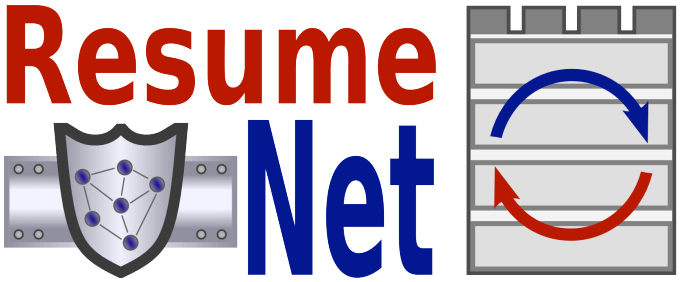

# Resilience and Survivability for future networking: framework, mechanisms, and experimental evaluation




| | |
|---|---|
| Deliverable number | 1.5a |
| Deliverable name | First interim strategy document for resilient networking |
| WP number | 1 |
| Delivery date | 31/08/2009 |
| Date of Preparation | 2/10/2009 |
| Editor | Paul Smith (ULanc) |
| Contributor(s) | Radovan Bruncak (ULanc), Christian Doerr (TuDelft), Ali Fessi (TUM), Michael Fry (USyd), David Hutchison (ULanc) Abdul Jabbar (KU), Merkouris Karaliopoulos (ETHZ), Marcus Schöller (NEC), Paul Smith (ULanc), James P.G. Sterbenz(KU) |
| Internal reviewer | Guy Leduc (ULg) |

## Summary

Resilience will need to be a key property of future Internet because of our unrelenting demand for Internet-based services, the challenging environments it will operate in, and the continued existence of intelligent adversaries. In this deliverable, we describe our current understanding, after one year in the ResumeNet project, about the challenge of making resilience an integral component of future networks. Our overall strategy to resilience is neatly summarized into the so-called $D^2R^2+DR$ acronym, standing for Defend, Detect, Remediate, Recover, Diagnose and Refine. This otherwise straightforward approach is only an abstraction of the actual tools that have to be in place to realize it. We first focus on what we consider fundamental components of our resilience framework: resilience metrics, understanding challenges, policies for resilience, and cross-layer information sharing and control. These are the kind of ingredients that can be used to inform the design and implementation of resilience mechanisms.

Based upon our current research findings, we reflect on the suitability of the strategy, and describe our on-going work populating the strategy with algorithms and mechanisms. This work, carried out within the context of WP2 and WP3 of the project, is presented in more detail in Deliverable D6.3 "Report of technical work in WP2 and WP3 during the 1st year of the project." Therefore, we limit ourselves to a summary of the undertaken work, pointing the reader to D6.3 for more details.

Finally, we discuss the experimental scenarios we will use to evaluate our research. The four study cases exemplify the applicability of the framework concepts and mechanisms and assess their effectiveness in widely variable networking scenarios. In the same time, they retrofit the overall framework design process allowing more quantitative characterization of framework components.

# Contents

# 1 Introduction

Resilience – the ability of a networked system to provide an acceptable level of service under adverse conditions – will need to be a key property of future Internet. This need is brought about by an increasing dependence on Internet-based services, which is unlikely to abate. Furthermore, as the Internet creeps into novel deployment settings, driven in large part by wireless technologies and the miniaturisation of devices, it will face new challenges stemming from the environment. Inevitably, this will be set against a background of malicious actors exploiting vulnerabilities in the Internet and associated socio-technical systems.

In the EU-funded ResumeNet project, we are investigating a framework and mechanisms for resilience in a future Internet. At the centre of the project is a straightforward strategy for building resilient networked systems: Initially, one must install appropriate *defensive* measures, e.g., configure firewalls and use appropriate redundancy and diversity of services, to ward foreseen challenges off. In many cases, there will be unforeseen events (or those that are too expensive to build defensively for) that will breach defensive measures and cause a degradation of service. Such challenges should be *detected* in real-time and the network dynamically adapted to *remediate* them. This implies an underlying monitoring system. Most likely, there will be a cost associated with remedying a challenge (e.g., sub-optimal paths are used to route around a malicious node); a *recovery* stage in our strategy reflects that we should disengage mitigation mechanisms when a challenge has abated. We assume the system is not perfect; therefore, we aim to *diagnose* shortcomings and *refine* the networked system. We call this strategy $D^2R^2 + DR$– *Defend*, *Detect*, *Remediate*, *Recover*, *Diagnose* and *Refine*.

In this deliverable, we describe our current understanding of how to build resilient networked systems, such as the future Internet. This understanding grows as we "drill down" into the implementation details of our otherwise straightforward strategy. Initially, we describe fundamental components of a framework for resilient networking:

- Appropriate *metrics* are indispensable for resilience; without them, we can neither specify nor measure it. We discuss some of the challenges in defining metrics for resilience and our initial work addressing them.
- The network response to detected challenges will be directed by *policies*. Here, we summarize the state-of-the-art regarding policy frameworks and how they are applied in a closely related project to ResumeNet to enable automatic mitigation of attacks. Then we present our initial understanding as to how we can apply policies for resilience.
- Applying appropriate defensive measures requires an *understanding* of the most probable high-impact *challenges* the network will face. We have developed a risk assessment strategy that can be used to identify such high-impact challenges. To aid this process we have developed a taxonomy of challenges, which is summarised here. We discuss issues relating to identifying measures of challenge occurrence and their impact.
- Because of the relative simplicity it affords, the strict layering of protocols has aided the Internet in its success. However, new wireless networking environments, for example, call for a break-down of the strict layering and intensive use of *cross-layer information sharing and control*. We discuss some of the challenges of doing this, and how we intend to model such interactions to aid the design of appropriate cross-layer mechanisms. Underpinning a cross-layer framework will be a distributed monitoring and measurement platform; we describe two promising technologies we are investigating for use in the project.

The research carried out during the first year of the ResumeNet project has allowed us to reflect on the suitability of the $D^2R^2 + DR$ strategy. We describe our thoughts on this and, in particular, discuss the nature of challenge detection, the time-scales that remediation could be invoked based upon information gleaned from detection, and the relationship between building strong defensive measures and the need for a distributed monitoring and detection platform. We continue with a discussion on how we are populating our resilience strategy with specific algorithms and mechanisms. To organise our efforts into approachable threads of investigation, we adopt two complementary approaches, namely *network* and *service* resilience. Network resilience is largely concerned with enabling resilience at OSI layers two through four, while service resilience addresses higher-layer issues, and investigates resilience in the context of peer-to-peer overlay networks and service virtualisation, for example. As mentioned earlier, this split is intended for organisational and presentational convenience, it is not architecture-driven. On the contrary, we consider it necessary to take a multi-level approach to resilience, where it is provided by a continuum of provider and consumer services throughout the layers.

Finally, we discuss how we will evaluate our research via four experimental scenarios that are likely to feature in a future Internet. We have tried to use scenarios that cover some of the most recent and promising networking paradigms: wireless multi-hop networks, opportunistic networks, SIP-based multimedia services, and ambient intelligence environments. In many cases, a significant inhibitor to the wider deployment of these networking paradigms are the challenges we aim to address in our experiments.

# 2 Framework for Resilient Networking

The resilience framework we introduce draws on four main components: understanding challenges, resilience metrics, policies, and cross-layer information sharing and control. All of them need to be specified and well understood to enable the realisation of the $D^2R^2 + DR$ strategy. Here, we describe our current understanding of each of these components. We lead-off with some reflections on the suitability of the resilience strategy.

## 2.1 Reflections on the $D^2R^2 + DR$ Strategy

One of the core components of our research in the ResumeNet project is to evaluate how appropriate the $D^2R^2+DR$ strategy and resilience framework are for building resilient networks. At the end of the first year of the project we can begin to reflect on this. Our understanding of the conditions under which remediation mechanisms should be invoked has matured. Initially, we considered it to be necessary to invoke a remedial action with a complete understanding of the challenge (i.e., an understanding of its root cause) – the rationale for this being the inappropriate behaviour of TCP in response to wireless losses [BV99]. Given our activities on evaluating risk when understanding challenges (see Section 2.2), we have relaxed this "requirement" to suggest that potentially beneficial remedial activities can be invoked without understanding a challenge's root cause, **but** this is done at the *risk* of undesirable behaviour occurring. This risk can be reduced as more is understood about the nature of a challenge. Specifically, the following aspects of a challenge can be used to inform the invocation of remediation:

- *Challenge's Symptoms*
  These are the measurable impacts of a challenge on the networked system that represent

a deviation from its normal behaviour. Anomaly detection techniques can be used, for example, to detect the symptoms of a challenge. An example of remediation action that can be invoked based upon a challenge's symptoms is the ReplEx algorithm [FKF06], which can be used to perform dynamic traffic engineering in multi-path networks based upon signs of congestion in routers.

