Resilience and Survivability in Communication Networks: Strategies, Principles, and Survey of Disciplines

Article *in* Computer Networks · June 2010

DOI: 10.1016/j.comnet.2010.03.005

CITATIONS
567

READS
3,132

7 authors, including:

James P.G. Sterbenz
University of Kansas
184 PUBLICATIONS **3,726** CITATIONS

SEE PROFILE

David Hutchison
Lancaster University
434 PUBLICATIONS **6,200** CITATIONS

SEE PROFILE

Egemen Kemal Cetinkaya
Verizon
60 PUBLICATIONS **1,754** CITATIONS

SEE PROFILE

Abdul Jabbar
Virtual University of Pakistan
66 PUBLICATIONS **1,939** CITATIONS

SEE PROFILE

Computer Networks 54 (2010) 1245–1265

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines

James P.G. Sterbenz [a,b,*], David Hutchison [b], Egemen K. Çetinkaya [a], Abdul Jabbar [a], Justin P. Rohrer [a], Marcus Schöller [c], Paul Smith [b]

[a] The University of Kansas, Lawrence, KS, USA
[b] Lancaster University, United Kingdom
[c] NEC Laboratories Europe, Heidelberg, Germany

ARTICLE INFO

Article history:
Available online 17 March 2010

Keywords:
Communication network
Future Internet resilience
Fault tolerance
Survivability
Disruption tolerance
Dependability
Reliability
Availability
Security
Performability
Critical infrastructure
Defence
Defense
Detection
Remediation
Recovery
Restoration
Diagnosis
Refinement
Metrics

ABSTRACT

The Internet has become essential to all aspects of modern life, and thus the consequences of network disruption have become increasingly severe. It is widely recognised that the Internet is not sufficiently resilient, survivable, and dependable, and that significant research, development, and engineering is necessary to improve the situation. This paper provides an architectural framework for resilience and survivability in communication networks and provides a survey of the disciplines that resilience encompasses, along with significant past failures of the network infrastructure. A resilience strategy is presented to defend against, detect, and remediate challenges, a set of principles for designing resilient networks is presented, and techniques are described to analyse network resilience.

1. Introduction and motivation

Networks in general, and the Global Internet in particular, have become essential for the routine operation of businesses and to the global economy. Consumers use the Internet to access information, obtain products and services, manage finances, and communicate with one another. Businesses use the Internet to transact commerce with consumers and other businesses. Governments depend on networks for their daily operation, service delivery, and response to disasters. The military depends on the Global Information Grid [1] to execute network centric operations and warfare. The Global Internet may thus be described as one of the *critical infrastructures* on which our lives and prosperity depends, along with transportation infrastructure, power generation and distribution grid [2]. Furthermore, many of these infrastructures have dependencies on one another [3]. The canonical example is that the Internet depends on the electrical grid for power, while the electrical grid increasingly depends on the Internet for SCADA (supervisory control and data acquisition) [4].

* Corresponding author at: The University of Kansas, Lawrence, KS, USA.
E-mail address: jpgs@ittc.ku.edu (J.P.G. Sterbenz).

1389-1286/$ - see front matter © 2010 Published by Elsevier B.V.
doi:10.1016/j.comnet.2010.03.005

However, the increased dependence on, and sophistication of services make the internet more vulnerable to problems. With a continuously increasing reliance come two related consequences: First, this reliance results in increasing consequences of disruption. Second, the increased consequences of disruption lead to networks becoming a more attractive target for cyber-criminals. We believe that *resilience* must be viewed as an essential design and operational characteristic of future networks in general, and the Global Internet in particular. The vulnerabilities of the current Internet and the need for greater resilience are widely recognised [4–8]. In our work, and in this paper, we define *resilience as the ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation*.

This paper provides a broad overview of the discipline of resilience and presents a systematic architectural framework based on several major research initiatives, and is organised as follows: Section 2 introduces the scientific disciplines that form the basis for resilience and presents them systematically. Section 3 describes a selection of significant past challenges and failures of the network infrastructure in the context of threats to the network. Section 4 presents strategies for achieving network resilience and survivability. Next, Section 5 describes a set of design principles to achieve resilience based on the strategy as well as from experience in the resilience disciplines. Section 6 describes the need for analysis of resilience along with a state space formulation and example application of a large-scale disaster. Finally, Section 7 summarises the main points of the paper and suggests further directions of research.

2. Resilience disciplines

There are a number of relevant disciplines that serve as the basis of network resilience, and for which our broad definition of resilience subsumes. Because these disciplines have developed independently over a number of decades, there is no established self-consistent schema and terminology. This section will introduce these disciplines and describe their organisation within the resilience domain, after introducing the important concept of the fault → error → failure chain.

2.1. Fault → error → failure chain

A *fault* is a flaw in the system that can cause an error [9,10]. This can either be an accidental design flaw (such as a software bug), or an intentional flaw due to constraints that permit an external challenge to cause an error, such as not designing a sufficiently rugged system due to cost constraints. A dormant fault may be triggered, leading to an active fault, which may be observable as an error. An *error* is a deviation between an observed value or state and its specified correct value or state [10–12] that may lead to a subsequent service failure [9]. A *service failure* (frequently shortened to *failure*) is a deviation of service from the desired system functioning to not meeting its specification or expectation [9,13,11,12,14]. Thus a fault *may* be triggered to cause an observable error, which *may* result in a failure if the error is manifest in a way that causes the system not to meet its service specification. This relationship is shown in Fig. 1; the boxes labelled *defend* and *detect* are part of the resilience strategy that will be explained in Section 4. For now, note that network defences may prevent challenges from triggering a fault and that many observable errors do *not* result in a failure. Disruption tolerance (Section 2.2.3) is one example of reducing the impacts of fault and errors on service delivery. Furthermore, challenges and errors can be detected, which also provides a basis for actions taken as part of a resilience strategy.

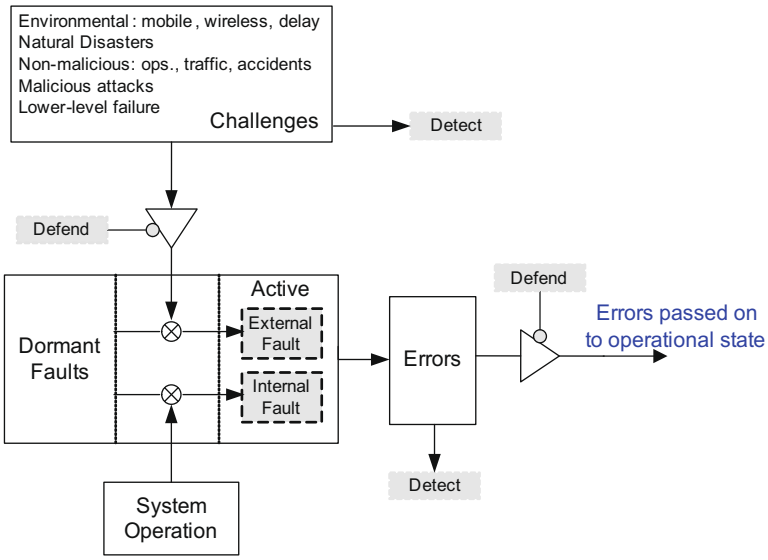


Fig. 1. Fault → error → failure chain.

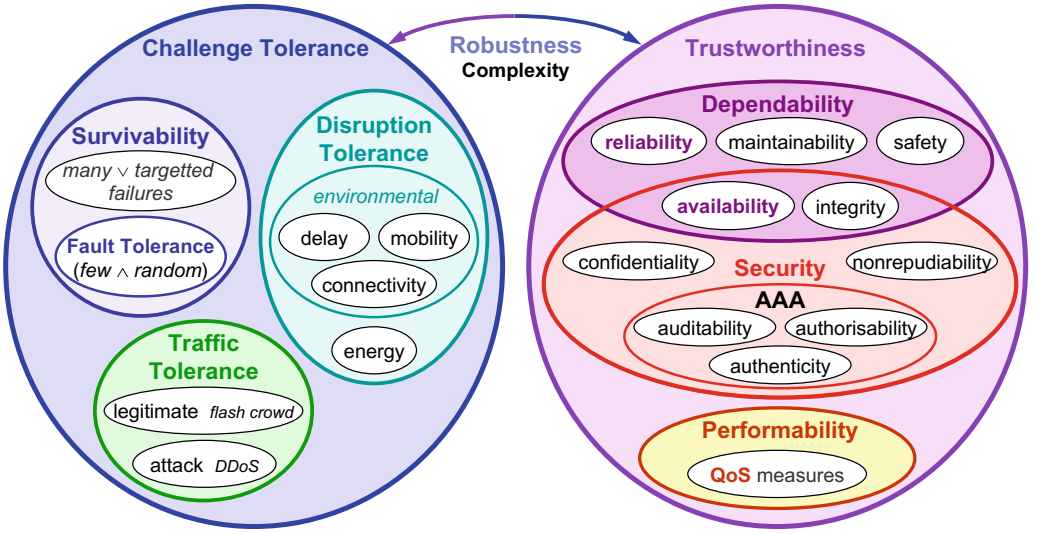


Fig. 2. Resilience disciplines.

2.2. Disciplines relating to challenge tolerance

At the highest level, we divide the disciplines into two categories, as shown in Fig. 2. On the left side are *challenge tolerance* disciplines that deal with the design and engineering of systems that continue to provide service in the face of challenges. On the right sides are *trustworthiness* disciplines that describe measurable properties of resilient systems. The relationship between these two is *robustness*, which formally is the performance of a control system when perturbed, or in our context, the trustworthiness of a system when challenged. Note that a *comprehensive* survey of these fields would occupy far more space than is available in this paper.

The first major subset of resilience disciplines deals with the problem of how to design systems to tolerate the challenges that prevent the desired service delivery. These challenges can be subdivided into (1) component and system failures for which *fault tolerance* and *survivability* are concerned, (2) disruptions of communication paths for which *disruption tolerance* is concerned, and (3) challenges due to the injection of traffic into the network, for which *traffic tolerance* is concerned.

2.2.1. Fault tolerance

Fault tolerance is one of the oldest resilience disciplines, and is defined as the ability of a system to tolerate faults such that service failures do not result [13,11,12]. While the use of redundancy to cover for failures in physical systems dates back centuries, perhaps the first reference in a computing context was the introduction of *N*-version programming in the context of the Babbage difference engine [15] in the mid 1800s. Fault tolerance emerged as a modern discipline in the 1950s when von Neumann and Shannon devised techniques to design reliable telephone switching systems out of relatively unreliable mechanical relays [16]. Fault tolerance was also applied to computer system design in the 1960s, particularly for mission critical systems used in defence and aerospace [17,18].

Fault tolerance relies on *redundancy* as a technique to compensate for the random uncorrelated failure of components. Fault tolerance techniques can be applied to both hardware, such as triple-modular redundancy [19], and to software, such as *N*-version programming [20] and recovery blocks [21], and are generally sufficient when applied to systems of limited geographic scope. Fault tolerance is not sufficient to provide coverage in the face of correlated failures, and therefore is necessary but not sufficient to provide resilience. Thus, fault tolerance can be considered a subset of survivability, which considers multiple correlated failures, as described in the next section.

It is important to note that the optical networking community uses the term survivability to mean link- and node-level fault tolerance. Techniques such as SONET/SDH automatic protection switching [22] and *p*-cycles [23] are fault tolerance techniques applied to a network graph. Note that shared link risk groups (SLRGs) [24] provide topological diversity, but not necessarily geographic diversity.

2.2.2. Survivability

The emergence of and dependence on the Internet lead to the realisation that new techniques were needed for *unbounded networks* that could be affected by correlated failures for which fault-tolerant design techniques are not sufficient. Survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of threats such as attacks or large-scale natural disasters. This definition captures the aspect of correlated failures due to an attack by an intelligent adversary [25,26], as well as failures of large parts of the network infrastructure [27,28].

In addition to the redundancy required by fault tolerance, survivability requires *diversity* so that the same fate is unlikely to be shared by parts of the system undergoing correlated failures.

While survivability is significantly more difficult to quantify than fault tolerance, it has been formalised as a set-theoretic and state-machine based formulation [29]:

$$\text{Survivability} = \{S, E, D, V, T, P\}$$

where S is the set of acceptable service specifications, E describes the ways in which the system can degrade based on external challenges, D are the practical values of E, V is the relative ordering of service values $S \times D$, $T \subseteq S \times S \times D$ is the set of valid transitions between service states S given a challenge D, and P are the service probabilities that some $s \in S$ must meet dependability requirements.

Survivability has also been quantified using multi-dimensional Markov chains to consider simultaneous failures [30], in which one dimension captures the failure of components and the other dimension the performability.

2.2.3. Disruption tolerance

Another major type of challenge that is unique to communication networks comes from challenges in the communication environment that make it difficult to maintain stable end-to-end connections between users. *Disruption tolerance* is the ability of a system to tolerate disruptions in connectivity among its components, consisting of the environmental challenges: weak and episodic channel connectivity, mobility, unpredictably-long delay, as well as tolerance of energy (or power) challenges.

There are three major contributors to the field of disruption tolerance. The first is motivated by dynamic network behaviour and began with wireless packet radio networks [31]. This led to further research in mobile ad hoc networks (MANETs) that have proposed forwarding and routing mechanisms for dynamic networks in which the connectivity among members is continually changing [32,33]. The second contributor was motivated by very large delays that traditional network protocols can not tolerate, specifically the satellite and space environment [34] and the Interplanetary Internet (IPN) [35]. This research led to more general notions of delay-tolerant networking [36] and disruption-tolerant networking in which stable end-to-end paths may never exist. Techniques that support such networks include communicating as far as possible but reverting to store-and-forward when necessary, and mobile nodes carrying information, called *store-and-haul* [26], *store-carry-forward* [37], or *ferrying* [38]. The third contributor is energy-constrained networks, exemplified by wireless sensor networks [39], in which nodes that have drained their battery can no longer contribute to network connectivity [40,41].

More recently, disruption-tolerant networking techniques have found application in a number of domain-specific scenarios, including vehicle ad hoc networks (VANETs) [42–44], weather disruption-tolerant networks [45], and highly-dynamic airborne networks [46].

