A COMPLEX NETWORKS APPROACH TO DESIGNING RESILIENT SYSTEM-OF-SYSTEMS

A Dissertation
Presented to
The Academic Faculty

by

Huy T. Tran

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2015

A COMPLEX NETWORKS APPROACH TO DESIGNING RESILIENT SYSTEM-OF-SYSTEMS

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Eric Feron
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Professor Jeff Shamma
School of Electrical and Computer Engineering
Georgia Institute of Technology (on leave)

Dr. Jean Charles Domerçant
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: September 25th 2015

ACKNOWLEDGEMENTS

I want to thank Professor Dimitri Mavris for providing guidance throughout this entire Ph.D. process. His technical and academic insights have been crucial to the development of this thesis and my own abilities. Perhaps more importantly though, his advice and focus on achieving a higher level of critical thinking has helped me to become a much more confident and capable researcher and individual. I would also like to thank Dr. Charles Domerçant, for spending countless hours meeting with me, helping me sort through my jumbled ideas, and providing suggestions for how to improve my thesis. Thanks to my other committee members as well, Professor Eric Feron, Professor Daniel Schrage, and Professor Jeff Shamma, for their thoughts and suggestions regarding my thesis.

Last but not least, thanks to all of my friends and family for supporting me through this journey from beginning to end. Thanks to the many friends I've made at Georgia Tech, who have provided countless laughs and outlets from work, in addition to being available whenever I needed someone to bounce ideas off of. Thanks for my brother for being there to support me whenever I needed an escape from working. Thanks to my dad for pushing me academically and intellectually, and showing me the benefits of hard work and perseverance. Thanks to my mom for always believing in me, supporting me, and pushing me to reach my goals. And finally, thanks to my fiancé, Katie, for never failing to be there for me in the most difficult and most enjoyable parts of this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

System-of-systems (SoS) are becoming increasingly networked in an effort to provide new capabilities and functions not possible with individual systems. However, this connectivity also introduces new vulnerabilities to SoS that must be considered. This thesis addresses this problem by investigating the effects of node failures on SoS networks, and identifying methods for mitigating the effects of those failures.

Two methods to account for potential SoS network threats are designing for robustness and designing for resilience. Robustness is a traditional design method, focused on designing systems that are insensitive to perturbations in operating conditions. This is a passive method, aiming to preemptively handle uncertainty through careful selection of system design parameters. There is growing interest in taking an alternative approach to handling uncertainty and potential failures, namely resilience. Resilience focuses on designing a system that can maintain or recover desired capabilities following a disturbance or threat event. This method takes an active approach to potential threats, focusing on the ability of a system to adapt to threats. This thesis hypothesizes that for SoS networks, resilience is a more cost-effective method than robustness for handling complex operating environments and unexpected threats.

Based on the hypothesis that SoS networks should be designed to be resilient, a methodology is developed for designing resilient SoS networks. This methodology includes a capability-based resilience assessment framework, used to quantify SoS resilience. A complex networks approach is used to generate potential SoS network designs, focusing on scale-free and random network topologies, degree-based and random rewiring adaptation, and targeted and random node removal threats. Statistical design methods, specifically response surface methodology, are used to evaluate SoS networks and provide an understanding of the advantages and disadvantages of potential designs. This analysis focuses on main factor effects and interactions. Linear regression is then used to model a continuous representation of the network design space, and determine optimally resilient networks for particular threat types.

The methodology is applied to an information exchange (IE) network model to demonstrate its use and identify resilient IE networks. IE networks provide a fundamental representation of important characteristics and processes occurring on SoS networks. Results show that optimally resilient network topologies are random for networks with adaptation, regardless of the threat type. However, the optimally resilient adaptation method sharply transitions from being fully random to fully degree-based as threat randomness increases.

Cost-benefit analysis of resilient and robust SoS networks is performed to test the hypothesis that a resilience-based approach is more appropriate for SoS than a robustness-based approach. A military command and control (C2) application is used for this analysis, due to the need for C2 networks that are resilient in the face of evolving threats and uncertain operating environments. This analysis identifies conditions within which resilient C2 networks are more cost-efficient than robust ones, based on the cost of rewiring network links relative to creating new links.

The primary contributions of this thesis are threefold: (1) a methodology for designing resilient SoS networks that provides a quantitative and exhaustive method for exploring and optimizing SoS network resilience, (2) an understanding of how IE networks should be designed for resilience, with respect to their initial topology and adaptation method, and (3) cost-benefit analysis comparing resilient and robust C2 networks showing that the most cost-efficient approach depends on the ratio of rewired link costs to new link costs.

CHAPTER I

INTRODUCTION TO SOS NETWORKS AND RESILIENCE

We are becoming increasingly reliant on systems that are networked together to provide services used in everyday life. The connectivity among these systems enables them to provide new capabilities and functions not previously possible. These types of integrated systems are called system-of-systems (SoS). Examples of SoS exist in many different domains, including intelligently networked devices, critical infrastructures, and networked military forces.

While networks are critical to the operation of many SoS, they also introduce vulnerabilities to those SoS. As SoS become more connected, the need to understand these vulnerabilities and the impacts they may have on SoS performance grows. This thesis aims to address this problem by investigating SoS vulnerabilities to network threats and ways to mitigate the effects of those threats.

This chapter motivates the need for SoS research, defines SoS and describes the role networks play in SoS, discusses challenges faced by SoS networks, and identifies methods to address challenges related to potential network threats. Two overarching research questions for this thesis are established from observations of SoS network challenges, and research objectives formulated to investigate those questions.

1.1 The Need for SoS Research

Advanced information and communications technologies have increased connectivity between systems with civil and military applications. For example, the continued development of Internet-capable devices has created a global market focused on the

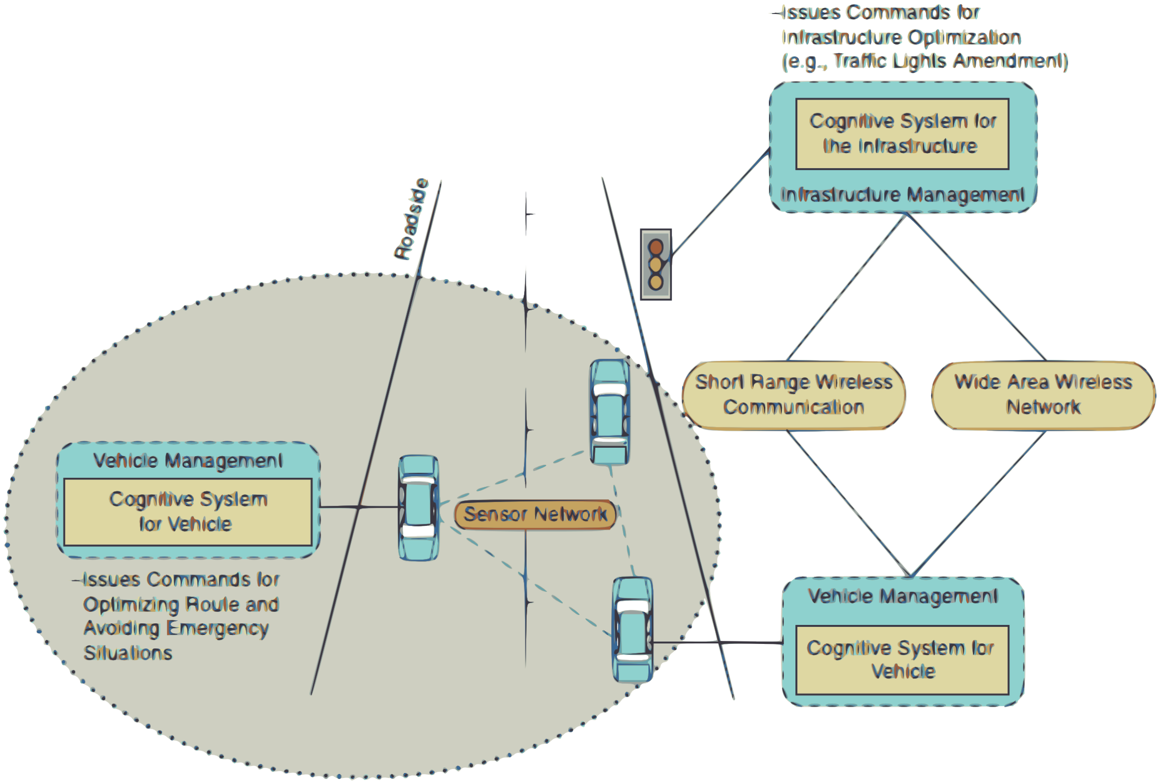

–Issues Commands for
Infrastructure Optimization
(e.g., Traffic Lights Amendment)
Cognitive System for the Infrastructure
Infrastructure Management
Roadside
Short Range Wireless Communication
Wide Area Wireless Network
Vehicle Management
Cognitive System for Vehicle
Sensor Network
–Issues Commands for
Optimizing Route and
Avoiding Emergency
Situations
Vehicle Management
Cognitive System for Vehicle

Figure 1: High-level view of functionality in the ITS [36].

Internet of Things (IoT). The IoT is based on the idea of connecting systems together (e.g. building, vehicles, mobile phones) so they can interact with each other to provide new benefits to individuals, organizations, and societies. Potential applications include health monitoring, domotics (intelligent homes), and manufacturing [16]. Another example of networked civil systems is the Intelligent Transportation System (ITS). The ITS program "aims to bring connectivity to transportation through the application of advanced wireless technology" in an effort to reduce traffic accidents and improve the efficiency of transportation systems. The vision of the ITS program is based on a connected vehicle environment, where vehicles, the infrastructure, and mobile devices are connected to each other [7].

Many military operations also utilize networked systems to achieve information superiority over constantly evolving threats. This transition towards a connected military force stems from the concepts of network-centric warfare (NCW) and network enabled capability (NEC) [13, 29, 5]. NCW and NEC aim to support and improve future military operations by "networking sensors, decision makers, and shooters..." [13]. These military SoS rely on technologies such as the United States (US) Army's Capability Set 13, which allows soldiers to connect to a mobile communications network, enhancing the connectedness of US forces and the situational awareness of soldiers. For example, capability Set 13 enables US troops still deployed in Afghanistan to maintain tactical communications while much of the current communications infrastructure is being removed from service [54]. Another technology enabling connectivity is the Nett Warrior system. This system uses smart phones as an "end-user device" to provide soldiers with a system for accessing and using available networks [37].

The proliferation of information and communications technologies has created highly connected SoS able to provide capabilities and functions not previously available. However, increased reliance on connectivity within SoS creates a need to improve our understanding of the role networks play in SoS, and potential vulnerabilities they may introduce to those SoS.

1.2 Defining Characteristics of SoS

It is important to define what an SoS is, specifically differentiating between SoS and systems. The Department of Defense (DoD) defines a system as "a functionally, physically, and/or behaviorally related group of regularly interacting or interdependent elements; that group of elements forming a unified whole" [90]. This definition, and others [2, 3, 74], focuses on a system being a collection of elements operating together to perform a desired function. Systems engineering (SE) focuses on the development of systems with consideration of the technical and managerial aspects of their entire life cycle, from conception to operation [6].

Though SE has provided valuable support in the design and analysis of past systems, future demands require a transition from focusing on the engineering of *individual* systems to the engineering of *multiple* systems integrated into an SoS. With respect to this new type of engineering challenge, "there seems to be little argument concerning present shortcomings in our abilities to deal with difficulties generated by increasingly complex and interrelated systems of systems" [65]. As such, a field of study addressing the engineering of SoS has emerged.

SoS engineering (SoSE) is still in an early stage of development, resulting in many proposed definitions of an SoS. Descriptions of SoS can be found as early as 1956 [50], with the modern view of SoS being introduced by Eisner, Marciniak, and McMillan in 1991 [42] and Shenhar in 1994 [101]. Maier provides a popular definition of an SoS as "an assemblage of components which individually may be regarded as systems, and which possesses two additional properties: operational independence of the components and managerial independence of the components" [74]. A more recent definition provided by Jamshidi emphasizes the networked aspect of SoS. This definition will be used for the purposes of this thesis:

> **Definition:** ...***systems of systems*** are large-scale integrated systems which are heterogeneous and independently operable on their own, but are *networked* together for a common goal - Jamshidi [60].

The Systems Engineering Guide for Systems of Systems summarizes major developments in the field and characteristics of SoS [90]. Commonly identified elements distinguishing an SoS from a system include the following [66]:

- Operational independence of elements - decomposition of an SoS into constituent systems would not render the constituent system inoperable.

- Managerial independence of elements - constituent systems that make up SoS can be separately acquired and integrated with the managerial function of the constituent systems remaining independent of the SoS.

- Evolutionary development - once an SoS is enabled, changes have to be made to the SoS as more knowledge is acquired and circumstances shift.
- Emergent behavior - properties and behaviors of an SoS develop from the interaction of system elements over time. These properties and events cannot be predicted or understood from the properties of single elements in the SoS.
- Geographical distribution - SoS are large geographically distributed assemblages.
- Interoperability - individual systems are typically designed in isolation, requiring the consideration of interoperability between systems.
- Complementarity - each system should complement the other systems within the SoS.
- Holism - the whole is the primary and often greater than the sum of its parts.

These characteristics of SoS show several differences between an SoS and a system, notably the inclusion of multiple systems (which themselves may be complex) in an SoS and the independence of those individual systems. These differences result in several distinctions between SE and SoSE, summarized in table 1 (aspects of SoSE focused on within this thesis are highlighted). The following observation identifies unique characteristics of SoS that are particularly relevant to this thesis:

> **Observation:** SoS are composed of multiple systems *networked* together to achieve a *capability* not possible with a single system. Complex interactions between systems often result in *emergent behavior* and require the consideration of *interoperability costs.*

Table 1: Major Drivers of SE and SoSE [50]

	SE	SoSE
Focus	Single complex system	**Multiple integrated complex systems**
Objective	Optimization	Satisficing, sustainment
Boundaries	Static	Dynamic
Problem	Defined	**Emergent**
Structure	Hierarchical	**Network**
Goals	Unitary	Pluralistic
Approach	Process	Methodology
Timeframe	System life cycle	Continuous
Centricity	Platform	**Network**
Tools	Many	Few

1.3 Representing SoS as Information Exchange Networks

Many SoS researchers have established the networked nature of SoS. Some researchers explicitly include the term “network” in their definition or description of SoS [23, 60, 101]. Other researchers investigate SoS by viewing them as a network [34, 44, 46, 50, 74]. Examples of network approaches to SoS include an SoS management framework based on network management principles [50], an SoS approach for analyzing civil transportation networks [34], and the application of SoS methods towards the study of critical infrastructure (CI) network interdependencies and resilience [44, 46].

Military organizations have also recognized the role that networks play in military SoS, resulting in the establishment of many network research efforts, such as the Network Science center at the US Military Academy at West Point. These efforts seek to improve the effectiveness of the US military through an understanding of how networks operate and how they can be used within military SoS. For example, the command and control (C2) of complex military operations can often be viewed as an SoS composed of systems such as unmanned aerial vehicles (UAVs), satellites, and data centers, as shown in fig. 2. Representing this SoS as a network allows analysts to focus on the overall structure of the SoS and the ability of systems to communicate necessary information throughout a mission.

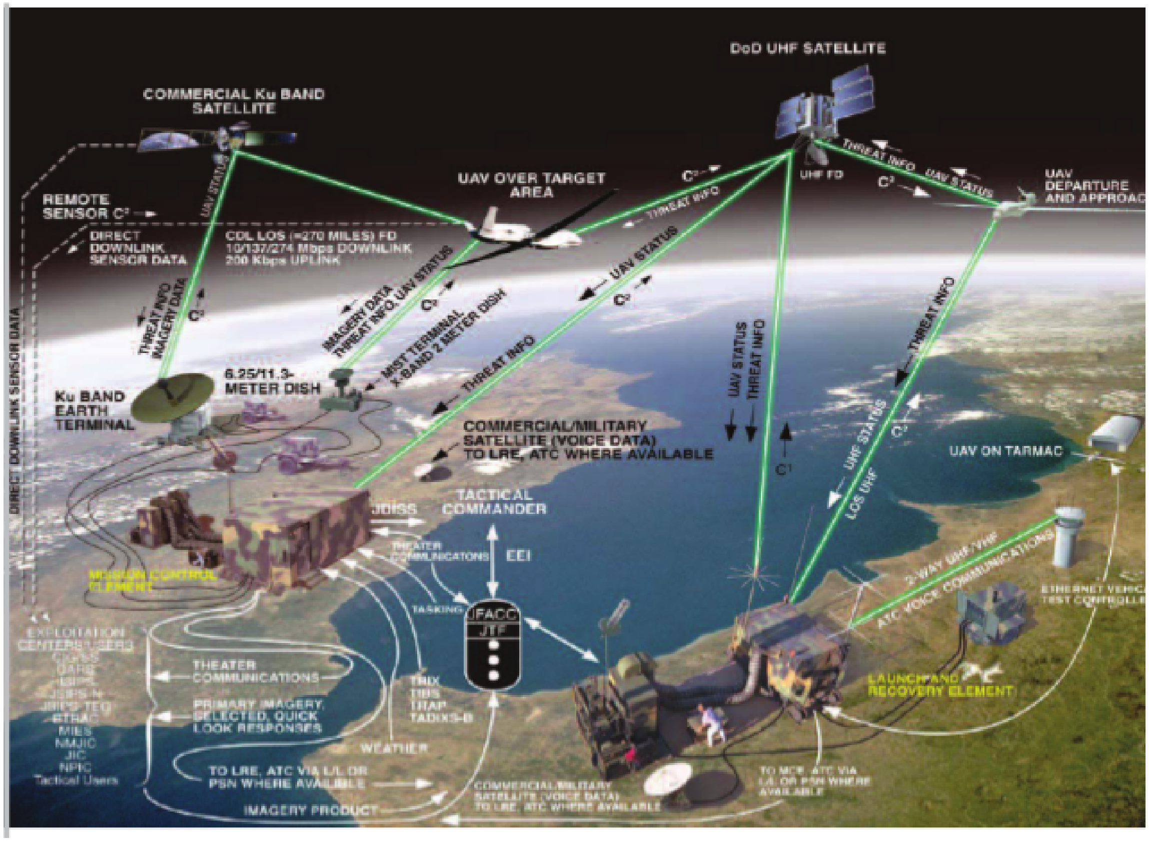


Figure 2: Military SoS network example, composed of several individual systems connected together to share information and coordinate necessary activities [1].

This thesis takes a network approach to the design of SoS, focusing on the structure of an SoS network and interactions between individual systems. Network nodes are defined to represent systems within an SoS, with links between nodes representing connections between systems. Examples of possible network connections include:

- Physical: Internet, transportation, power grid, communication, sensors
- Information: social, organizational, financial
- Decision: command structure, control architectures, organizational

Focusing on SoS networks, no emphasis is placed on the design of individual systems. Rather, this thesis focuses on how those individual systems should interact with each other to provide desired overall SoS capabilities. Since SoS show evolutionary development, this thesis also considers how network structures defining those interactions should evolve over time.

Since many of the discussed SoS networks enable some sort of information exchange among systems, this thesis uses a message passing network as an example of an SoS network. Examples of information exchange (IE) SoS networks include Internet-enabled smart devices used to improve the efficiency of everyday activities, large-scale enterprise organizations using technological networks to distribute information, and military C2 networks sharing necessary information to develop situational awareness. These information-centric SoS capture the fundamental nature of many SoS networks. Nodes in an IE network represent systems, with links representing data or message passing links between those systems (e.g., professional relationships between individuals in an enterprise organization or communications links between military systems).

1.4 SoS Network Challenges

Networks are essential for the operation of many SoS, particularly those relying on connectivity to enable information exchange critical to their functionality. However, the increasing *complexity and uncertainty* of SoS threats makes it difficult to provide uninterrupted network connectivity. This difficulty is especially seen in the military domain, where military missions have transitioned from conventional to asymmetric warfare. Conventional warfare typically involved long, multi-year missions that gave military forces time to design systems for the mission at hand. In contrast, asymmetric warfare is characterized by quickly evolving threats that are "unpredictable and unprecedented". Improvised explosive devices (IEDs) are an example of an adversary changing its tactics to create a new threat to soldiers [81]. These unpredictable threats can result in system failures and unacceptable performance at the SoS level

[32]. The uncertainty surrounding future threats makes it difficult to predict future needs of military systems [33]. This problem is made even more difficult by the fact that increasingly stringent military budgets require SoS to be cost efficient, as well as effective [81]. Therefore, there is a need to design SoS able handle a variety of threats, and in the case of an unanticipated threat, be able to recover from it.