- *Challenge's Root Cause*
  This is one reason why we are observing the symptoms of a challenge. For example, a server may be overloaded with requests for service. This could be due to a flash crowd event or a DDoS attack; one may want to remedy these challenges differently. Work carried out in the EU IST-FET-funded ANA project [ANA] is addressing this problem. There, they use a supervised Naïve Bayes estimator to determine the root cause of abnormal traffic patterns at run-time [MPH08], with a direct impact on the selected remedy.

- *Challenge's Impact*
  This is a measure of the negative impact a challenge may have or has on the network. For example, this may be the delay incurred by services as a consequence of a challenge. Also, we may want to consider the socio-technical impact of a challenge; for example, its monetary cost. Our work on resilience metrics, described in Section 2.3 of this document, will influence how we understand the impact of a challenge.

Therefore, using a biological analogy, we could identify two types of remediation strategies corresponding to two different time-scales of invocation: *reflex* and *conscious*. Reflex remediation is invoked based upon a challenge's symptoms and can therefore be initiated rapidly, much like moving one's hand away from a hot surface. In may cases, this may be the most appropriate action to take, but in some cases it could be wrong. Conscious remediation is based upon understanding a challenge's root cause, its impact on the system, and the potential consequences of the remedy on the system and other services, for example. Clearly, this will take longer to determine and is more (computationally) expensive, but should lead to less risky remedies. We envisage remediation to be an iterative process, i.e., we may initially invoke reflexive remedies, and adjust this strategy after some consideration with a greater understanding of the challenge.

From the project outset, we have assumed the need for a distributed monitoring and assessment platform that is used to detect anomalies. This is because no defensive measures can provide perfect protection. Defences will be breached, perhaps because of the unforeseen severity of challenges, and remediation will be necessary. During the first year of the project, we have re-visited this issue, and we believe that there is much research needed to understand the trade-off between the cost and efficacy of building strong defensive measures (e.g., through resilient structures and security mechanisms) on the one hand, and applying resources to building a distributed monitoring and detection platform on the other. We may need both. A simple way to formulate this problem is to ask the question: "given strong enough defensive measures, to what extent do I need a measurement and detection platform?" Our work on understanding challenges, using risk to determine where best to apply resources, can help to resolve this question. Also, the project's work on resilience metrics may help us as we consider this problem. We intend to devote some resources to looking further into achieving resilience by means of defensive measures, including appropriate system structuring to help isolate and localise (i.e., minimise the spread of) the ill-effects of challenges.

## 2.2 Understanding Challenges and Risks

The available resources for building components of future Internet will be finite. Mechanisms that will enhance the resilience of the future Internet incur a cost, both monetary and computational. To make the best use of limited resources and ensure appropriate defensive resilience measures are installed, it is necessary to understand what are the probable *and/or* high-impact challenges that may occur. One way of determining the high-impact challenges a networked system may face is to conduct a risk assessment that aims to elicit the *critical assets* associated with it and the most probable challenges, or threats, that may occur. Given these two parameters, it is possible to determine a measure of risk or exposure by, for example, taking the product of the challenge probability and the cost of the asset being compromised – $exposure = challenge\_prob \times asset\_cost$. In the security field, a number of risk management approaches have been proposed that can be used to determine the high-impact threats [CRA, ABPW99].

We have developed a seven-stage risk assessment process, depicted in Figure 1, that can be used to determine the high-impact challenges a networked system may face. Similar to earlier work, it is based upon understanding the critical assets associated with a networked system (Stage 1 in Figure 1). We conducted a focus group with users of a rural wireless mesh network [IBPR08] and found their notion of assets to be wide-ranging [BIR+08], including the safety of minors and unfettered use of the network. For a network provider, assets could include switches and routers, for example. The next stage in our risk assessment process involves understanding the cost of an asset being compromised. For example, for a network provider the loss of a core switch may lead to not meeting an SLA, and a monetary penalty being incurred. With an understanding of the critical assets and their cost if compromised, the next stages aim to determine an understanding of the networked system – modern software engineering approaches decompose the provisioning of assets into multiple sub-systems and services; in this phase these are identified. With an understanding of the system, we aim to determine the challenges it will face and the system faults (in security parlance, the system's vulnerabilities) that could be triggered by these challenges. We have identified the following classes (or taxonomy) of challenges that can be used to guide an assessor:

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

**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).

**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
- high-delay paths either due to length (e.g., satellite) or as a result of episodic connectivity

are gathered in the communication environment category.

**Human Mistakes** Human mistakes describe non-malicious errors that are made by people that interact with the system, such as device misconfigurations or operations not following policies. These can become more pernicious if the parties involved try to cover up their mistakes.

**Malicious Attacks** Malicious attacks from intelligent adversaries pose a threat to system performance and form a group of challenges to networked systems.

**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.

**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 behavior of the provider service. As service usage can be vertical, i.e., using a lower layer service, as well as horizontal, i.e., client-server based or peer service, interoperability faults also fall into this category. Last, failing of the provider service due to an unidentifiable challenge is covered in this category.

Given the challenges and system faults, we aim to determine the probability of a system failure (this is discussed further below), which can be used to determine a measure of exposure.

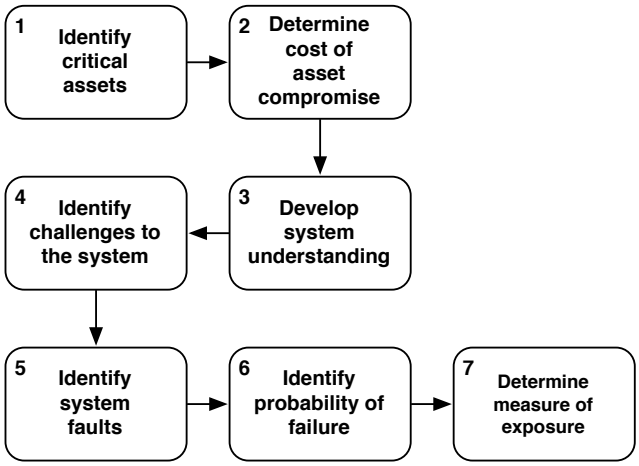



Figure 1: A risk assessment process for determining high-impact challenges

One of the major challenges when trying to understand the risks a system will face is determining meaningful measures for the occurrence probability of a challenge and its impact. These can be expressed qualitatively, e.g., high, medium or low, or quantitatively, e.g., $10,000 loss of earnings. To determine the probability of a challenge occurring, one can conduct an analysis of the system, using, for example, STRIDE [HLOS06] or attack/fault trees [Ves87, Sch99, Coo07], or use advisories to learn of prevalent threats [CER, SAN, MS]. These approaches will play a key role in understanding the challenges a system will face, but they have drawbacks. Advisories can relate to similar but not exactly the same system as the one that is being engineered to be resilient, so could be misleading. The complexity of networked systems and the wide range of potential challenges to its operation could render the system analysis intractable.

A complementary approach towards determining meaningful challenge occurrence probabilities consists in developing a chronology of the system. This could include information relating to challenges, their impact on the system, and the efficacy of defensive and remedial measures. Under this approach more accurate probabilities of challenge occurrence and impact can be developed over time. when compared with those derived via off-line analysis or potentially misleading advisories. The implementation of such a system has architectural implications on the future Internet. Maintaining this information has privacy and security implications; a trade-off will need to be made between usability and privacy preservation (e.g., via anonymisation), especially if such information is to be shared across organisational boundaries. Furthermore, there will be resource overhead that must be managed, for example, in summarising the raw events that will be added to the chronology and storing them for analysis. Another key question that will need to be addressed is which system features should appear in the chronology that indicate the system is challenged and are useful for generating statistics for risk assessment. Our work on building a distributed information store, described in Section 3.2, will address some of these concerns.

Finally, there are cases where no chronological data are available regarding the impact challenges can have on a system. In these cases, analytical modelling of the system and the challenges it may face can provide reasonable and fast estimates of their impact on its normal operation. This approach is taken in assessing the impact of node misbehaviour on the performance of opportunistic and wireless mesh and opportunistic networks – we have begun to model the affects of selfish node behaviour in opportunistic networks given different forwarding strategies [Kar09]. Since there are very few, if any, instances of commercial deployment of these networks, measurement data and operational experience with them are also limited. We briefly refer to this work in Section 3.2, pointing the reader to deliverable D6.3 [Con09] for further details.