2.2.4. Traffic tolerance

The last major challenge category is that caused by the injection of traffic into the network. *Traffic tolerance* is the ability of a system to tolerate unpredictable offered load without a significant drop in carried load (including congestion collapse), as well as to isolate the effects from cross traffic, other flows, and other nodes. In defining traffic as a challenge, we mean traffic beyond the design parameters of the network in its normal operation (Section 4). Traffic challenges can either be unexpected but legitimate such as from a flash crowd [47], or malicious such as from a distributed denial-of-service (DDoS) attack [48]. It is important to note that while DDoS detection is an important endeavour, network resources are impacted regardless of whether traffic is malicious or not. Furthermore, a sufficiently sophisticated DDoS attack is indistinguishable from normal traffic, and thus traffic tolerance mechanisms are important whether or not attack detection mechanisms are successful.

2.3. Disciplines relating to trustworthiness

Trustworthiness is defined as the assurance that a system will perform as expected [49], which must be with respect to measurable properties. The trustworthiness disciplines therefore measure service delivery of a network, and consist of (1) dependability, (2) security, and (3) performability.

2.3.1. Dependability

Dependability is the discipline that quantifies the reliance that can be placed on the service delivered by a system [50,9], and consists of two major aspects: availability and reliability. Important to both of these aspects are the expected values of the failure and repair density functions. The basic measures of dependability are the MTTF (mean time to failure), which is the expected value of the failure density function, and the MTTR, which is the expected value of the repair density function. The mean time between failure is the sum of these two [51]:

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

Availability is *readiness* for usage, which is the probability that a system or service will be operable when needed, and is calculated as

$$A = \text{MTTF}/\text{MTBF}$$

Reliability is *continuity of service*, that is the probability that a system or service remains operable for a specified period of time:

$$R(t) = \Pr[\text{no failure in}[0, t]] = 1 - Q(t)$$

where $Q(t)$ is the failure cumulative distribution function.

These notions of dependable systems have been codified by IFIP WG 10.4 [14] and ANSI T1A1 [52] and are commonly applied to network dependability. These notions of reliability are also applied to fibre-optic links as a measure of fault tolerance [53,54].

The relative importance of availability and reliability depend on the application service. Availability is of primary importance for transactional services such as HTTP-based Web browsing: as long as the server is usually up, it matters less if it fails frequently as long as the MTTR is very short. On the other hand, reliability is of prime importance for session- and connection-oriented services such as teleconferencing: to be useful, the session must remain up for a specified period of time requiring a long MTTF.

Additionally, there are several other aspects of dependability [9]: *Maintainability* is the aptitude to undergo repairs and evolutions. *Safety* is dependability with respect to catastrophic failures [55]. Safety is of particular concern for critical infrastructures such as the power grid and nuclear power plants, and their interdependence on the Internet and SCADA (supervisory control and data acquisition) networks. *Integrity* is an aspect of dependability that is more commonly associated with security, which is described next.

2.3.2. Security

Security is the property of a system, and the measures taken such that it protects itself from unauthorised access or change, subject to policy [56]. Security properties include AAA (*authenticity*, *authorisability*, *auditability*), *confidentiality*, and *nonrepudiability*. Security shares with dependability the properties of *availability* and *integrity* [14]. In the context of trustworthiness, we are concerned with the measurable properties of the security aspects [57,58]. We can also consider security to be related to a level of self-protection [59], which is an enabling principle for resilience.

2.3.3. Performability

Performability [60] is the property of a system such that it delivers performance required by the service specification, as described by QoS (quality of service) measures such as delay, throughout or goodput, and packet delivery ratio [61–66]. Performability extends the binary dependability concepts to the *range* of degradable systems.

2.4. Robustness and complexity

Two disciplines lie outside challenge tolerance and trustworthiness, but describe their relationship to one another (robustness) and overall characteristics (complexity).

2.4.1. Robustness

Robustness is a control-theoretic property that relates the operation of a system to perturbations of its inputs [67,68]. In the context of resilience, robustness describes the trustworthiness (quantifiable behaviour) of a system in the face of challenges that change its behaviour. Note that the term robustness is frequently used in a much less precise manner that is synonymous with resilience, survivability, or security.

2.4.2. Complexity

Complexity refers to the ways in which large numbers of systems interact, resulting in emergent behaviour [69]. Complexity science has an important relationship to resilience and the robustness of systems, because resilience mechanisms such as self-organisation and autonomic behaviour increase complexity, and increased complexity may result in greater network vulnerability.

3. Challenges and past failures

This section describes the challenges to the Global Internet and interdependent networks such as the PSTN that motivate the need for resilience. Interwoven among challenge types is a selected set of past events and failures that have seriously affected these networks. Challenges are any characteristic or condition that impacts the normal operation of the network, consisting of unintentional mis-configuration or operational mistakes, large-scale natural or human-caused disasters, malicious attacks from intelligent adversaries, environmental challenges (mobility, weak channels, unpredictably long delay, constrained energy), unusual but legitimate traffic load, and service failures at a lower level.

3.1. Unusual but legitimate traffic load

A non-malicious request for service that places a greater (or different along some other dimension) than the normal operation has been engineered to cope with, is a challenge to the network. This is commonly caused by a *flash crowd* [47] when an event triggers a large volume of service requests beyond the normal load. In addition to affecting the target of the flash crowd, the network as a whole can be affected, particularly by cross traffic near the target [70]. A secondary effect of many of the challenges listed in the following subsections is a flash crowd due to emergency response and the population trying to obtain news about what has happened.

3.2. Accidents and human mistakes

Accidents and mistakes are non-malicious, made by people who interact with the system, such as unintentional device mis-configuration or not following correct policy. These may occur during system design or operation and can become more pernicious if the parties involved try to cover up their mistakes. Sometimes accidents and mistakes can have very significant consequences, as described in the following paragraphs.

A human error caused Google to flag every search result with the message "This site may harm your computer". Usually only sites that are suspected of installing malicious software are kept in the database. On 01 February 2009 the '/' string was mistakenly inserted into this list, which expanded to all URLs and caused the erroneous flagging [71].

A large-scale blackout affected much of the Northeastern United States and the province of Ontario on 14 August 2003, affecting 50 million people. Many interrelated factors came into play in turning a few operational problems into a major failure. Three power-plant outages on a hot summer day resulted in insufficient reactive reserves being available. At about the same time, automatic alarm software processes were left turned off due to human error. The result was that when three high-voltage power-lines went down due to insufficient tree-trimming practices the load was not properly rerouted and instead caused a large portion of the power grid (15 more high-voltage lines) in northern Ohio to collapse. This caused surges and cascading failures in neighbouring connected grids, spreading faster than automatic protection relays were able to trip. Eventually enough failsafes and line-outages occurred, thus isolating the less robust portions of the grid and stopping the cascading failures, at the same time

partitioning the grid into various islands. Full service was not restored in some areas for over a week. The costs, both in terms of repairs and lost productivity, were on the order of 10 billion dollars [72]. This large-scale power failure had a significant impact on the interrelated Internet infrastructure, when over 2000 globally advertised prefixes had severe outages of 2 hours or longer, affecting nearly 50% of all Internet autonomous systems [73]. This is the canonical example of infrastructure interdependence with the internet relying on the power grid for equipment to stay operational, while at the same time many SCADA power-control systems are communicating using Internet-based services. A lack of understanding of the complexity of the grid as a whole, as well as a number of human mistakes both in planning and in remediation decisions contributed to the extent and severity of this blackout.

On 8 May 1988 a fire at the Illinois Bell switching office in Hinsdale caused severe damage to phone services in Northern Illinois in the US. The outage affected local, long-distance, mobile telephone, 800 service, and air-traffic control communication between Midway and O'Hare Airports in Chicago and the FAA Center in Aurora, Illinois. It took until the end of May to restore service. Although the PTSN (public switched telephone network) contained hardware and link redundancy, services failed because both the primary and the backup system were located in the same building, and were both destroyed in the fire resulting from a lightning strike. The Hinsdale fire is a canonical example of how fault tolerance alone is not sufficient for resilience. A significant fraction of PSTN failures are due to accidents and human mistakes [74].

On 18 July 2001 a train derailed in the Howard Street Tunnel in Baltimore, Maryland in the Northeast US. The subsequent fire caused fibre backbone disruptions experienced by seven major ISPs [75,76]. This tunnel was a convenient place to route fibre conduits under the city of Baltimore. Most traffic was rerouted, but resulted in congestion on alternative links, causing a noticeable slowdown for a significant portion of the Internet. New fibre strands were laid to restore physical capacity within 36 hours. In the case of both the Baltimore tunnel and Hinsdale fires, design choices had been made that resulted in redundant systems being deployed; however without geographic diversity the redundant infrastructure shared the same fate.

While Pakistan Telecom was attempting to comply with a Government mandate to block a particular YouTube video on 24 February 2008, they advertised their own AS as the shortest path to a portion of YouTube's IP-address space. This advertisement went out not only to downstream providers within Pakistan, but also to their upstream provider, Pacific Century CyberWorks (PCCW). At that time PCCW was not filtering for bogus prefix advertisements such as this, and it propagated the advertisement to the rest of the world, causing most HTTP requests for YouTube world-wide to be directed to Pakistan Telecom. Over the next couple of hours, several attempts were made by YouTube to compete with the bogus route advertisements using more specific prefixes, but the situation did not return to normal until PCCW disconnected Pakistan Telecom entirely. While the global scope of this hijacking was most-likely accidental, it clearly demonstrates the vulnerability of BGP to route-spoofing and still presents a major challenge to resilient Internet operation [77–79].

In 2009 a small Czech provider (AS number 47868) using a MikroTik router caused severe problems to the BGP infrastructure [80,81]. An administrator mis-configured the BGP settings, which triggered a bug in Cisco's BGP implementation. The effect of this problem was observed world-wide. The administrator's intention was to prepend his own AS number once in order to increase the length of the BGP path via this AS. This is a common technique that operators use in order to avoid attracting traffic from peer providers. Unfortunately the administrator assumed that the configuration syntax followed Cisco's IOS syntax, used in the dominant router platform. The syntax of a MikroTik router does not take the AS number that should be prepended as an argument, but rather as an integer indicating how often the AS number should be prepended, which led to $47868 \bmod 256 = 252$ times prepending the AS number. Such very long paths should be filtered by the receiving routers but actually only few routers were configured in this way. A Cisco router receiving an update message which contains a route with 255 ASes would reset itself while processing this message due to an interoperability bug in IOS. This combination of mis-configuration of a rarely used router, in conjunction with poorly configured routers accepting excessively long AS paths and the implementation bug in Cisco's IOS, led to a ten-fold increase in global routing instability for about an hour.

3.3. Large-scale disasters

Large-scale disasters may result from either natural causes or from human mistakes. In either case they are a unique category of challenges because they result in correlated failure over a large area, such as destroying hardware while simultaneously preventing operators from performing normal functions and restricting information access by decision-makers, resulting in poor remediation choices. Examples of large-scale disasters include hurricanes, earthquakes, ice storms, tsunamis, floods, and widespread power outages.

On 29 August 2005, Hurricane Katrina caused massive destruction in Louisiana and Mississippi in the Southeast US, and disrupted communications with 134 networks [82,83]. Many of these disruptions were due to power outages, and the majority of them were restored within a ten day period. One link of the Abilene Internet2 research network was also taken down, but sufficient capacity was available in alternate paths to reroute traffic. There was also significant disruption to the PSTN due to the increased traffic and destruction of cellular-telephony towers. The disaster-recovery efforts which occurred following the hurricane also highlighted the challenges resulting from incompatible communication equipment used by different first responders (local police, fire, state police, coast guard, national guard, etc.).

On 26 December 2006, and continuing for two days, an earthquake with a number of major aftershocks took place

near Hengchun Taiwan that damaged the submarine cables that provide Internet connectivity between Asia and North America [84]. 1200 prefix ranges became temporarily unreachable, China's internet access capacity was reduced by 74%, and Hong Kong's Internet access was completely disabled. While BGP was able to automatically reroute some of the traffic, it did so without any knowledge of the underlying physical topology or link utilisation, resulting in traffic between China and Taiwan crossing the Pacific Ocean twice. Manual traffic engineering was required to reroute traffic via Korea and Japan, instead of via the US.

Other events that could cause catastrophic power grid, radio-communication, and communication network failures over a large area include radiation from solar-induced geomagnetic storms [85] (with a predicted peak in 2012) and attacks from electromagnetic pulse (EMP) weapons [86]. Finally, a serious pandemic could have severe impact if people are unable or afraid to operate and maintain critical infrastructure including the Global Internet [87,88]; this was a concern that fortunately did not come to pass in the 2009–2010 H1N1 influenza pandemic, but may be inevitable in a future avian or swine flu pandemic.

3.4. Malicious attacks

Malicious attacks come from cyber-criminals, for reasons including terrorism, information warfare among nations, political groups, or competing businesses, as well as from recreational crackers including script kiddies. These challenges may destroy or damage critical components in the network infrastructure with the intent of disrupting network services, or disable network infrastructure as collateral damage.

The 9/11 terrorist attacks of 11 September 2001 to New York were relatively localised and not targetted against the network infrastructure per se, but many subscriber lines were affected, because 1–2% of all Internet prefix blocks were unreachable at the peak of the disruption [89]. Most of these were restored within a day or two, and the impact to core Internet infrastructure services such as DNS and BGP was minimal. The more noticeable effects were due to flash crowds for news-related Web sites, which were quickly overloaded, seeing in some cases 350% of their normal traffic load. DNS, however, saw only normal load due to caching at end systems, and the Internet itself saw lower than normal aggregate amounts of traffic, presumably because many people were preoccupied watching the televised coverage and using the Web less than usual. The 7/7 terrorist attacks against London Transport on 7 July 2005 did not directly damage network infrastructure other than that used by the London Underground, but also induced significant traffic on the mobile telephone network and Internet as people tried to get news, and impaired first responder ability to communicate with one another [90].

Malicious attacks that exploit protocols or software vulnerabilities include distributed denial-of-service (DDoS) campaigns that are frequently intended to harm an individual, organisation, corporation, or nation. When the Estonian government decided to move a Soviet war memorial, a DDoS attack was launched from IP-addresses within Russia. The politically-motivated attacks targeted a variety of Estonian government and business Web sites. Estonia severed its Internet connection to the rest of the world to stop the effects of attacks [91].