Potential threats to SoS networks create a need to understand the effects of system (i.e., node) failures on the ability of SoS networks to function. However, simply understanding these impacts is not enough. That understanding should then be used to identify methods for mitigating the effects of potential system failures. The National Research Council describes this need in its report on networks and the Army, claiming "there is a clear need to better understand and design networked systems that are both robust to variations in the components (including localized failures) and secure against hostile intent" [79]. This observed need results in the first overarching research question for this thesis (RQ 1):

> **Research Question 1:** What happens when SoS network nodes fail and how can we mitigate the effects of those failures?

Answering this question is a difficult task because of the uncertainty regarding potential threats, as well inherent complexities of SoS networks. One challenge to answering this question is the fact that SoS are often composed of a *mix of legacy and newly developed systems*. This integration of new and old systems may result in situations where systems are not being used for the purposes they were designed for [32]. These systems may then be operating outside of designed operating ranges, increasing their probability of failure. There is also a lack of data characterizing the performance of new systems, particularly when used in large-scale SoS. This lack of data "creates a large degree of uncertainty in the reliability of the overall SoS architecture in fulfilling the required capability needs" [33].

Another challenge associated with SoS networks is *interoperability*. Interoperability is a key enabler for the successful operation of SoS, allowing heterogeneous systems to link together and communicate with each other. However, providing high levels of system interoperability and connectivity requires large investments in system acquisition and operation [35]. The cost of high interoperability requirements creates a difficult challenge for SoS designers, as they must balance the benefits of increased system connectivity with the costs of providing that level of interoperability. The following observation summarizes challenges to designing SoS networks and answering research question one:

> **Observation:** Designing SoS networks able to mitigate the effects of potential node failures is challenging, because SoS networks face *uncertain and evolving threats*, are composed of *legacy and new systems*, and require consideration of system *interoperability costs.*

1.5 Mitigating SoS Network Threats with Resilience

A literature review is performed to identify methods for designing SoS networks capable of mitigating potential network threats. These methods should be cost effective because of cost implications related the use of legacy and new systems in SoS, as well as subsequent interoperability issues for those systems. This review seeks to answer the second overarching research question for this thesis (RQ 2):

> **Research Question 2:** What is the most cost effective method for designing SoS networks that can mitigate the effects of potential network threats?

Two promising methods for addressing this issue are designing for *robustness* and designing for *resilience*. These methods focus on designing a system, or SoS, to provide desired capabilities in a wide range of conditions, in spite of potential component or system failures.

A traditional method for handling potential threats is to design a system that is *robust* to perturbations in the operating environment. Robustness can be defined in the context of SoS as the following:

> **Definition:** ***Robustness***, in the context of an SoS here, is the reduced sensitivity of SoS performance to variations in individual system performances that could potentially generate cascading effects across an SoS network [33].

A similar definition of robustness claims that a system property or capability is robust if it is invariant to perturbations [15]. Traditional robust design uses probabilistic methods to identify design parameter settings that make a system insensitive to noise factors. Various techniques such as Taguchi arrays and robust parameter design support this approach [78]. Robustness can also be achieved by overdesigning a system to reduce its probability of failure [15, 81]. This process may involve the use of high reliability components or materials to ensure performance and reduce system uncertainty. Excessively high design requirements can also be established to reduce failure probabilities. Another common method of achieving robustness is through system or functional redundancy [15]. Redundancy uses backup systems or parallel functional paths to maintain capabilities in spite of system failures.

Despite a long history of designing for robustness, there are several disadvantages to taking such an approach for the design of SoS. For one, overdesign and redundancy are no longer affordable for organizations facing increasingly strict budgets and pressure for cost efficiency [81]. This is especially true when considering SoS, due to the scale and breadth of systems used in an SoS [109]. The use of legacy systems also makes it difficult to incorporate robust design, since improving previous designs may require lengthy and expensive retrofits to previously acquired and active systems. The independence of systems within an SoS also limits the control SoS designers have over the development of component systems [32]. These observations regarding the applicability of robust design for SoS are summarized as follows:

> **Observation:** Designing for robustness is not as well-suited for SoS applications as it is for systems design, due to the use of legacy systems and inability to control the design of new systems used within SoS. Additionally, unknowns threats, complex emergent behaviors, and cost restrictions make it difficult to limit system failure probabilities.

Instead of focusing on how to design an SoS that is insensitive to failure, a better approach may be to assume that at some point a failure will occur. In fact, some researchers acknowledge that eventual failures within an SoS are unavoidable, regardless of preparations taken to limit such events, due to their complexities and emergent behaviors [73]. Therefore, designers should instead focus on how an SoS will adapt to failures, while using remaining operational systems. This approach focuses on how to improve the ways in which currently available systems are used within an SoS, rather than improving the design of the individual systems themselves. Designing for resilience focuses on the ability to adapt to and recover from potential failures or disruptions.

Resilience is an active method of handling potential threats, focusing on adaptability and capability recovery. In comparison, robustness is a passive method focused on preemptively integrating defensive measures into a system to prevent or mitigate the effects of anticipated threats. Neches defines a resilient system as being "effective in a wide range of situations, readily adaptable to others through reconfiguration or replacement, with graceful and detectable degradation of function" [80]. This definition and others from the literature [53, 73, 104, 112, 81] identify several characteristics of resilient systems:

- the ability to provide desired capabilities in a variety of conditions

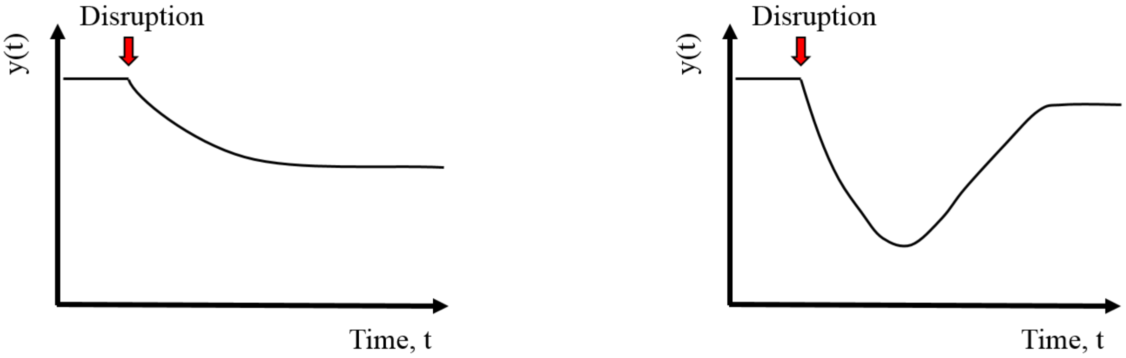
- graceful and detectable degradation of function or capability when faced with a disruption
- the ability to maintain or recover degraded capabilities when faced with anticipated and unexpected disruptions in a timely manner
- the ability to adapt to evolving threats and operating conditions, often through reconfiguration and replacement
- affordable and effective performance

Based on these characteristics, the following definition of resilience (within the context of SoS) is provided for the purposes of this thesis:

> **Definition:** ***Resilience*** is the ability to *maintain or recover desired capabilities in a timely manner* when faced with a threat or disturbance, through well-informed design and *adaptation*.

Since this definition focuses on the ability to maintain desired capabilities, clarification is given regarding the difference between SoS capability and performance. An SoS capability is defined to be a specific function that an SoS is desired to be able to provide. SoS capability is time invariant if no changes to the SoS occur. For example, an IE network may have the capability to transfer C messages per second (i.e., a throughput rate of C data packets per second). This capability does not change if no changes are made to the network.

In comparison, SoS performance is defined as a time-varying measure of how well an SoS is achieving a desired capability. Returning to the IE network example, though the network is capable of passing C messages per second, at any given time instant t, the actual observed message transfer rate of the network (i.e., its performance) is some value $y(t)$ that may slightly differ from C. Figure 3 shows a notional comparison of SoS performance over time following a single disruption event, for robust and resilient



(a) (b)

Figure 3: Notional comparison of performance for a robust SoS (a) and a resilient SoS (b), where $y(t)$ is the SoS performance level at time t (e.g., the number of successfully transfered messages in an IE network). The robust design is more insensitive to the disruption than the resilient design. However, the resilient design is able to adapt to the disruption, allowing it to recover nearly all lost performance levels.

SoS designs. The SoS capability for this example would be the initial performance level seen before the disruption.

Sterbenz et al. describe the need for resilience, claiming that "resilience must be viewed as an essential design and operational characteristic of future networks in general, and the Global Internet in particular" [104]. The DoD has also recognized the importance of resilience, creating an Engineered Resilient Systems (ERS) initiative focused on developing methods and tools for designing and analyzing resilient systems [80, 57, 81]. Neches and Madni describe DoD systems as being "called upon to perform increasingly complex missions in a variety of operational environments. They need to be rapidly fieldable, affordably adaptable, and effective" [81]. These three design goals, *affordable*, *adaptable*, and *effective*, form the basis of the ERS initiative and provide guidelines for the design of resilient systems. Uday and Marais also argue for a focus on SoS resilience, specifically noting the difficulties of a reliability-based approach for SoS [109].

A review of the literature identifies designing for resilience as a promising method to handle potential threats faced by SoS networks, while accounting for inherent difficulties associated with SoS design. Based on this review, the following hypothesis is formed in response to research question two:

> **Hypothesis 2:** Resilience is a more cost effective method than robustness for designing SoS networks able to mitigate potential threats, due to a focus on adaption within resilience.

1.6 Summary and Research Objectives

Increased connectivity among systems has created networks of systems, or SoS, able to deliver more capabilities and functionality than individual systems. However, the dependence of SoS on networks has created a need to understand what happens when network nodes fail, and develop methods to design SoS networks able to mitigate the effects of those failures. This need is reflected in research question one, the first overarching research question for this thesis.

Two methods for designing such SoS networks are designing for robustness and resilience. Robustness is a passive approach to handling threats and failures, while resilience is a more active approach to this problem. Research question two is concerned with which approach is best suited for SoS networks. This thesis hypothesizes that resilience is a more cost effective approach than robustness for SoS networks, due to challenging aspects of SoS and the focus on adaptation within resilience (HYP 2).

Three overall research objectives are defined for this thesis. When completed, these objectives should answer the two overarching research questions for this thesis (RQs 1 and 2), and test hypothesis two. The first research objective (RO 1) is defined as follows:

> **Research Objective 1:** Develop a methodology for assessing and designing resilient SoS networks.

The developed methodology should provide guidelines for quantitatively assessing resilience, defining potential SoS network designs, and comparing the resilience of those networks. The methodology should also consider network adaptation, due to the focus on adaptation within resilience. Therefore, the methodology must meet the following requirements to satisfy research objective one:

- Assessments of resilience should be *quantitative and capability-based.*
- Design alternatives and threats should focus on the *networked* nature of SoS (i.e., they should focus on network topologies and network threats)
- *Network adaptation* should be considered as a mechanism for mitigating potential threats.

The second research objective (RO 2) is to use the developed methodology to answer research question one, and is defined as follows:

> **Research Objective 2:** Use the developed methodology to determine the impact of node failures on SoS networks and identify network designs that are resilient to potential threats.

The third research objective (RO 3) is to answer research question two and test hypothesis two, by comparing resilient and robust SoS network designs as follows:

> **Research Objective 3:** Perform cost-benefit analysis on resilient and robust SoS network designs, to compare resilience-based and robustness-based approaches to designing SoS networks.

The primary contributions of this thesis are threefold. First, a methodology is developed for designing resilient SoS networks that expands on previously existing methods for evaluating resilience and exploring adaptive network designs. Second, the methodology is used to investigate and optimize the resilience of adaptive IE networks facing potential network threats, contributing to our understanding of how network adaptation can be used to improve resilience. Third, cost-benefit analysis of resilient and robust military SoS networks identifies scenarios in which one approach may be preferred over another, advancing our knowledge of the benefits of resilience relative to robustness.

Chapter 2 presents a literature review of relevant work that has been done to address the design of resilient SoS networks. This review is used to identify significant gaps in the literature, as well as provide background knowledge of relevant concepts and methods. Chapter 3 provides an overview of the methodology developed by this thesis. Chapter 4 describes a framework for assessing system resilience, used in the first step of the developed methodology. Chapter 5 describes a complex networks approach for generating SoS network alternatives, used in the second step of the methodology. Chapter 6 describes the use of statistical design methods for evaluating SoS network alternatives, used in the third and final step of the methodology. Chapter 7 uses the developed methodology to explore and optimize the space of potential SoS network designs. Chapter 8 performs a cost-benefit analysis comparing a resilient SoS network design to robust designs. Chapter 9 summarizes the developed methodology, results from implementation of the methodology, contributions of this thesis, and potential extensions of this work.

1.7 Representative Test Problem: IE Network Model

An IE network model is created as a test problem to provide a platform for testing various steps in the methodology as it is developed. This test problem focuses on message passing within a network, due to the importance of information exchange in SoS networks. The model is based on a message passing network model used to study the robustness of organizational networks [38] (discussed in section 2.3.3), and is adapted to consider network adaptation as a means for resilience.

The model represent systems within an SoS as network nodes, and communication paths between systems as network links. Nodes are defined as active or inactive to represent system failures or attacks. All nodes begin as active in a simulation. Nodes become inactive and are removed from the network if they fail or are attacked throughout a simulation. Links connected to inactive nodes are also removed.

The model represents information exchange between systems through message passing between nodes. Every active node has a probability, μ, of generating a new message at every time step in a simulation of the model (i.e., μ defines the message generation rate of a node). The message generation rate of the entire network is defined by μN, where N is the size of the network. Every generated message, M_{ij}, has a source node i, the node that created the message, and a target node j, the node that the source node will try to send the message to. Messages represent information required by the target node to perform its necessary functions. For example, a message may contain the location of a downed soldier in a military search-and-rescue scenario. The target node in this example may be a helicopter tasked with retrieving the soldier. The message generation rate defines how demanding the network task environment is. High values of μ represent networks that generate and require a large amount of information exchange to properly function. Low values of μ represent networks that generate a small amount of information and can perform well with limited information exchange.

The model is simulated from an initial simulation time, t_0, to a final time, t_{final}. Simulations progress in discrete time steps, $t_0, t_1, \ldots, t_{final}$, where each step is arbitrarily defined to represent one second in time. Time steps can be defined to represent any time length desired. The message generation rate is adjusted throughout a simulation to ensure that the expected number of messages created each time step by an entire network stays constant as nodes are removed. The message generation rate at time t, μ_t, is calculated as

$$\mu_t = \frac{\mu_0 N_0}{N_t}, \tag{1}$$

where μ_0 is the initial message generation rate, N_0 is the initial number of active nodes (i.e., the initial network size), and N_t is the number of active nodes at time t.

Message targets are selected uniformly at random from the set of active nodes in the network excluding the source node, such that the probability of selecting node j as a target, $P(j)$, is

$$P(j) = \frac{1}{N_t - 1}. \tag{2}$$

Message target selection can be adapted to give preference to certain nodes in the network using a weighting function, such as the softmax function. The softmax function is a generalization of the logistic function, and is commonly used in game theory for determining the probability that a player selects a given action based on its expected reward. Using this function, the probability of selecting node j as a target, $P(j)$, can be defined as

$$P(j) = \frac{\exp\left(d_{ij}/\tau\right)}{\sum_{k=1}^{N_t} \exp\left(d_{ik}/\tau\right)}, \tag{3}$$

where d_{ij} is the geodesic distance between the source node i and possible target node j and τ is the temperature parameter. Variation of the temperature parameter defines how far information must generally travel in the network to maintain system functionality. High temperatures ($\tau \to \infty$) would give all nodes the same probability of being selected (i.e., target selection is uniformly random), representing networks where information does not have to travel far in the network. Low temperatures ($\tau \to 0^+$) would give preference to nodes far from the source node, representing networks where information has to travel far in the network.

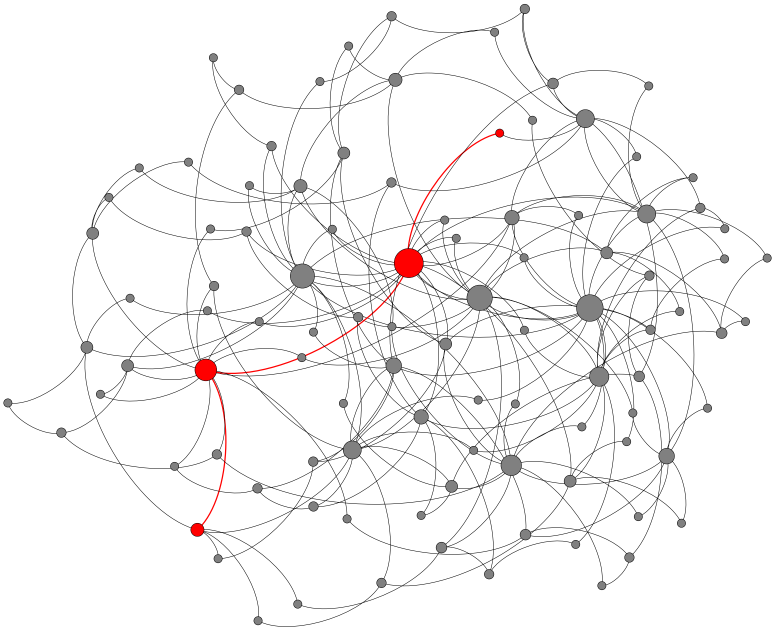


Figure 4: Example of message sending in a network with $N = 100$ nodes. The source node, target node, and shortest path between those nodes are highlighted. Node sizes are scaled by degree (i.e., number of incident links).

Messages are sent from their source node to their target node using the shortest path in the network, as shown in fig. 4. Each time step in a simulation of the model, every message is forwarded from its current node position to the next neighboring node in the shortest path from it to the target node of the message. Nodes are assumed to have complete knowledge of the current network topology and set of active nodes, enabling them to determine the shortest path to a target node at any given time. A message is lost if no path exists between the source node and target node, the target node becomes inactive before receiving the message, or a node becomes inactive while holding the message.

The capability, C, of a network is defined by its ability to receive messages over time, such that

$$C = \mu \times N \times V, \tag{4}$$

where $\mu \times N$ is the expected number of messages generated by the network in a given time step and V is the value of each message (assuming all messages have the same value). The expected number of generated messages determines an IE network's capability because the purpose of an IE network is to enable information exchange through message passing. Therefore, the more messages a network is expected to generate, and ultimately receive, the more capable the network is. This capability definition assumes that messages are eventually able to reach their target location in a normal operating scenario.

The performance, $y(t)$, of a network (i.e., the network's ability to provide its desired capability) is then defined as the total number of messages actually received at time t, such that

$$y(t) = \sum_{i=1}^{N_t} R^i(t), \tag{5}$$

where $R^i(t)$ is the number of messages received by node i at time t.

Time sensitivity is also considered in determining IE network performance, through a time sensitivity parameter Δ, where $0 \leq \Delta \leq 1$. For some networks, the time taken to receive a message does not affect the value, or usefulness, of the message. These networks are referred to as time insensitive IE networks. However, most SoS networks are used to exchange information that is time sensitive. Returning to the search-and-rescue example, if the downed soldier is in a hostile environment, the time taken to rely a message containing the location of the soldier to a rescue helicopter will strongly affect the benefit gained from receiving that message. Information exchange in an enterprise organization is also likely to be time sensitive. Many organizations depend on up to date information to make decisions regarding day-to-day processes. If required information takes too long to arrive, decisions may be made using inaccurate or irrelevant data.

IE network performance, $y(t)$, is adjusted to account for time sensitivity by defining the value of a received message as Δ^d, where Δ is a time sensitivity parameter and d is the travel time for the message (i.e., the time between when the message was created and received). The performance of an entire network is then

$$y(t) = \sum_{i=1}^{N_t} \sum_{j=1}^{R^i(t)} \Delta^{d^i_j}, \tag{6}$$

where d^i_j is the travel time for the jth message received by the ith node at time t. Since messages travel one "hop" in the network at each time step, message travel time is equivalent to the geodesic distance between the source and target node of a message if no changes occur to the network while the message travels. Setting $\Delta = 1$ makes eq. (6) equivalent to eq. (5). Decreasing Δ increases the importance of message travel time, i.e., decreasing Δ increases message time sensitivity. Therefore, if $\Delta < 1$, messages with longer travel times contribute less to network performance than those with short travel times. Network capability, C, is also adjusted for message travel time by defining the value of a message to be $V = \Delta^{\langle d \rangle}$, where $\langle d \rangle$ is the network average path length (i.e., the expected travel time of a message).

