.085	0.9
Gale-force winds	0.1	0.0	0.9
Power outage	0.8	0.2	0.0
Radio interference	0.02	0.05	0.4
Jamming attack	0.01	0.99	0.0
ARP storm	0.4	0.2	0.4
Malicious user behaviour	0.0	0.1	0.9
Objects in LOS	0.1	0.90	0.0

an impaired link service, and only in 1.5 % of the time, rain will cause an unacceptable service provisioning. For the last challenge in Table III, the milk truck blocking a link while stopping in Wray, enough data is already available to give more justifiable figures. The truck usually stops for 15 to 20 minutes. According to our definition, this would lead to an impaired service. In some cases, the truck would stop for longer than 20 minutes, leading to an unacceptable service. In any case, the milk truck will cause a degradation of the Internet connectivity every day. These figures should be derived from an understanding of the system and its associated faults, and the nature of the challenge, i.e., the information derived in the previous steps of our process.

Step 4: Output: The probability of a challenge leading to a service failure will directly feed into Step 5 of this process. Together with the output of Step 1 and Step 3 the measure of exposure can be calculated.

Step 5: Determine a measure of exposure and order the challenge-fault pairs accordingly, placing the high impact challenges at the top of this list.

In Step 2 of this process the cost of a particular mode of compromise for an asset will have been identified. Based on this, we would like to compute a numeric value of exposure – a measure to assess the impact a challenge has on network assets. We determine a measure of exposure using the equation below; $challenge\_prob$ is the probability that a challenge will occur (from Step 4), $compromise\_prob$ is the probability that a compromise will occur to an asset, which is based on the likelihood of a failure (from Step 5), and $impact$ is the cost associated with an asset being compromised, from Step 1.

$$exposure = (challenge\_prob \times compromise\_prob) \times impact$$

It is clear that using this strategy to determine a measure of risk will yield similar values for high-probability, low-impact events and low-probability, high-impact events. Depending on which scenarios the network engineer would like to account for, exposure measures could be filtered out.

High impact challenges in the Wray WMN: A convenient way of depicting the results for the Wray network analysis is shown in Fig. 3. The connections on the left-hand side of the graph (from the *asset* column to the *compromise* column) are annotated with the impact a particular compromise to an asset could have on the studied stakeholder. The annotations on the connections of the right-hand side of the graph show the product of the challenge probability and the challenge leading to a particular compromise.

Table IV shows the sorted exposure values for the high impact challenge scenarios in relation to the Internet connectivity asset. It can be seen the "Obstacle in LOS" challenge and severe rain are the two most significant, followed by signal interference and gale-force winds. An engineer wishing to improve the resilience of the Wray WMN could use this information to select appropriate mechanisms, e.g., introduce multiple paths to avoid service degradation due to signal blocking or attenuation, or improve known faults.

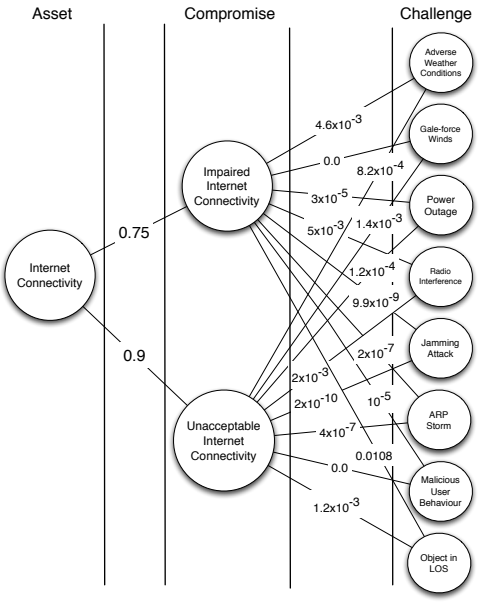


Fig. 3. An example exposure graph for the Wray WMN use-case study

TABLE IV
THE CHALLENGE SCENARIOS SORTED ON EXPOSURE

Challenge	Compromise	Exposure
Obstacle in LOS	Impaired	8.1×10^{-3}
Adverse weather condition	Impaired	3.5×10^{-3}
Interference	Unacceptable	1.8×10^{-3}
Gale-force wind	Unacceptable	1.26×10^{-3}
Obstacle in LOS	Unacceptable	1.08×10^{-3}

Step 5: Output: The last stage of our process outputs a list of challenge-fault pairs ordered by their exposure. The network engineer will select and implement protection or isolation mechanisms for the high impact challenges to maximize network resilience.

Risk assessment process summary

The described process provides a guideline for network engineers to acquire information that is needed to design and integrate resilience mechanisms, which can protect a network from the most severe challenges. Adding mechanisms to a network as a result of this process requires repetition of steps 2 to 5. This is because new features not only provide higher resilience against certain challenges, but also have the potential to introduce new faults and hence new threats to the network. Ideally, the measure of exposure after introducing new defensive mechanisms and remedies will be lower than before. In addition, this process should be repeated during network operations as new system faults and new challenges are discovered.