3.5. Environmental challenges

Challenges to the communication environment include 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. These challenges are addressed by disruption-tolerant networks (DTNs), as described in Section 2.2.3.

3.6. Failures at a lower layer

If any of these challenges causes a service failure at a particular layer, that failure becomes a challenge to any higher level service which depends on the correct behaviour of the layer that fails. This class of failures may induce recursive failing of services until the error can be contained within a higher-level service. Therefore, symptoms of a challenge can often be seen by multiple services. For example, a fibre cut causes the physical lightpath service to fail, resulting in the failure of all link-layer connections that rely on that lightpath; however, remediation may occur at a higher layer by re-routing traffic across an alternative fibre.

In January 2008, cable cuts in the Mediterranean Sea caused substantial disruptions in connectivity to Egypt and India, with lesser disruptions to Afghanistan, Bahrain, Bangladesh, Kuwait, Maldives, Pakistan, Qatar, Saudi Arabia, and the United Arab Emirates. More than 20 million Internet users were affected. The cause of the cuts is assumed to be unintentional, such as a boat anchor or natural wear and abrasion of the submarine cable against rocks on the sea floor [92–94].

Submarine cable cuts are a common occurrence, with an average of one cut occurring every three days worldwide. Most of these cuts occur as uncorrelated random events and go unnoticed by end users due to redundancy in the infrastructure. However in cases such as the Taiwan earthquake, and the Mediterranean cable cuts (as well as the Baltimore tunnel fire), multiple correlated link failures caused major outages, emphasising that redundancy for fault tolerance is not sufficient for resilience; geographic diversity for survivability is also needed.

3.7. Summary of challenges and past failures

While the Global Internet *as a whole* has proven to be relatively resilient to challenges discussed in this section due to its scope and geographic diversity, there is reason for concern. Some of these challenges have had a significant regional impact, and analyses of vulnerabilities indicate that a coordinated attack on the Internet infrastructure such as DNS root servers and IXPs (Internet exchange points such as MAE East, West, and Central) could have severe consequences.

4. ResiliNets framework and strategy

This section describes frameworks and strategies for network resilience. First, several important previous frameworks are reviewed that consider dependability, survivability, and performability. Then, the comprehensive ResiliNets framework and strategy is presented, which draws heavily on these frameworks, as well as past work in the disciplines presented in Section 2.

4.1. Previous strategies

There have been several systematic resilience strategies, presented in the following subsections: ANSA, T1, CMU-CERT, and SUMOWIN.

4.1.1. ANSA

The Advanced Networked Systems Architecture (ANSA) project [95] covered a number of aspects of large system design, including dependability. The dependability management strategy consists of eight stages (based on [96]): fault confinement, fault detection (properly error/failure detection), fault diagnosis, reconfiguration, recovery, restart, repair, and reintegration. The ANSA framework defines *expectation regions* in a two-dimensional *value* × *time space* to describe acceptable service, and thus considers performability. Service failures are a mismatch between an occurrence in this space and the expectation regions.

4.1.2. T1

T1A1.2 Working Group of Alliance for Telecommunications Industry Solutions (ATIS) on network survivability performance has developed a multilevel framework for network survivability [97] with four layers: *physical* consisting of infrastructure with geographic diversity for survivability; *system* consisting of nodes and links with protection switching for survivability; *logical* consisting of capacity on the system layer; and *service* consisting of voice and data circuits with dynamic routing and reconfiguration for survivability. This framework quantifies service outages as a (*U,D,E*) triple, in which *U* is the *unservability* (an inverse dependability metric such as unavailability or failure), *D* is the *duration* in time, and *E* is the *extent* over various parameters including geographic area, population, and services. Severity categories of this triplet mapped into three-dimensional space are categorised as *minor*, *major*, or *catastrophic*.

4.1.3. CMU-CERT

The CERT Coordination center at CMU proposed a four-step strategy [25] consisting of the "three R's": (1) resistance (traditional security, diversity, redundancy, specialization, trust validation, and observed stochastic properties); (2) recognition (analytical redundancy and testing, intrusion monitoring, system behaviour, integrity monitoring); (3) recovery (redundancy, diverse location of information resources, contingency planning and response teams); followed by (4) adaptation and evolution.

4.1.4. SUMOWIN

The Survivable Mobile Wireless Networking (SUMOWIN) project [26] explored mechanisms and strategies for disruption tolerance (before the term was generally adopted) in environments where stable end-to-end connectivity was not achievable. The strategy consists of three components: (1) Maintain connectivity when possible using techniques such as adaptive transmission power control and highly-dynamic MANET techniques (*eventual stability*); (2) Use new forwarding and routing techniques that do not require routing convergence to move data towards the destination, using store-and-forward techniques such as store-and-forward buffering and *store-and-haul* (store-carry-forward or ferrying) towards the destination when stable end-to-end paths cannot be maintained (*eventual connectivity*); (3) Use innovate technologies such as satellite networks and active (adaptable programmable) networking to establish connectivity and maintain stealth.

4.2. ResiliNets

The ResiliNets initiative [98] has developed a framework for resilient networking [99], initially as part of the Autonomic Network Architecture (ANA) [100,101] and Postmodern Internet Architecture (PoMo) [102,103] projects, serving as the basis of the ResumeNet (Resilience and Survivability for Future Networking: Framework, Mechanisms, and Experimental Evaluation) project [104,105]. This initiative was heavily influenced by the frameworks described above, and can be viewed as a successor and synthesis of all of them. The ResiliNets framework is described by a set of axioms and a strategy in the rest of this section, and by a set of design principles described in Section 5. As appropriate, cross references will be given by axiom (**A*n***), strategy (**S*n***), or principle (**P*n***) number.

4.3. ResiliNets axioms

Axioms provide the basis for any systematic framework; we present four basic self-evident tenets that form the basis for the ResiliNets strategy.

A0. ***Faults are inevitable***; *it is not possible to construct perfect systems, nor is it possible to prevent challenges and threats.*

It is not possible to construct fault-free systems, for two reasons: First, internal faults are those that arise from within a given system due to imperfect designs, and while it is theoretically possible to use formal methods to design a provably correct system, this remains impractical for large complex systems and networks for the foreseeable future. Second, external faults are exercised by challenges from outside the system, and it is neither possible nor practical to predict all such challenges (present and future) and design defences against them. Threat and challenge models improve the ability to prevent external faults, but do not eliminate them.

A1. ***Understanding normal operation is necessary***, *including the environment, and application demands. It is only by understanding normal operation that we have*

any hope of determining when the network is challenged or threatened.

We define *normal operation* to be the state of the network when there are no adverse conditions present. This loosely corresponds to the conditions for which the current Internet and PSTN are designed, when the network is not under attack, the vast majority of network infrastructure is operational, and connectivity is relatively strong. As an example, the public switched telephone network (PSTN) is designed to handle normal time-of-day fluctuations of traffic, and even peak loads such as Mother's day. These predictable application demands are within normal operation. On the other hand, PSTN call load during a disaster such as 9/11 or Hurricane Katrina, as well as flash crowds to an obscure Web site represent traffic that is beyond normal operation. It is essential to understand normal operation to be able to *detect* when an adverse event or condition occurs (**S2**).

A2. ***Expectation and preparation for adverse events and conditions is necessary***, *so that defences and detection of challenges that disrupt normal operations can occur. These challenges are inevitable.*

We define an *adverse event or ongoing condition* as challenging (Section 3) the normal operation of the network. We can further classify adverse events and conditions by severity as mild, moderate, or severe, and categorise them into two types:

(1) Anticipated adverse events and conditions are ones that we can predict based either on past events (such as natural disasters), and attacks (e.g. viruses, worms, DDoS) or that a reasoned threat analysis would predict might occur.
(2) Unanticipated adverse events and conditions are those that we cannot predict with any specificity, but for which we can still be prepared in a general sense. For example, there will be new classes of attacks for which we should be prepared.

It is necessary to expect adverse events and conditions in order to design resilient networks, and thus this axiom motivates the *defend* and *detect* aspects of the resilience strategy (**S1**, **S2**).

A3. ***Response to adverse events and conditions is required for resilience***, *by remediation ensuring correct operation and graceful degradation, restoration to normal operation, diagnosis of root cause faults, and refinement of future responses.*

While it is necessary to expect adverse events and conditions, it is just as important to take action when challenges do occur. This motivates the *remediation* aspect of the resilience strategy (**S3**).

4.4. ResiliNets strategy

The resilience axioms motivate a strategy for resilience, based in part on the previous strategies (Section 4.1), developed as part of the ResiliNets [98], ANA [100], and ResumeNet [104] projects.

4.4.1. Introduction

We begin with a motivating example and analogy of a mediaeval castle that is designed to be resilient in structure and operations. Castles generally contain both inner and outer thick walls, perhaps additionally protected by a moat or by being located on a mountain; this is a structural or *passive defence*. Additionally, guards are patrolling the perimeters and checking the credentials of visitors, an *active defence*. While the defences are intended to resist attack, they may be penetrated. An advancing army might be successful in launching a trebuchet projectile that blasts a hole in the castle wall. The guards *detect* this adverse event, and respond by sending subjects to the inner wall, and repel the advancing forces, perhaps by pouring boiling oil over the hole to prevent the enemy from entering; these are *remediation* actions. Once the enemy forces have been repelled, the outer wall must be repaired, *recovering* to the initial pre-attack state.

It is also important to analyse what went wrong. A *diagnosis* of why the projectile was able to penetrate the wall may indicate a design flaw in the wall materials or thickness. Furthermore, an analysis of the entire battle may *refine* a number of aspects of the entire process, indicating ways to improve detection (for example establishing remote outposts to observe enemy movement) and remediation (improving fighting techniques).

This example not only motivates the ResiliNets strategy presented next, but also indicates that many of these ideas are very old, at least in a general context predating their application to networks by several centuries.

We formalise this as a two-phase strategy that we call $D^2R^2 + DR$, as shown in Fig. 3. At the core are passive structural defences. The first active phase, D^2R^2: *defend*, *detect*, *remediate*, *recover*, is the inner control loop and describes a set of activities that are undertaken in order for a system to rapidly adapt to challenges and attacks and maintain an acceptable level of service. The second active phase DR: *diagnose*, *refine*, is the outer loop that enables longer-term evolution of the system in order to enhance the approaches to the activities of phase one. The following sections describe the steps in this strategy. As appropriate, principles in Section 5 will be referenced as **P*n*** that are motivated by the strategy. All of these strategy steps require that the proper design tradeoffs be made (**P6, P7, P8**).

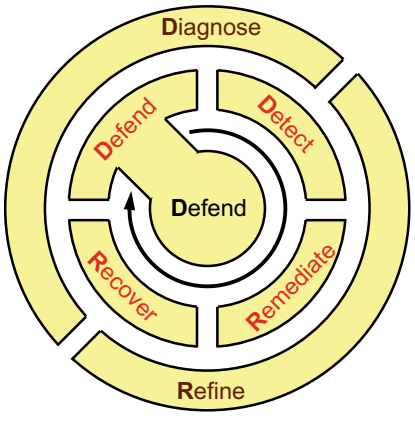


Fig. 3. ResiliNets strategy.

4.4.2. D^2R^2 inner loop

The first strategy phase consists of a passive core and a cycle of four steps that are performed in real time and are directly involved in network operation and service provision. In fact, there is not just one of these cycles, but many operating simultaneously throughout the network for each resilient system, triggered whenever an adverse event or condition is detected.

S1. ***Defend*** *against challenges and threats to normal operation.*

The basis for a resilient network is a set of defences that reduce the probability of a fault leading to a failure (fault tolerance) and reduce the impact of an adverse event on network service delivery. These defences are identified by developing and analysing threat models (**P3**), and consist of a passive and active component.

Passive defences are primarily structural, suggesting the use of trust boundaries (**P5**), redundancy (**P11**) and diversity (**P12**). The main network techniques are to provide geographically diverse redundant paths and alternative technologies such as simultaneous wired and wireless links, so that a challenge to part of the network permits communication to be routed around the failure [106–108].

Active defences consist of self-protection (**P9**) mechanisms operating in the network that defend against challenges, such as firewalls that filter traffic for anomalies and known attack signatures [109], and the eventual connectivity paradigm (**P10**) that permits communication to occur even when stable end-to-end paths cannot be maintained. Clearly, defences will not always prevent challenges from penetrating the network, which leads to the next strategy step: detect.

S2. ***Detect*** *when an adverse event or condition has occurred.*

The second step is for the network, as a distributed system as well as individual components such as routers, to detect challenges and to understand when the defence mechanisms have failed. There are three main ways to determine if the network is challenged. The first of these involves understanding the service requirements (**P1**) and normal operational behaviour (**A1, P2**) of a system and detecting deviations from it – *anomaly detection* based on metrics (**P4**) [110]. The second approach involves detecting when errors occur in a system, for example, by calculating cyclic-redundancy checks (CRCs) to determine the existence of bit errors that could lead to a service failure. Finally, a system should detect service failures; an essential facet of this is an understanding of service requirements (**P1**). An important aspect of detecting a challenge is determining its nature, which requires context awareness (**P14**). For example, in an environmental monitoring system a sensor may be reporting anomalous readings; this could be as a result of the observed environment (e.g., there is a flood event) or device failure. Detection along with this understanding will inform an appropriate remediation strategy [111].

S3. ***Remediate*** *the effects of the adverse event or condition.*

The next step is to remediate the effects of the detected adverse event or condition to minimise the effect on service delivery. The goal is to do the best possible at all levels after an adverse event and during an adverse condition. This requires adaptation (**P17**) and autonomic behaviour (**P16**) so that corrective action can be taken at all levels (**P13, P15**) without direct human intervention, to minimise the impact of service failure, including correct operation with graceful degradation of performance.

A common example of remediation is for dynamic routing protocols to reroute around failures [112,32,45,113,114] and for adaptive applications and congestion control algorithms to degrade gracefully from acceptable to impaired service (Section 6). There may be a number of strategies that can be used to remediate against a given challenge; a key problem is determining the most appropriate one to take [115,116].