Figure 5 shows the effect of changing Δ on IE network performance, for various message travel times. For $\Delta = 1$, the network performance $y(t) = 100$ regardless of message travel times (there are 100 nodes in the network and each node receives one message at time t). As Δ is decreased from one, network performance decreases since each message has a travel time $d > 0$. For a given $\Delta < 1$, as message travel time increases, network performance decreases. Equation (6) therefore captures time sensitivity by reducing the value of messages as travel time increases. Table 2 summarizes important model parameters used by the IE network model.

This model of information exchange in a network is simple enough to be computationally inexpensive and scaled to large SoS networks, but includes enough dynamic processes to study fundamental behaviors of those networks. The model also captures the primary characteristics of SoS, and is therefore used as a representative test problem for investigating SoS network resilience. The following summarizes several

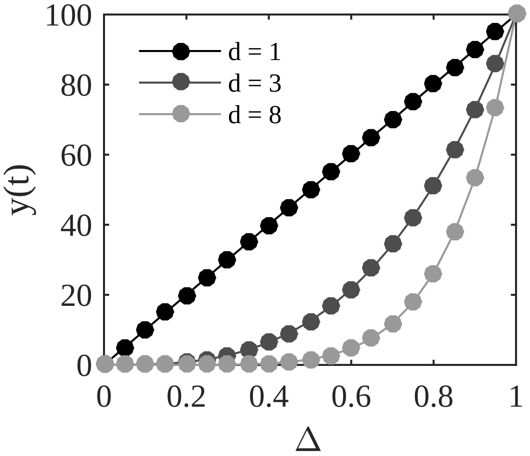


Figure 5: Effect of changing Δ on network performance, for a network with $N = 100$ nodes, where each node successfully receives one message at time t, and each message has a travel time specified by d.

Table 2: IE network model parameters

Parameter	Description
N	Initial network size
L	Initial number of links in a network
$[t_0\ t_{final}]$	Simulated scenario time interval (in seconds)
μ	Message generation rate
Δ	Time sensitivity

SoS characteristics found in the model:

- The network is composed of many individual systems (i.e., nodes) that interact with each other.
- These systems are networked together to quickly disseminate necessary information, providing a level of information exchange not possible with individual systems.
- Emergent behavior likely exists, due to potential complexities of the network structure, making it difficult for analysts to predict the behavior of the overall network from properties of individual systems.
- The structure of the network may evolve over time as systems become inactive.

CHAPTER II

LITERATURE REVIEW

This chapter reviews the literature to identify existing methods that may be used to design resilient SoS networks, which may provide an answer to research question two. This review also seeks to determine the current state of our understanding regarding network resilience, in an attempt to answer research question one. This review focuses on work from three research domains: SoSE, resilience engineering, and complex networks.

2.1 SoSE Methods for Resilience

Uday and Marais develop a method called stand-in redundancy to improve SoS resilience [109]. Their method assumes a hierarchical representation of an SoS, based on a functional decomposition of desired capabilities. The lowest level of this hierarchy represents the single-system functions within the SoS (e.g. a UAV providing imaging of an assigned surveillance area). The middle level represents multi-system capabilities, C. Multi-system capabilities are those achieved through the use of multiple single-system functions (e.g., imaging and target identification through the collaboration of a UAV and satellite). The highest level represents the overall SoS capability.

Using this functional decomposition of an SoS operation, the authors apply combinatorial optimization to re-task existing systems to missing functions. Following an attack or system failure, some previously satisfied functions may no longer have a system assigned to them. Re-assigning existing systems to cover those functions allows the SoS to continue its operation and provide its desired capabilities. This optimization is performed with respect to constraints on target levels of performance and reliability for each capability. Performance and reliability levels are calculated using estimates of system reliability and the conditional probability that systems are able to provide their functions given that they are fully operable.

Researchers have also developed methods to model SoS networks and account for the interdependencies within them. Filippini and Silva [46] provide a framework for analyzing SoS network resilience based functional dependencies among systems. Eusgeld, Nan, and Dietz discuss modeling alternatives for SoS, focusing on interdependent critical infrastructures [44]. Davendralingam and DeLaurentis develop a method for architecting an SoS that is robust to perturbations [33].

Limitations of SoSE Methods for Resilience

The stand-in redundancy approach to SoS resilience is quantitative and capability-based. The method is also adaptive, through re-tasking of component systems to necessary functions following system failures. However, this method assumes the existence of a central controller able to optimize and re-task every component system within an SoS network. SoS networks are often composed of independently operated and managed systems, making it difficult to provide a single entity with global control of the SoS network. This method also assumes that level of performance can be determined through probabilistic assumptions of system and function reliabilities. These reliabilities are not easy to obtain, especially given the many interdependencies and emergent behavior of SoS networks. A more thorough method would simulate the performance of the SoS and consider the impacts of the network structure on that performance.

Other work on SoS resilience is also limited in its applicability towards the research objectives for this thesis. The framework presented by Filippini and Silva has limited considerations of adaptation as a response to a disturbance. Work by Eusgeld et al. focuses on the representation and modeling of network interdependencies, but does not offer suggestions for how to account for those interdependencies and design resilient SoS. Work by Davendralingam and DeLaurentis focuses on SoS robustness rather than resilience.

2.2 Resilience Engineering Methods for Resilience

Resilience engineering is a growing field of research that has seen much progress in recent years. Interest in resilience as a concept of its own, separate from safety, reliability, and robustness, stems from the observation that today's operating environments are becoming increasingly complex and volatile. The uncertainty surrounding possible threats to critical systems requires system designers to develop systems that can not only absorb disturbances, but also adapt to them, effectively and affordably [80]. Several frameworks for resilience assessment and design exist in the literature. This section discusses the most relevant frameworks with respect to designing SoS networks.

A System Resilience Framework

Vugrin et al. propose a capability-based framework for assessing the resilience of critical infrastructure and economic systems [112]. Their framework consists of two main components: a quantitative method for measuring the impact of disturbances on performance and recovery costs, and a qualitative method for assessing important aspects of a system that determine its resilience.

The authors propose quantitative metrics for resilience using measurements of system performance over time. They consider resilience costs through measurement of recovery effort over time. Figure 6 graphically shows their interpretation of system resilience and recovery effort. Systemic impact, SI, is calculated as an integration-based metric,

$$SI = \int_{t0}^{tf} [TSP(t) - SP(t)]dt, \tag{7}$$

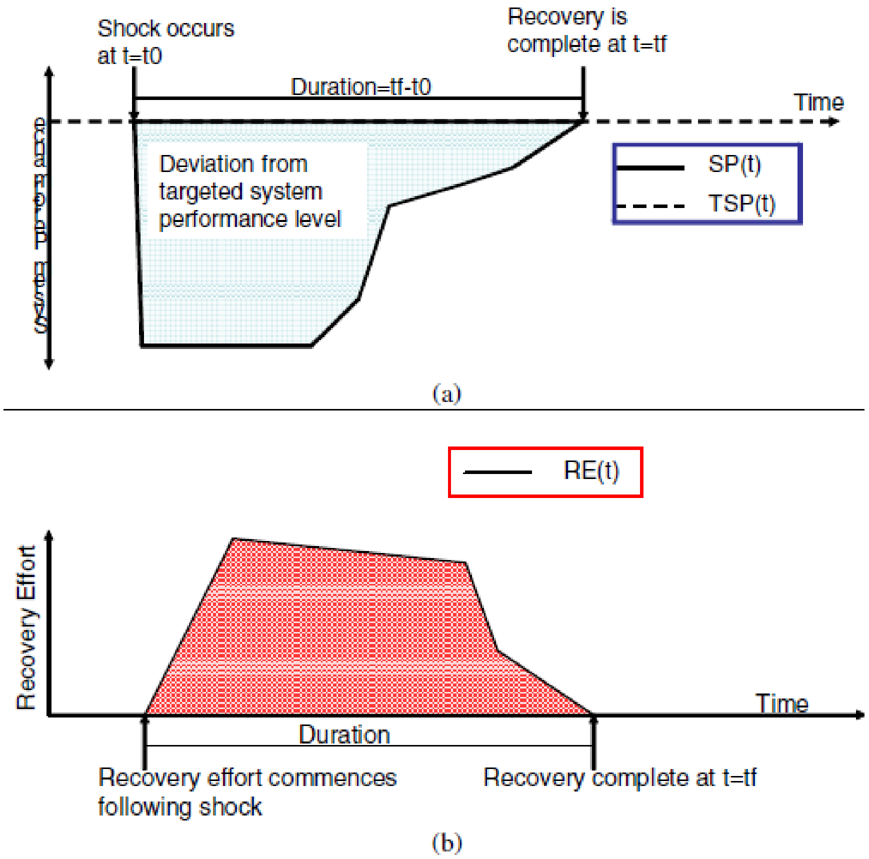


Figure 6: (a) System performance and (b) recovery effort as a function of time [112].

where TSP is the targeted system performance, SP is the system performance, $t0$ is the time of the disruption event, and tf is the time when system recovery is complete. Total recovery effort, TRE, is similarly calculated as

$$TRE = \int_{t0}^{tf} [RE(t)]dt, \tag{8}$$

where RE is the recovery effort. The authors propose two additional metrics that focus on a weighted sum of the systemic impact and total recovery effort.

Their framework identify three qualitative system capacities that determine system resilience: *absorptive capacity*, *adaptive capacity*, and *restorative capacity*. These capacities account for the dynamic nature of resilience and the role recovery speed plays in system resilience. The absorptive capacity “is the degree to which a system can automatically absorb the impacts of system perturbations and minimize consequences with little effort." This capacity is endogenous to the system. System redundancy is described as a method of improving a system's absorptive capacity. The adaptive capacity "is the degree to which the system is capable of self-organization for recovery of system performance levels." This capacity is an endogenous, dynamic capability of a system. System replacement is described as a method of improving a system's adaptive capacity. The restorative capacity "is the ability of a system to be repaired easily." System repairs are typically performed by an external entity, therefore making this capacity exogenous to a system. System health monitoring is described as a method of improving a system's restorative capacity.

The authors apply their framework to assess the resilience of critical infrastructure systems to an earthquake. However, they only show qualitative results describing the ability of emergency and postal/shipping services to handle an earthquake disruption.

Limitations of the System Resilience Framework

This framework addresses many of the desired aspects of resilience. However, the framework is limited in its ability to quantitatively compare the resilience of multiple system designs. Integrating system performance and cost with respect to time provides quantitative values for comparison, but makes it difficult to differentiate between systems that may have similar integrated values yet vastly different dynamic behaviors. For example, consider two systems designs, one that shows high initial performance degradation but quickly recovers nearly all of that performance over time, and another that shows no initial performance degradation but slowly loses performance over time (see fig. 7). Integration may assign similar values of resilience to these two systems, despite differences in their response to a disruption.

The concept of resilience capacities begins to address this issue, but defined in a qualitative way. Turnquist and Vugrin extend this framework with a stochastic

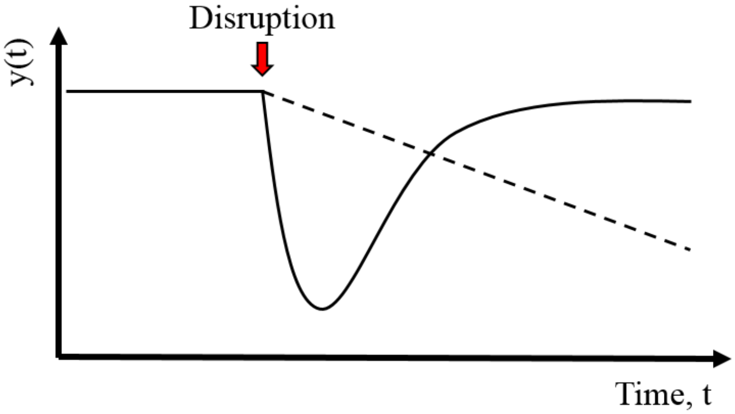


Figure 7: Notional comparison of system performance, $y(t)$, for two systems with different responses to a disruption. The recovering system (solid line) shows a large initial degradation in performance following the disruption, but is able to quickly recover most of the lost performance. The degrading system (dashed line) shows a slow, gradual decline in performance following the disruption, but is not able to recover any lost performance over time. These performance data show different responses to a disruption but have notionally similar integration values.

optimization model (applied to infrastructure distribution networks), but only consider three methods of improving resilience: increasing distribution center capacities (improving system capabilities), providing backup distribution centers (redundancy), and investing in faster recovery (improving system recovery capabilities) [108]. These methods focus on overdesigning systems and providing redundancy, ignoring potential adaptation methods.

TIRESIAS

TIRESIAS is a resilience framework proposed by Balchanos in his dissertation investigating complex dynamic system resilience [20]. The core of the framework is a set of capability-based metrics proposed for quantified resilience comparisons between potential system designs [20, 18, 19]. These metrics are used to provide a detailed characterization of the dynamic aspects of system performance following a disruption event. This characterization relies on measurements of system performance over time (see fig. 8), which can be used to compare the absorptive and restorative capacities of

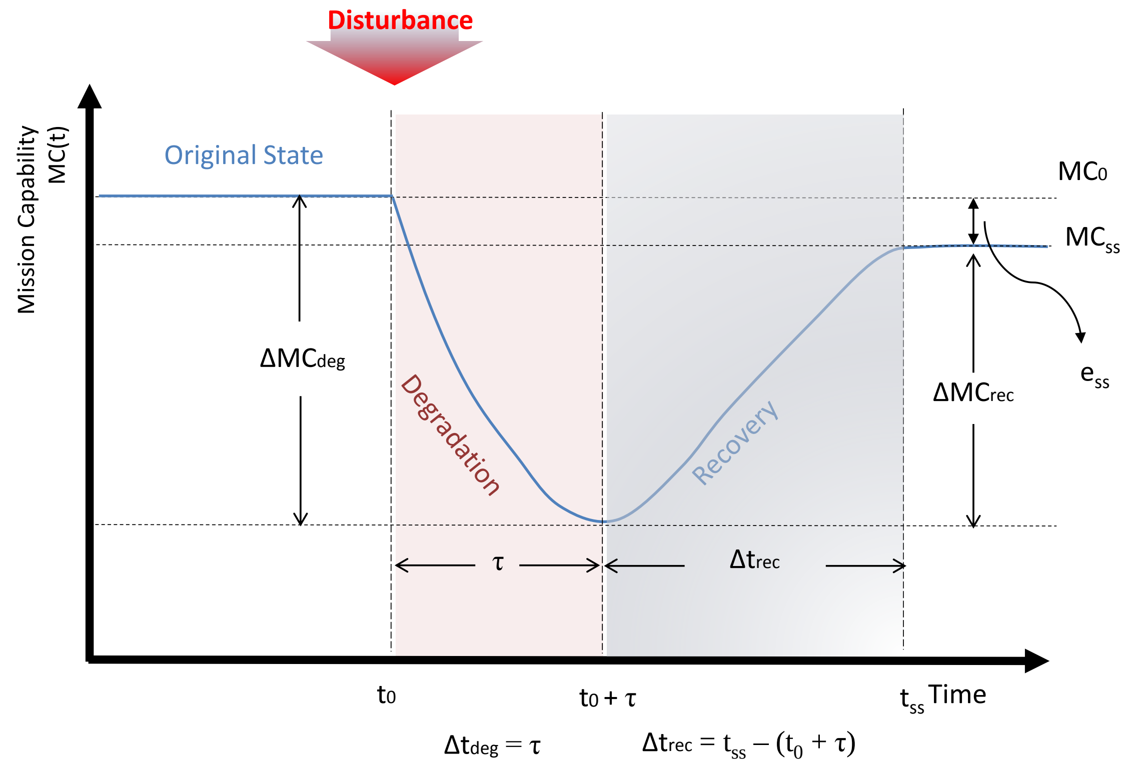


Figure 8: Notional diagram of system performance over time used for quantitative resilience metrics [19].

potential designs. Using fig. 8 as a reference, the ability of a system to absorb a disturbance is captured by the average degradation rate, ADR, and the time-averaged performance degradation, tMC_{deg}, calculated as

$$ADR = \frac{MC_0 - MC_{min}}{t_{min} - t_0} \tag{9}$$

$$tMC_{deg} = \frac{1}{t_{min} - t_0} \int_{t_0}^{t_{min}} [MC_0 - MC(t)]dt, \tag{10}$$

where MC_0 is the original desired mission capability or performance level, MC_{min} is the minimum mission performance level, t_{min} is the time at which MC_{min} is reached, and t_0 is the time at which the disturbance occurs.

The ability of a system to restore capabilities lost due to a disturbance is captured by the average recovery rate, ARR, and time-averaged performance recovery, tMC_{rec}

$$ARR = \frac{MC_{SS} - MC_{min}}{t_{SS} - t_{min}} \tag{11}$$

$$tMC_{rec} = \frac{1}{t_{SS} - t_1} \int_{t_1}^{t_{SS}} [MC_{SS} - MC(t)] \, dt. \tag{12}$$

where t_1 is the start time of the system recovery.

Limitations of TIRESIAS

Balchanos et al. demonstrate the use of this framework on a chilled water network [18] and a networked unmanned aerial vehicle (UAV) surveillance simulation [19]. However, both of these demonstrations focus on assessing system resilience, rather designing resilient SoS networks. As such, they provide limited guidance as to how SoS networks should be structurally defined, and how adaptation should be incorporated to achieve resilience.

ResiliNets

Sterbenz et al. propose the ResiliNets framework for assessing and designing resilient networks [104]. Their framework is based on a set of qualitative axioms (e.g. faults are inevitable) and a two-phase strategy for resilience called D^2R^2 + DR. This strategy proposes that a resilient network be designed with the ability to *defend* itself (passively and actively), *detect* disruptions, perform *remediation* actions, and *recover* its lost capabilities. This phase defines the D^2R^2 part of the strategy. The system would then *diagnose* what went wrong and *refine* itself to evolve and enhance resilience, defining the DR part of the strategy. The authors define a set of qualitative enablers and desired behaviors for resilience. Resilience enablers include connectivity, redundancy, and diversity. Resilience behaviors include self-organization, adaptability, and evolvability. A resilience state space is proposed for the analysis of system resilience, tracking the operational state of a system and its ability to provide desired service parameters (see fig. 9).

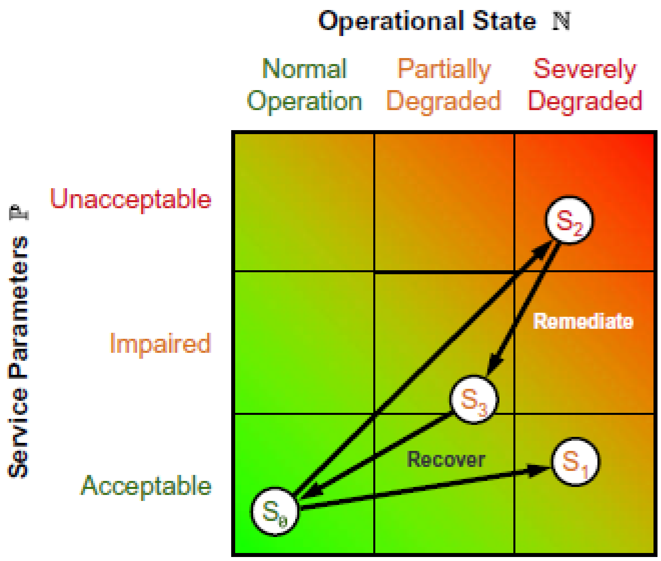


Figure 9: Resilience state space for the analysis of a system's resilience following a disruption event [104].

Sterbenz et al. apply the ResiliNets framework to study the resilience of various Internet topologies to node failures in [103]. They show results using network structural properties (i.e., largest component size and clustering coefficient) as service parameters and link failure probability as the operational state for several Internet service provider topologies. They also show results using simulated packet delivery ratio (service parameter) as a function of node and link failures (operational state). Nodes and links are removed from networks in a malicious or non-malicious manner. Malicious disruptions remove nodes or links based on degree and betweenness. Non-malicious disruptions remove nodes or links uniformly at random. Area-based disruptions are also modeled, where failures occur in a geographic region surrounding an initial point. Results show that targeted attacks are more disruptive to service parameters than random failures, and it is important to consider physical and logical networks when evaluating Internet resilience.

Limitations of ResiliNets

The ResiliNets framework provides a method for comparing the resilience of potential network designs. This framework can be quantitative and capability-based, as demonstrated by use of simulated packet delivery ratio for measuring provided network service. However, ResiliNets is mostly limited to being an assessment framework, with conceptual suggestions for methods to improve network resilience. Additionally, the work by Sterbenz et al. in [103] focuses on assessing the resilience of static network topologies without considering how a network might adapt to improve its resilience.