## 2.3 Resilience Metrics

The aim of a resilient networked system is to provide an acceptable level of service when under duress from challenges. A fundamental component of this is a specification of the desired level of service, and its (relative) priority. This requires appropriate metrics that can be both used in policies to express desired service, and to measure the resilience of a networked system. As no metrics to express levels of desired service at a low granularity are universally accepted (beside service-level agreements that are typically used between providers), we will focus on topological metrics, which are well-defined, in this work. However, even the tasks of choosing appropriate topological metrics for assessing and measuring resilience is challenging, for a variety of reasons:

### 2.3.1 Resilience Metrics Challenges

There is a tension between metrics that are well understood to be measured, analyzed and interpreted structurally and technologically in a networking environment, and those which are commonly specified and meaningful to end-users and network providers. Consider the case where an end-user subscribes to a video-conferencing application. The application will have certain requirements, such as a minimum available bandwidth and maximum latency thresholds. Furthermore, the end-user will have further resilience requirements, for example, that the service has 99.999% availability. For a network provider to realise these high-level requirements is complicated, because it relies on the provider being able to translate them

into lower-level requirements of their hardware and software infrastructure. This would require pervasive deployment of QoS-enabled infrastructure, standardized classifications and interfaces to specify the service requirements from the link- to application-layer, and a means to synthesize higher-level services (e.g., highly available video-conferencing at good quality) from lower-level measures (e.g., 300 kbps path guaranteed at 5ms delay or 500 kbps at 7ms delay). This step becomes increasingly difficult when services, as specified by end-users, become more abstract.

Ideally, the metrics used in a good classification system need to be as orthogonal as possible, i.e., each metric should measure an entirely different aspect of the network topology and its usage scenario. While the theoretic demand on resilience metrics is clear, in practice, finding such a perfect orthogonal assessment system is problematic. Consider the example of two topological metrics, hop count (of the shortest path) and the betweenness of nodes and links. It is easy to see that the results of both metrics is highly correlated (in the average case), as the route that each shortest path takes also increases the betweenness of the links and nodes along this path. Other relationships are more subtle in nature, i.e., the dependency might only exist to some degree (for example, between clustering coefficients and average node coreness, average neighbour degree and assortativity coefficient, etc.) and under when the graph has certain topological characteristics [JU08]. Using covariance calculations, it is possible to determine for a specific set of metrics how much overlap exists for a given network topology in conjunction with a service. However, from these sample calculations, globally valid conclusions cannot yet be drawn. For a selected set of topology metrics, it is possible to prove their interdependency and non-orthogonality. However, effort is required to expand this work to the large array of currently used and recently proposed resilience metrics.

Metrics used for resilience assessment should be general enough and universally applicable to allow for comparisons of network environments across varying network topologies and service demands. To date, a number of networking metrics have been proposed, of which many are targeted towards aspects and features observed for a particular application context. To allow for direct comparisons between systems (and thereby indirectly also allowing for resilience engineering – the process of upgrading the network infrastructure to possess higher levels of resilience), universally utilized metrics are necessary.

### 2.3.2 Multi-level Resilience Quantification Framework

The objective of this avenue of research into resilience metrics is to develop a framework to quantify network resilience. Since ResumeNet proposes multi-level resilience, we propose a framework that is applicable at the boundary of any two layers in the network stack. The fundamental concept in this approach is to quantify resilience as a measure of service degradation in the presence of challenges (perturbations) to the operational state of the network, therefore aiming to address the first challenge we discussed earlier.

**Methodology** Consider the boundary $B_{ij}$ between any two adjacent layers $L_i$, $L_j$. In other words, $i - j = 1$. In order to characterize the state of the network below the boundary $B_{ij}$, we define a set of $k$ operational metrics $\mathbb{N} = \{N_1, N_2, \ldots, N_k\}$. Similarly, to characterize the service from layer $j$ to layer $i$ , we define a set of $l$ service parameters $\mathbb{P} = \{P_1, P_2, \ldots, P_l\}$. For a given layer boundary, we divide the operational and service space in to three regions, based on a number of scenario-specific factors. These are termed as normal, partially degraded, and severely degraded for the operational space and acceptable, impaired, and unacceptable for the service space. Resilience $R_{ij}$ at the boundary $B_{ij}$ is then evaluated as the transition of

the network through this state space. The exact formulation to derive the $R_{ij}$ as a function of $\mathbb{N}$ and $\mathbb{P}$ remains a part of the future work. As an example, in the simplest case $R_{ij}$ is the slope of the curve obtained by plotting $\mathbb{P}$ vs. $\mathbb{N}$ on a multivariate piecewise axis, as shown in the preliminary test results below.

**Multi-level Resilience** In the multi-level analysis, 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 procedure is repeated until we reach the application layer $k$ and the resilience evaluated at the boundary $B_{k-1,k}$ is considered the overall resilience $R$ of the network. Of course, the resilience evaluation across multiple layer boundaries is context specific. A generic evaluation of resilience across different context remains an open question that will be explored in the future.

**Steady State vs. Transient Analysis** The proposed methodology supports two modes of resilience evaluation: steady-state analysis and transient analysis. In the steady-state analysis we evaluate the long term view of the network resilience by conducting either theoretical calculations or network simulations to understand the impact of perturbations in the operational state (due to challenges and attacks) on the service parameters. This leads to best, worst, and average case resilience measures. In the transient analysis, we observe the instantaneous state of the network and plot the service parameters of the network in real-time as the network challenges are countered by detection, defence, remediation, and recovery mechanisms. In this case, resilience is characterized by the state transitions that occur in real time. The use of rigorous and formal discrete state analysis remains a part of the future work.

**Framework Challenges** There are several challenges that need to be addressed in the proposed framework. One of the most significant challenges is the selection of metrics that characterize the operational state of the network at any given layer boundary. In the absence of easy-to-use independent metrics, we must consider the interdependence between several metrics. In the transient analysis, the number of states may explode if each individual operational point of the network is considered a separate state. However, combining individual instances of network operations into aggregate states using sensible thresholds could limit the number of states. Lastly, resilience evaluation is context based – a universal resilience measure across a wide range of contexts remains a challenge.

**Preliminary Test Results** As discussed above, we conducted initial studies at the $B_{23}$ boundary between the link layer 2 and the network layer 3. In this case, a set of vertices $V$ and edges $E$ and link failures $f$ characterize the operational state of the network. Since we consider only link failures, we chose a single operational metric $f$ to represent the number of link failures. The service across this boundary is hop-by-hop paths, in particular, a connected graph. In order to characterize this service, we chose the two initial service parameters as the average node degree $d$ and the relative size of the largest connected component $lc$ [1]. Figure 2 shows the steady state resilience of the Sprint network to link failures as degradation in the service from acceptable to unacceptable region. The region boundaries in both the operational and service dimensions are arbitrarily chosen. We show the best, worst, and average case depending upon

[1] These parameters represent an initial pick chosen to simply illustrate the proposed methodology. Further work is needed to narrow down the exact set of metrics that quantify a given graph.

the location of the link failures. In the best case scenario, we see that the network remains in the acceptable service region even when the network is partially degraded. In the work case, the Sprint network provides unacceptable service in the presence of a single link failure.

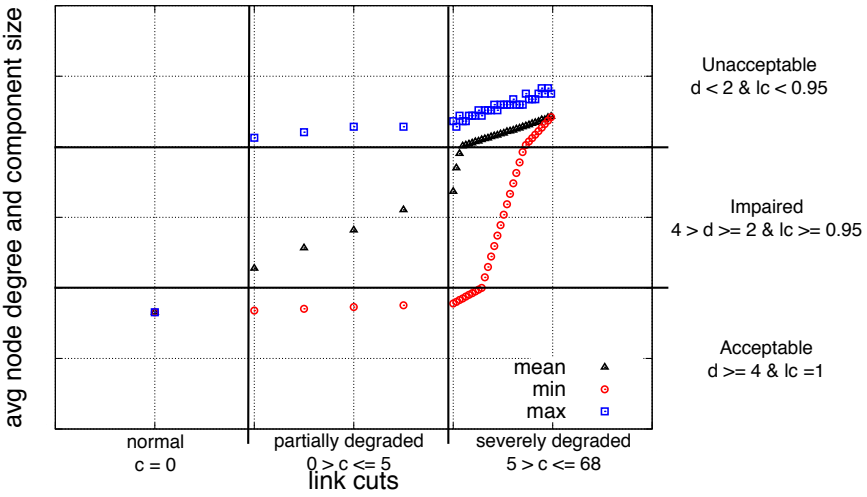



Figure 2: Resilience at boundary $B_{23}$ for the Sprint network

### 2.3.3 Robustness Quantification and Optimization Framework

After the first component of the framework has primarily focused on an investigation of service degradation at the boundaries of the network level stack to perturbations, the second component of the overall framework in ResumeNet will further investigate the concrete root cause for the degradation, i.e., pinpoint the successful resilience (or its lack thereof) to specific parts of the network graph, and therefore provide deeper insights into the robustness of a network and potential robustness optimizations.