S4. ***Recover*** *to original and normal operations.*

Once the challenge is over after an adverse event or the end of an adverse condition, the network may remain in a degraded state (Section 6). When the end of a challenge has been detected (e.g., a storm has passed, which restores wireless connectivity), the system must recover to its original optimal normal operation (**P2**), since the network is likely not to be in an ideal state, and continued remediation activities may incur an additional resource cost. However, this may not be straightforward. For example, it may not be clear when to revoke a remediation mechanism that is attributed to a particular challenge, as it may be addressing another problem.

4.4.3. DR outer loop

The second phase consists of two background operations that observe and modify the behaviour of the D^2R^2 cycle: diagnosis of faults and refinement of future behaviour. While currently these activities generally have a significant human involvement, a future goal is for autonomic systems to automate diagnosis and refinement (**P16**).

S5. ***Diagnose*** *the fault that was the root cause.*

While it is not possible to directly *detect* faults (Section 2.1), we may be able *diagnose* the fault that caused an observable error. In some cases this may be automated, but more generally it is an offline process of root-cause analysis [10]. The goal is to either remove the fault (generally a design flaw as opposed to an intentional design compromise) or add redundancy for fault tolerance so that service failures are avoided in the future. An example of network-based fault diagnosis is the analysis of packet traces to determine a protocol vulnerability that can then be fixed.

S6. ***Refine*** *behaviour for the future based on past D^2R^2 cycles.*

The final aspect of the strategy is to refine behaviour for the future based on past D^2R^2 cycles. The goal is to learn and reflect on how the system has defended, detected, remediated, and recovered so that all of these can be improved to continuously increase the resilience of the network.

This is an ongoing process that requires that the network infrastructure, protocols, and resilience mechanisms be evolvable (**P17**). This is a significant challenge given the current Internet hourglass waist [117] of IPv4 BGP, and DNS, as well as other mechanisms (e.g. NAT) and protocol architectures (e.g. TCP and HTTP) that are entrenched and resist innovation.

5. ResiliNets design principles

The past experience of the resilience disciplines, basis of the axioms, and synthesis of the $\mathbf{D^2R^2 + DR}$ strategy leads to a set of principles for the design of resilient networks and systems. There is a careful balance to be struck in any system of design principles: a large enough number to provide specific guidance in the architecture and design of resilient networks, but a small enough number to be manageable without being overwhelming. The resilience principles are shown in Fig. 4, clustered in the major categories prerequisites, tradeoffs, enablers, and behaviour.

5.1. Prerequisites

Five principles span the domain of prerequisites necessary to build a resilient system.

P1. Service requirements of applications need to be determined to understand the level of resilience the system should provide. In this sense, resilience may be regarded as an additional QoS property along with conventional properties such as performance.

P2. Normal behaviour of the network is a combination of design and engineering specification, along with monitoring while unchallenged to learn the network's normal operational parameters [118,119]. This is a fundamental requirement (**A1**) for detecting (**S2**) challenges [120,121].

P3. Threat and challenge models [122,123] are essential to understanding and detecting potential adverse events and conditions. It is not possible to understand, define, and implement mechanisms for resilience that defend against, detect, and remediate (**S1–3**) challenges without such a model.

P4. Metrics quantifying the service requirements and operational state are needed to measure the operational state (in the range normal ↔ partially-degraded ↔ severely-degraded) and service state (in the range acceptable ↔ impaired ↔ unacceptable) to detect and remediate (**S1–2**) and quantify resilience to refine future behaviour (**S6**). This is discussed further in Section 6.

P5. Heterogeneity in mechanism, trust, and policy are the realities of the current world. No single technology is appropriate for all scenarios, and choices change as time progresses. Therefore it is increasingly unrealistic to consider the set of global networks as a homogeneous internetwork. The Global Internet is a collection of realms [124–126] of disparate technologies [102]. Furthermore, realms are defined by trust and policy, across which there is *tussle* [127]. Resilience mechanisms must deal with heterogeneous link technologies, addressing, forwarding, routing, signalling, traffic, and resource management mechanisms. Resilience mechanisms must also explicitly admit trust and policy tussles. These realms can also serve to enhance resilience by giving self-protection (**P9**) boundaries in which to operate.

5.2. Design tradeoffs

Three principles describe fundamental tradeoffs that must be made while developing a resilient system.

P6. Resource tradeoffs determine the deployment of resilience mechanisms. The relative composition and placement of these resources must be balanced to optimise resilience and cost. The maximum availability of a particular resource serves as a constraint in these optimizations. Resources to be traded against one another include bandwidth, memory [128], processing, latency [129], energy, and monetary cost. Of particular note is that maximum resilience can be obtained with unlimited cost, but there are cost constraints that limit the use of enablers such as redundancy and diversity (**P11–P12**).

P7. Complexity of the network results due to the interaction of systems at multiple levels of hardware and software, and is related to scalability. While many of the resilience principles and mechanisms increase this complexity, complexity itself makes systems difficult to understand and manage, and thereby threatens resilience. The degree of complexity [130–132] must be carefully balanced in terms of cost vs. benefit, and unnecessary complexity should be eliminated.

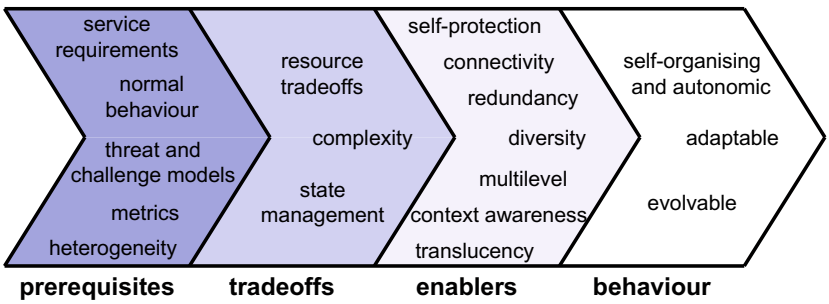


Fig. 4. Resilience principles.

P8. State management is an essential part of any large complex system. It is related to resilience in two ways: First, the choice of state management impacts the resilience of the network. Second, resilience mechanisms themselves require state and it is important that they achieve their goal in increasing overall resilience by the way in which they manage state, requiring a trade-off among the design choices. Resilience tends to favour soft, distributed, inconsistency-tolerant state rather than hard, centralised, consistent state, but careful choices must be made in every case.

5.3. Enablers

Seven principles are enablers of resilience that guide network design and engineering.

P9. Self-protection and security are essential properties of entities to defend against challenges (**A2**) in a resilient network. Self-protection is implemented by a number of mechanisms, including but not limited to mutual suspicion, the AAA mechanisms of authentication, authorisation, and accounting, as well as the additional conventional security mechanisms of confidentiality, integrity, and nonrepudiation.

P10. Connectivity and association among communicating entities should be maintained when possible based on eventual stability, but information flow should still take place even when a stable end-to-end path does not exist based on the eventual connectivity model [133,26]; the use of disruption-tolerant networking (DTN) techniques such as partial paths [134], store-and-forward with custody transfer [35,36], and store-and-haul [26,37,38] permit this.

P11. Redundancy in space, time, and information increases resilience against faults and some challenges if defences (**S1**) are penetrated. Redundancy refers to the replication of entities in the network, generally to provide fault tolerance. In the case that a fault is activated and results in an error, redundant components are able to operate and prevent a service failure. Spatial redundancy examples are triple-modular redundant hardware [19] and parallel links and network paths. Examples of temporal redundancy are erasure coding [135] consisting of repeated transmission of packets, periodic state synchronisation, and periodic information transfer (e.g. digital fountain [136]). Information redundancy is the transmission or storage of redundant information, such as forward error correction (FEC). It is important to note that redundancy does not inherently prevent the redundant components from sharing the same fate.

P12. Diversity is closely related to redundancy, but has the key goal to avoid fate sharing. Diversity in space, time, medium, and mechanism increases resilience against challenges to particular choices [137–139,24]. Diversity consists of providing alternatives so that even when challenges impact particular alternatives, other alternatives prevent degradation from normal operations. Diverse alternatives can either be simultaneously operational, in which case they defend (**S1**) against challenges [140], or they may be available for use as needed to remediate (**S3**) [141]. *Spatial diversity* requires redundancy of a degree at least equal to the degree of diversity, and can be categorised as *topological diversity* across the (logical) topology of the network [142,24], and *geographic diversity* across the physical topology of the network [45,106]; note that topologically diverse links or nodes may be physically co-located. *Temporal diversity* is intentional variance in the temporal behaviour of a component or protocol, such as variation in timing of protocol state transitions to resist traffic analysis. *Operational diversity* refers to alternatives in the architecture and implementation of network components and protocols, that is the avoidance of monocultures, and may be categorised as: *implementation diversity* that prohibits systems from exhibiting the same error caused by an implementation fault (e.g. *N*-version programming [20]), by deploying systems software from multiple vendors and routers from different hardware vendors; *medium diversity* that provides choices among alternative physical media through which information can flow, such as wired and wireless links; and *mechanism diversity* that consists of providing alternative mechanisms, such as FEC and ARQ-based error control.

Fig. 5 shows an example of several kinds of diversity. Communicating subscribers are multihomed to service providers that are diverse in both geography and mechanism. Protection against a fibre cut is provided by the wireless access network; protection against wireless disruptions such as weather or jamming is provided by the fibre connection.

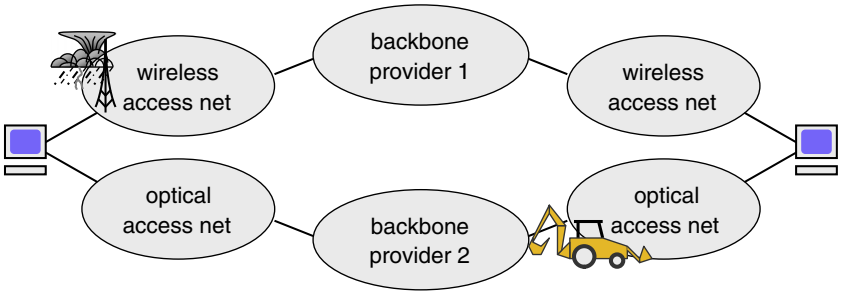


Fig. 5. Diversity example.

Table 1
Levels and selected resilience mechanisms.

Level	Mechanism
Application	Adaptive applications
Transport	Eventual connectivity, erasure codes
Internetworking	Heterogeneity, realm diversity
Path	Multipath spreading, medium diversity
Topology	*k*-connected, geographic graph diversity
Links and nodes	Link error control, fault tolerance
Physical channel	Robust coding

P13. Multilevel resilience is defined with respect to protocol layer, protocol plane, and hierarchical network organisation [97,143,144]. Multilevel resilience is needed in three orthogonal dimensions: *Protocol layers* in which resilience at each layer provides a foundation for the next layer above; *planes*: data, control, and management; and *network architecture* inside-out from fault tolerant components, through survivable subnetwork and network topologies, to the global internetwork including attached end systems.

A given level is designed to be as resilient as is practicable given cost constraints (**P6**); this provides a foundation for the next level, as shown in Table 1. At the physical layer, robust coding optimises the probability of bits arriving uncorrupted to their next hop, but this will not be perfect in hostile environments. Therefore, at the link layer, we apply error control to provide as reliable a link as practical, but this will not be perfect, and links may be cut or intermittently connected. We ensure that geographically and technologically diverse links and nodes are available (**P12**). This permits us to construct network topologies that have not only redundancy but geographic diversity so that link cuts do not partition the network. This in turn provides the foundation for the network layer to constructs paths even when links are down and part of the topology is not stable, by using techniques such as multipath and fast-reroute. A heterogeneous (**P5**) set of subnetworks form the Global Internet, over which a resilient end-to-end transport layer uses principles such as eventual connectivity (**P10**) and erasure coding to transfer data end-to-end, even when end-to-end stable paths cannot be maintained. Finally, adaptive applications should be tolerant of poor end-to-end transport.

P14. Context awareness is needed for resilient nodes to monitor the network environment (channel conditions, link state, operational state of network components, etc.) and detect adverse events or conditions [145,146,45]. Remediation (**S3**) mechanisms must take the current context of system operation into account.

P15. Translucency [147,129] is needed to control the degree of abstraction vs. the visibility between levels. Complex systems are structured into multiple levels to abstract complexity and separate concerns. In the case of networks this consists of three multilevel dimensions: layer, plane, and system organisation. While this abstraction is important, an opaque level boundary can hide too much and result in suboptimal and improper behaviour based on incorrect implicit assumptions about the adjacent level [46]. Thus it is important that level boundaries be translucent in which cross-layer control loops allow selected state to be explicitly visible across levels; *dials* expose state and behaviour from below; *knobs* influence behaviour from above [102].

5.4. Behaviour needed for resilience

The last group of three principles encompass the behaviours and properties a resilient system should possess.

P16. Self-organising and autonomic behaviour [148,149,101] is necessary for network resilience that is highly reactive with minimal human intervention. A resilient network must initialise and operate itself with minimal human configuration, management, and intervention. Ideally human intervention should be limited to that desired based on high-level operational policy. The phases of autonomic networking consist of: (1) initialisation consisting of *auto-configuration* of network components and their *self-organisation* into a network [150,69]; (2) steady-state normal operation consisting of *self-managing* with minimal human interaction dictated by policy and *self-optimising* to dynamic network conditions, and (3) steady-state expecting faults and challenges consisting of *self-diagnosing* [151] and *self-repair*.

P17. Adaptability to the network environment is essential for a node in a resilient network to detect, remediate, and recover from challenges. Resilient network components need to adapt their behaviour based on dynamic network conditions, in particular to remediate (**S3**) from adverse events or conditions, as well as to recover (**S4**) to normal operations. At the network level, programmable and active network techniques enable adaptability [152,153].

P17. Evolvability [154] is needed to refine (**S6**) future behaviour to improve the response to challenges, as well as for the network architecture and protocols to respond to emerging threats and application demands. Refinement of future behaviour is based on reflection on the inner strategy loop: the defence against, detection, and remediation of adverse events or conditions and recovery to normal operation (**S1–4**). Furthermore, it is essential that the system can cope with the evolution and extension of the network architecture and protocols over time, in response to long term changes in user and application service requirements, including new and emerging applications technology trends, as resource tradeoffs change, and as attack strategies and threat models evolve.