System resilience factor

Francis and Bekera expand on other resilience metrics by incorporating a recovery time factor to capture temporal aspects of resilience [48]. They define a system resilience factor, ρ_i, as follows

$$\rho_i \left(S_p, F_r, F_d, F_0\right) = S_p \frac{F_r}{F_0} \frac{F_d}{F_0}, \tag{13}$$

where S_p is a recovery time factor, F_r is the recovered system performance level, F_d is the degraded system performance level, and F_0 is the original performance level. They also account for uncertainty in event occurrences through consideration of event probabilities. Hence, their method can be seen as a resilience-based risk assessment.

Limitations of the system resilience factor

The system resilience factor enables design comparisons and aids decision-making processes by providing a single value for system resilience. However, their resilience metric does not consider the intermediate variation of system performance during recovery or the ability to adapt to multiple disruptions over time. There is also limited guidance on how to determine recovered performance levels and recovery time in the presence of volatile data.

2.3 Complex Network Resilience

Given the importance of networks to SoS, network methods for resilience are also reviewed. This review focuses on work from the field of complex networks. These studies are motivated by the observation that many complex, natural networks display a surprising amount of resilience to node failures. These failures, often random in occurrence, rarely destabilize networks to the point of total failure. Researchers aim to understand this inherent resilience of complex networks and identify conditions that leave networks vulnerable to failures.

This section begins with definitions of basic network terminology, followed by an introduction to the field of complex networks. The section concludes with a review of complex network methods for resilience that are relevant to this thesis.

2.3.1 Network Terminology

A network is a set of connected items. These items are known as nodes or vertices, and the connections between them as links or edges. The terms network and graph are often used interchangeably, with networks typically being defined by nodes and links, and graphs being defined by vertices and edges. Networks are typically associated with real systems while graphs are simply mathematical representations composed of vertices and edges [14]. Basic network terms are defined to clarify concepts commonly used in network research. A more detailed discussion of basic graph theory concepts is given by Barabási in his introductory book to Network Science [21].

The *total number of nodes* in a network is denoted by N. The set of all nodes in a network is denoted by $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, where n_i is the i^{th} node in the network. The *total number of links* in a network is denoted by L. The density, D, of a network is defined as the ratio of the number of links to the total possible number of links, such that

$$D = \frac{2L}{N\,(N-1)}. \tag{14}$$

Links in a network can be *undirected* or *directed*. An undirected link represents a two-way connection (i.e. the connection runs in both directions), while a directed link represents a one-way connection. An arrow pointing in the direction of the connection typically designates the direction of a directed link. A research collaboration network is an example of an undirected network, since research collaborations are typically bilateral partnerships. The WWW is an example of a directed network, since hyperlinks uni-directionally take a user from one web page to another. Two nodes i and j are *neighbors* if they are connected by a link l_{ij}. For a directed network, the link l_{ij} refers to a link from node i to node j.

The *degree*, k, of a node is the number of links connected to, or incident on, that node. For directed networks, the *in-degree* of a node (number of incoming links) is distinguished from its *out-degree* (number of outgoing links). The degree of a node (researcher) in a research collaboration network is the number of direct collaborators that node has. The out-degree of a node (webpage) in the WWW is the number of webpages that node directly links to. The degree of node i in an undirected network is denoted by k_i. For a directed network, the in-degree of node i is denoted by k_i^{in} and the out-degree by k_i^{out}. The total degree of node i in a directed network, k_i, is given by

$$k_i = k_i^{in} + k_i^{out}. \tag{15}$$

The *average degree* of a network is the average of node degrees over the entire network. The average degree of an undirected network is given by

$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i = \frac{2L}{N}. \tag{16}$$

The average degree of a directed network is given by

$$\langle k \rangle = \frac{1}{N}\sum_{i=1}^{N} k_i^{in} = \frac{1}{N}\sum_{i=1}^{N} k_i^{out} = \frac{L}{N}. \tag{17}$$

The average degree of a network is useful for providing a general description of the network. However, average degree is insufficient when considering networks with skewed proportions of high or low node degrees. The *degree distribution*, p_k, of a network is a more descriptive property for many real networks. The degree distribution describes the probability that a randomly chosen node in a network has a degree of k. Equivalently, p_k is the fraction of nodes in the network with a degree of k, such that

$$p_k = \frac{N_k}{N}, \tag{18}$$

where N_k is the number of nodes with degree k. Since p_k is a probability, the degree distribution sums to one (i.e. $\sum_{k=1}^{\infty} p_k = 1$).

Networks are often represented by adjacency matrices. The *adjacency matrix*, A, of a network of size N is an $N \times N$ matrix whose elements are

$$A_{ij} = \begin{cases} 1 & \text{if there is a link from node } i \text{ to node } j \\ 0 & \text{if there is not a link from node } i \text{ to node } j. \end{cases}$$

The adjacency matrix of an undirected network is symmetric (i.e., $A_{ij} = A_{ji}$). Links in a network can be *unweighted* or *weighted.* A weighted link has a weight, w_{ij}, associated with itself. Weighted networks are used to represent networks where links have varied weights or strengths, such as a communication network where each link has some data transmission or bandwidth capability. The elements of an adjacency matrix for a weighted network are the weights of that link (i.e., $A_{ij} = w_{ij}$).

Path lengths within a network are often used to describe the structure and assumed capabilities of a network. A *path* in a network is a route used to get from one node to another, traveling along links in the network. The *geodesic distance* between nodes i and j, d_{ij}, is the shortest path between those nodes (i.e. the path that uses the least number of links). The geodesic distance between two unconnected nodes (i.e. two nodes that have no path between them) is infinite. Breadth first search (BFS) is a commonly used algorithm to determine the geodesic distance between two nodes [21]. BFS calculates the distance between two nodes by identifying the direct neighbors of a starting node, then the neighbors of those neighbors, and continuing until the target node is reached.

The *diameter*, d_{max}, of a network is the largest geodesic distance between any two nodes in the network, calculated as

$$d_{max} = \max_{i,j \in \mathcal{N}, i \neq j} d_{ij}. \tag{19}$$

The *average path length* of a network, $\langle d \rangle$, is the average geodesic distance between all possible node pairs in a network. Average path length is also known as *characteristic path length*. Average path length can be used as a measure of network efficiency. Average path length is calculated by

$$\langle d \rangle = \frac{1}{N(N-1)} \sum_{i,j \in \mathcal{N}, i \neq j} d_{ij}. \tag{20}$$

Since average path length diverges for networks with unconnected nodes, the *inverse average path length* is also used to represent network efficiency. Inverse average path length is calculated by

$$\langle d \rangle' = \frac{1}{\langle d \rangle} = \frac{1}{N(N-1)} \sum_{i,j \in \mathcal{N}, i \neq j} \frac{1}{d_{ij}}. \tag{21}$$

The centrality of a node is useful for identifying how important it is to the connectivity of a network. The *betweenness centrality* of a node i, b_i, can be defined by

$$b_i = \sum_{j,k \in \mathcal{N}, j \neq k} \frac{\sigma_{jk}(i)}{\sigma_{jk}}, \tag{22}$$

where σ_{jk} is the number of shortest paths between nodes j and k and $\sigma_{jk}(i)$ is the number of shortest paths between j and k that passes through i. A similar calculation for *edge betweenness centrality*, b_e, can be defined using the number of paths that includes an edge e.

A network is *connected* if a path exists between every pair of nodes in the network. A *connected component* of a network is a subset of nodes in the network that is itself connected, but would no longer be so if any other node was added to the subset. A network composed of two components could become connected by the addition of a properly placed link. A *giant component* is a component that contains a large fraction of the nodes in a network [i.e., a component whose size is $O(N)$]. The *connectance* of a network is the ratio of links in the network to the possible number of links in the network (i.e., $\frac{L}{N}$). Connectance is commonly used in ecological network studies [40, 41].

A *regular* network is a network where every node has the same degree. A *complete* network is a network where every node is connected to each other, such that

$$L = L_{max} = \binom{N}{2} = \frac{N(N-1)}{2}. \tag{23}$$

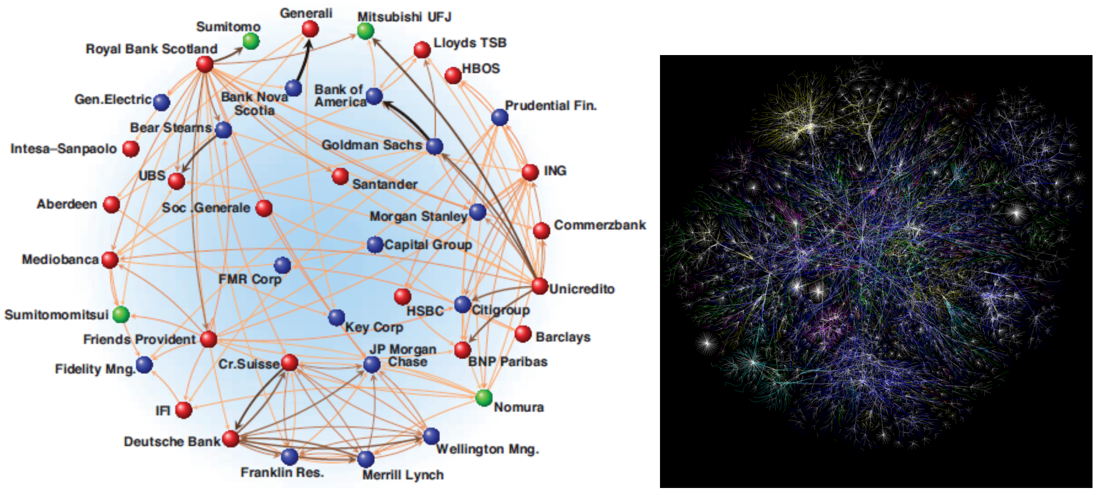
2.3.2 Background on Complex Networks

Many of the recent advances in network research, including those related to network resilience, focus on real world networks. These studies often take a statistical approach to modeling the evolution of real network topologies and understanding dynamic processes occurring on those networks. This research is commonly referred to as complex networks research, or network science. A 2005 NRC report on networks and the Army defined network science as, "the study of network representations of physical, biological, and social phenomena leading to predictive models of these phenomena" [79]. The growing presence of networks across multiple research domains has led this field to be highly interdisciplinary, applying methods and data from various research areas. Several review papers summarize influential work in complex networks research [85, 8, 24].

There is no clear definition of what a complex network is. Alderson loosely defines a complex network as a "network system with (1) a large number of components (complexity of size), (2) intricate relationships among components (complexity of interconnection), or (3) many degrees of freedom in the possible actions of components (complexity of interaction)" [14]. Complex networks can also be defined as networks with properties found in real world networks [67], such as the "small-world" property [113] and a power-law degree distribution [22]. These properties result in non-trivial network topologies not found in simple networks such as regular, latticed, and random networks. Boccaletti et al. add the consideration of network evolution, defining complex networks as "networks whose structure is irregular, complex and dynamically evolving in time" [24]. For this work, a complex network is defined as a network displaying properties and processes seen in real world networks, with a structure that evolves over time.

There are many examples of complex networks. Four categories of commonly studied complex networks are: social, information, technological, and biological networks [85]. Examples of complex networks (see fig. 10) include networks of terrorist interactions [21], research collaborations [83, 82, 84, 86], relationships between financial institutions [99], the World Wide Web (WWW) [9], airline networks [17], the Internet [45], and metabolic networks [63].

Complex networks research differs from traditional network research in three primary ways: (1) it focuses on *real world networks* rather than simple networks with trivial topologies; (2) it considers the *evolution* of a network's structure over time; and



(a) (b)

Figure 10: Network representations of (a) an international financial network [99] and (b) the Internet (links connect IP addresses) [4].

(3) it looks beyond the structural topology of a network to consider *coevolutionary network dynamics* [89].

The increasing availability of data characterizing real networks has enabled researchers to move beyond studies of simple, abstract networks to studies of complex, naturally occurring networks. However, visualizing these large networks (which can have up to millions of nodes), and attempting to derive meaningful insights from their corresponding data is a difficult process, requiring new methods and techniques for network analysis. A statistical approach enables researchers to study the structure of a large network without needing to clearly visualize it. A traditional approach might ask, "What does the network look like?" A statistical approach may alter the question to be, "What are the statistical properties of the network, and how can they be used to characterize the structure of the network?" Many network scientists use a network's degree distribution [22], diameter [10], and average path length [110] to characterize its structure.

Complex networks research also aims to understand how networks evolve over time. Previous research often viewed network topologies as static, limiting insights to be focused on the current state of a network. Focusing on the evolution of a network provides an understanding of the processes driving the formation the network, enabling new insights into the behaviors and dynamics of complex systems.

A growing area of research in complex networks considers coevolutionary behaviors of networks [52]. Dynamic processes occurring on a network often affect the evolution of a network's structure (e.g., bottlenecks preventing information information flow in a network may result in topological changes to alleviate those bottlenecks) . In turn, the evolution of a network's structure often affects process occurring on the network (e.g., topological changes to alleviate bottlenecks may have unintended consequences on connectivity in a different part of the network). The study of coevolutionary networks jointly considers the dynamics *on* a network (processes occurring on the network) and the dynamics *of* a network (evolution of the network's structure). Network resilience can be considered a coevolutionary network dynamic, since it considers changes to a network's structure following node failures or attacks, how those changes affect the ability of a network to provide desired capabilities, and how a network adapts its structure to recover lost capabilities.

Complex Network Topologies

Many real, complex networks have been found to display a scale-free structure, or topology [22, 45, 63, 8, 85]. Finding structural similarities between these networks is particularly interesting, since they represent many different types of systems (e.g. biological, information, and social systems) that would seem to be controlled by different formation processes and dynamics. Engineered systems and SoS, such as C2 networks, have also been found to have scale-free topologies [61, 51, 106]. Due to their prevalence in real world networks, many studies of complex network resilience focus on scale-free networks.

Scale-free networks have a degree distribution that follows a power law, such that the probability, p_k, that a node has a degree of k, scales as

$$p_k \sim k^{-\gamma}, \tag{24}$$

where the degree exponent, γ, typically falls in the range of $2 < \gamma < 3$. This class of networks is heterogeneous with respect to node degree, as they contain many nodes with a degree lower than the average and several nodes with a degree much higher than the average. These highly connected nodes act as network hubs, efficiently providing connectivity within the network. For this reason, scale-free networks can also be described as hub and spoke networks. The term scale-free refers to the lack of an intrinsic scale regarding the expected degree of a randomly chosen node in the network, resulting from the large range in node degrees seen in scale-free networks [21].

A commonly used model for generating scale-free networks is the Barabási-Albert (BA) preferential attachment model [22]. The model begins with a small network of size m_0, with links arbitrarily assigned such that each node has at least one link (this thesis assumes the m_0 nodes are fully connected). A new node is then added to the network at each time step. Each new node links with m existing nodes in the network, where the probability, $\prod(k)$, that an existing node with degree k is linked with is

$$\prod(k) = \frac{k}{\sum_{i=1}^{N(t)} k_i}, \tag{25}$$

where $N(t)$ is the size of the network at the current time step t and k_i is the degree of node i. This model includes two important network mechanisms, growth and preferential attachment, hypothesized by Barabási and Albert to be driving forces in the formation of scale-free networks. Network growth is captured by growing the network from size m_0 to N, one node at a time. Preferential attachment is captured by the "preference" a new node gives to existing highly connected nodes when selecting who to link with.

In contrast, a random network is one where each node pair is independently linked with the same probability p, creating a topological structure that appears to be truly random. Random networks are also referred to as Erdős-Rényi (ER) networks, due to the influential work of Erdős and Rényi establishing a branch of mathematics focused on these networks [21]. Random networks have a degree distribution that follows the binomial distribution, or in the limit of large network size, N, the Poisson distribution, such that

$$p_k = \binom{N}{k} p^k \left(1-p\right)^{(N-k)} \simeq \frac{\lambda^k e^{-\lambda}}{k!}, \tag{26}$$

where $\lambda = \langle k \rangle$ in the limit of large N. The degree distribution of a random network peaks at $\langle k \rangle$. This class of networks is homogeneous with respect to node degree, as most nodes have a degree near the average. This homogeneity results in few, if any, network hubs.

A model for generating random networks is the $G(N, L)$ model, referred to as the ER model (for its creators Erdős and Rényi) within this thesis [43]. The ER model randomly links L node pairs in a network of size N. This model does not include network growth or preferential attachment mechanisms, since the network begins with all N nodes and randomly adds L links to the network. Figure 11 shows a notional comparison of scale-free and random networks.

2.3.3 Complex Network Methods for Resilience

Many studies of complex network resilience expand on the work of Albert, Jeong, and Barabási characterizing the effects of targeted and random node removal on scale-free and random networks [10] (note the correction submission for their original paper [11]). The authors consider a random network, created using the ER model [43], and scale-free networks, created using the BA preferential attachment model [22]

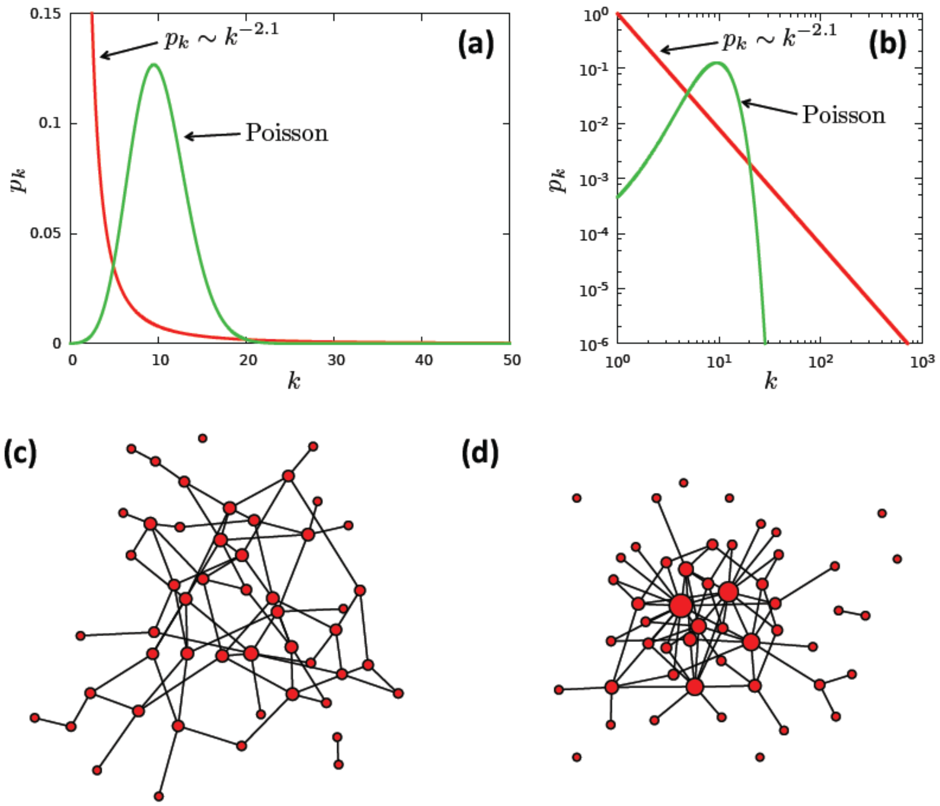


Figure 11: Comparison of theoretical degree distributions for scale-free (power law) and random networks (Poisson) with $\langle k \rangle = 10$, shown with (a) linear and (b) log-log scales. Comparison of (c) random and (d) scale-free network topologies with $\langle k \rangle = 3$ [21].

and data of the Internet and WWW. Targeted attacks are simulated by removing the highest degree nodes from a network. Node degrees are recalculated following each node removal. These attacks represent a malicious adversary seeking to damage a network by targeting specific nodes. Random failures in a network are simulated by removing nodes uniformly at random. Network resilience is quantified by tracking the changes network structural properties as a function of the fraction of nodes removed, f.