Due to the difficulties in finding appropriate service metrics as discussed above, the second part of the framework therefore, for the moment, focuses on topological metrics due to their exact and well-defined nature. However, initial investigation has shown that the topological approach can also be scaled and applied towards more user-centric, service-level metrics in the future.

**Exactness and hard guarantees** One of the main advantages of the second framework component is its ability to provide exact results which provide hard quality guarantees to the network designer and operator. In other words, for any type of network topology and any specifiable challenge scenario (ranging from for example random failures, natural disasters to intentional attacks on the network structure), its robustness quantification can determine **provable** best, worst and average performance levels that can be expected from the network. We believe that providing such hard guarantees (in contrast to a maximum and minimum estimate as obtained after $k$ probabilistic simulation runs) will be an important requirement of future potential adopters of the ResumeNet project results, i.e., a network designer who is committing effort to design a resilient network structure requires a level of assurance that the resulting topology will react within the specified parameters under any foreseeable circumstances.

**Measures of Uncertainty** As each class of challenge to a network (equipment failures, natural disasters, etc.) will have different levels of impact on the overall system, it is important to further quantify the level of uncertainty that will be faced with each challenge type and network topology, e.g., in the case of a heavy electromagnetic storm one needs to determine the different outcomes between the provable worst case (the storm is forming in a highly populated area or a core part of the network critical to communication with the highest possible impact on the network) and the best case (the storm is in a sparsely populated or low traffic location). This difference or, in other words, the envelope between the best and worst case situations is also an important measure and metric to network resilience, as a network operator might want to design a network to simultaneously bound the overall impact of challenges, as well as their variability to make challenges and their impact more predictable. Figure 3 shows such a case study for the Sprint and Geant network for an exemplary topological metric. As can be seen from the graph, while on average (e.g., random equipment failures) both commercial networks perform reasonably well to challenges, both network show serious deficiencies in the worst-case situations (e.g., a co-ordinated attack on 2 to 4 links). Based on these insights, the robustness quantification and optimization framework can then provide recommendations on how to reduce the overall vulnerability and uncertainty for specific classes of challenges.

**Robustness Optimization** The final contribution of the second framework component is its ability to derive optimization instructions that will strengthen overall resilience, as well as reduce the amount of uncertainty. This can be calculated on a per component basis (given that $k$ nodes or links are added, what is the best incremental placement) or an alternative network topology can be derived that would maximize robustness. Figure 3 shows an example: The green Lattice graph is a theoretical alternative investigated by the Robustness Quantification and Optimization Framework to the Geant network that contains a comparable number of links, yet provides much better overall resilience and far less total uncertainty than the original network.

## 2.4 Policies

In the ResumeNet project, policies will play a key role in determining how remedies are selected in response to detected challenges. Policies have been defined by M. Sloman et al. [DLSD01] as "rules that govern the choices in behaviour of a system." Later work by Strassner [Str03] defined *policy-based network management* as "the usage of rules to manage the states of managed entities and to accomplish decisions." An important property of policies is that they can be changed at run-time, e.g., in order to adjust to new business goals of the company.

There has been significant work on policy-based network management, which we briefly introduce before presenting the specifics of network resilience policies. [BX02] provides a good overview of policy-based network management and its state-of-the art (until 2002). In particular, they use the definition of Sloman and mention that "the management system is tasked with: the transformation of human-friendly management goals to syntactical and verifiable rules governing the function and the status of the network." This breakdown of human-friendly management goals to low-level technical policies that can be directly applied at network devices is also followed by Strassner and is called the *policy continium* [Str03]. Strassner defines a framework for policy-based network management using tuples consisting of respectively an **E**vent, a **C**ondition and an **A**ction (ECA tuples). Using ECA for expressing policies seems to be a well-accepted approach in the community, since it is simple and powerful. Furthermore, Strassner defines the *DEN-ng* model [Str03], which provides instructions on

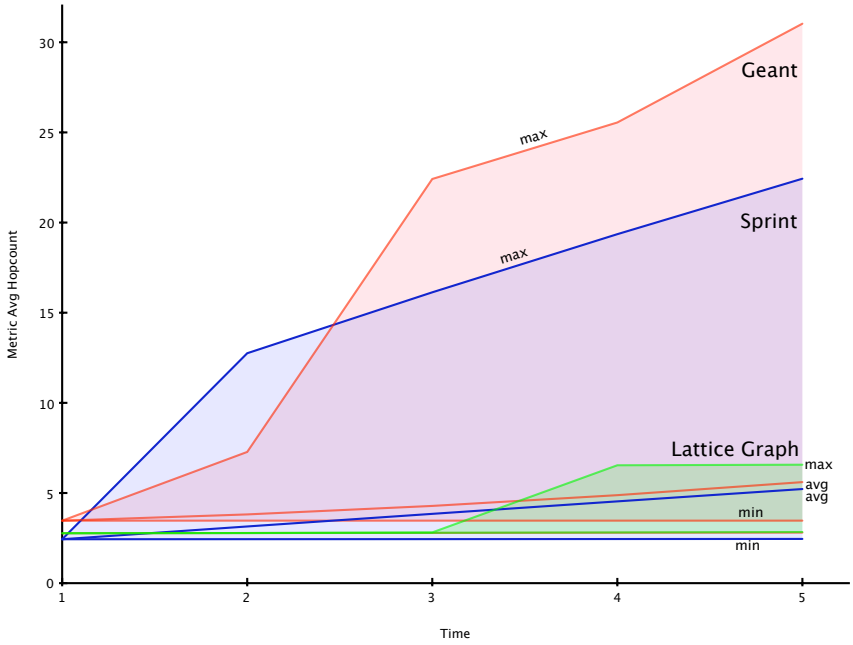



Figure 3: Provable best, worst and average case performance boundaries for the Sprint, Geant and a comparable Lattice network structure

how to express ECA tuples. DEN-ng differs from other policy frameworks, e.g., the Common Information Model (CIM) which allows for the sets of events, conditions or actions to be the null set as well. This is not possible with the DEN-ng model. Sloman et al. defined Ponder, which is a Policy specification language [DDLS01, LLS04, TDLS09]. The IETF defined a framework for policy-based network management [RFC2748, RFC2753], which is a management-agent model with Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs). The protocol standardised by the IETF is the Common Open Policy Services (COPS) (although today Netconf could be more appropriate). Other languages for expressing policies are, e.g., the Policy description language [LBN99] and the Web Service Policy Language (WSPL) [And04].

A first step towards network resilience policies has been taken by the EU FP7-funded INTERSECTION project. This project is considering approaches to intrusion tolerance for multi-domain heterogeneous networks. It has developed a framework for intrusion tolerance that is similar to our resilience strategy – $D^2R^2 + DR$ – and populates some of the details it masks. One of the key components of the framework is a *reaction* component that can be used to make policy-based decisions on how to remediate attacks. Based upon policies, the reaction component tries to resolve the following three issues:

1. *Determine the most appropriate remedy to deploy in response to an alert.*
   Based upon the type of attack, different remedies are employed, e.g., for a DDoS SYN attack, if bandwidth starvation is occurring on the path towards a server under attack, then traffic is rate-limited; otherwise for a low-volume SYN attack, a proxy is deployed that uses SYN-cookies to only allow legitimate TCP connection requests. It is intended the severity of the attack is expressed in an IDMEF [DCF07] document that is made available to the reaction component. Furthermore, the operational constraints of a remedy, its deployment cost and the value of the resource to be protected are considered when choosing an appropriate remedy.

2. *Determine where to deploy remedies.*
   The reaction component has to be kept abreast of the existence and capabilities of devices (e.g., routers) that can be configured to enact remedies. It is intended that targets for reconfiguration (devices) can be placed into domains and sub-domains, such that a particular type of remedy can be deployed in one sub-domain, and another remedy in a different sub-domain.

3. *Determining the appropriate configuration parameters for deployed remedies.*
   This will be based upon the severity of the attack that has being detected and the normal behaviour of the network and associated services, i.e., the normal or maximum functional load. Furthermore, this includes parameters for determining when to withdraw a remedy based upon an understanding of normal behaviour, e.g., traffic rates.

Obligation policies which define which actions to perform in certain circumstances, are being used, and the project is considering Ponder [TDLS09] to determine whether policies can be formulated, so the issues specified above can be addressed. However, there a few shortcomings of the INTERSECTION approach, which needs extension for usage within ResumeNet and the broader range of challenges it considers: In order to statically associate a remediation strategy with a challenge requires a *very detailed understanding* of this challenge and a *perfectly matching remediation strategy*. Applying rate limiting in the case of a DDoS attack is an example for such a static binding of a specific remediation strategy to a challenge. But, if more than one remediation strategy can be applied to a challenge and determining which of them fits an adverse condition best is difficult or impossible, such a static binding cannot be applied. Moreover, such a static binding of remediation strategy to challenges only allows the system to deal with anticipated challenges. New kind of challenges, like new types of attacks, require a more generic approach to resilience policies. In the extreme we have to define policies which allow the system to choose from a large set of remediation strategies. At first as no a priori knowledge of the adverse condition is available but the service is regarded critical, trial-and-error could be the only option. Later, as the system has tested and evaluated several remediation cycles for an adverse condition, learning methods are envisioned to provide more fine-grained guiding of the remediation selection process. Therefore, resilience policies have to incorporate *policy weights* which represent the success rate of the specified remediation strategy in comparison with other strategies. In other words, the policy weight indicates how successful the application of a remediation strategy has been in the past.