6. Resilience analysis

In this section, we address the fundamental question of *how to measure and quantify network resilience*. To develop a comprehensive understanding of network resilience, methodologies are needed to measure the resilience (or lack thereof) of a given network and evaluate the benefit of proposed architectures, principles, and mechanisms. Traditionally, both resilience mechanisms and measures

have been domain specific as well as challenge specific. For example, existing research on fault tolerance measures such as reliability and availability targets single instances of random faults, such as topology based survivability analysis, considering node and link failures [155–157]. More recently, generic survivability frameworks consider network dynamics in addition to infrastructure failures [29,158,159]. Survivability can be quantified based on availability and network performance models [160–162,30,163] using the T1A1.2 working group definition of survivability [52,13]. Resilience can be quantified as the transient performance and availability measure of the network when subjected to challenges outside of the design envelope [164]. Service oriented network measures include *user lost erlangs* (measuring the traffic capacity lost during an outage) and *unservability* [165,54]. Based on the common distinction in the industry between equipment vendor, service provider, and end user, specific metrics have been developed for each domain. In the field of network security, the commonly taken approach is to perform a vulnerability analysis [166,167,55] in order to determine how a network responds to security risks. Resilience evaluation is more difficult than evaluating networks in terms of traditional security metrics, due to the need to evaluate the ability of the network to continue providing an acceptable level of service, while withstanding challenges as defined in Section 2.3.1 [167,168].

Evaluating network resilience in this way effectively quantifies it as a measure of service degradation in the presence of challenges (perturbations) to the operational state of the network. Hence, the network can be viewed (at any layer) as consisting of two orthogonal dimensions as shown in Fig. 6: one is the operational state of the network, which consists of its physical infrastructure and their protocols; the second dimension is the services being provided by the network and its requirements [168].

Formally, let resilience be defined at the boundary B_{ij} between any two adjacent layers L_i, L_j. In order to characterise the state of the network below the boundary B_{ij}, let there be a set of k *operational metrics* $\mathbb{N} = \{N_1, N_2, \ldots, N_k\}$. To characterise the service from layer j to layer i, let there be a set of l *service parameters* $\mathbb{P} = \{P_1, P_2, \ldots, P_l\}$. Note that both of these dimensions are multi-variate and there is a clear mapping between the operational metrics and service parameters. Simply put, for a given set of operational conditions, the network provides a certain level of service for a given application. Thus, the *state* of a network is an aggregate of several points in this two-dimensional space.

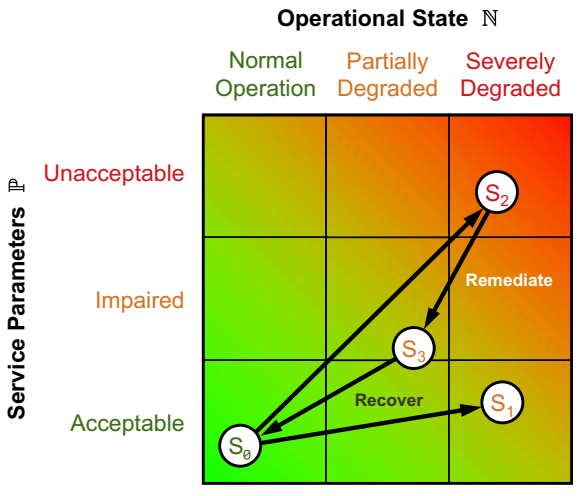


Fig. 6. Resilience state space.

In order to limit the number of states, the operational and service space of the network may be divided into three regions each as shown in Fig. 6. The network operational space is divided into *normal*, *partially-degraded*, and *severely-degraded* regions. Similarly, the service space is divided into *acceptable*, *impaired*, and *unacceptable* regions. While an arbitrary number of such regions is possible, one of the primary goals of this work is to achieve tractable yet useful solutions, and this set of nine (3×3) regions provides the necessary abstraction while limiting the number total regions. Each region may contain multiple states if the service demands such a fine granularity. In the limiting case, each region represents just one state.

6.1. State transitions and resilience evaluation

Applying the *fault* → *error* → *failure* chain (Section 2.1), adverse events are manifest as degradations in the operational condition of the network. Hence, when such events degrade the operational state of the network, the level of service being provided degrades as well resulting in *state transitions*, e.g., $S_0 \rightarrow S_1$ and $S_0 \rightarrow S_2$. The boundaries of each state and the number of states are determined by the application being supported as well as the expected service. Resilience 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 R_{ij} as a function of $\mathbb{N}$ and $\mathbb{P}$. In the simplest case R_{ij} is the slope of the curve obtained by plotting $\mathbb{P}$ vs. $\mathbb{N}$ on a multi-variate piecewise axis. For example, when comparing two services over a given network, the service with a smaller slope $(S_0 \rightarrow S_1)$ is considered more resilient as shown in Fig. 6. There are two potential issues with this approach: (1) The number of metrics in each dimension may be very large and (2) there may be state explosion due to the number of quantifiable network states. Fortunately, in this context, both problems can be avoided. First, the number of metrics can be limited to those that affect the service under consideration. Secondly, the number of states are limited by the granularity of the service differentiation required as illustrated below. Finally, all transients in network operation (originating from network dynamics) that result in a predetermined service level are aggregated into one state. This approach significantly reduces the number of states.

A network's resilience is then evaluated based on the range of operational conditions for which it stays in the acceptable service region (irrespective of the specific state), and to declare a network *resilient* implies that it stays in the acceptable service region across a wide range of operational conditions. An ideal network would be one that stays in the acceptable service region (all its states distributed within the acceptable service region) for all network conditions under the presence of any number of

challenges, but given that ideal systems are impractical, the objective is to design networks that stay in the acceptable state as long as possible and degrade gracefully in the face of increasingly severe challenges.

Fig. 7 shows the relationship of the $D^2R^2 + DR$ strategy (Section 4) to the 2-dimensional state space. The left part shows the fault → error → failure chain (Fig. 1) with defence resisting the activation of faults and propagation of errors to service failures, as previously described. Challenges and errors can be detected to initiate remediation. The right side of the figure shows the operational dimension $\mathbb{N}$ and service dimension $\mathbb{P}$ as ranges that are degraded when errors propagate to operational state and when the effects of the operational state impact the service

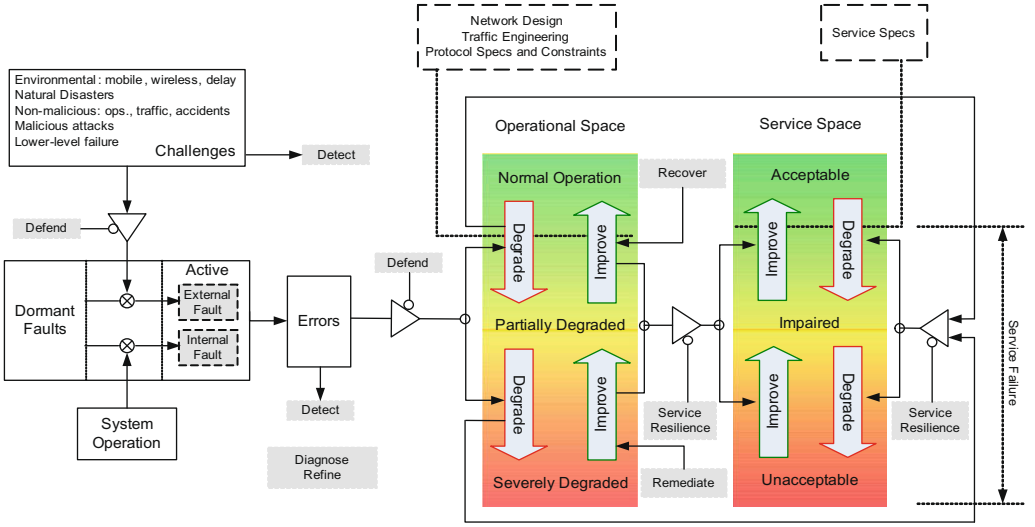


Fig. 7. ResiliNets strategy block diagram.

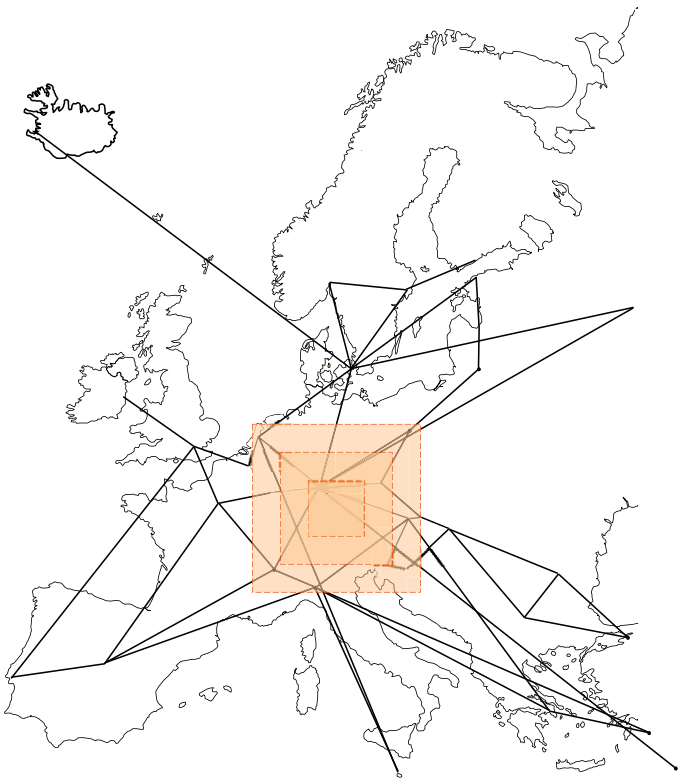


Fig. 8. Simulated area-based challenges for GÉANT2 topology.

state ($S_0 \rightarrow S_2$ trajectory in Fig. 6). Remediation mechanisms help drive the operational state towards improvement ($S_2 \rightarrow S_3$ trajectory), and service resilience resists the effects of degraded operational state from impairing the service ($S_0 \rightarrow S_1$ trajectory). Recovery moves the operational state back to normal operation at the end of the inner strategy loop ($S_3 \rightarrow S_0$ trajectory). Diagnosis and refinement are the outer loop, used to improve this entire process.

6.2. Resilience analysis scenario

To perform this evaluation, challenges are applied to the network that induce failures in links, nodes, or protocols. As an example consider the application of the state space to the link/topology boundary in Table 1. The horizontal $\mathbb{N}$ axis represents metrics that describe the operational state of the network in terms of link failures (below the link/topology service line). Zero (or a small number) of failures are within normal operation, and some larger number of failures are defined as being severely degraded. Challenges to the network that cause links to fail then cause the state to move from normal operations S_0 to the right. Fig. 8 shows an example network to which we might apply such a challenge: the GÈANT2 research network.

The vertical $\mathbb{P}$ axis is the service delivered above the link/topology line, which is a connected diverse topology, and this is measured by metrics such as number of partitions, k-connectedness, and total graph diversity [106]. Consider the example of an area-based challenge [27,28] shown by the squares in Fig. 8, which might be the result of a major power failure or storm. As the impacted area increases, the topology goes from acceptable S_0 upwards, resulting in an impaired topology measured as loss of diversity (the smallest square), to an unacceptable topology that is partitioned (the largest square). This particular example would follow a trajectory $S_0 \rightarrow S_2$ in Fig. 6, however if more links existed (for example from the UK to Iceland and Moscow to Turkey), a trajectory closer to $S_0 \rightarrow S_1$ would result since the network would not partition. This analysis can continue up the levels, and be performed using a combination of analysis and simulation, particularly at the higher levels to determine the effects on end-to-end protocols and applications.

7. Summary and research directions

The Internet has become essential to all aspects of modern life, and thus the consequences of network disruption have become increasingly severe. It is widely recognised that the Internet is not sufficiently resilient, survivable, or dependable, and that significant research, development, and engineering are necessary to improve the resilience of its infrastructure and services. Substantial research has previously gone into the disciplines that are part of resilience, and it is now possible to relate these to one another and derive an architectural framework to improve the resilience of existing networks, to guide the inclusion of resilience as a first-class design property in the evolving and future Internet.

This paper has presented a systematic architectural framework that unifies resilience disciplines, strategies, principles, and analysis. The D^2R^2 + DR (defend, detect, remediate, recover + diagnose, refine) ResiliNets strategy leads to a set of design principles that guide the analysis and design of resilient networks. The principles encompass prerequisites, tradeoffs, enablers, and behaviours for resilient network architecture and design. We believe this to be the most comprehensive resilience framework to date, which builds upon and unifies previous frameworks such as ANSA, T1, CMU-CERT, and SUMOWIN.

While most of the technical areas covered by the ResiliNets framework are the subject of intense investigation by a large number of researchers, some are more mature than others. We believe that significantly more work needs to be done in the areas of structural defences and remediation mechanisms, understanding and defining resilience metrics, and the refinement aspects of the outer control loop. Future Internet research and infrastructure programmes such as FIND [169], FIRE [170], and GENI [171], and projects within these programmes such as PoMo [102,103], ResumeNet [104,105], and GpENI [172,173] provide a fertile context in which to perform this research and experimentation.

Acknowledgments

This research was supported in part by the National Science Foundation FIND (Future Internet Design) Program under Grant CNS-0626918 (Postmodern Internet Architecture) and the European Commission under Grants EU FP6-IST-27489 (Autonomic Network Architecture) and FP7-224619 (ResumeNet). The authors would like to thank members of these projects for their input into the ideas expressed in this paper, and also the reviewers, whose comments have helped to improve this paper.