Results show that the average path length (referred to by the authors as the network diameter) for random and scale-free networks stays fairly constant with f for random node removals. However, average path length is slightly higher for random networks than scale-free. The increased resilience of scale-free networks to random

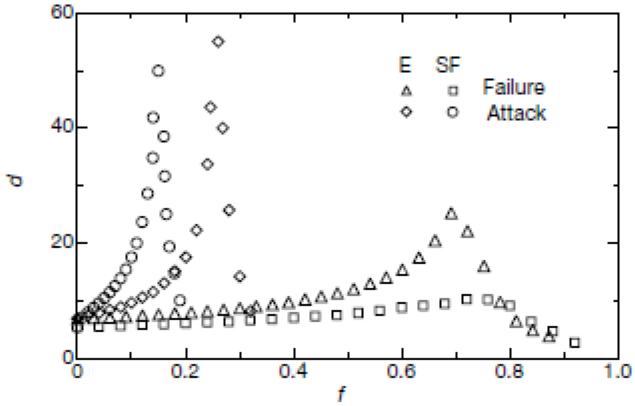


Figure 12: The effect of attack (targeted) and failure (random) on average path length, d, as a function of the fraction of nodes removed, f, for (a) random (E) and scale-free (SF) networks [10].

node removals is due to the heterogeneity of scale-free networks; since most nodes have a relatively low degree, randomly removing a node has little probability of greatly affecting the connectivity of the network. Since most nodes in random networks have similar degrees, randomly removing a node in these networks has more impact on network connectivity than with scale-free networks. Figure 12 shows results for scale-free and random network resilience, measured by average path length.

Switching from random to targeted node removals shows no substantial change in network resilience for random networks. These networks are insensitive to node removal type because of their node homogeneity. In comparison, the average path length of scale-free networks drastically increases with f for targeted node removals. Scale-free networks are not resilient to targeted attacks because they contain highly connected network “hubs,” which have a significant impact on overall network connectivity when removed.

The authors also consider the effect of node removals on the size of the largest component and the average size of isolated components. Isolated components are all components other than the largest one. These results explain the fragmentation process of networks as f is increased. The primary conclusion of the study is the observation that scale-free networks are more tolerant to random failures than random networks, but more susceptible to targeted attacks.

Broder et al. perform a similar study in which they evaluate the size of the largest component of the WWW network following removal of high degree nodes [27]. In contrast to Albert et al., they conclude that scale-free networks are resilient to targeted attacks, since significant damage to the giant component does not occur until all nodes with in-degree of five or higher are removed. Newman explains that there is no conflict between these results because of the highly skewed degree distribution for the WWW data used by Broder et al. [85]. This distribution results in the fraction of nodes with in-degree higher than five being a small portion of the overall network, meaning that only a small fraction of nodes had to be attacked before the giant component collapsed.

Several other studies investigate the resilience and robustness of real and modeled complex networks. Jeong et al. examine protein networks, correlating the degree of a node to the phenotypic effect of removing that node [62]. Dunne et al. examine the resilience of food web networks (i.e., species predator-prey networks), and show that networks with higher connectance are more robust to highest degree and random node removal [41]. Newman et al. study the prevention of a virus outbreak on email networks, and show that selectively protecting about 10% of the nodes in an email network can provide near immunity to a global virus outbreak [87]. Dodds et al. focus on information exchange in an organizational network subject to information overload (congestion) and node removal [38].

Holme et al. investigate the vulnerability of two real networks and several modeled networks to node and edge removal [59]. Nodes are removed based on their initial degree (ID), recalculated degree (RD), initial betweenness (IB), and recalculated betweenness (RB). Recalculation is done following every node removal step. Similar targeting strategies are used to remove edges from a network. The degree of an edge k_e connecting nodes i and j is defined by the product of the degrees of

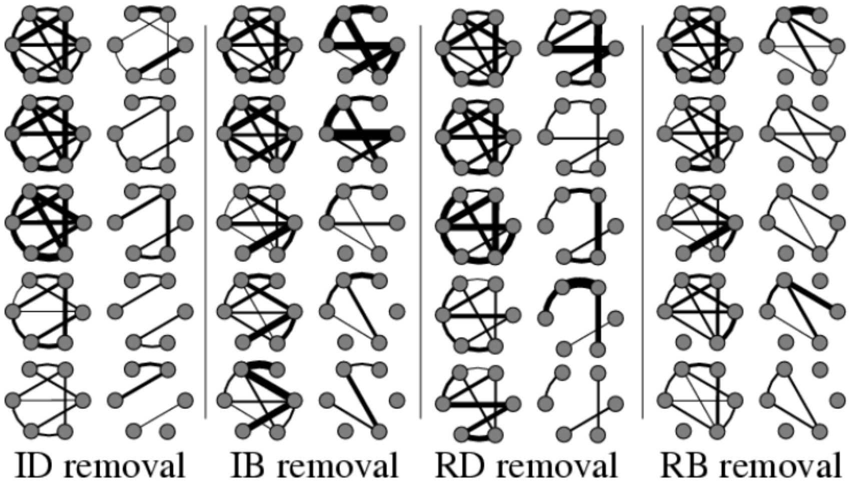


Figure 13: Comparison of targeting strategies applied to the Watts-Strogatz small-world network model. Edges are removed from the network following the images moving down each column, beginning with the top left image for each targeting strategy [58].

the connected nodes (i.e., $k_e = k_i k_j$). The four edge targeting strategies are shown in fig. 13. Their support previous claims that random networks are more robust to targeted node removal than scale-free networks.

Limitations of Complex Network Methods for Resilience

These studies of network resilience provide significant insights into the ability of complex networks to maintain desired structural properties following node or edge removal. However, several assumptions made in these studies limit their use towards the design and understanding of resilient SoS networks.

One limitation is the use of *network structural properties* to assess resilience. The performance of complex networks is intuitively related to their structure or topology. However, focusing on structural properties does not fully capture the concept of resilience, as defined for this work (see section 1.5). Resilience focuses on the ability of a system to maintain desired *capabilities* following disruptions. While network structure and connectivity likely contribute to these capabilities, there are many other factors that may determine the provided capabilities of an SoS network. Ignoring these possible factors and assuming that connectivity directly translates to capability can be misleading when designing SoS networks.

Focusing on structural properties also assumes a *static* view of resilience. The resilience of a system is not only determined by the capabilities maintained in the presence of a threat, but also the capabilities recovered following the threat disturbance, with consideration of the time elapsed during this entire process. Using network structural properties to evaluate resilience does not fully consider this aspect of resilience, since there is no time consideration when determining the effect of a removed node on network structure.

These studies also do not consider network adaptation and reconfiguration. The ability of a resilient system to adapt to potential threats is a defining characteristic that distinguishes it from a robust system. These studies are more accurately described as studies of network *robustness*.

Percolation Theory Approaches to Resilience

Complex network researchers have also applied concepts from percolation theory to study network resilience. This approach focuses on identifying the phase transition of a network from a non-percolating to percolating state, in the limit of infinite network size. A network is said to be percolating if a randomly selected node has high probability of being in a giant or infinitely sized component. Percolation models are based on bond or site percolation. Site percolation randomly designates nodes as "occupied" (functional) or "unoccupied" (failed). A resilient network maintains a giant component of occupied nodes for a high fraction of unoccupied nodes [85].

Cohen et al. use percolation theory to derive analytic expressions for the percolation critical threshold, q_c, of the Internet and similarly structured networks. They assume that the phase transition of a network occurs when, on average, a node i connected to a node j that is in the largest component, is also connected to at least one other node. The authors then consider the uniformly random breakdown of a fraction q of nodes in a network (i.e., the probability that a node becomes unoccupied is q). Using derived expressions for the phase transition as a function of q, they identify that networks with a power law degree distribution and degree exponent $\gamma \leq 3$ (where $p_k \sim k^{-\gamma}$) have a phase transition that occurs at $q \leq 0$. This result suggests that networks with the described power law degree distribution always have a giant component. Since real world scale-free networks have been shown to have a power law degree distribution with degree exponent $\gamma \leq 3$, this result agrees with other work showing scale-free networks to be robust to random node failures.

Callaway et al. [28] extend the work of Cohen et al. [30] by no longer restricting breakdown probabilities to being uniformly random, instead allowing them to be a function of node degree. This extension allows the consideration of targeted node attacks. Their method uses generating functions to derive expressions for the phase transition (see Newman et al. [88] for a description of this approach). Their results show that networks with power law degree distributions are highly susceptible to targeted node removal, as was shown by Albert et al. and others. Cohen et al. similarly expand their previous work to include targeted attacks and reach the same conclusion [31]. Figure 14 shows the critical threshold of power law networks as a function of power law degree exponent. Schwartz et al. later consider the percolation of directed networks [98] and Serrano and Boguna the percolation of clustered networks [100].

Limitations of Percolation Theory Approaches to Resilience

These studies are significant because they arrive at similar conclusions to those made by Albert et al. [10], Broder et al. [27], and others regarding the resilience of complex networks. However, these studies maintain many of the same issues limiting the use of other network resilience studies for designing resilient networked SoS. These studies focus on network structural properties, take a static view of resilience, and do not

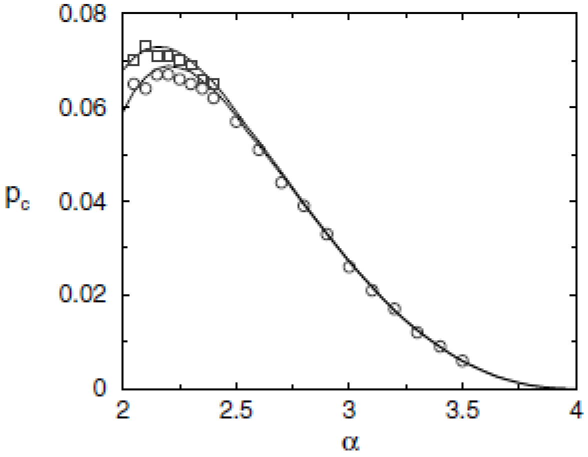


Figure 14: Critical threshold, p_c, as a function of degree exponent (shown here as α), for power law networks subjected to targeted node removal [31].

consider network adaptation.

Random Rewiring Approaches to Resilience

Schneider et al. [97] and Herrmann et al. [55] address some limitations of previous work on network resilience by applying random link rewiring to improve network resilience to node removal. They use a robustness measure, R, defined as

$$R = \frac{1}{N} \sum_{Q=1}^{N} s(Q), \tag{27}$$

where $s(Q)$ is the fraction of nodes in the largest component after removing $Q = qN$ nodes. The fraction of removed nodes, q, consists of the most connected nodes in the network, representing targeted attacks. This approach to measuring network robustness is more descriptive than percolation theory methods since it considers the partial breakdown of the giant component.

Schneider et al. use this robustness measure to propose a random rewiring algorithm for mitigating the effects of targeted attacks [97]. The algorithm consists of randomly selecting two links in a network, e_{ij} and e_{kl}, and swapping those connections to result in links e_{ik} and e_{jl}. The swap is only performed if network robustness, calculated using eq. (27), is improved. The process is repeated with other randomly

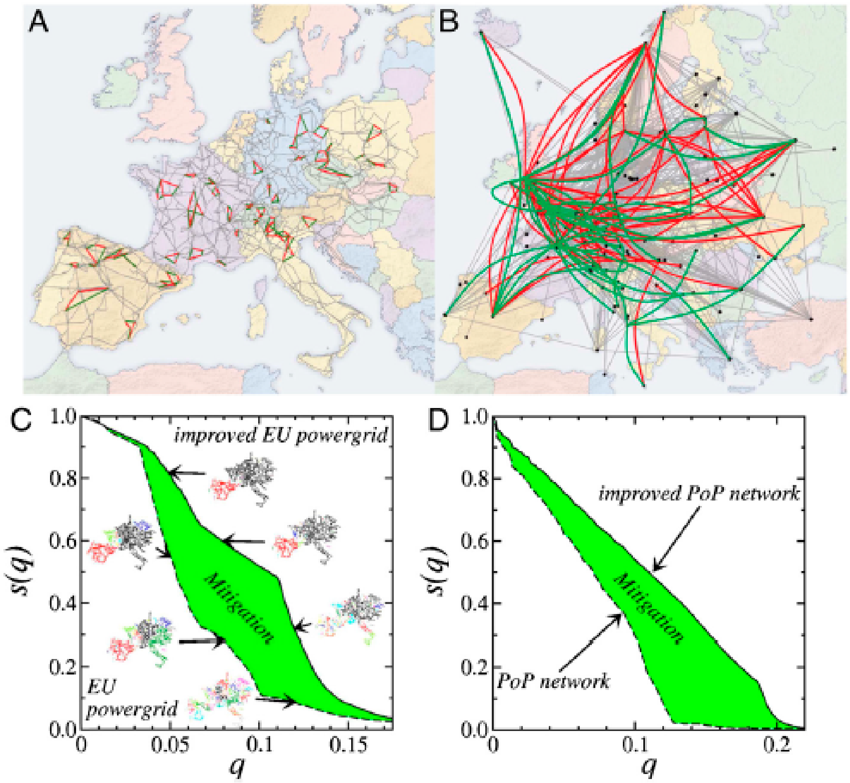


Figure 15: EU power grid (structure shown in A) and point of presence Internet networks (structure shown in B) are shown to have improved robustness following link swaps. (C) shows improvements to the EU grid, (D) shows improvements to the Internet networks [97].

selected pairs of links until no significant improvement is achieved within a series of subsequent links swaps. This method considers connection costs by maintaining the initial number of network links. The authors apply this method to a European power supply network and service provider representation of the Internet (see fig. 15). They also consider the case of designing a new network using their link swap algorithm, while maintaining a given degree distribution. Their results show a unique “onion-like” structure to their designed robust networks. Herrmann et al. provide an expanded discussion of this novel onion-like network structure in [55]. Louzada et al. apply a similar link swap algorithm to improve the robustness of air transportation networks [71].

Ventresca et al. focus on the resilience of a network against multi-strategy sequential attacks. Their approach allows an attacker to alter its targeting strategy during successive attacks, choosing between six possible centrality measures (betweenness centrality, closeness centrality, degree centrality, PageRank, Kleinberg's Authority scores, and leverage centrality). Attackers either randomly choose their targeting strategy at each iteration or use a greedy approach based on ranked effects for each strategy. Graph degree distribution entropy, largest component size, efficiency (similar to average path length), and pairwise connectivity are used as measures of network resilience. They allow networks to adapt through random rewiring, in an effort to mitigate the effects of node removals. Their rewiring strategy allows the random re-addition of $k-1$ links following the removal of a node with k links. Results show that maintaining a betweenness targeting strategy is generally more effective at damaging a network than the other strategies considered, including the greedy multiple strategy case. Random rewiring-based adaptation slows down network destruction for all cases considered.

Limitations of Random Rewiring Approaches to Resilience

These studies add random link rewiring to previously discussed work on complex network resilience, partially accounting for the adaptive aspect of resilience. However, the link swap algorithm proposed by Schneider et al. [97] does not fully capture the concept of resilience through adaptation, since they propose link swaps as a method of preemptively improving a network's structure. This approach is more focused on robustness, since the network is still unable to adapt itself once an attack occurs. In fact, the authors define their metric to determine accepted link swaps as a robustness measure. Their robustness measure also does not consider the time aspect of resilience (i.e., how quickly a system loses capabilities and how long it takes to recover following an attack).

Ventresca et al. [111] propose a more dynamic defense against attacks by allowing networks to add links following each attack. Their method more closely considers resilience by incorporating adaptation. However, their analysis of how well adaptive networks perform still focuses on the structure of the network following each attack.

A Method for Organizational Network Resilience

Dodds et al. explore information exchange in an organizational network subjected to information overload (congestion) and node removal [38]. They develop a model to create organizational networks based on a backbone hierarchical network. Information exchange in the network is modeled by specifying a message initiation rate, μ, and target locality, ζ. The authors define an ultrarobust network to be one that is robust to network congestion and node removal. Network congestion is varied by increasing the message initiation rate and distance messages must be passed across the network (i.e., decreasing target locality). Nodes are removed based on rank, neighbors cascading outwards from a random star point, connectivity, and uniformly random selection. Congestion robustness is measured by the maximum congestion centrality, where congestion centrality is the probability that any message will be processed by a node. Network robustness to node removals is measured by the fractional size of the largest component in the network. Results show that a class of network topologies, called multiscale networks, is the only network class studied to show the property of ultrarobustness.

Limitations of A Method for Organizational Network Resilience

This study provides insights into the robustness of organizational networks, but as with previous complex network studies, does not fully consider resilience. The networks studied are static and unable to adapt to attacks. Additionally, network connectivity is used to evaluate the robustness of a network to node removal. This approach does not account for the actual system capabilities. Congestion robustness is a capability-based measure, but this measure is only used to study network congestion as the operational environment is changed (i.e., robustness to variations in operating conditions).

2.4 Gaps in the Reslience Literature

A review of the literature reveals certain gaps and limitations regarding our understanding of resilient SoS networks, and the availability of methods for designing them. Most of the work from the SoS community focuses on robustness, rather than resilience, or provides limited guidance as to how resilient SoS networks should be designed and assessed. The resilience engineering community has proposed many frameworks for assessing resilience. However, these frameworks are often qualitative, or limited in their application to real data. These framework also focus on resilience assessments, rather than identifying novel ways to improve resilience.

Most work in the complex networks community also focus on network *structural properties.* While network path lengths and connectivity can typically be assumed to play a large role in how well an SoS network performs, there are other factors that can affect SoS capabilities. Assuming that an SoS network is resilient because it is connected can lead to poorly informed design choices. Complex networks research is also limited in its application of network adaptation for resilience. Instead, many studies focus on robustness to node removals.

The identified gaps in the literature result in the following observation:

> **Observation:** Methods developed by the SoS, resilience engineering, and complex networks communities are insufficient for meeting the research objectives established for this thesis, due to incomplete quantitative capability-based assessments of resilience and limited consideration of network adaptation for resilience.

Chapter 3 describes the methodology developed to satisfy the research objectives for this thesis and address the issues raised by this observation.

CHAPTER III

METHODOLOGY OVERVIEW AND REPRESENTATIVE TEST PROBLEM

A methodology for designing **re**silient **S**o**S** **net**works (ReSSNET) is developed to answer the primary research question of how to mitigate network threats (RQ 1) and test the hypothesis that designing for resilience rather than robustness is better for SoS networks (HYP 2). This methodology satisfies the first research objective for this thesis, by providing *quantitative, capability-based assessments* of resilience, accounting for the *networked* nature of SoS, and considering *network adaptation* as a response to potential threats.

This chapter describes a framework for developing the ReSSNET methodology, high-level research questions related to methodology steps, and a representative SoS network test problem used throughout this thesis.

3.1 ReSSNET Methodology Overview

A framework for developing the methodology is the Top-Down Design Decision Support Process in the Integrated Product and Process Development (IPPD) methodology [75]. The Decision Support Process is a general design method applicable to many types of problems, regardless of their domain. This process is summarized on the left side of fig. 16.

The first two steps of the process are outside the scope of this thesis, as this work assumes that the need and problem have already been established with proper requirements analysis and definition of desired system functions. The final step of the process, make a decision, is also outside the scope of this thesis, since it is an

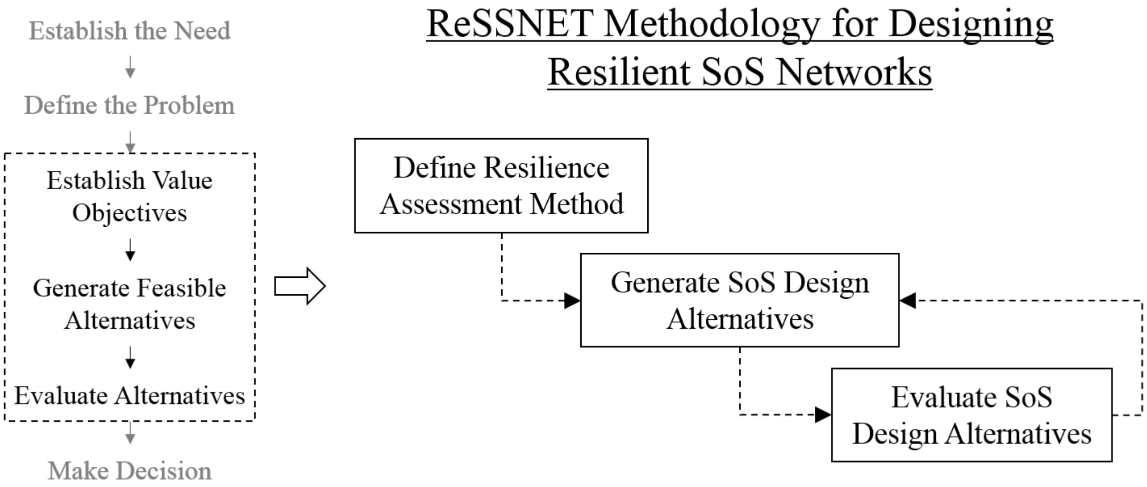

Establish the Need
Define the Problem
Establish Value Objectives
Generate Feasible Alternatives
Evaluate Alternatives
Make Decision
ReSSNET Methodology for Designing Resilient SoS Networks
Define Resilience Assessment Method
Generate SoS Design Alternatives
Evaluate SoS Design Alternatives

Figure 16: Overview of the ReSSNET methodology for designing resilient SoS networks.

exhaustive step that is very specific to the application problem being considered. This thesis focuses on steps 3-5: establishing metrics to assess performance, generating alternatives, and evaluating those alternatives.