An example currently under investigation is a set of policies for unanticipated events which leads to the following behaviour:

1. if a service fails or operates due to an unknown reason, restart the service;

2. if the failure persists, restart the service from a different code-base (exploit operational diversity);

3. if the failure persists, check for other failing services communicating with the same peer service. If more services experience the same failure, restart peer service – from different code-base, if applicable;

4. if other services can communicate with the peer normally, determine services on which the failing service depends and identify erroneous behaviour. Apply this process to any service which operates erroneously.

There are numerous pitfalls for such a strategy, e.g., zero-day attacks which cannot be fixed by the system itself and would lead to an even more severe adverse condition as the system is constantly restarting services and analysing the status of dependent services.

## 2.5 Multi-level Resilience

The design of the Internet protocol suite is strictly hierarchical, with protocols organised into layers. While this strict layering has contributed significantly to the success of the Internet, it is becoming increasingly well-understood that it has limitations [JIZ92]; especially in emerging network types that will play an important role in a future Internet, such as mobile and ad-hoc wireless networks [BCD06]. The drawbacks stem from the fact that interoperability between layers (information sharing and control) to enable better performance and resilience is not readily supported. Consequently, a number of cross-layer architectures [TS07, LBS02] and bespoke scenario-specific optimisations [RPSM07, GQ08] have been proposed. Remediation of a challenge or number of challenges at the same time can be complicated. For example, Jabbar et al. [JRO+09] propose a mechanism that uses weather information to pro-actively route around regions of precipitation in millimeter wave wireless mesh networks, whose performance degrades significantly in heavy rain, to reduce packet loss. At the same time, a set of malicious nodes could be orchestrating a wormhole attack, which invokes another perhaps contradictory route-around mechanism. Clearly, these mechanisms have to be co-ordinated. Multiple interactions of control loops, which interfere with each other, naturally might cause instabilities in the network [KK05].

As a starting point to address this problem regarding cross-layer interactions, a theoretical cross-layer control framework needs to be developed. This framework will be used for calculating and reasoning about cross-layer interactions. Resilience developers will be studying challenges. They will be developing resilience mechanisms, which might require the employment of cross-layer interactions. The system for calculating and reasoning will help to understand fundamentally the interactions [LMMR05, LMM+06]. The intention is to enable developers of resilience mechanisms to be able to formally describe the interactions, find a trade-off, and avoid the undesirable effects being invoked by the resilience mechanisms. For the formalisms, evaluation models, and assessment methodologies, we might apply theories from system theory, formal methods [KDG08], mathematical logic, fuzzy logic [CCLK08], Markov chain models [FvdS07], and Petri-nets, for example. As a starting point, we are developing a notation that can be used to help reason about the cost versus benefit trade-offs associated with the various approaches to cross-layering.

Underpinning a framework for cross-layer information sharing and control will be a distributed monitoring platform. We have investigated potential options for this; two interesting candidates are the ISS Framework [SFH07] and the ANA monitoring architecture [GGH+].

At the core of the ISS Framework lies a distributed Information Sensing (IS) architecture, which supports a network-wide knowledge plane. The framework, which is presented in more detail in [SFH07], is a unified approach to managing cross-layer and network-context information sharing, with an event-notification system that decouples the information collection process from the consumption of this information. An event composition service enables event consumers to specify interests in event abstractions that are semantically meaningful to it. These events are logical composites of information sources. Sources may be simple primitives, such as protocol state, or more complex measures such as packet loss, flow rate, etc. An event consumer may also act as an event producer, enabling more complex, cascading event struc-

tures. The framework also enables information exchange over the network among distributed instances of the framework, and delivery of the information in a uniform manner, independent of location. A prototype implementation of the IS event registration and notification system has been tested with a number of hand-coded scenarios [SFH07]. These include typical cross-layer scenarios, such as the solution to the canonical TCP congestion event problem in wireless networks [CSN01]. These results encourage further use of the framework, and we have begun determining the suitability of the ISS Framework code-base.

The ANA monitoring architecture [GGH$^+$] acknowledges that there may be a number of different mechanisms for obtaining monitoring information that are suitable in different contexts, and that there is often replication of measurement tasks. Consider a node in an overlay network trying to determine its relative network position to its peers, it could either directly probe these peers (e.g., using ICMP) or use a virtual co-ordinate system, such as Vivaldi [DCKM04]. There are trade-offs associated with using each of these approaches – direct probing is more expensive and will need replicating for each peer, but virtual co-ordinate systems, while more efficient, are known to perform poorly in wireless settings. The ANA monitoring architecture introduces an intermediary component, called Orchestration, that selects appropriate measurement tools based upon a client's monitoring requests that specify parameters, such as accuracy of the desired measurement and timeliness of its delivery. To enable re-use of measurement information, a Multi-Compartment Information Sharing (MCIS) component is used to store measurement data in a distributed manner, and allows range-based queries to be executed over it. In the ResumeNet project, we will develop a component that is similar to the MCIS, which can be used to store information regarding challenges, their detection, and subsequent remediation. This could be used as a basis for the longer-term evolution of the system – the *diagnosis* and *refinement* stages of our strategy. Currently, we are seeking access to the implementation of the ANA monitoring architecture. When this becomes available – it is still in development – we will investigate its suitability for use within the project.

# 3 Towards a Realisation of our Resilience Strategy

As mentioned earlier, in the ResumeNet project, we have adopted two complementary approaches to realising resilience that use the $D^2R^2+DR$ strategy as a basis: *service* and *network* resilience. We describe our on-going activities in this area; for a more detailed description of these activities please see [Con09].

## 3.1 Service Resilience

Because of the ossification of the Internet, network support for resilience is unlikely to be forthcoming in the near future. Thus, applications need to deploy their own resilience mechanisms at the application layer, and aim to hide the impact of challenges arising on the lower layers. Unnecessary dependencies on other intermediary services components, e.g., addressing services and resolving a service location, may be avoided as well, where possible.

Although the mechanisms that are currently investigated in ResumeNet for service resilience can be considered as a complementary set of *proactive* and *reactive* mechanisms, the focus is much more on those which are proactive, since an ideal goal would be to reduce the Time-To-Repair (TTR) to zero. Some proactive mechanisms can achieve this, e.g., redundancy and fault tolerance. Other mechanisms are a mixture of proactive and reactive, by having a prepared set of remediation and recovery patterns that can be applied as soon as a challenge

is detected and therefore minimise the TTR.

Some of the solutions proposed today that can provide remediation from challenges occurring in lower layers are, e.g., the SCTP transport protocol [Ste07] and the Host Identity Protocol (HIP) [MN06]. SCTP supports multi-homing and failover from one IP address to another. HIP offers a constant host identifier on top of IP to the application. Therefore, when a failover to another IP address occurs the application should not notice the change. However, both of these solutions have their limitations in terms of resilience, despite being designed with it in mind. SCTP can perform only a soft failover, i.e., there has to be always at least one established flow. HIP requires DNS extensions and rendezvous-points, which can be single points of failure.

Based on these thoughts, we aim at studying the benefits of the deployment of further innovative methods for providing better abstraction from the underlying layers as well as reducing dependancies on other services and intermediary components in the network. Some of these methods are, e.g., system virtualization and overlay networks, in particular, peer-to-peer (P2P) networks.

P2P overlays provide promising resilience properties, such as data and link redundancy, and the autonomic recovery from local failures. Operating system virtualization provides the ability to migrate, duplicate or start new services, which can be used in case a challenge in the underlying resources is detected, for example, the hard disk breaks down, the network connectivity becomes degraded, or a power failure requires the system to run on battery. Migration of virtual machines is usually performed within a subnetwork. However, this does not allow for geographic diversity. Therefore, we will consider scenarios for migrating virtual machines across the Internet. Geographic diversity was considered to be prohibitively expensive, since it was not always possible or easy to host services anywhere in the world. Today, this is becoming increasingly common. Therefore, we are considering geographic diversity as one of the requirements for resilient services.