References

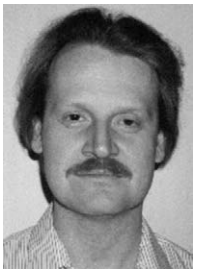
[1] Protecting America's Infrastructures, Report, President's Commission on Critical Infrastructure Protection, 1997.
[2] S. Rinaldi, J. Peerenboom, T. Kelly, Identifying, understanding, and analyzing critical infrastructure interdependencies, IEEE Control Systems Magazine 21 (6) (2001) 11–25, doi:10.1109/37.969131. ISSN 0272-1708.
[3] K. Stouffer, J. Falco, K. Kent, Guide to supervisory control and data acquisition (SCADA) and industrial control systems security, Special Publication NIST-SP-800-82-2006, National Institute of Standards and Technology (NIST), 2006.
[4] A Roadmap for Cybersecurity Research, Technical Report, Department of Homeland Security (DHS), 2009.
[5] S. Goodman, H. Lin, Toward a Safer and More Secure Cyberspace, National Academies Press, 2007.
[6] F. Schneider, Trust in Cyberspace, National Academies Press, 1999.
[7] UK Resilience Homepage, <http://www.cabinetoffice.gov.uk/ukresilience.aspx>, 2010.
[8] European Information Society, <http://ec.europa.eu/information_society/policy/nis/strategy/activities/ciip/index_en.htm>, 2010.
[9] J.-C. Laprie, Dependability: basic concepts and terminology, Draft, IFIP Working Group 10.4 – Dependable Computing and Fault Tolerance, 1994.
[10] M. Steinder, A. Sethi, A survey of fault localization techniques in computer networks, Science of Computer Programming 53 (2) (2004) 165–194.

[11] T1A1.2 Working Group, ATIS Telecom Glossary 2000, American National Standard for Telecommunications T1.523-2001, Alliance for Telecommunications Industry Solutions (ATIS), 2001.
[12] 1037C, Telecommunications: Glossary of Telecommunication Terms, 1996.
[13] T1A1.2 Working Group, Reliability-related metrics and terminology for network elements in evolving communications networks, American National Standard for Telecommunications T1.TR.524-2004, Alliance for Telecommunications Industry Solutions (ATIS), 2004.
[14] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, Transactions on Dependable and Secure Computing 1 (1) (2004) 11–33. January–March, ISSN 1545-5971.
[15] D. Lardner, Babbage's calculating engine, Edinburgh Review 59 (120) (1834) 263–327.
[16] E. Moore, C. Shannon, Reliable circuits using less reliable relays, Journal of the Franklin Institute 262 (3) (1956) 191–208.
[17] W. Pierce, Failure-tolerant Computer Design, Academic Press, 1965.
[18] A. Avizienis, Design of fault-tolerant computers, in: 1967 Fall Joint Computer Conference, vol. 31 of *AFIPS Conf. Proc.*, Thompson Books, 1967, pp. 733–743.
[19] R. Lyons, W. Vanderkulk, The use of triple-modular redundancy to improve computer reliability, IBM Journal of Research and Development 6 (2) (1962) 200–209.
[20] L. Chen, A. Avizienis, N-version programming: a fault tolerance approach to the reliable software, in: Proceedings of the Eighth International Symposium on Foult-Tolerant Computing, Toulouse, France, 1978.
[21] B. Randell, System structure for software fault tolerance, in: Proceedings of the International Conference on Reliable Software, ACM, New York, NY, USA, 1975, pp. 437–449.
[22] G. Ellinas, T. Stern, Automatic protection switching for link failures in optical networks with bi-directional links, in: Proceedings of the Global Telecommunications Conference (GLOBECOM), vol. 1, 1996, pp. 152–156, doi: 10.1109/GLOCOM.1996.594351.
[23] W.D. Grover, D. Stamatelakis, Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration, in: Proceeding of the IEEE International Conference on Communications (ICC'98), vol. 1, 1998, pp. 537–543.
[24] J. Strand, A. Chiu, R. Tkach, Issues for routing in the optical layer, IEEE Communications Magazine 39 (2) (2001) 81–87, doi:10.1109/35.900635. ISSN: 0163-6804.
[25] R. Ellison, D. Fisher, R. Linger, H. Lipson, T. Longstaff, N. Mead, Survivable network systems: an emerging discipline, Tech. Rep. CMU/SEI-97-TR-013, Software Engineering Institute, Carnegie Mellon University, 1997.
[26] J.P.G. Sterbenz, R. Krishnan, R.R. Hain, A.W. Jackson, D. Levin, R. Ramanathan, J. Zao, Survivable mobile wireless networks: issues, challenges, and research directions, in: WiSE '02: Proceedings of the Third ACM Workshop on Wireless Security, ACM Press, New York, NY, USA, 2002, pp. 31–40, ISBN: 1-58113-585-8.
[27] R.A. Mahmood, Simulating Challenges to Communication Networks for Evaluation of Resilience, 2009.
[28] B. Bassiri, S.S. Heydari, Network survivability in large-scale regional failure scenarios, in: Proceedings of the Second Canadian Conference on Computer Science and Software Engineering (C3S2E), ACM, New York, NY, USA, 2009, pp. 83–87, ISBN 978-1-60558-401-0, doi: <http://doi.acm.org/10.1145/1557626.1557639>.
[29] J.C. Knight, E.A. Strunk, K.J. Sullivan, Towards a rigorous definition of information system survivability, in: Proceedings of the DARPA Information Survivability Conference and Exposition DISCEX III, Washington DC, 2003, pp. 78–89.
[30] P.E. Heegaard, K.S. Trivedi, Network Survivability Modeling, Computer Networks 53(8) (2009) 1215–1234, ISSN 1389-1286, doi: 10.1016/j.comnet.2009.02.014, Performance Modeling of Computer Networks: Special Issue in Memory of Dr. Gunter Bolch.
[31] J. Westcott, G. Lauer, Hierarchical routing for very large networks, in: Proceedings of the IEEE Military Communications Conference (MILCOM), vol. 2, 1984, pp. 214–218, doi: 10.1109/MILCOM.1984.4794999.
[32] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: SIGCOMM '94: Proceedings of the Conference on Communications Architectures, Protocols and Applications, ACM, New York, NY, USA, 1994, pp. 234–244, ISBN: 0-89791-682-4.
[33] D.B. Johnson, Routing in ad hoc networks of mobile hosts, in: Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications (WMCSA), IEEE Computer Society, Washington, DC, USA, 1994, pp. 158–163, ISBN: 978-0-7695-3451-0, doi: <http://dx.doi.org/10.1109/WMCSA.1994.33>.
[34] R.C. Durst, G.J. Miller, E.J. Travis, TCP extensions for space communications, in: MobiCom '96: Proceedings of the Second Annual International Conference on Mobile Computing and Networking, ACM Press, New York, NY, USA, 1996, pp. 15–26, ISBN: 0-89791-872-X.
[35] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, H. Weiss, Delay-tolerant networking: an approach to interplanetary internet, IEEE Communications Magazine 41 (6) (2003) 128–136. ISSN: 0163-6804.
[36] K. Fall, A delay-tolerant network architecture for challenged internets, in: SIGCOMM '03: Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, New York, NY, USA, 2003, pp. 27–34, ISBN: 1-58113-735-4.
[37] Q. Li, D. Rus, Sending messages to mobile users in disconnected ad hoc wireless networks, in: Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom), ACM, New York, NY, USA, 2000, pp. 44–55, ISBN: 1-58113-197-6, doi: <http://doi.acm.org/10.1145/345910.345918>.
[38] W. Zhao, M. Ammar, E. Zegura, A message ferrying approach for data delivery in sparse mobile ad hoc networks, in: Proceedings of the Fifth ACM International Symposium on Mobile Ad hoc Networking and Computing (MobiHoc), ACM, New York, NY, USA, 2004, pp. 187–198, ISBN: 1-58113-849-0, doi: <http://doi.acm.org/10.1145/989459.989483>.
[39] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[40] S. Singh, M. Woo, C. Raghavendra, Power-aware routing in mobile ad hoc networks, in: Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom), ACM, New York, NY, USA, 1998, pp. 181–190, ISBN: 1-58113-035-X, doi: <http://doi.acm.org/10.1145/288235.288286>.
[41] R. Shah, J. Rabaey, Energy aware routing for low energy ad hoc sensor networks, in: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), vol. 1, 2002, pp. 350–355, doi: 10.1109/WCNC.2002.993520.
[42] F. Li, Y. Wang, Routing in vehicular ad hoc networks: a survey, IEEE Vehicular Technology Magazine 2 (2) (2007) 12–22, doi:10.1109/MVT.2007.912927. ISSN: 1556-6072.
[43] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, MaxProp: routing for vehicle-based disruption-tolerant networks, in: Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM), 2006, ISSN: 0743-166X, doi: 10.1109/INFOCOM.2006.228.
[44] J. Ott, D. Kutscher, Drive-thru internet: IEEE 802.11b for "Automobile" Users, in: 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 1, 2004, p. 373, ISSN: 0743-166X, doi: 10.1109/INFCOM.2004.1354509.
[45] A. Jabbar, J.P. Rohrer, A. Oberthaler, E.K. Çetinkaya, V. Frost, J.P.G. Sterbenz, Performance comparison of weather disruption-tolerant cross-layer routing algorithms, in: Proceedings of the IEEE INFOCOM 2009, the 28th Conference on Computer Communications, ISSN: 0743-166X, 1143–1151, 2009.
[46] J.P. Rohrer, A. Jabbar, E. Perrins, J.P.G. Sterbenz, Cross-layer architectural framework for highly-mobile multihop airborne telemetry networks, in: Proceedings of the IEEE Military Communications Conference (MILCOM), San Diego, CA, USA, 2008.
[47] J. Jung, B. Krishnamurthy, M. Rabinovich, Flash crowds and denial-of-service attacks: characterization and implications for CDNs and web sites, in: Proceedings of the 11th International Conference on World Wide Web (WWW), ACM, New York, NY, USA, 2002, pp. 293–304, ISBN: 1-58113-449-5, doi: <http://doi.acm.org/10.1145/511446.511485>.
[48] J. Mirkovic, P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, SIGCOMM Computer Communication Review 34 (2) (2004) 39–53. ISSN: 0146-4833, doi: <http://doi.acm.org/10.1145/997150.997156>.
[49] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, Technical Research Report TR 2004-47, Institute for Systems Research, the University of Maryland, 2004.
[50] P.A. Lee, T. Anderson, Fault Tolerance: Principles and Practice, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1990, ISBN: 0387820779.
[51] R. Billinton, R. Allan, Reliability Evaluation of Engineering Systems, Plenum Press, New York, 1992.