Figure 16 shows how the Decision Support Process is adapted to the problem of designing resilient SoS networks. Since the objective is to design for resilience, a method for assessing SoS resilience is defined. SoS design alternatives are then generated and evaluated using the defined resilience assessment method. If the generated alternatives are found to be insufficient for the problem of interest, new alternatives should be generated based on lessons learned from the previous iteration of the methodology. The loop between steps two and three of the methodology represents the possibility of such an iteration.

3.2 High-level Research Questions

Several high-level research questions must be answered to develop the desired methodology. The first question focuses on identifying a method for assessing SoS resilience. To enable SoS design comparisons and trade-off studies, the method should provide a single metric, or set of metrics, for quantitatively assessing resilience. For example, these metrics would provide a method for quantifying the ability of an IE network to maintain a high message passing rate in the presence of node failures. This need results in the following research question:

> **Research Question 3:** What metric, or set of metrics, should be used to quantitatively assess SoS resilience?

Once a method for assessing resilience is identified, SoS design alternatives need to be generated. To fully explore the problem of SoS network resilience, we need to identify methods for defining initial network topologies and adaptation methods, as well as potential threats. For example, these methods would provide a set of potential IE network designs and threats to consider. This need results in the following research question:

> **Research Question 4:** How should we define potential SoS network topologies, threats, and adaptation methods?

The final step in the methodology is to evaluate SoS design alternatives. This step focuses on identifying trade-offs between alternatives and understanding what types of scenarios those alternatives are best suited for. For example, this step would identify what types of IE networks should be used in certain situations. The following research question addresses this step in the methodology:

> **Research Question 5:** What methods should be used to compare SoS design alternatives and understand their advantages and disadvantages?

CHAPTER IV

A FRAMEWORK FOR ASSESSING SOS RESILIENCE

The first step in the ReSSNET methodology requires a method for quantitatively assessing SoS resilience. The selected method should provide a metric, or set of metrics, for assessing resilience to enable design trade-off studies. This need is reflected in research question three (repeated here for convenience):

> **Research Question 3:** What metric, or set of metrics, should be used to quantitatively assess SoS resilience?

This chapter describes a capability-based framework that provides a set of metrics for quantifying resilience and an experiment performed to test the metrics proposed by the framework.

Method Requirements and Alternatives

To answer research question three, we must establish what it means to be resilient and specify criteria for achieving resilience. Section 1.5 describes several defining characteristics of resilience and resilient systems. These characteristics apply to systems and SoS, and are summarized in the following observations:

- Resilience is *capability-based*, since resilient systems are able to maintain or recover desired capabilities.
- Resilience is *dynamic*, since resilient systems are able to absorb, recover from, and adapt to threats.
- Resilient systems provide *graceful, or smooth*, transitions from degraded to recovered states.

- Resilience is *time-dependent*, since recovery time is typically important for mission success.

- Resilience is *disruption-dependent*, since the resilience of a system depends on the disruption it faces.

In addition to these observations, we also note the importance of providing quantitative assessments of resilience to enable design trade-off studies. These observations are used to derive a set of requirements for selecting the method to assess SoS resilience. The selected method must be:

- Quantitative

- Capability-based

- Dynamic (account for the ability to smoothly and quickly absorb, recover from, and adapt to threats)

- Disruption-based

A review of the literature shows many methods and frameworks for assessing system resilience (see chapter 2). For this thesis, the most relevant of these methods are the system resilience framework [112], TIRESIAS [20], ResiliNets [104], and complex network structural properties [10, 59].

The *system resilience framework* [112] provides a partially quantitative method for evaluating resilience. This method directly accounts for system capabilities through calculations of systemic impact, but does not provide detailed quantitative calculations to differentiate between how a system absorbs, recovers from, and adapts to a threat. Instead, the characterization of these resilience capacities is limited to qualitative comparisons. The use of performance and cost measurements over time does account for dynamic system behavior. However, smooth capability transitions are not accounted for, since integration removes some of the dynamics of capability data. The generalized nature of the framework allows any type and number of threats to be considered.

TIRESIAS is a quantitative framework for assessing system resilience [20]. This framework extends the system resilience framework by adding quantitative metrics to characterize how well a system absorbs and recovers from a disruption. These metrics allow a more refined comparison between potential systems that also considers the dynamic behavior of a system following a disruption. However, these metrics are limited in their ability to characterize the smoothness of capability transitions and adaptation to repeated threats.

The *ResiliNets* framework proposes a resilience state space to evaluate the resilience of a potential system facing a given threat. Desired capabilities are used as the service parameter in the state space. The state space provides a dynamic consideration of system performance by tracking how the system behaves throughout an engagement. However, there is no suggested quantification of the various capacities of resilience or ability to provide smooth transitions from degraded to recovered states.

The *system resilience factor* extends other resilience metrics by explicitly considering recovery time, in addition to recovered and degraded performance levels. However, there is limited guidance for how to calculate recovery time and recovered performance levels for systems facing multiple threats with volatile performance data.

Assessing resilience with *network structural properties* is quantitative, but does not directly account for system capabilities. While good network connectivity often indicates good network performance, a more thorough approach would focus on actual system capabilities, whether through real data or simulated performance. This approach also does not fully consider the dynamic behavior of a system following a threat or disturbance, since network structures only change immediately before and after threat events.

Table 3: Comparison of resilience evaluation methods

Requirement	System Resilience	TIRESIAS	ResiliNets	Resilience Factor	Network Structural
Quantitative	Partially	Yes	Partially	Yes	Yes
Capability-based	Yes	Yes	Yes	Yes	No
Dynamic	Partially	Partially	Partially	Partially	No
Disruption-based	Yes	Yes	Yes	Yes	Yes

Table 3 compares potential resilience assessment methods with respect to the defined requirements for use in this methodology. Since none of the methods meets the established requirements, a gap in the literature is identified as follows:

> **Observation:** There is a gap in the resilience engineering literature for a capability-based resilience assessment framework that satisfies the established requirements for assessing SoS resilience.

4.1 A Capability-based Resilience Assessment Framework

This thesis develops a new method for assessing system, or SoS, resilience. The method is presented as a framework to provide a structured process for assessing resilience, and fills the identified gap in the resilience engineering literature. This framework enables quantitative comparisons of potential system designs, with respect to their resilience to a set of identified disruptions. For example, the framework could be used for the conceptual design of a resilient communications network, where multiple recovery actions (e.g., network adaptation) are being considered. Assessing the resilience of these potential network designs enables decision makers to quantitatively compare the benefits and expected performance of each design.

The framework is composed of five steps, shown in fig. 17. This section provides a description of each step in the framework. Particular attention is given to the final step of the framework, which develops of a set of resilience factors, a system resilience metric R, and a system total resilience metric R_{total}. These metrics are the primary

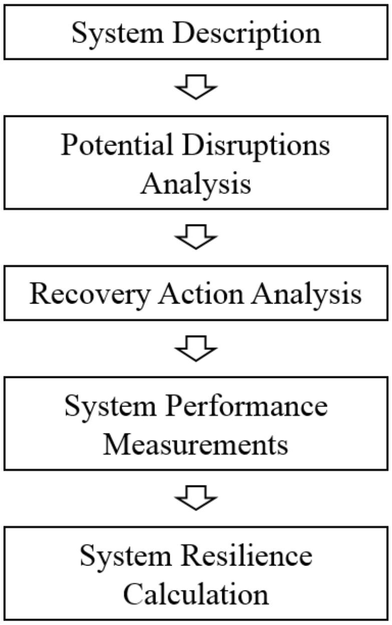

System Description
Potential Disruptions Analysis
Recovery Action Analysis
System Performance Measurements
System Resilience Calculation

Figure 17: Overview of the capability-based resilience assessment framework.

contribution of the framework and are therefore given their own section, section 4.2. These metrics directly answer research question three. The framework is applicable to systems and SoS (it does not differentiate between the two), and is therefore described with respect to systems for convenience.

System Description

The framework begins with a definition of the systems being considered and their desired capabilities or performance levels. These systems may be currently existing ones with suspected vulnerabilities, future systems being considered in a design study, or any other systems requiring a resilience assessment. The primary limitation on the type of system that may be considered is the ability to attain measurements of system performance over time for the scenarios of interest. System capability is defined as a time invariant property of a system (assuming no changes to the system itself) that represents its ability to provide some desired functionality. System performance is defined as a time varying measure of a system's ability to provide its desired capability.

For the IE network model used in this thesis, systems are defined as IE networks, with network capability defined as the ability to receive messages. IE network performance is therefore the number of received messages at a given time.

Potential Disruptions Analysis

When assessing the resilience of a system, an analyst must consider what the system is resilient to, since “the resilience of a system can be measured only in terms of the specific threat (input)... different attacks would generate different consequence (output) trajectories for the same resilient system” [53]. Therefore, the framework requires identification of potential threats, or disruptions, to be considered.

For example, consider a structure designed for use in areas known for strong tornado activity. Since this structure is designed in anticipation of destructive wind speeds, it will likely be resilient to scenarios with low to moderate wind speeds. However, this same structure may not be as resilient to flooding, since it was not designed with such disruptions in mind. Stating that a system is simply resilient does not provide enough information regarding the expected performance of the system. A more complete statement would include what the system is resilient to; high wind speeds in this example.

Potential threats for the IE network model are node failures or attacks on nodes and links.

Recovery Action Analysis

Since resilience focuses on the ability to adapt and respond to threats, a definition of potential recovery actions is also required. These recovery actions enable a system to respond to threats and recover lost capabilities. Some recovery actions may even improve system capabilities relative to normal operating conditions, providing a level of anti-fragility to the system [105].

Potential recovery actions for the IE network model include network adaptation and link redundancy. Network adaptation would allow a network to regain connectivity following node or link attacks and maintain the ability to share information throughout the network.

System Performance Measurements

This framework uses measurements of system performance over time to calculate resilience, since temporal aspects of a system's response to disruptions play a large role in resilience [53]. These measurements are required to be equally spaced in time, such that they can be represented as time series data. This requirement enables the use of digital signal processing and time series methods for analyzing system performance data and calculating resilience.

Two approaches to attaining system performance data are using recorded data from actual system operations and modeling and simulation. If possible, real system data should be used, provided that the data is complete and has been accumulated over periods of time in which the system faced disruptions of interest. However, such data is often highly corrupted or simply not available, particularly for large, complex systems and SoS. Additionally, resilience assessments may be needed for system design trade-off studies. The systems of interest in such studies typically do not exist. This thesis uses modeling and simulation to produce system performance data, in the absence of actual system data. This approach enables consideration of multiple system designs, disruption types, and recovery actions. Simulation studies can also capture stochasticity in system operations, for example through the use of simulation replications or Monte Carlo simulation. Many complex systems and SoS have randomness in certain aspects of their operation, which can strongly affect the impact of potential disruptions on a system and ability of the system to respond. The stochastic nature of system operations can therefore affect the resilience of a system, and should be considered in any resilience assessment.

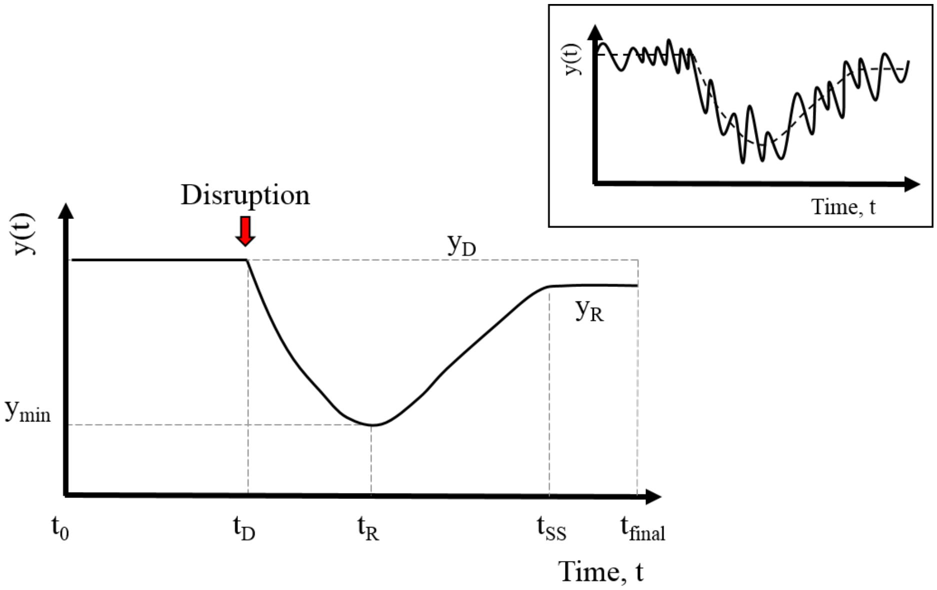


Figure 18: Notional plot of system performance data. The inset figure shows a notional example of volatile (solid line) and smoothed (dashed line) performance data.

Figure 18 shows an example of notional system performance data, where $y(t)$ represents the performance of a system at time t. Important times and performance values in this notional example are:

- t_0 = start time of the period of interest
- t_D = time of the disruption event
- t_R = time when the recovery action is implemented
- t_{SS} = time when performance recovery has reached steady-state
- t_{final} = end time of the period of interest
- y_D = desired performance level
- y_R = recovered performance level
- y_{min} = minimum performance level

The notional data in fig. 18 shows clear trends and smooth transitions in system performance. However, actual measured or simulated data is often volatile or noisy, due to the stochastic nature of many real or simulated processes (see the inset of fig. 18). To simplify the analysis and focus on general trends in noisy performance data, a data smoothing method is desired for the calculation of resilience metrics. The following research question addresses this need:

> **Research Question 3.1:** What method should be used to smooth system performance data?

The primary requirement for a smoothing method for resilience assessments is the ability to remove unnecessary noise in the data while still capturing general trends, particularly large peaks due to sudden increases or decreases in the data. The method should also be computationally inexpensive to aid in automation of resilience calculations for large simulation-based design studies. Since the framework requires performance data that is equally spaced in time, digital signal processing methods are considered.

A simple method of digital filtering replaces each data value with a linear combination of nearby values [93]. Assume data is provided as a series of N equally spaced values where f_i is the value associated with the ith data point, $i = 1 \ldots N$. For performance data, f_i would represent the system performance at the ith time step. Each data value f_i can then be replaced by g_i, a linear combination of nearby values such that

$$g_i = \sum_{j=-M}^{M} c_j f_{i+j}, \tag{28}$$

where M is the moving window or interval half-width and c_j are weighting coefficients. The number of data points included in the moving window is defined by $2M + 1$.

A moving average filter replaces f_i with the average of the values in the moving window, such that the weighting coefficients in eq. (28) are defined as

$$c_j = c = \frac{1}{2M+1}. \tag{29}$$

The only tuning parameter for this type of filter is the window half-width M. This filter introduces no bias into the data if the underlying functional shape of the data is constant or linear. However, if the functional shape is non-linear, as is likely with performance data for SoS scenarios, this filter will introduce undesirable bias into the data. Additionally, though moving average filters are able to capture general trends in the data, such as the area and location of peaks, they often smooth peaks too much when the width of the peak is on the order of the window size [64].

An alternative data filter is the Savitzkey-Golay (S-G) filter [95, 93, 96]. S-G filters use least-squares polynomial fitting to smooth data. For each data point i with corresponding value f_i, this method fits an nth order polynomial

$$p_i(t) = \sum_{k=0}^{n} a_k t^k = a_0 + a_1 t + a_2 t^2 + \cdots + a_n t^n, \tag{30}$$

to all $2M + 1$ data points in the moving window surrounding the ith data point. The data value f_i is then replaced by $p_i(t = i)$, the value of the p_i polynomial at position $t = i$. A new polynomial, $p_{i+1}(t)$, is used to replace the next data value f_{i+1}. Linear matrix inversion is used to determine a set of filter coefficients c_j that enables the use of eq. (28) with least-squares polynomial smoothing, avoiding the process of calculating a new polynomial for each data point. The only tuning parameters for S-G filters are the polynomial order n and window half-width M.

S-G filters improve the ability of simple moving average filters to maintain the shape of narrow peaks in the data, at the expense of removing less noise [64, 91]. Figure 19 shows a comparison of moving average and S-G filters. The moving average filter removes more noise from the data than the S-G filter, while maintaining the

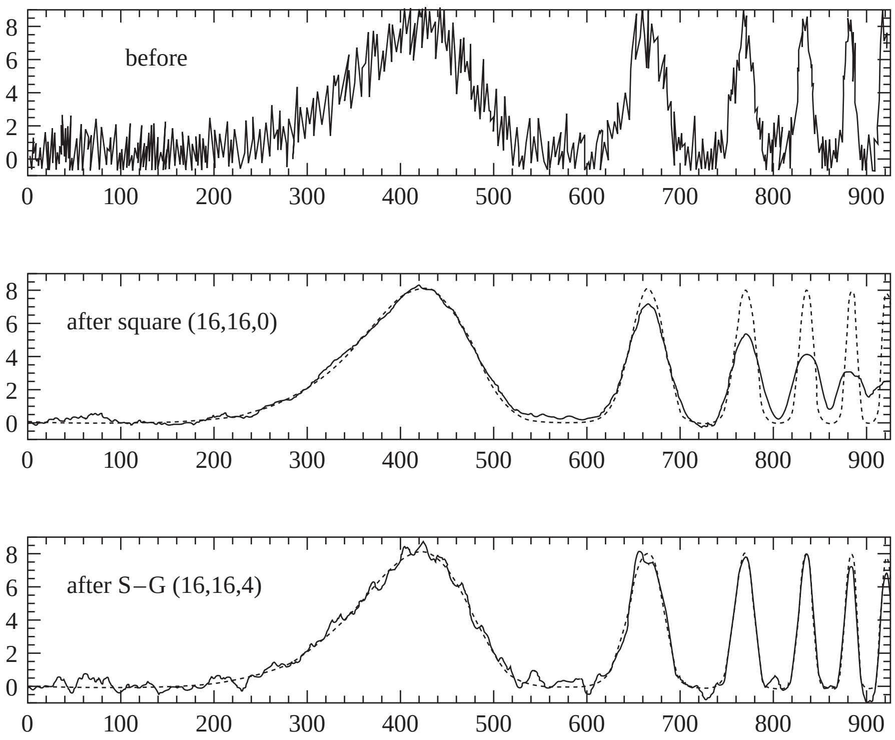


Figure 19: Comparison of (top) raw synthetic data with additive Gaussian white noise, (middle) moving average smoothed data, and (bottom) S-G filter smoothed data [93]. Data is smoothed with a window size of 33 for both filters (i.e., $M = 16$), and a polynomial order of four for the S-G filter (i.e., $n = 4$)

location of peaks. However, the S-G filter captures peak heights better than the moving average filter, especially for narrow width peaks. Since system degradation and recovery may produce large, narrow peaks in performance data, S-G filters are preferred for smoothing performance data. S-G filters are therefore selected for use in the resilience assessment framework, summarized in the following response to research question 3.1:

> **Response to RQ 3.1:** Savitzky-Golay filters are used to smooth performance data, due to their ability to capture desired trends in noisy and peaked data, while still removing undesired noise.

For the remaining discussion of the assessment framework, $y_{raw}(t)$ represents raw performance data and $y(t)$ represents the raw data smoothed using an S-G filter.

Smoothed data is used for all resilience metric calculations. Raw data is only used for calculation of the volatility factor. This thesis uses $n = 3$ and $m = 5$ for all resilience calculations, based on experimentation with performance data from the IE network model.

System Resilience Calculation

The final step in the framework is to calculate system resilience, using performance data. A resilience metric, R, is defined to quantify a system's resilience to a single disruption event. Providing a single metric for resilience enables quantitative comparisons and trade-offs studies between potential system designs. However, systems may also face multiple disruption events within a time period of interest. To account for such scenarios, a total resilience metric, R_{total}, is also defined. Section 4.2 defines these metrics and describes steps used to calculate them.

4.2 Resilience Metric Definitions

The resilience metric R is developed from the requirements set for a method to assess SoS resilience and established characteristics of resilience. This metric is based on the integration metric proposed by Vugrin et al. [112] and the resilience factor proposed by Francis and Bekera [48] (see section 2.2). The integration metric provides a quantitative method of capturing the total performance maintained by a system throughout a scenario. The metric proposed by Francis and Bekera additionally considers recovery time, as well degraded and recovered system capabilities. Therefore, combining these metrics accounts for the overall performance of a system, how quickly it recovers lost capabilities, the absorptive capacity of a system, and the restorative capacity.