To underpin our work on peer-to-peer networks and system virtualization, it is necessary to monitor supported services and continuously have an up-to-date view of the performance and the Quality of Experience (QoE) provided. Therefore, it is necessary to collect diagnostic information, and if needed perform the appropriate correlation for detecting abnormal behaviour and performing an appropriate remediation strategy, e.g., migrating a service hosted by a virtual machine, or in the worst case stopping the old service, e.g., if it has been compromised, and starting a new service, eventually in another location in the network.

## 3.2 Network Resilience

Building a resilient networked system means adding resilience functionality to the services the communication system provides. In line with our strategy, we therefore exploit a cross-layer approach to *defend* against challenges, *detect* challenges and service failures, and to adapt and evolve the system, in order to *remediate* challenges and *refine* the network.

**Defensive Measures** The first task when building a resilient network is to design an architecture that a) is optimal for its expected "normal" state; b) can tolerate a list of foreseen potential adversarial events, such as those derived via the risk assessment approach discussed in Section 2.2. It is also necessary to develop tools based upon sound metrics, to efficiently evaluate the proposed architecture. We are investigating three forms of defensive measures, ranging from longer-term off-line decisions to real-time mechanisms:

1. Design an "optimal" topology: for a given set of access nodes, potential transmission and transport technologies, define the best possible way to interconnect the nodes, on a permanent or non-permanent basis, so that a prescribed set of demand scenarios can be accommodated.

2. Define and tune the routing and addressing mechanisms that will allow the connections to take place with the best possible conditions, and react as fast as needed to potential failure situations.

3. Define and implement the best possible curing policy in a fast-evolving and sometimes very aggressive environment, so that the communications and the content stored in the network remain free from viruses and protected from other potential attacks.

**Challenge and Failure Detection** Despite having built defensive measures into the networked system, challenges will trigger faults and cause service failures. There are two main reasons for this: firstly, we do not know how to design or implement a fault-free complex system, which includes the defensive measures, nor are we able to operate and maintain such a system without introducing new faults. Secondly, we can not foresee all possible challenges a system will face in a deployment scenario. Therefore, we argue that to be able to determine when a system is challenged and a particular mitigation strategy has been effective, it is necessary to understand the normal behaviour of a networked system. Clearly, this is non-trivial. We are investigating ways to make understanding normal behaviour of a system tractable, based upon an understanding of the high-impact challenges (see Section 2.2), and develop tools for specifying normal behaviour. This will be done in the context of a small number of well-defined challenge scenarios.

It will be necessary to automatically remedy challenges; this comes at the risk of applying inappropriate mechanisms and further reducing the level of service provided by the network. To enable the appropriate automatic selection of mitigation strategies, it is necessary to detect the symptoms of a challenge, understand as well as possible its root cause, and gain a notion of its impact on the system. We are investigating a distributed challenge detection system based on the understanding of normal behaviour that can be used to determine this information. An essential feature of a distributed resilient system is a repository to store, for example, detection and mitigation events. This distributed information store is used during fault localization and fault verification, and provides input to the remediation strategy selection logic. Moreover, this stored information can be used, for instance, to enable long-term learning and evolution of the system. We will investigate appropriate interaction models (e.g., publish/subscribe) and implementation approaches for such a distributed information repository.

**Adaptation and Evolution Framework** Having detected a degradation of service within the network, an adaptation framework is triggered to select and deploy a remediation strategy (see the upper part in Figure 4). Such a strategy can be as simple as turning some control knobs to adjust configuration parameters of the affected services. The other extreme would be to change the complete communication subsystem architecture, i.e., using an autonomic networking approach with functional composition to re-build the protocol stack. Which strategy to apply for a given challenge in a specific context is task of the remediation decision engine. Deciding which strategy to apply will be based on the detected challenge, the system's operational context, the network operator's system policies, and experience from past decisions. In order to enable learning over time, information about the selected strategy is stored in the

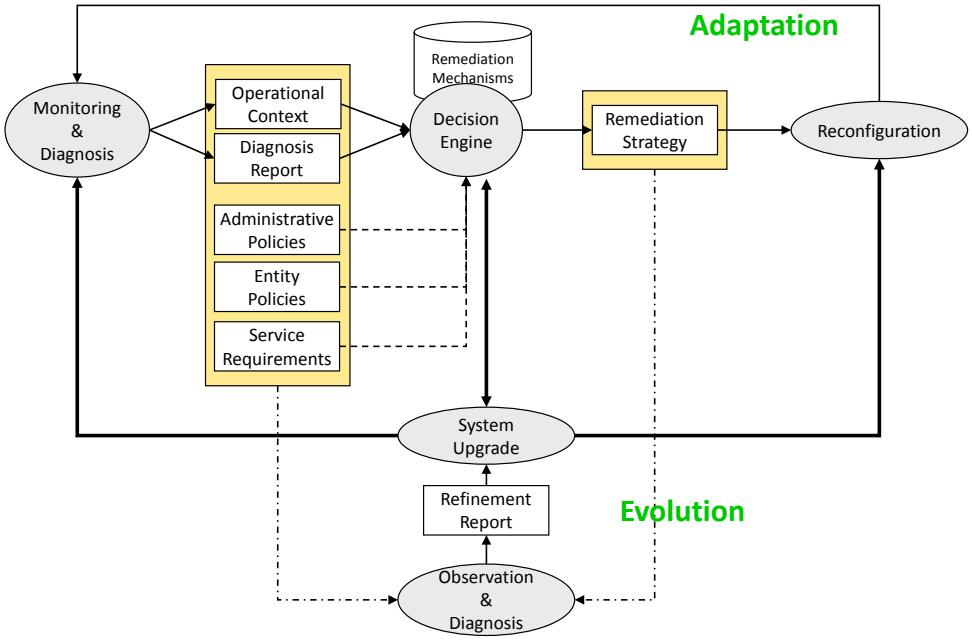



Figure 4: The adaptation and evolution framework, including challenge detection

distributed information store, together with pointers to the related challenge. A measurement unit, which verifies the effectiveness of a deployed remediation strategy, does not only close the real-time control loop but also make its assessment of the strategy's effectiveness persistent by storing the measurement result in the same store. Feeding this information into long-term learning algorithms is expected to provide a basis to more readily select the right remediation strategy in the future. This off-line learning is implemented in an evolution framework (see the lower part in Figure 4). The work on the adaptation and evolution framework begins officially in M13 of the project lifetime.

# 4 Experimental Evaluation

In the experimentation part of the project, the aim is to exemplify our approach to resilience in concrete study cases. Study cases could be viewed as *networking technology, service provision scenario, challenge set* tuples. We have tried to cover some of the most recent and promising networking paradigms, their common feature being high decentralization in the way they operate and are managed. Interestingly, their deployment is in many cases hindered by the lack of solid solutions to challenges of various types such as a) deliberate attacks to the network infrastructure; b) lack of incentives for collaboration; c) management mechanisms that could cross various heterogeneous networking settings.

Although work on experimentation begins in the second half of the project lifetime, significant effort has been devoted so far to the more detailed definition of the experimentation scenarios and the respective testbed developments, where appropriate. This work is directly affected by the progress made on the framework (WP1) and mechanism (WP2-WP3) aspects of the project. The aim is to accommodate the maximum and most important elements in the experimental evaluation. In the following, we briefly present the four experimentation study cases in the project, as well as the current status of work.

## 4.1 Coping with Node Misbehaviour in Wireless Multi-hop Networks

The first study case investigates the challenge of node misbehaviour in wireless multi-hop networks. As the demand for wireless access grows, several types of these networks become interesting, for example, wireless metropolitan networks. The decentralized nature of these networks gives many degrees of freedom to users; yet, they rely on the full co-operation of nodes, which cannot be taken for granted. Both selfish and malicious behaviours threaten the normal operation of these networks, even their existence as such. In our experimentation, we consider several types of misbehaviour. Jamming and generation of dummy traffic are two instances of malicious node behaviour. On the other hand, selfish node behaviour can be expressed as: a) greedy over-usage of the common transmission medium (due to the distributed nature of the the MAC layer); b) refusal to participate in the data forwarding process. The latter is the focus of our current work.

We have carried out analysis looking at the total achievable throughput in a wireless mesh network when a certain number of nodes are misbehaving. To gain a measure of performance degradation, we compare the achievable throughput in an ideal scenario, where a central entity with full knowledge about the network applies interference-aware routing and optimal scheduling of transmissions at MAC layer. Our analysis will be compared against experiments on the ETH Zürich wireless testbed (TIKnet) [TIK], through which we can assess the validity of the model, and the impact of radio propagation aspects. This work will draw on and, at the same time, inform the work we are doing on resilience metrics.

As a second step, we will experiment in the testbed with a reciprocation mechanism we are developing. The protocol under design draws on game-theoretic principles and, more specifically, on the field of mechanism design without money. The aim is to offer incentives to the nodes so they co-operate and the network is functional. This protocol will be one of the *defensive* mechanisms in the $D^2R^2 + DR$ strategy, which should be active for such networks.