[52] T1A1.2 Working group, enhanced network survivability performance, Technical Report T1.TR.68-2001, Alliance for Telecommunications Industry Solutions (ATIS), 2001.
[53] M. Clouqueur, W. Grover, Availability analysis of span-restorable mesh networks, IEEE Journal on Selected Areas in Communications 20 (4) (2002) 810–821.
[54] W.D. Grover, Mesh-based Survivable Networks, Prentice-Hall PTR Pearson, Upper Saddle River, NJ, 2003, 2004.
[55] D.M. Nicol, W.H. Sanders, K.S. Trivedi, Model-based evaluation: from dependability to security, IEEE Transactions on Dependable and Secure Computing 01 (1) (2004) 48–65. ISSN: 1545-5971, doi: <http://doi.ieeecomputersociety.org/10.1109/TDSC.2004.11>.
[56] C. Landwehr, Computer security, International Journal of Information Security 1 (1) (2001) 3–13.
[57] R.M. Savola, Towards a taxonomy for information security metrics, in: Proceedings of the ACM workshop on Quality of Protection (QoP), ACM, New York, NY, USA, 2007, pp. 28–30, ISBN: 978-1-59593-885-5, doi: <http://doi.acm.org/10.1145/1314257.1314266>.
[58] R.B. Vaughn, R. Henning, A. Siraj, Information assurance measures and metrics: state of practice and proposed taxonomy, in: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS), IEEE Computer Society, Washington, DC, USA, 2003, ISBN: 0-7695-1874-5, 331.3.
[59] R. Shirey, Internet Security Glossary, Version 2, RFC 4949 (Informational), 2007.
[60] J. Meyer, Performability: a retrospective and some pointers to the future, Performance Evaluation 14 (3–4) (1992) 139–156.
[61] A. Campbell, G. Coulson, D. Hutchison, A quality of service architecture, SIGCOMM Computer Communication Review 24 (2) (1994) 6–27. ISSN: 0146-4833, doi: <http://doi.acm.org/10.1145/185595.185648>.
[62] C. Aurrecoechea, A. Campbell, L. Hauw, A survey of QoS architectures, Multimedia Systems 6 (3) (1998) 138–151.
[63] J.S. Turner, Design of an integrated services packet network, SIGCOMM Computer Communication Review 15 (4) (1985) 124–133. ISSN: 0146-483, doi: <http://doi.acm.org/10.1145/318951.319028>.
[64] J. Gruber, L. Nguyen, Performance requirements for integrated voice/data networks, IEEE Journal on Selected Areas in Communications (JSAC) 1 (6) (1983) 981–1005. ISSN: 0733-8716.
[65] N. Yeadon, F. Garcia, D. Hutchison, D. Shepherd, Filters: QoS support mechanisms for multipeer communications, IEEE Journal on Selected Areas in Communications 14 (7) (1996) 1245–1262, doi:10.1109/49.53636. ISSN: 0733-8716.
[66] A. Lazar, G. Pacifici, Control of resources in broadband networks with quality of service guarantees, IEEE Communications Magazine 29 (10) (1991) 66–73, doi:10.1109/35.99263. ISSN: 0163-6804.
[67] E. Jen, Robust Design: A Repertoire of Biological, Ecological, and Engineering Case Studies, Oxford University Press, 2005.
[68] W. Willinger, J. Doyle, Robustness and the Internet: Design and Evolution, Oxford University Press, 2005.
[69] M. Prokopenko, F. Boschetti, A.J. Ryan, An information-theoretic primer on complexity, self-organisation and emergence, Complexity 15 (1) (2009) 11–28, doi:10.1002/cplx.20249.
[70] L. Xie, P. Smith, D. Hutchison, M. Banfield, H. Leopold, A. Jabbar, J.P.G. Sterbenz, From detection to remediation: a self-organized system for addressing flash crowd problems, in: ICC '08: Proceedings of IEEE International Conference on Communications, Beijing, China, 2008, pp. 5809–5814.
[71] M. Mayer, This site may harm your computer on every search result?!?!, 2009, <http://googleblog.blogspot.com/2009/01/this-site-may-harm-your-computer-on.html>.
[72] B. Liscouski, W.J. Elliot, Final report on the August 14, 2003 blackout in the United States and Canada: causes and recommendations, Tech. Rep., US – Canada Power System Outage Task Force, 2004.
[73] J.H. Cowie, A.T. Ogielski, B. Premore, E.A. Smith, T. Underwood, Impact of the 2003 blackouts on internet communications, Preliminary Report, Renesys Corporation (updated March 1, 2004), 2003.
[74] D. Kuhn, Sources of failure in the public switched telephone network, Computer 30 (4) (1997) 31–36, doi:10.1109/2.585151. ISSN: 0018-9162.
[75] H.C. Styron, CSX Tunnel Fire: Baltimore, MD, US Fire Administration Technical Report USFA-TR-140, Federal Emergency Management Administration, Emmitsburg, MD, 2001.
[76] M.R. Carter, M.P. Howard, N. Owens, D. Register, J. Kennedy, K. Pecheux, A. Newton, Effects of catastrophic events on transportation system management and operations, Howard Street Tunnel Fire, Baltimore City, Maryland – July 18, 2001, Tech. Rep., US Department of Transportation, ITS Joint Program Office, Washington DC, 2002.
[77] M.A. Brown, Pakistan hijacks YouTube, 2008, <http://www.renesys.com/blog/2008/02/pakistan_hijacks_youtube_1.shtml>.
[78] C. Meinel, Attacking and Defending the Internet with Border Gateway Protocol (BGP), Online Article, Cisco, 2008.
[79] M.A. Brown, E. Zmijewski, Pakistan Telecom Hijacks YouTube: Or how to SYN-flood DOS yourself while annoying everyone on the planet, Presentation, Renesys Corporation, Taipei, 2008.
[80] E. Zmijewski, Reckless Driving on the Internet, 2009a, <http://www.renesys.com/blog/2009/02/the-flap-heard-around-the-worl.shtml>.
[81] E. Zmijewski, Longer is Not Always Better, 2009b, <http://www.renesys.com/blog/2009/02/longer-is-not-better.shtml>.
[82] T. Davis, H. Rogers, C. Shays, Others, A Failure of Initiative: The Final Report of the Select Bipartisan Committee to Investigate the Preparation for and Response to Hurricane Katrina, Congressional Report H. Rpt. US House of Representatives, Washington, DC, 2006, pp. 109–377.
[83] J. Cowie, A. Popescu, T. Underwood, Impact of Hurricane Katrina on Internet Infrastructure, Report, Renesys, 2005.
[84] Y. Kitamura, Y. Lee, R. Sakiyama, K. Okamura, Experience with restoration of Asia Pacific network failures from Taiwan earthquake, IEICE Transactions on Communications E90-B (11) (2007) 3095–3103.
[85] Severe Space Weather Events: Understanding Societal and Economic Impacts, Workshop Report, National Research Council, 2008.
[86] Report of the Commission to Assess the Threat to the United States from Electromagnetic Pulse (EMP) Attack, Report, Critical National Infrastructures, 2004.
[87] Pandemic Influenza Preparedness, Response, and Recovery Guide for Critical Infrastructure and Key Resources, CI/KR Guide, Department of Homeland Security (DHS), 2006.
[88] Pandemic Influenza Impact on Communications Networks Study, Unclassified, Department of Homeland Security (DHS), 2007.
[89] C. Partridge, P. Barford, D.D. Clark, S. Donelan, V. Paxon, J. Rexford, M.K. Vernon, The Internet Under Crisis Conditions: Learning from September 11, Report 10659, National Research Council, 2003.
[90] Report of the 7 July Review Committee, Report, Greater London Authority, 2006.
[91] M. Lesk, The New Front Line: Estonia under Cyberassault, IEEE Security and Privacy 5 (2007) 76–79. ISSN: 1540-7993, doi: <http://doi.ieeecomputersociety.org/10.1109/MSP.2007.98>.
[92] A. Popescu, B. Premore, E. Zmijewski, Impact of the Middle East Cable Breaks: A Global BGP Perspective, Presentation, Renesys Corp., San Jose, CA, 2008.
[93] Wikipedia: 2008 Submarine Cable Disruption, 2008, <http://en.wikipedia.org/wiki/2008_submarine_cable_disruption>.
[94] Mediterranean Fibre Cable Cut – a RIPE NCC Analysis, 2008, <http://www.ripe.net/projects/reports/2008cable-cut/>.
[95] N. Edwards, Building Dependable Distributed Systems, Technical Report APM.1144.00.02, ANSA, 1994.
[96] D.P. Siewiorek, R.S. Swarz, Reliable Computer Systems: Design and Evaluation, second ed., Digital Press, 1992.
[97] T1A1.2 Working Group, Network Survivability Performance, Technical Report T1A1.2/93-001R3, Alliance for Telecommunications Industry Solutions (ATIS), 1993.
[98] J.P.G. Sterbenz, D. Hutchison, ResiliNets: Multilevel Resilient and Survivable Networking Initiative Wiki, 2008, <http://wiki.ittc.ku.edu/resilinets>.
[99] M. Schöller, J.P.G. Sterbenz, A. Jabbar, D. Hutchison, First draft of the Resilience and Security Framework, 2006, <http://www.ana-project.org/deliverables/2006/D.3.2.-Resilience.pdf>.
[100] Autonomic Network Architecture Wiki, 2006, <http://www.ana-project.org/>.
[101] G. Bouabene, C. Jelger, C. Tschudin, S. Schmid, A. Keller, M. May, The autonomic network architecture (ANA), IEEE Journal on Selected Areas in Communications (JSAC) 28 (1) (2010) 4–14, doi:10.1109/JSAC.2010.100102. ISSN: 0733-8716.
[102] B. Bhattacharjee, K. Calvert, J. Griffioen, N. Spring, J.P.G. Sterbenz, Postmodern Internetwork Architecture, Technical Report ITTC-FY2006-TR-45030-01, Information and Telecommunication Center, 2335 Irving Hill Road, Lawrence, KS 66045-7612, 2006.
[103] J.P.G. Sterbenz, B. Bhattacharjee, K. Calvert, J. Griffioen, N. Spring, PoMo: Postmodern Internetwork Architecture Wiki, 2008, <http://wiki.ittc.ku.edu/pomo>.

[104] ResumeNet Wiki, 2009, <http://www.resumenet.eu/project/index>.
[105] M. Schöller, P. Smith, C. Rohner, M. Karaliopoulos, A. Jabbar, J.P.G. Sterbenz, D. Hutchison, On realising a strategy for resilience in opportunistic networks, in: Proceedings of the EU Future Network and Mobile Summit, Florence, Italy, in press.
[106] J.P. Rohrer, A. Jabbar, J.P.G. Sterbenz, Path diversification: a multipath resilience mechanism, in: Proceedings of the Seventh International Workshop on the Design of Reliable Communication Networks (DRCN), Washington, DC, USA, 2009a.
[107] D. Guidoni, R. Mini, A. Loureiro, On the Design of Resilient Heterogeneous Wireless Sensor Networks based on Small World Concepts, Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET) 54 (8) (2010) 1266–1281.
[108] P.M.S. del Río, J.A. Hernández, J. Aracil, J.E.L. de Vergara, J. Domzal, R. Wojcik, P. Cholda, K. Wajda, J.P. Fernández-Palacios, Ó.G. de Dios, R. Duque, A reliability analysis of double-ring topologies with dual attachment using p-cycles for optical metro networks, Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET) 54 (8) (2010) 1328–1341.
[109] S. Bellovin, W. Cheswick, Network firewalls, IEEE Communications Magazine 32 (9) (1994) 50–57, doi:10.1109/35.312843. ISSN: 0163-6804.
[110] Z. Li, Y. Gao, Y. Chen, HiFIND: a high-speed flow-level intrusion detection approach with DoS resiliency, Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET) 54 (8) (2010) 1282–1299.
[111] K. Park, S. Pack, T. Kwon, An adaptive peer-to-peer live streaming system with incentives for resilience, Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET) 54 (8) (2010) 1316–1327.
[112] J. Moy, Experience with the OSPF Protocol, RFC 1246 (Informational), 1991.
[113] A. Kvalbein, A.F. Hansen, T. Cicic, S. Gjessing, O. Lysne, Multiple Routing Configurations for Fast IP Network Recovery, IEEE Transactions on Networking 17 (2) (2008) 473–486.
[114] M. Menth, M. Hartmann, R. Martin, T. Cicic, A. Kvalbein, Loop-free alternates and not-via addresses: a proper combination for IP fast-reroute?, Computer Networks: Special Issue on Resilient and Survivable Networks (COMNET) 54 (8) (2010) 1300–1315.
[115] P. Cholda, A. Mykkeltveit, B. Helvik, O. Wittner, A. Jajszczyk, A survey of resilience differentiation frameworks in communication networks, IEEE Communications Surveys Tutorials 9 (4) (2007) 32–55, doi:10.1109/COMST.2007.4444749. ISSN: 1553-877X.
[116] M. Menth, M. Duelli, R. Martin, J. Milbrandt, Resilience analysis of packet-switched communication networks, IEEE/ACM Transactions on Networking 17 (6) (2009) 1950–1963, doi:10.1109/TNET.2009.202098. ISSN: 1063-6692.
[117] V. Cerf, E. Cain, The DoD internet architecture model, Computer Networks 7 (1983) 307–318.
[118] K. Ku, Z.-L. Zhang, S. Bhattacharyya, Profiling internet backbone traffic: behavior models and applications, in: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), ACM, New York, NY, USA, 2005, pp. 169–180, ISBN: 1-59593-009-4, doi: <http://doi.acm.org/10.1145/1080091.1080112>.
[119] P. Smith, D. Hutchison, M. Banfield, H. Leopold, On understanding normal protocol behaviour to monitor and mitigate the abnormal, in: Proceedings of the IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation (MonAM), Tuebingen, Germany, 2006, pp. 105–107.
[120] A. Lakhina, M. Crovella, C. Diot, Mining anomalies using traffic feature distributions, in: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), ACM, New York, NY, USA, 2005, pp. 217–228, ISBN 1-59593-009-4, doi: <http://doi.acm.org/10.1145/1080091.1080118>.
[121] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Computing Surveys 41 (3) (2009) 1–58. ISSN: 0360-0300, doi: <http://doi.acm.org/10.1145/1541880.1541882>.
[122] C. Alberts, A. Dorofee, J. Steven, C. Woody, Introduction to the OCTAVE Approach, Tech. Rep., 2003, URL: <http://www.cert.org/octave/approach-intro.pdf>.
[123] S. Hernan, S. Lambert, T. Ostwald, A. Shostack, Threat Modeling: Uncover Security Design Flaws Using The STRIDE Approach, MSDN Magazine.
[124] D. Clark, K. Sollins, J. Wroclawski, D. Katabi, J. Kulik, X. Yang, R. Braden, T. Faber, A. Falk, V. Pingali, M. Handley, N. Chiappa, New Arch: Future Generation Internet Architecture, Technical Report, DARPA, MIT, ISI, 2003.
[125] D. Clark, R. Braden, A. Falk, V. Pingali, FARA: reorganizing the addressing architecture, SIGCOMM Computer Communication Review 33 (4) (2003) 313–321. ISSN: 0146-4833.
[126] J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, A. Warfield, Plutarch: an argument for network pluralism, in: Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA), ACM, New York, NY, USA, 2003, pp. 258–266, ISBN: 1-58113-748-0, doi: <http://doi.acm.org/10.1145/944759.944763>.
[127] D.D. Clark, J. Wroclawski, K.R. Sollins, R. Braden, Tussle in cyberspace: defining tomorrow's internet, IEEE/ACM Transactions on Networking 13 (3) (2005) 462–475. ISSN: 1063-6692, doi: <http://dx.doi.org/10.1109/TNET.2005.850224>.
[128] J. Paul Nussbaumer, B.V. Patel, F. Schaffa, J.P.G. Sterbenz, Networking requirements for interactive video on demand, IEEE Journal on Selected Areas in Communications 13 (1995) 779–787.
[129] J.P.G. Sterbenz, J.D. Touch, High-Speed Networking: A Systematic Approach to High-Bandwidth Low-Latency Communication, first ed., Wiley, 2001.
[130] M.H. Behringer, Classifying network complexity, in: Proceedings of the ACM Workshop on Re-architecting the Internet (ReArch), ACM, New York, NY, USA, 2009, pp. 13–18, ISBN: 978-1-60558-749-3, doi: <http://doi.acm.org/10.1145/1658978.1658983>.
[131] M.S. Blumenthal, D.D. Clark, Rethinking the design of the internet: the end-to-end arguments vs. the brave new world, ACM Transactions on Internet Technology 1 (1) (2001) 70–109. ISSN: 1533-5399, doi: <http://doi.acm.org/10.1145/383034.383037>.
[132] J.C. Doyle, D.L. Alderson, L. Li, S. Low, M. Roughan, S. Shalunov, R. Tanaka, W. Willinger, The "Robust Yet Fragile" Nature of the Internet, Proceedings of the National Academy of Sciences of the United States of America 102 (41) (2005) 14497–14502. ISSN: 00278424.
[133] A. Vahdat, D. Becker, Epidemic routing for partially-connected ad hoc networks, Technical Report CS-200006, Duke University, 2000.
[134] S. Heimlicher, M. Karaliopoulos, H. Levy, T. Spyropoulos, On leveraging partial paths in partially-connected networks, in: Proceeding of the 28th IEEE Conference on Computer Communications (INFOCOM), Rio de Janeiro, Brazil, 2009.
[135] A.J. McAuley, Reliable broadband communication using a burst erasure correcting code, SIGCOMM Computer Communication Review 20 (4) (1990) 297–306. ISSN: 0146-4833.
[136] J.W. Byers, M. Luby, M. Mitzenmacher, A. Rege, A digital fountain approach to reliable distribution of bulk data, SIGCOMM Computer Communication Review 28 (4) (1998) 56–67. ISSN: 0146-4833, doi: <http://doi.acm.org/10.1145/285243.285258>.
[137] A.K. Miu, H. Balakrishnan, C.E. Koksal, Improving loss resilience with multi-radio diversity in wireless networks, in: Proceedings of the 11th ACM MOBICOM Conference, Cologne, Germany, 2005.
[138] F. Junqueira, R. Bhagwan, K. Marzullo, S. Savage, G.M. Voelker, The phoenix recovery system: rebuilding from the ashes of an internet catastrophe, in: Proceedings of the Ninth Workshop on Hot Topics in Operating Systems (HotOS IX), Lihue, Hawaii, 2003.
[139] E. Ayanoglu, I. Chih-Lin, R. Gitlin, J. Mazo, Diversity coding: using error control for self-healing in communication networks, in: Proceedings of the Ninth Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOMM), vol. 1, 1990, pp. 95–104, doi: 10.1109/INFCOM.1990.91238.
[140] J.P. Rohrer, R. Naidu, J.P.G. Sterbenz, Multipath at the transport layer: an end-to-end resilience mechanism, in: RNDM'09 – International Workshop on Reliable Networks Design and Modeling, St. Petersburg, Russia, 2009b.
[141] M. Motiwala, M. Elmore, N. Feamster, S. Vempala, Path splicing, in: SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, ACM, New York, NY, USA, 2008, pp. 27–38, ISBN: 978-1-60558-175-0.
[142] H. Han, S. Shakkottai, C.V. Hollot, R. Srikant, D. Towsley, Multipath TCP: a joint congestion control and routing scheme to exploit path diversity in the internet, IEEE/ACM Transactions on Networking 14 (6) (2006) 1260–1271. ISSN: 1063-6692.
[143] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemenfe, M. Ravera, A. Jajszczyk, D. Janukowicz, K.V. Doorselaere, Y. Harada, Resilience in multilayer networks, IEEE Communications Magazine 37 (8) (1999) 70–76, doi:10.1109/35.783128. ISSN: 0163-6804.
[144] D. Medhi, D. Tipper, Multi-layered network survivability-models, analysis, architecture, framework and implementation: an overview, in: Proceedings of the DARPA Information Survivability