However, these metrics do not consider the ability of a system to smoothly provide desired capabilities (i.e., penalize volatile systems). Additionally, Francis and Bekera provide limited guidance for how to determine certain values within their recovery time calculation. The proposed metric R addresses these limitations by incorporating a volatility factor and providing a traceable method for calculating a recovery time factor. The proposed metric explicitly accounts for desired characteristics of a resilient system through a set of resilience factors, which are used to calculate the system resilience, R, as

$$R = \begin{cases} \sigma\rho[\delta + \zeta + 1 - \tau^{(\rho-\delta)}] & \text{if } \rho - \delta \geq 0 \\ \sigma\rho\,(\delta + \zeta) & \text{otherwise,} \end{cases} \tag{31}$$

where $0 \leq R \leq \infty$, and R has a reference value of two for a normal operating scenario. A normal operating scenario is defined as one in which the system shows no loss of desired performance over time [i.e., $y(t)$ is some constant C for all t].

This metric is based on the integration of performance data, defined as the *total performance factor*, σ, similar to other resilience metrics in the literature [92, 94, 112]. Integration-based resilience is then modified by incorporating a set of resilience factors to explicitly account for various aspects of resilience not considered by simple integration. The *recovery factor*, ρ, accounts for the end state of the system (i.e., its recovered capability or performance level). The *absorption factor*, δ, accounts for the ability of the system to absorb the effects of a disruption. A *volatility factor*, ζ, accounts for the volatility of performance data, representing the ability of the system to smoothly transition from one state to another. A normalized *recovery time factor*, τ, accounts for temporal aspects of the system response through calculation of the time required to reach steady-state following a disruption. The influence of τ decreases as the recovered performance level (i.e., $\rho - \delta$) decreases. The conditional statement in eq. (31) ensures that systems are only rewarded for quickly reaching steady-state (i.e., having low values of τ) if their recovered performance level is better than their minimum performance. The following sections describe the calculation of each resilience factor (refer to fig. 18 for descriptions of variables used in the calculations that are obtained from performance data).

Total Performance Factor

The *total performance factor*, σ, accounts for the total performance maintained by a system throughout the time period of interest. This factor is calculated as

$$\sigma = \frac{\sum_{t=t_0}^{t_{final}} y(t)}{y_D(t_{final} - t_0)}, \tag{32}$$

for discrete time performance data, and as

$$\sigma = \frac{\int_{t=t_0}^{t_{final}} y(t)}{y_D(t_{final} - t_0)}, \tag{33}$$

for continuous time data, where $0 \leq \sigma \leq \infty$, and σ has a value of one in a normal operating scenario. This metric rewards systems able to provide high performance levels *at any point of time* during a scenario. However, it does not account for *when* that performance is provided (see section 2.2).

Absorption, Recovery, and Recovery Time Factors

The ability of a system to absorb the impacts of a disruption and recover lost capabilities in a timely manner is critical to its resilience [112, 48]. The *absorption factor*, δ, captures how well a system is able to absorb a disruption and reduce its impact on system performance. This factor is calculated as

$$\delta = \frac{y_{min}}{y_D}, \tag{34}$$

where $0 \leq \delta \leq \infty$, and δ has a value of one in a normal operating scenario. The minimum of the performance data is used for y_{min}.

The *recovery factor*, ρ, captures how well a system is able to recover lost capabilities. This factor is calculated as

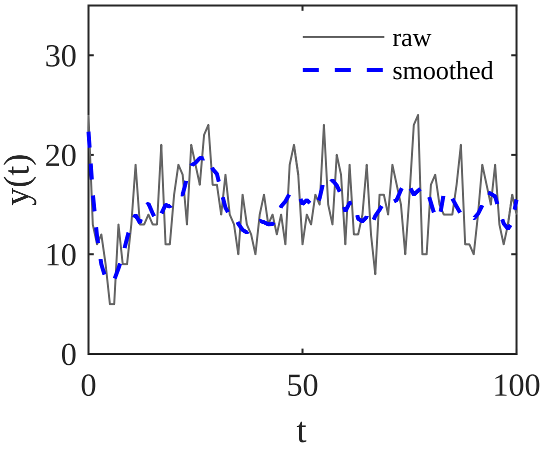


Figure 20: Example showing raw performance data, $y_{raw}(t)$, smoothed with an S-G filter $(n = 3, M = 5)$ to give $y(t)$. The smoothing reduces much of the noise in the raw data; however, a method for calculating the recovery time and recovered capability or performance level is still required.

$$\rho = \frac{y_R}{y_D}, \tag{35}$$

where $0 \le \rho \le \infty$, and ρ has a value of one in a normal operating scenario. Determining the recovered performance, y_R, is a simple task when analyzing a small number of scenarios with clearly defined recovery values, such as that seen in fig. 18. One can perform a visual inspection of the data, or if automated calculation is desired, select the final performance as the recovered value. However, for automated analysis of data sets where the system does not recover to a clearly defined value (e.g., those from stochastic simulation studies), determining y_R is a non-trivial task. This difficulty can exist even when using data smoothed with an S-G filter (see fig. 20).

To account for these situations, y_R is defined to be the steady-state performance (i.e., recovered capability) of the system following a disruption event and possible implementation of a recovery action. Using the steady-state performance provides a more accurate representation of what level the system has recovered to than, for instance, using the final observed system performance. The use of steady-state performance creates the following research question:

Research Question 3.2: What method should be used to estimate the steady-state time and value of system performance data?

The following are requirements for selecting a steady-state estimation method to be used in the resilience assessment framework:

- Accuracy - The method should accurately estimate steady-state time and value.
- Robustness - The method should work well with a variety of system types.
- Easy automation - The method should be easily automated (requiring little user input) and computationally inexpensive for use with large simulation design studies.

The problem of finding the steady-state recovered performance level is similar to that of the initial transient for steady-state estimation in stochastic simulation output analysis [69]. Consider a discrete-time, stochastic process $Y_1, Y_2, \ldots$ from the output of a single run of a non-terminating simulation. Each random variable Y_i represents the output of the simulation at time step i; for performance data, Y_i represents the performance of the system at the ith time step. Let $F_i\left(y \mid I\right) = P\left(Y_i \leq y \mid I\right)$, where y is a real number in the set of possible values for Y_i, $i = 1, 2, \ldots$, and I represents the initial conditions of the simulation. Then $F(y)$ is the steady-state distribution of the stochastic process $Y_1, Y_2, \ldots$ if $F_i\left(y \mid I\right) \rightarrow F(y)$ as $i \rightarrow \infty$. The steady-state random variable Y is the random variable with distribution $F(y)$. Steady-state estimation for simulation output analysis attempts to estimate the time index k at which $F_k\left(y \mid I\right)$ is approximately the same as $F_{k+1}\left(y \mid I\right), F_{k+2}\left(y \mid I\right), \ldots$, or equivalently Y_k has approximately the same distribution as $Y_{k+1}, Y_{k+2}, \ldots$. This estimation does not mean that the observed output values $y_k, y_{k+1}, \ldots$ from a given simulation run are equal; it instead means that the distributions of those corresponding random variables are approximately equal.

Assuming a steady-state random variable Y of a stochastic process, the steady-state mean $\nu = E(Y)$. Since most simulations have initial conditions different from those at steady-state, the sample mean, $\bar{Y}(N)$, calculated from a sample size N of observed output values, is a biased estimater of ν (i.e., the mean of observed output values from a simulation run of length N is a biased estimator of the true steady-state mean). The sample mean is calculated as

$$\bar{Y}(N) = \frac{\sum_{i=1}^{N} Y_i}{N}, \tag{36}$$

where Y_i is the ith observed value and N is the sample size, or number of observations. The difficulty of estimating ν given a sample of simulation output data is often referred as the problem of the initial transient or initialization bias.

The most common method for dealing with the problem of the initial transient is determining a simulation warm-up period, deleting the data determined to be within that warm-up period, and using the remaining data to estimate ν [69]. Using the warm-up period deletion method, the simulation steady-state mean ν can be estimated as the truncated mean, $\bar{Y}(N,d)$, calculated as

$$\bar{Y}(N,d) = \frac{\sum_{i=d+1}^{N} Y_i}{N-d}, \tag{37}$$

where the steady-state time index d determines the end of the warm-up period and $d+1$ determines the beginning of the truncated data. The truncated data includes all Y_i where $i > d$.

The length of the warm-up period, d, can be used to estimate the time when steady-state is reached for system performance data. The truncated steady-state mean, $\bar{Y}(N,d)$, can then be used as the recovered system performance level, y_R. Simulation warm-up period estimation methods are therefore reviewed for use in the resilience assessment framework.

Hoad, Robinson, and Davies [56] provide a comparison of methods for estimating the length of the warm-up period for simulation output data. They review 44 methods from the literature, which are categorized as being graphical methods, heuristic methods, statistical methods, initialization bias tests, and hybrid methods. Their criteria for evaluating the methods are accuracy and robustness of the method, ease of automation, generality, and computer time. Since these criteria match the requirements set for a steady-state estimation method for this framework, and their study provides a thorough comparison of methods, their results are used to answer research question 3.2.

The results from Hoad, et al. identify the MSER-5 method as the method that best satisfies their criteria for a warm-up estimation method. The authors find MSER-5 to be a general method (i.e., not model or data type specific) that is easily automated, as it does not require many input parameters or user actions. They find the method to be robust and effective for the considered cases. Similar conclusions are drawn by White and Spratt in a previous comparison of warm-up estimation methods [116].

The Marginal Standard Error Rules (MSER) method is a heuristic method for estimating the length of the warm-up period in simulation data [114, 115, 69]. This method estimates the length of the simulation warm-up period, d^*, using the MSER statistic, as

$$d^* = \underset{d \geq 0}{\arg\min}\, \mathrm{MSER}(N, d) \tag{38}$$

$$\mathrm{MSER}(N, d) = \frac{1}{(N-d)^2} \sum_{i=d+1}^{N} (Y_i - \bar{Y}(N, d))^2. \tag{39}$$

For system performance data, N is the number of observed performance values and Y_i is the ith performance value. White and Robinson [115] provide a description of the intuition behind this method.

MSER-m modifies MSER by applying eq. (39) to batch means data, rather than raw simulation output data [115]. This modification groups the data into batches of m observations each, and calculates steady-state from the mean value of each batch. A batch size of $m = 5$ has shown improved performance for some cases compared to MSER and other methods [116, 56]. However, using batched instead of raw data reduces the resolution with which steady-state can be identified. Since the assessment framework does not specify a required length of simulation data, using batched data may not provide the desired level of resolution for steady-state estimates. Additionally, the benefits of MSER-5 over MSER are not clearly established for a wide variety of cases. Therefore, MSER is selected for calculating steady-state time.

One issue with using the MSER method with performance data is that typical applications of the method assume steady-state is reached within the first half of the data (i.e., $d^* \leq N/2$). A reason for this assumption is that the MSER statistic naturally drops to very low values for the last few observations in a data set, since very few data points are used to calculate the mean-squared error for those observations (see fig. 21). This drop in MSER causes the algorithm to inaccurately estimate steady-state within the final few observations. Since systems are not guaranteed to reach steady-state in the first half of a data set, the last 10 percent of performance data points are defined as a buffer region for steady-state calculations. These observations are not considered as possible values for d^*, preventing the algorithm from incorrectly estimating steady-state at the end of the data set.

A second issue with the MSER method is that it does not determine if steady-state has actually been reached within a data set. A required MSER statistic threshold, $\text{MSER}_{\text{required}}$, is defined to account for this. If $\text{MSER}(N, d) > \text{MSER}_{\text{required}}$ for all $d = 0, 1, \ldots N$ outside of the buffer region, then the framework assumes steady-state is not reached. If steady-state is not reached, y_R is defined as the mean performance of the buffer region. This thesis uses $\text{MSER}_{\text{required}} = 0.1$ for all resilience calculations with the IE network model, based on experimentation with performance data from the model.

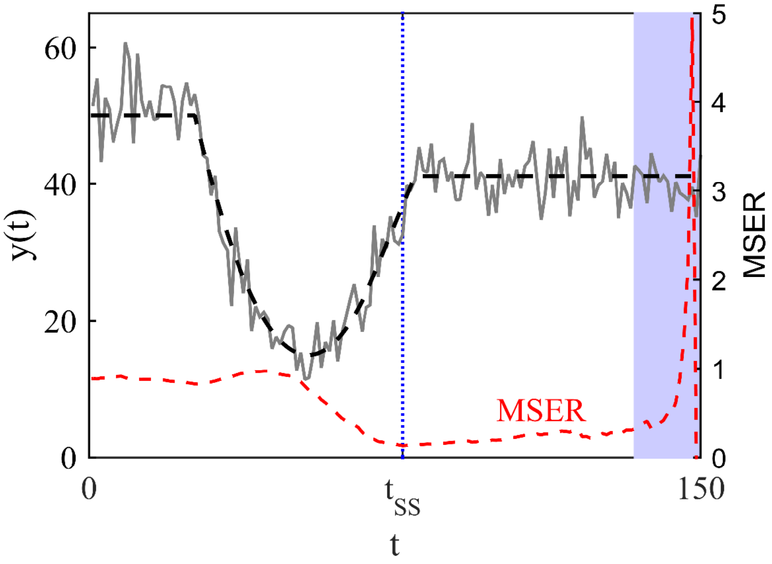


Figure 21: Example use of MSER to determine the steady-state time, t_{SS}, for notional system performance data with additive Gaussian white noise. The solid gray line shows the noisy performance data, with the dashed black line showing the true signal for the performance data (data with no additive white noise). The dashed red line shows the MSER value at each possible truncation time t. The dashed vertical line shows the selected steady-state time. The shaded area shows the buffer region not considered by the algorithm.

A third modification to the MSER method is the use of another MSER threshold, $\text{MSER}_{\text{threshold}}$, which defines the minimum MSER required to assume steady-state has been reached (i.e., steady-state is reached once $\text{MSER}(N, d) \leq \text{MSER}_{\text{threshold}}$). This threshold is implemented to account for cases where performance values decrease so slowly over time that the algorithm estimates steady-state at the end of the data set even though very little change is seen in performance values past the first observation where the threshold is reached. This thesis uses $\text{MSER}_{\text{threshold}} = 0.0001$ for all resilience calculations with the IE network model, based on experimentation with performance data from the model.

Figure 21 shows an example use of the MSER method for estimating steady-state time in notional performance data. The estimated steady-state time is $t_{SS} = 77$. The true steady-state time (determined using the true signal with no additive noise) is 80, demonstrating the accuracy of this method for this example case. The following answer is given to research question 3.2:

> **Response to RQ 3.2:** The MSER method is used to estimate steady-state time and value of system performance data, due to its accuracy, robustness, and ability to be automated for large-scale simulation studies.

Using the MSER method, the steady-state recovery time of a system, t_{ss}, is defined as the time associated with the $d^* + 1$ observation in performance data. For example, for performance data beginning at time $t = 1$ and incremented in one second time steps (i.e., $t = 1, 2, \dots$), if $d^* = 10$ then $t_{ss} = 11$. The steady-state performance of the system, y_R, is defined as the truncated mean following steady-state,

$$y_R = \frac{1}{N - d^*} \sum_{t=t_{ss}}^{t_{final}} y(t). \tag{40}$$

Steady-state recovery time is used to define the *recovery time* factor, τ, which captures the speed with which a system recovers to steady-state. This factor accounts for the ability of a system to not only respond to a disruption and recover lost capabilities, but to also do so in a timely manner. The recovery time factor is calculated as the time until steady-state recovery is reached, relative to the total time period considered,

$$\tau = \frac{d^*}{N}, \tag{41}$$

where $0 \leq \tau \leq 1$, and τ has a value of zero in a normal operating scenario.

Volatility Factor

The ability of a system to provide smooth transitions from degraded to recovered states is accounted for with the *volatility factor*, ζ. A system with highly volatile performance data, such as that in fig. 21, sees large and sudden drops in performance over time. This type of system is defined to be less resilient than a system that produces similar, but smoother performance data. Signal-to-noise ratio ($\mathrm{SNR}_{\mathrm{dB}}$) is used to quantify volatility in performance data, calculated as

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log_{10} \frac{P_s}{P_n} \tag{42}$$

$$P_s = \sum_{t=t_0}^{t_{final}} S(t)^2 \tag{43}$$

$$P_n = \sum_{t=t_0}^{t_{final}} N(t)^2, \tag{44}$$

where P_s is the signal power, P_n is the noise power, $S(t)$ is the true signal, and $N(t)$ is signal noise. The true signal is approximated with smoothed performance data, such that $S(t) = y(t)$. Signal noise is approximated as the difference between the raw data, $y_{raw}(t)$, and the smoothed data, using

$$N(t) = y_{raw}(t) - y(t) \tag{45}$$

Figure 22 shows approximated performance noise for the raw and smoothed performance data from fig. 20.

$\mathrm{SNR}_{\mathrm{dB}}$ is converted to the volatility factor, ζ, using a logistic function transformation as follows,

$$\zeta = \frac{1}{1 + \exp\left[-0.25 \left(\mathrm{SNR}_{\mathrm{dB}} - 15\right)\right]}, \tag{46}$$

where $0 \leq \zeta \leq 1$, and ζ has a value of one in a normal operating scenario. The function steepness (set to -0.25) and offset (set to 15) are set to ensure that performance data with no volatility provides a volatility factor of one, with a gradual decline in ζ as volatility is increased.

Accounting for volatility in this manner requires systems being compared to have

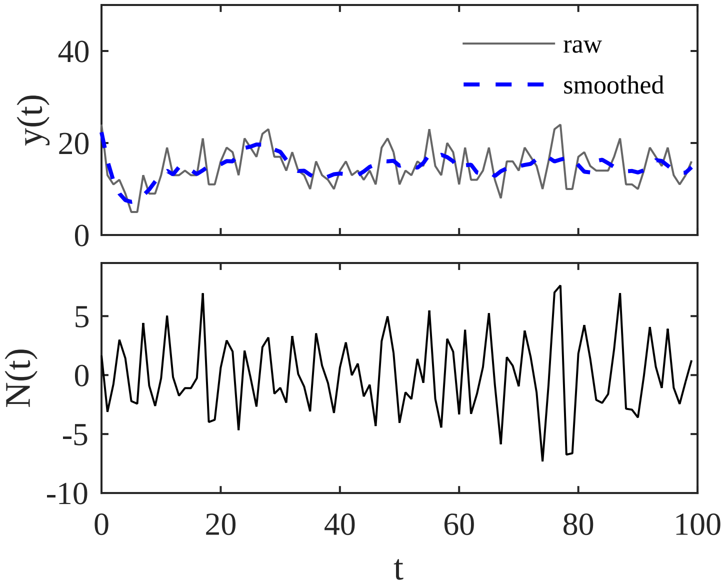


Figure 22: Example plot of (top) raw and smoothed performance data, with (bottom) corresponding approximations for performance noise. The raw data is smoothed using an S-G filter ($n = 3$ and $m = 5$).

similar performance measurement noise. For example, noise due to sensor measurements needs to be comparable between systems, to prevent penalizing one system more than another due to measurement noise, rather than response volatility.