## 4.2 Coping with Node Misbehaviour in Opportunistic Networks

Opportunistic networks depart from the assumption of an end-to-end connection. Instead, information is stored, carried, and forwarded to other nodes that are expected to bring the information closer to the destination(s) as they pass by. They offer an interesting aspect to resilience: even though an opportunistic network is by definition resilient to link and node failures, resource limitations require the careful selection of when and where to replicate and forward data. This selection process, also known as congestion control, can easily be disturbed by misbehaving nodes. They can choose to not collaborate in forwarding data, or introduce (useless) data into the network using other node's resources, for example. Therefore, we consider opportunistic networks as a challenging study case for the interaction between challenge detection and the adaptation and evolution framework in order to perform congestion control, even with misbehaving nodes.

The experimental work will be conducted on the Haggle architecture [NGR09], because it introduces a novel and well-structured design space for distributed congestion control. At the core of the data-centric communication architecture is a relation graph between data and users' interests, which is maintained by every node. Relations are weighted according to matching metadata and other parameters, such as the lifetime of data. Weighted relations allow setting thresholds and ranking of the relevance of data for a particular user. These parameters, and the choice of forwarding strategy, build an interesting design space for distributed congestion control and opens interesting research questions. We intend to investigate the design space

of the weighting function, resolution parameters, and choice of forwarding strategy; in a first step to better understand the choice of parameters in general, and then in the context of misbehaving nodes. We analyze *precision* and *recall* scores of received data as a benchmark for the performance of the data dissemination. The experiments will be conducted in the Haggle testbed running nodes in a virtual environment.

### 4.3 Reliable Signalling for Multimedia Sessions using Peer-to-Peer Networks

We have been running experiments on PlanetLab [PLA] with Cooperative SIP (CoSIP) [COS] signalling, where we have a SIP [RSC+02] server and a Distributed Hash Table (DHT) [KAD, SMK+01] in the background. This approach differs from most others for P2P-based SIP signalling, in the sense that it is a hybrid approach with a server and DHT together. The server is needed for bootstrapping the security mechanisms, e.g., X.509 certificates, which allow for P2P authentication and integrity of the data stored in the DHT. The server is also used for signalling under normal operation, which usually provides better lookup performance than the DHT.

A straightforward approach for signalling with a server and a DHT would be to initiate the signalling with the server first. The server will respond fast under normal operation. The SIP User Agents (UAs) wait until the server responds or until a timeout occurs. When the timeout is fired, signalling via the DHT can start. This would mean that the session setup would take the time required to signal using the DHT plus the timeout interval. Nevertheless, it will still be successful, which is not the case if the server is down, and there is no DHT as a backup. Therefore, signalling via the DHT when the server is unreachable can be considered as a remediation mechanism. The recovery will occur when the server becomes available and back to operation. One potential issue with this first approach is that many DHT algorithms provide better connectivity when the DHT is used frequently for lookups. Some DHT algorithms perform lazy maintenance (e.g., refresh routing tables once an hour) and profit from the lookup to update their routing table entries. This is the case, e.g., for the Kademlia DHT algorithm [KAD]. Some other algorithms, e.g., Chord [SMK+01] or Pastry [RD01], perform stabilisation frequently (e.g., refresh routing tables every five minutes). Therefore, this has to be considered when designing such a co-operative system. It has to be ensured that the DHT signalling works fine under normal conditions and not only when the server fails. In fact, this can be compared to the fire alarm tests or power outage tests, which are required occasionally. Otherwise, the fire alarms or the measures for coping with power outages might fail when they are really needed.

A better signalling approach would be to perform the session setup initiated by signalling to the server as well as in the DHT in parallel. This means that in case the server is not reachable, the session setup will take the time required required to signal using the DHT but no additional timeout interval. Using this approach, the DHT is also used frequently, which can ensure its continuous functionality. This approach can be considered as *proactive*. Regardless of whether a challenge is detected or not, the signalling is always performed in parallel to the server and the DHT. The result is a lower Mean-Time-To-Recovery (MTTR) at the cost of additional signalling overhead. However, this cost can be optimised.

To summarise, the approach followed for resilient SIP signalling is inline with the $D^2R^2 +$ DR strategy. However, we have a strong focus on proactive mechanisms. Proactive means that defence is consolidated, and that mechanisms for remediation and recovery are always ready to be applied as soon as a challenge is detected in order to achieve the best possible

MTTR, which should be zero.

## 4.4 Addressing challenges in Ambient Intelligence (AmI) Environments

Ambient Intelligence (AmI) can be characterized using the image of people surrounded by smart and intuitive interfaces embedded in everyday objects around them, and an environment recognising and responding to the presence of individuals in an invisible way. To provide AmI, built on ubiquitous computing, ubiquitous communication, and intelligent user interfaces, the Object Naming Service is proposed as a mechanism leveraging DNS to discover information about a product and related services from the Electronic Product Code, a component of the EPCglobal Network. In this vision of the Internet of things, bar-codes are replaced by contactless chips (RFID) on all manufactured products, giving access via the Internet to dynamic data updated on each object (data on the origin, shipment of the merchandise, traceability, etc.). The ICOM platform, built for mass distribution enterprises, is a middleware aiming to be the crossroads for all these RFID data, gathered through EPCglobal standards compliant infrastructure, and transiting towards the enterprises' information systems, or trading partners' networks. Based on open source modules and complemented with components developed by France Telecom, this publish-subscribe environment aims to become a platform offer for shared services.

Overcoming resilience challenges is one of the conditions of success for this environment. Data reliability at the source is the first and basic consideration. High availability of the platform, even in the event of bizarre traffic conditions and/or abnormal situations is another requirement. As enterprises entrust their data to an external infrastructure, the guarantee of data confidentiality is unavoidable. Another cause for concern is the threat of DDoS attacks saturating the enterprises' servers in their information systems. This non-exhaustive list of challenges will be studied in the light of ResumeNet's $D^2R^2 + DR$ strategy, and particularly through its Detection (monitoring/surveillance, event correlation, etc.) and Remediation (dynamic policy change, etc.) steps.

# 5 Conclusions and Outlook

Resilience will need to be a key property of a future Internet because of our unrelenting demand for Internet-based services, the challenging environments it will operate in, and the continued existence of intelligent adversaries. We propose that a holistic and systematic approach to resilience is required that aims to consider the complete socio-technical system and the wide range of challenges it may face. In this deliverable, we have described our current understanding of how to build resilient networked systems, such as the future Internet, that uses a superficially straightforward conceptual strategy, called $D^2R^2 + DR$. All this strategy says is that a resilient networked system must carry defensive mechanisms, be able to detect and remedy challenges that breach defences, and recover to a normal state of operation when a challenge has abated. Furthermore, it acknowledges the network should improve over time via a reflective mechanism of diagnosing past failings, which informs refinement.

We have described our current understanding of a number of fundamentals for resilience that constitute part of our resilience framework. Namely, that metrics are essential for us to be able to specify and measure resilience. We describe the challenges of defining resilience metrics and our current status addressing them; much further work is needed. It is essential to make appropriate use of the limited resources available for resilience. To do this, we need to understand the high-impact challenges that will occur, which requires sound probabilities of challenge occurrence and measures of impact. Here, we summarise a risk assessment strategy and a taxonomy of challenges that can be used as part of the assessment process. Managing complexity will be a significant challenge in a future Internet. Some of this complexity could occur from multiple cross-layer control loops that cause the network to become unstable. We aim to develop formal methods for resilience engineers to reason about such control loops, so as to improve confidence in a system's potential stability. Policies will be used to direct system adaptation to mitigate challenges; we have described our understanding of the state-of-the-art in policy frameworks, their use in a complimentary project to ResumeNet, and how we may use policies to mitigate challenges we cannot determine the nature of.

A discussion of our two complimentary threads of research that aim to populate our strategy for resilience with specific algorithms and mechanisms is given. This work will build upon the framework presented. We are at a relatively early stage in our understanding of how to approach ensuring resilience for a future Internet. Our work will continue in the context of strong experimental scenarios that we believe will feature in a future Internet. Through these scenarios we will evaluate the validity of our strategies for resilience and what we understand to be fundamental to it.

## References

[ABPW99] C. Alberts, S. Behrens, R. Pethia, and W. Wilson. Operationally Critical Threat, Asset, and Vulnerability Evaluation Framework, version 1.0. Technical Report CMU/SEI-99-TR-017, Carnegie Mellon University, June 1999.

[ANA] The EU IST-FET-funded Autonomic Network Architecture (ANA) Project. http://www.ana-project.org.

[And04] Anne H. Anderson. An Introduction to the Web Services Policy Language (WSPL). In *POLICY '04: Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*, page 189, Washington, DC, USA, 2004. IEEE Computer Society.