IV. Conclusion

We have presented an asset-based risk assessment process to identify the high impact challenges a network may face. The process is based on three main pillars: 1) understanding the cost of an asset's compromise; 2) gaining knowledge about the technical implementation of these assets including vulnerabilities which can lead to a degradation of delivered service; and 3) identification of the likelihood of a challenge occurring which can trigger such faults, and their impact on the network. This process has to be executed at network design time and repeated during the network's operation, as new challenges and vulnerabilities become apparent. The result of this risk assessment process is an ordered list of challenge-fault pairs that can be used by network engineers to realise resilience mechanisms for those with highest potential impact. In addition to this main result, the process provides additional information to the network engineer, including priorities for remediation, a dependency graph which can inform the design of generic remedies, and a vulnerability report which guides the implementation of the protection and isolation mechanisms.

Open research questions

While our risk assessment approach sets forth a concrete base for quantitatively assessing the impact of challenges, there are certain aspects that necessitate further research: First, further work has to be carried out on reliably estimating the *probability of challenges occurring*. This includes the availability and analysis of network monitoring information. In addition, information from other sources has to be gathered and analysed, as our case study suggests: as humid weather conditions and heavy winds are common phenomena in the north of England, signal attenuation or broken antennas are likely events; whereas crime statistics show that vandalism is an unlikely occurrence. The set of external data sources we have been looking at has been limited so far. Understanding which additional *context information* could prove useful is an open research question.

Second, an important part of the approach we adopt is to *quantify the impact of challenges* on the systems and services that support an asset, i.e., determine the likelihood of a failure occurring. To do this, we again could use historical data to gain an inference about the probability of a failure occurring in light of a challenge. However, the context of past events may be different to that of the network under examination, which may lead to an inaccurate understanding. One way to improve confidence in the probability of a failure occurring is to develop simulations of the network, including (statistical) fault models, and exercise challenges against it. Before developing simulations of challenges, we first need to develop a taxonomy to systematically describe them in enough detail. An ideal formulation would involve directly translating a high-level description of a challenge into a simulation model. This is a subject of further research in the resilience domain. Analytical models would also be of use in this task, whereby analysis can track or reasonably abstract the complexity of the problem.

Finally, our work has focused on understanding the horizontal and vertical dependencies between services and systems that form a network. This offers a potentially limited view of the dependencies a network has. For instance, further research could investigate approaches to explicate inter-dependencies between networks and the power grid. This could yield interesting insights that would prove informative when determining how networks fail.

Acknowledgements

The research presented in this paper has been funded by the European Commission in the context of the Research Framework Program Seven (FP7) project ResumeNet (Grant Agreement No. 224619).

References

[1] ANSI/ATIS T1A1.2 Working group, "Reliability/availability framework for ip-based networks and services," Tech. Rep. 70, American National Standards Institute, August 2001.

[2] N. Edwards, "Building dependable distributed systems," tech. rep., ANSA, February 1994.

[3] J. P.G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET)*, vol. 4, pp. 1245–1265, 2010.

[4] C. Alberts, S. Behrens, R. Pethia, and W. Wilson, "Operationally Critical Threat, Asset, and Vulnerability Evaluation Framework, version 1.0," Tech. Rep. CMU/SEI-99-TR-017, Carnegie Mellon University, 1999.

[5] C. Alberts, A. Dorofee, J. Steven, and C. Woody, "Introduction to the OCTAVE Approach," tech. rep., Carnegie Mellon University, 2003.

[6] K. Vajanapoom, D. Tipper, and S. Akavipat, "A risk management approach to resilient network design," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on*, pp. 622 –627, oct. 2010.

[7] J. Ishmael, S. Bury, D. P. Pezaros, and N. J. Race, "Deploying Rural Community Wireless Mesh Networks," *IEEE Internet Computing Magazine*, vol. 12, pp. 22–29, July–August 2008.

[8] J. P. G. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, Q. Shi, and J. P. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation (invited paper)," *Springer Telecommunication Systems*, 2011. accepted March 2011.

[9] S. Bury, J. Ishmael, N. J. P. Race, P. Smith, and M. Rouncefield, "Towards an understanding of security concerns within communities," *Wireless and Mobile Computing, Networking and Communication, IEEE International Conference on*, vol. 0, pp. 478–483, 2008.

[10] P. Smith, A. Schaeffer-Filho, A. Ali, M. Schöller, N. Kheir, A. Mauthe, and D. Hutchison, "Strategies for network resilience: capitalising on policies," in *Proceedings of the 4th international conference on Autonomous infrastructure, management and security*, AIMS'10, pp. 118–122, Springer-Verlag, 2010.

[11] C. Kappler, P. Mendes, C. Prehofer, P. Pöyhönen, and D. Zhou, "A framework for self-organized network composition," in *Autonomic Communication* (M. Smirnov, ed.), vol. 3457 of *Lecture Notes in Computer Science*, pp. 261–266, 2005.

[12] I. Katzela and M. Schwartz, "Schemes for fault identification in communication networks," *Networking, IEEE/ACM Transactions on*, vol. 3, pp. 753 –764, dec 1995.

[13] W. E. Vesely, *Fault Tree Handbook*. Nuclear Regulatory Commission, 1987.

[14] M. J. Cooper, *Event Tree Analysis*. Brunel Technical Press, 2007.

[15] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover Security Design Flaws Using The STRIDE Approach," *MSDN Magazine*, November 2006.

[16] C. Doerr and J. M. Hernandez, "A computational approach to multi-level analysis of network resilience," in *Proceedings of the 2010 Third International Conference on Dependability*, DEPEND '10, (Washington, DC, USA), pp. 125–132, IEEE Computer Society, 2010.