Conference and Exposition (DISCEX), vol. 1, 2000, pp. 173–186, doi: 10.1109/DISCEX.2000.824977.
[145] A.K. Dey, Understanding and using context, Personal Ubiquitous Computing 5 (1) (2001) 4–7. ISSN: 1617-4909, doi: <http://dx.doi.org/10.1007/s007790170019>.
[146] D. Salber, A.K. Dey, G.D. Abowd, The context toolkit: aiding the development of context-enabled applications, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, 1999, pp. 434–441, ISBN: 0-201-48559-1, doi: <http://doi.acm.org/10.1145/302979.303126>.
[147] F. Foukalas, V. Gazis, N. Alonistioti, Cross-layer design proposals for wireless mobile networks: a survey and taxonomy, IEEE Communications Surveys Tutorials 10 (1) (2008) 70–85, doi:10.1109/COMST.2008.4483671. ISSN: 1553-877X.
[148] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, F. Zambonelli, A survey of autonomic communications, ACM Transactions on Autonomous and Adaptive Systems 1(2) (2006) pp. 223–259, ISSN 1556-4665, doi: <http://doi.acm.org/10.1145/1186778.1186782>.
[149] J.O. Kephart, Research challenges of autonomic computing, in: Proceedings of the 27th International Conference on Software Engineering (ICSE), ACM, New York, NY, USA, 2005, pp. 15–22, ISBN: 1-59593-963-2, doi: <http://doi.acm.org/10.1145/1062455.1062464>.
[150] R. Krishnan, R. Ramanathan, M. Steenstrup, Optimization algorithms for large self-structuring networks, in: Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 1, 1999, pp. 71–78, doi: 10.1109/INFCOM.1999.749254.
[151] D.D. Clark, C. Partridge, J.C. Ramming, J.T. Wroclawski, A knowledge plane for the internet, in: Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications SIGCOMM, ACM, New York, NY, USA, 2003c, pp. 3–10, ISBN: 1-58113-735-4, doi: <http://doi.acm.org/10.1145/863955.863957>.
[152] K.L. Calvert, S. Bhattacharjee, E.W. Zegura, J.P.G. Sterbenz, Directions in active networks, IEEE Communications 36 (10) (1998) 72–78.
[153] A.W. Jackson, J.P.G. Sterbenz, M.N. Condell, R.R. Hain, Active network monitoring and control: the SENCOMM architecture and implementation, in: DARPA Active Networks Conference and Exposition (DANCE), IEEE Computer Society, Los Alamitos, CA, USA, 2002, pp. 379–393, ISBN: 0-7695-1564-9, doi: <http://doi.ieeecomputersociety.org/10.1109/DANCE.2002.100%3509>.
[154] S. Ratnasamy, S. Shenker, S. McCanne, Towards an evolvable internet architecture, SIGCOMM Computer Communication Review 35 (4) (2005) 313–324. ISSN: 0146-4833, doi: <http://doi.acm.org/10.1145/1090191.1080128>.
[155] S. Liew, K. Lu, A framework for network survivability characterization, in: SUPERCOMM/ICC'92: Proceedings of IEEE International Conference on Communications, 1992, ICC 92, Conference Record, Discovering a New World of Communications, 1992, pp. 405–410.
[156] S. Liew, K. Lu, A framework for characterizing disaster-based network survivability, IEEE Journal on Selected Areas in Communications 12 (1) (1994) 52–58.
[157] A. Antonopoulos, Metrication and performance analysis on resilience of ring-based transport network solutions, in: GLOBECOM'99: Global Telecommunications Conference, vol. 2, 1999, pp. 1551–1555.
[158] Q. Gan, B. Helvik, Dependability modelling and analysis of networks as taking routing and traffic into account, in: NGI '06: Proceedings of the Conference on Next Generation Internet Design and Engineering, 2006.
[159] W. Molisz, Survivability function – a measure of disaster-based routing performance, IEEE Journal on Selected Areas in Communications 22 (9) (2004) 1876–1883.
[160] Y. Liu, V. Mendiratta, K. Trivedi, Survivability analysis of telephone access network, in: Proceedings of the 15th International Symposium on Software Reliability Engineering, IEEE Computer Society Washington, DC, USA, 2004, pp. 367–378.
[161] P. Heegaard, K. Trivedi, Survivability Quantification of Communication Services, in: The 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Anchorage, Alaska, USA, 2008.
[162] P. Heegaard, K. Trivedi, Survivability Quantification of real-sized networks including end-to-end delay distributions, in: Proceedings of the Third International Conference on Systems and Networks Communications, IEEE Computer Society, 2008b, pp. 50–55.
[163] K. Trivedi, D. Kim, A. Roy, D. Medhi, Dependability and security models, in: Proceedings of the International Workshop of Design of Reliable Communication Networks (DRCN), IEEE, 2009, pp. 11–20.
[164] K. Trivedi, D. Kim, R. Ghosh, Resilience in computer systems and networks, in: Proceedings of the 2009 International Conference on Computer-Aided Design (ICCAD), IEEE, 2009.
[165] A. Zolfaghari, F.J. Kaudel, Framework for network survivability performance, IEEE Journal on Selected Areas in Communications (JSAC) 12 (1) (1994) 46–51.
[166] G. Qu, R. Jayaprakash, S. Hariri, C. Raghavendra, A framework for network vulnerability analysis, in: CT '02: Proceedings of the First IASTED International Conference on Communications, Internet, Information Technology, St. Thomas, VI, USA, 2002, pp. 289–298.
[167] S. Hariri, G. Qu, T. Dharmagadda, M. Ramkishore, C.S. Raghavendra, Impact analysis of faults and attacks in large-scale networks, IEEE Security and Privacy 01 (5) (2003) 49–54. ISSN: 1540-7993, doi: <http://doi.ieeecomputersociety.org/10. 1109/MSECP.2003.1236235>.
[168] A.J. Mohammad, D. Hutchison, J.P.G. Sterbenz, Towards quantifying metrics for resilient and survivable networks, in: Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP), 2006, pp. 17–18.
[169] NSF NeTS FIND initiative website, <http://www.nets-find.net>, 2008.
[170] FIRE: future internet research and experimentation, <http://cordis.europa.eu/fp7/ict/fire/>, 2009.
[171] GENI: Global environment for network innovations, <http://www.geni.net/>, 2009.
[172] J.P.G. Sterbenz, D. Medhi, B. Ramamurthy, C. Scoglio, D. Hutchison, B. Plattner, T. Anjali, A. Scott, C. Buffington, G.E. Monaco, D. Gruenbacher, R. McMullen, J.P. Rohrer, J. Sherrell, P. Angu, R. Cherukuri, H. Qian, N. Tare, The great plains environment for network innovation (GpENI): a programmable testbed for future internet architecture research, in: Proceedings of the Sixth International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom), Berlin, Germany, 2010.
[173] J.P.G. Sterbenz, D. Medhi, G. Monaco, B. Ramamurthy, C. Scoglio, B.-Y. Choi, J.B. Evans, D. Gruenbacher, R. Hui, W. Kaplow, G. Minden, J. Verrant, GpENI: Great Plains Environment for Network Innovation, <http://wiki.ittc.ku.edu/gpeni>, 2009.



Dr. James P.G. Sterbenz is Associate Professor of Electrical Engineering & Computer Science and on staff at the Information & Telecommunication Technology Center at The University of Kansas, and is a Visiting Professor of Computing in InfoLab 21 at Lancaster University in the UK. He received a doctorate in computer science from Washington University in St. Louis in 1991, with undergraduate degrees in electrical engineering, computer science, and economics. He is director of the ResiliNets research group at KU, PI for the NSF-funded FIND Postmodern Internet Architecture project, lead PI for the GpENI (Great Plains Environment for Network Innovation) international GENI and FIRE testbed, co-I in the EU-funded FIRE ResumeNet project, and PI for the US DoD-funded highly-mobile airborne networking project. He has previously held senior staff and research management positions at BBN Technologies, GTE Laboratories, and IBM Research, where he has lead DARPA- and internally-funded research in mobile, wireless, active, and high-speed networks. He has been program chair for IEEE GI, GBN, and HotI; IFIP IWSOS, PfHSN, and IWAN; and is on the editorial board of *IEEE Network*. He has been active in Science and Engineering Fair organisation and judging in Massachusetts and Kansas for middle and high-school students. 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 IEEE, ACM, IET/IEE, and IEICE. His research interests include resilient, survivable, and disruption tolerant networking, future Internet architectures, active and programmable networks, and high-speed networking and systems.

Dr. 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. He has recently focused his research efforts towards network resilience. He has completed many UK, European and industry-funded research contracts and published many papers as well as writing and editing books on these and related areas. He has been an expert evaluator and member or chair of various advisory boards and committees in the UK (EPSRC, DTI, OFTEL, e-Science, UKLight, UKCRC, JISC, DC-KTN) and within the EU through several Framework Programmes. Also, he has served as member or chair of numerous TPCs (including the flagship ACM SIGCOMM and IEEE Infocom), and of journal editorial boards. He is an editor of the renowned Lecture Notes in Computer Science and of the Wiley CNDS book series.

Egemen K. Çetinkaya is a Ph.D. candidate in the department of Electrical Engineering and Computer Science at The University of Kansas. He received the B.S. degree in Electronics Engineering from Uludag University (Bursa, Turkey) in 1999 and the M.S. degree in Electrical Engineering from University of Missouri-Rolla in 2001. He held various positions at Sprint as a support, system, design engineer from 2001 until 2008. He is a graduate research assistant at the KU Information & Telecommunication Technology Center (ITTC). His research interests are in resilient networks. He is a member of the IEEE Communications Society, ACM SIGCOMM, and Sigma Xi.

Abdul Jabbar is a Ph.D. candidate in the department of Electrical Engineering and Computer Science at The University of Kansas. He received the B.S. degree in Electrical Engineering from Osmania University (India in 2001), and the M.S. degree in Electrical Engineering from The University of Kansas in 2004, for which he received the Moore award for best M.S. thesis. He is a graduate research assistant at the KU Information & Telecommunication Technology Center (ITTC). His research focus is on resilience strategies, mechanisms, and evaluation methodologies. His interests also include topology modeling and analysis, highly-dynamic networks, fixed-wireless technologies, and the future Internet. He is a member of IEEE Communications and Computer Societies and ACM SIGCOMM.

Justin P. Rohrer is a Ph.D. candidate in the department of Electrical Engineering and Computer Science at The University of Kansas. He received the B.S. degree in Electrical Engineering from Rensselaer Polytechnic Institute in 2004. He is a graduate research assistant at the KU Information & Telecommunication Technology Center (ITTC) and an ITTC Graduate Fellow from 2004–2006. He received the best paper award at the International Telemetering Conference in 2008. His research focus is on resilient and survivable transport protocols. His interests also include highly-dynamic mobile networks, simulating network disruptions, and developing the GpENI network testbed for the GENI program. Previous research has included weather disruption-tolerant mesh networks and free-space-optical metropolitan networks. He is a member of the IEEE Communications and Computer Societies, ACM SIGCOMM, and is an officer of the Kansas City section of the IEEE Computer Society.

Dr. Marcus Schöller is a research scientist at NEC Laboratories Europe, Germany. He received the diploma in computer science at University of Karlsruhe, Germany, in 2001 and his doctorate in engineering in 2006 on robustness and stability of programmable networks. Afterwards he held a postdoc position at Lancaster University, UK, focusing his research on autonomic networks and network resilience. Marcus is currently working on the EU FP7 projects ResumeNet, with a focus on future network architecture with resilience as a key property, and the EU FP7 BeFemto project, which investigates next generation LTE-A femtocell technologies and business opportunities. His interests also include network and system security, intrusion detection, self-organization of networks, future network architectures, mobile networks including mesh and opportunistic networks.

Dr. Paul Smith is a senior research associate at Lancaster University's Computing Department. He submitted his Ph.D. thesis in the area of programmable networking resource discovery in September 2003, and graduated in 1999 with an honours degree in Computer Science from Lancaster. Paul 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 mis-configurations. In particular, his work has focused on the rich set of challenges that face community-driven wireless mesh networks and how they can be tackled. He is currently working on an EU FP7 project called ResumeNet, which is investigating a framework and mechanisms for future Internet resilience.