[BCD06] E. Borgia, M. Conti, and F. Delmastro. Mobileman: integration and experimentation of legacy mobile multihop ad hoc networks. *Communications Magazine, IEEE*, 44(7):74–79, July 2006.

[BIR+08] Sara Bury, Johnathan Ishmael, Nicholas J. P. Race, Paul Smith, and Mark Rouncefield. Towards an Understanding of Security Concerns within Communities. In *WIMOB '08: Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*, pages 478–483, Washington, DC, USA, 2008. IEEE Computer Society.

[BV99] Saad Biaz and Nitin H. Vaidya. Discriminating Congestion Losses from Wireless Losses using Inter-Arrival Times at the Receiver. In *ASSET '99: Proceedings of the 1999 IEEE Symposium on Application - Specific Systems and Software Engineering and Technology*, page 10, Washington, DC, USA, 1999. IEEE Computer Society.

[BX02] R. Boutaba and J. Xiao. Network management: State of the art. pages 127–146, 2002.

[CCLK08] Chao-Lieh Chen, Syue-You Chen, Jeng-Wei Lee, and Yau-Hwang Kuo. Hierarchical cross-layer fuzzy control for compromise of multiple objectives in wireless mobile networks. In *Mobility '08: Proceedings of the International Conference on Mobile Technology, Applications, and Systems*, pages 1–7, New York, NY, USA, 2008. ACM.

[CER] CERT. http://www.cert.org.

[Con09] The ResumeNet Consortium. Report of technical work in WP2 and WP3 during the 1st year, September 2009.

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

[COS] CoSIP: An Architecture for highly-available and secure SIP services. http://www.cosip.org/.

[CRA] CRAMM. http://www.cramm.com.

[CSN01] Song Ci, Hamid Sharif, and Guevara Noubir. Improving performance of MAC layer by using congestion control/avoidance methods in wireless network. In *SAC '01: Proceedings of the 2001 ACM symposium on Applied computing*, pages 420–424, New York, NY, USA, 2001. ACM.

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

[DCKM04] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2004. ACM.

[DDLS01] Nicodemos Damianou, Naranker Dulay, Emil Lupu, and Morris Sloman. The ponder policy specification language. In *Policies for Distributed Systems and Networks: International Workshop, POLICY 2001*, pages 18–38, Bristol, UK, 2001.

[DLSD01] N. Dulay, E. Lupu, M. Sloman, and N. Damianou. A Policy Deployment Model for the Ponder Language. pages 14–18, 2001.

[FKF06] Simon Fischer, Nils Kammenhuber, and Anja Feldmann. REPLEX: dynamic traffic engineering based on wardrop routing policies. In *CoNEXT '06: Proceedings of the 2006 ACM CoNEXT conference*, pages 1–12, New York, NY, USA, 2006. ACM.

[FvdS07] F. Fu and M. van der Schaar. A new theoretic foundation for cross-layer optimization. Technical report, UCLA, December 2007.

[GGH+] Vera Goebel, Bamba Gueye, Theus Hossmann, Guy Leduc, Sylvain Martin, Christoph Mertz, Ellen Munthe-Kaas, Thomas Plagemann, Matti Siekkinen, and Dorota Witaszek. Integrated Monitoring Support in ANA. http://www.ana-project.org/deliverables/2008/ana-d3.7-final.pdf.

[GQ08] R. Gunasekaran and Hairong Qi. XLRP: Cross Layer Routing Protocol for Wireless Sensor Networks. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 2135–2140, 31 2008-April 3 2008.

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

[IBPR08] Johnathan Ishmael, Sara Bury, Dimitrios P. Pezaros, and Nicholas J. P. Race. Deploying rural community wireless mesh networks. In *IEEE Internet Computing Magazine*, volume 14, 2008.

[JIZ92] J.Crowcroft, I.Wakeman, and Z.Wang. Layering considered harmful. *IEEE Network Magazine*, 6(1), 1992.

[JRO+09] Abdul Jabbar, Justin P. Rohrer, Andrew Oberthaler, Egemen K. Çetinkaya, Victor S. Frost, and James P.G. Sterbenz. Performance comparison of weather disruption-tolerant cross-layer routing algorithms. In *INFOCOM 2009. The 28th Conference on Computer Communications. IEEE*, pages 1143–1151, April 2009.

[JU08] Almerima Jamakovic and Steve Uhlig. On the relationships between topological measures in real-world networks. *Networks and Heterogeneous Media*, 3(2):345–359, 2008.

[KAD] Kademlia: A Design Specification. http://xlattice.sourceforge.net/components/protocol/kademlia/specs.html.

[Kar09] Merkourios Karaliopoulos. Assessing the vulnerability of DTN data relaying schemes to node selfishness. *to appear in IEEE Communications Letters*, 2009.

[KDG08] Dzmitry Kliazovich, Michael Devetsikiotis, and F. Granelli. *Heterogeneous Next Generation Networking: Innovations and Platforms*, chapter Formal Methods in Cross Layer Modeling and Optimization of Wireless Networks: State of the Art and Future Directions. IDEA Group Inc., 2008.

[KK05] V. Kawadia and P.R. Kumar. A cautionary perspective on cross-layer design. *Wireless Communications, IEEE*, 12(1):3–11, Feb. 2005.

[LBN99] Jorge Lobo, Randeep Bhatia, and Shamim Naqvi. A policy description language. In *Sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference*, pages 291 – 298, Orlando, Florida, United States, 1999.

[LBS02] L.-A. Larzon, U. Bodin, and O. Schelen. Hints and notifications [for wireless links]. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 2, pages 635–641 vol.2, Mar 2002.

[LLS04] Leonidas Lymberopoulos, Emil Lupu, and Morris Sloman. PONDER policy implementation and validation in a CIM and differentiated services framework. In *9th IEEE/IFIP network operations and management symposium (NOMS 2004)*, pages 31–44, Seoul, South Korea, 2004.

[LMM+06] A. Lachenmann, P.J. Marron, D. Minder, M. Gauger, O. Saukh, and K. Rothermel. Tinyxxl: Language and runtime support for cross-layer interactions. In *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society on*, volume 1, pages 178–187, Sept. 2006.

[LMMR05] A. Lachenmann, P.J. Marron, D. Minder, and K. Rothermel. An analysis of cross-layer interactions in sensor network applications. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pages 121–126, Dec. 2005.

[MN06] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423 (Informational), May 2006.

[MPH08] A. K. Marnerides, D. P. Pezaros, and D. Hutchison. Detection and Mitigation of Abnormal Traffic Behavior in Autonomic Networked Environments. In *4th ACM International Conference on emerging Networking EXperiments and Technologies, ACM CoNEXT Student Workshop*, Madrid, Spain, December 2008.

[MS] Microsoft Security Notification Service. http://www.microsoft.com/security.

[NGR09] Erik Nordström, Per Gunningberg, and Christian Rohner. A search-based network architecture for mobile devices. Technical report, Uppsala University, Department for Information Technology, 2009.

[PLA] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. http://www.planet-lab.org/.

[RD01] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, pages 329–350, London, UK, 2001. Springer-Verlag.

[RPSM07] H. Rutagemwa, S. Pack, Xuemin Shen, and J.W. Mark. Robust Cross-Layer Design of Wireless-Profiled TCP Mobile Receiver for Vertical Handover. *Vehicular Technology, IEEE Transactions on*, 56(6):3899–3911, Nov. 2007.

[RSC+02] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol. RFC 3261 (Proposed Standard), June 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621.

[SAN] SANS Institute. http://www.sans.org.

[Sch99] Bruce Schneier. Attack trees. *Dr Dobb's Journal*, 24(12), December 1999.

[SFH07] M. Sifalakis, M. Fry, and D. Hutchison. A Common Architecture for Cross Layer and Network Context Awareness. In D. Hutchison and R. Katz, editors, *Second International Workshop on Self-Organizing Systems (IWSOS)*, pages 103–118, September 2007.

[SMK+01] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, 2001.

[Ste07] R. Stewart. Stream Control Transmission Protocol. RFC 4960 (Proposed Standard), September 2007.

[Str03] John Strassner. *Policy-Based Network Management: Solutions for the Next Generation*. Morgan Kaufmann Series in Networking, 2003.

[TDLS09] Kevin Twidle, Naranker Dulay, Emil Lupu, and Morris Sloman. Ponder2: A policy system for autonomous pervasive environments. In *The Fifth International Conference on Autonomic and Autonomous Systems*. IEEE, 2009.

[TIK] TIK Wireless Testbed. http://tiknet.ee.ethz.ch/doku.php.

[TS07] Geethapriya Thamilarasu and Ramalingam Sridhar. Toward building a multi-level robust intrusion detection architecture for distributed mobile networks. In *ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, page 5, Washington, DC, USA, 2007. IEEE Computer Society.

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