Total Resilience Metric

Scenarios with multiple disruptions over time (see fig. 23) are also considered. Since the resilience metric, R, is defined for a scenario with a single disruption, multiple disruption scenarios are split into N_{epochs} epochs, where each epoch is defined to contain a single disruption event and the subsequent recovery action. System resilience for the ith epoch, R_i, is calculated using eq. (31). System *total resilience*, R_{total}, is then calculated for the entire scenario using an exponentially weighted mean of R_i from each epoch, as

$$R_{total} = \sum_{i=1}^{N_{epochs}} w_i R_i, \tag{47}$$

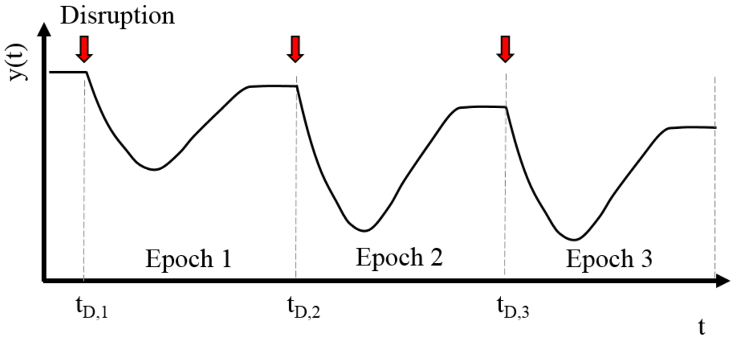


Figure 23: Notional example of performance data for a system subjected to multiple disruptions, with a recovery action implemented after each disruption. This scenario is split in three epochs for calculation of $R_1, R_2, R_3, \text{and} R_{total}$.

where $0 \leq R_{total} \leq \infty$. The weights, w_i, are defined as normalized coefficients of an exponentially weighted moving average filter,

$$w_i = \frac{\alpha(1-\alpha)^{N_{epochs}-i}}{\sum_{j=1}^{N_{epochs}} \alpha w_j} = \frac{(1-\alpha)^{N_{epochs}-i}}{\sum_{j=1}^{N_{epochs}} w_j}, \tag{48}$$

with smoothing factor α, where $0 \leq \alpha \leq 1$. Higher values of α increase the weight given to later values of R_i (i.e., R_i values associated with epochs that occur towards the end of the scenario are more heavily weighted than those occurring earlier in the scenario). A weighted mean is used to give preference to systems that improve their resilience over time, representing the ability of a system to adapt to disruptions. This weighting incorporates the adaptive capacity of resilience into the framework. All calculations in this thesis use $\alpha = 0.06$, a value commonly used in the analysis of time series data.

4.3 EXP 1: Resilience Metric Comparison

Purpose of the experiment: Test hypothesis 3 that the developed resilience metric captures the desired aspects of resilience better than an integration-based metric commonly used in the literature.

Having developed a resilience metric, R, and total resilience metric, R_{total}, the following hypothesis (HYP 3) is formed in response to research question three:

> **Hypothesis 3:** The resilience metric R provides a quantitative metric for assessing SoS resilience that captures the desired aspects of resilience better than an integration-based metric commonly used in the literature.

This hypothesis focuses on R, rather R_{total}, because R is used to calculate R_{total}. Additionally, most resilience metrics from the literature focus on a single threat event, providing no comparison metrics for R_{total}.

Experimental Setup

Experiment one (EXP 1) is performed to test this hypothesis. This experiment compares R and a resilience metric based on performance integration. Integration is used for comparison because it is often used in the literature for quantifying system resilience [92, 94, 112], and like R, provides a single value for resilience. These comparisons focus on the ability of both metrics to capture expected trends and desired aspects of resilience for notional data, based on the definition of resilience accepted for this thesis. The integration metric, I, is calculated as

$$I = \sum_{t=t_0}^{t_{final}} y(t). \tag{49}$$

A logistic function is used to create notional performance data for comparing R and I. The "S"-curve shape of this function notionally captures the shape of performance data for systems undergoing degradation and recovery processes. This data would represent the number of received messages in the IE network example problem. Data representative of a system facing a single disruption with no recovery action is generated with

$$y(t) = A_1 + \frac{K_1 - A_1}{1 + \exp\left[B_1\left(t - x_1\right)\right]} + N(0, \sigma), \tag{50}$$

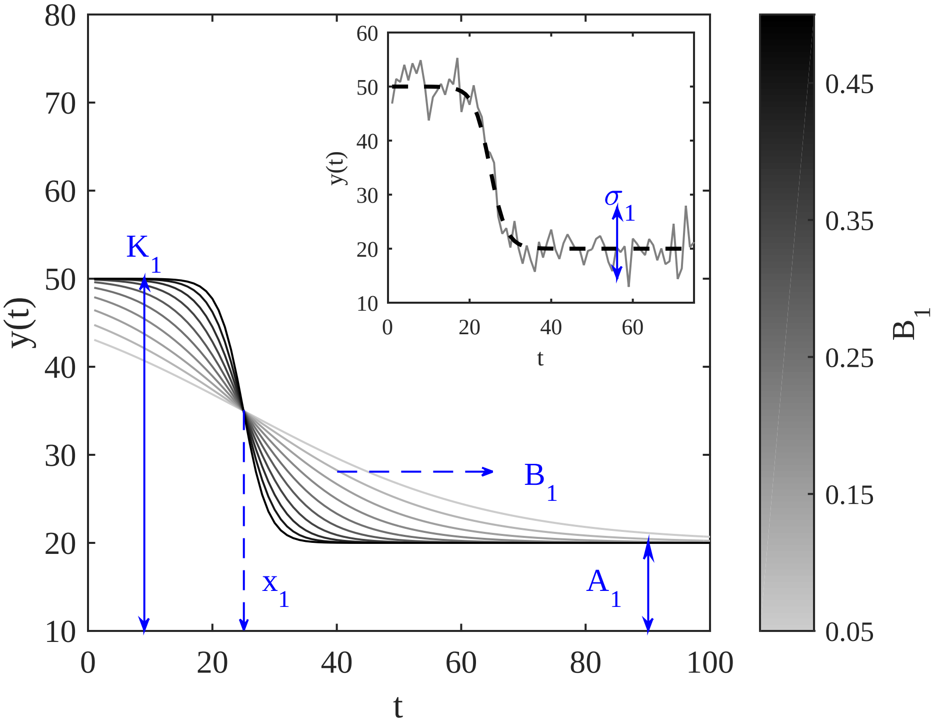


Figure 24: Notional performance data created using eq. (50), with the inflection steepness, B_1, varied $(A_1 = 20, K_1 = 50, x_1 = 25, \sigma = 0)$. The inset figure shows how white noise $(\sigma_1 = 3)$ is used to model performance volatility, where the dashed line shows the corresponding data with no volatility (i.e., $\sigma_1 = 0$).

where A_1 determines the minimum performance level (i.e., the lower asymptotic limit), K_1 determines the initial performance level (i.e., the upper asymptotic limit of the function), B_1 determines the inflection steepness, and x_1 determines the location of the inflection point on the x-axis. Volatility is modeled as Gaussian white noise, added to the data by drawing from a normal distribution with mean of zero and standard deviation σ. Figure 24 shows the function shape with B_1 varied.

A recovery action is added to notional performance data using

$$y(t) = A_2 + \frac{K_2 - A_2}{1 + \exp\left[B_2\left(t - x_2\right)\right]} + N(0, \sigma) \tag{51}$$

for the recovery portion of the data, where a negative value is used for B_2 to create an increasing function. Equation (50) is still used for the degradation portion of

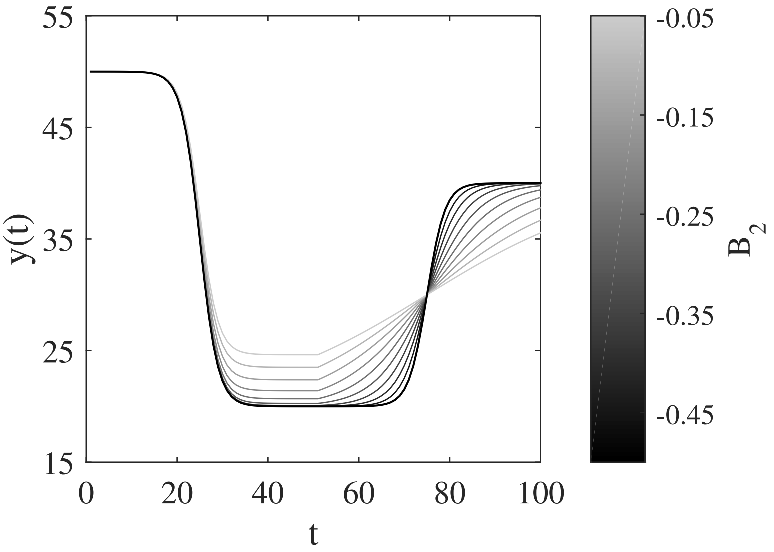


Figure 25: Capability data with a recovery action, where the recovery steepness, B_2, is varied ($K_1 = 50, B_1 = 0.5, x_1 = 25, A_2 = 20, K_2 = 40, x_2 = 70, \sigma = 0$). Data for $0 \leq t \leq 50$ is created using eq. (50), while data for $50 < t \leq 100$ is created using eq. (51).

the data, with A_1 adjusted to ensure a smooth transition between the recovery and degradation portions of the data. Figure 25 shows an example of data with a recovery action added and B_2 varied.

Equations (50) and (51) are used to generate notional performance data for comparisons between R and the integration metric, I. Logistic function parameters are parametrically varied to compare the behavior of these two metrics as the shape of the data changes. These comparisons provide an understanding of how well each metric is able to capture desired resilience trends for potential performance data shapes. Results focus on scenarios with a recovery action (e.g., fig. 25), since the ability to recover is an important characteristic of a resilient system. Table 4 shows logistic function parameters varied for this experiment. Similar results are seen for cases without a recovery action (e.g., fig. 24), when corresponding parameters of the degradation function are varied. Resilience calculations for R use data smoothed by an S-G filter ($n = 3$ and $m = 5$). Resilience calculations for I use raw performance data.

Table 4: Logistic function parameters settings for EXP 1

Parameter	Range	Description
K_2	$[20, 50]$	Determines recovered performance level
x_2	$[70, 90]$	Determines location of inflection point
B_2	$[-0.5, -0.05]$	Determines inflection steepness
σ_2	$[0, 3]$	Determines performance volatility

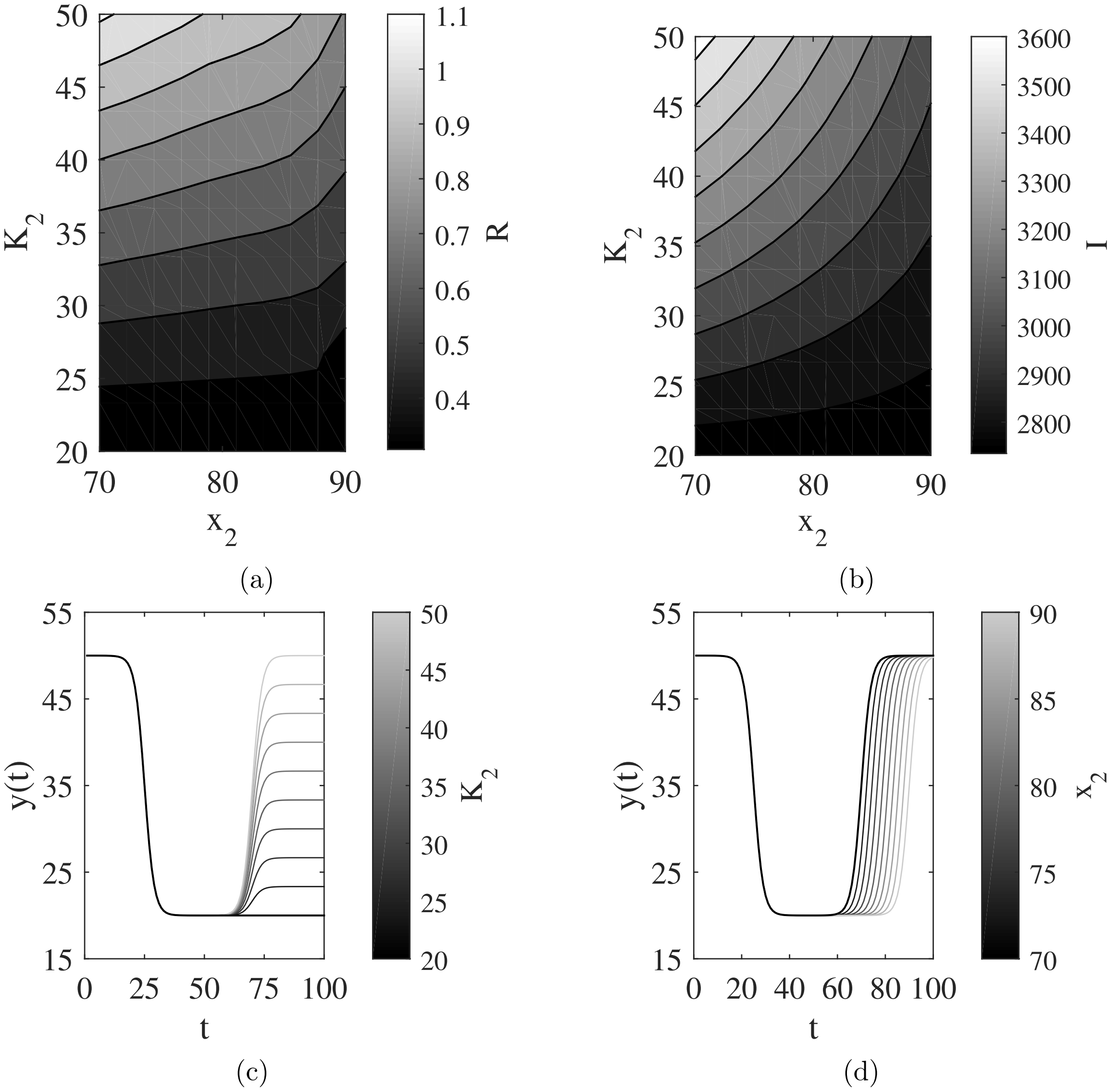
Experiment Results

Figure 26 shows a comparison of R and I for scenarios where the recovered performance level, K_2, and the location of the recovery inflection point, x_2, are varied. Both metrics capture expected resilience trends, since R and I increase as K_2 is increased and x_2 decreased. Increasing K_2 should improve resilience because systems that recover to a higher performance level are more resilient, assuming that all other factors are held constant. Decreasing x_2 should improve resilience because systems that recover faster are more resilient.

Figure 27 shows that R and I have different trends when the recovery steepness, B_2, and volatility, σ_2 are varied. Considering scenarios with no volatility added (i.e., $\sigma_2 = 0$), resilience measured by R increases as B_2 decreases. However, the opposite trend is seen when resilience is measured with I. This difference in trends is primarily due to the recovery factor, σ, used in the calculation of R. This recovery factor rewards systems that are able to recover to a higher performance level, such as those with low B_2 values in these scenarios. The recovery time factor, τ, also rewards scenarios with low B_2 values because of their fast recovery times. In comparison, the integration calculation of I only considers the total performance maintained throughout the entire scenario. Therefore, I rewards scenarios with higher B_2 values due to their ability to provide performance in the middle of the time period considered, despite lower recovered performance levels. Given these differences, R better captures resilience as defined for this work, since it focuses on the ability to recover lost capabilities.

This preference of R over I depends on the definition of resilience used. For example, there may be situations where the ability to absorb a disruption is more important than the ability to recover from it. Should an analyst wish to focus on particular aspects of resilience (e.g., absorption or recovery), weights can be added to the resilience factors to adjust R for specific needs.

R and I also show different trends when the volatility in performance data is varied using σ_2. Increasing σ_2 decreases resilience measured by R, due to the inclusion of the volatility factor, ζ, in the calculation of R. This volatility factor penalizes systems with highly volatile performance data, such as those represented by scenarios with high values of σ_2. In comparison, resilience measured by I is minimally affected by changes to σ_2. These results demonstrate the ability of R to account for performance volatility, a factor not considered by integration.



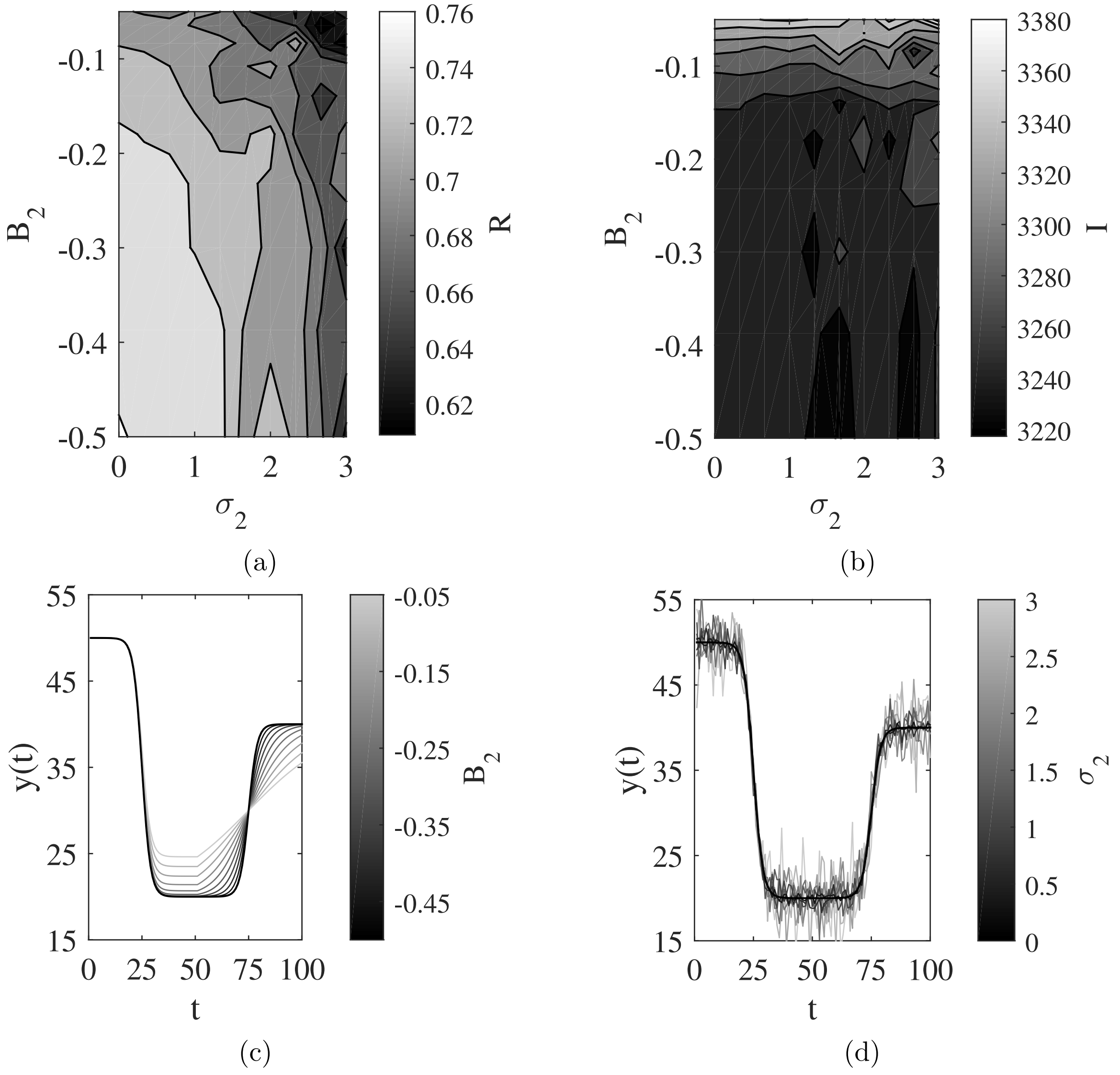
(a)

(b)

(c)

(d)

Figure 26: Resilience results for (a) R and (b) I for scenarios with K_2 and x_2 varied. These results are calculated using performance data shown in (c-d), where (c) K_2 is varied and (d) x_2 is varied $(K_1 = 50, B_1 = 0.5, x_1 = 25, A_2 = 20, \sigma = 0)$.



(a)

(b)

(c)

(d)

Figure 27: Resilience results for (a) R and (b) I for scenarios with B_2 and σ_2 varied. These results are calculated using performance data shown in (c-d), where (c) B_2 is varied and (d) σ_2 is varied ($K_1 = 50, x_1 = 25, B_1 = 0.5, A_2 = 20$).

Discussion of Results

The resilience metric R developed for the assessment framework is shown to better quantify resilience (as defined for this work) than the integration metric I. While other resilience metrics exist in the literature, some form of integration is found to be commonly used by the resilience engineering community. The following summarizes the outcome of experiment one, with respect to the hypothesis it aims to test:

> **Outcome of EXP 1:** Hypothesis 3 is confirmed since the resilience metric R better captures desired aspects of resilience than the integration metric I.

Since the resilience metric R captures the desired aspects of resilience, the developed assessment framework is used to assess SoS resilience. The following response is given to research question three:

> **Response to RQ 3:** The capability-based resilience assessment framework provides a set of metrics, R and R_{total}, that should be used to quantitatively assess SoS resilience.

4.4 Summary of Method for Resilience Assessment and Results

The capability-based resilience assessment framework provides a method for performing the first step of the ReSSNET methodology (see fig. 28). The following summarizes research questions, responses, and experiments from this chapter:

- RQ 3: What metric, or set of metrics, should be used to quantitatively assess SoS resilience?

- Response to RQ 3: The capability-based resilience assessment framework provides a set of metrics, R and R_{total}, that should be used to quantitatively assess SoS resilience.

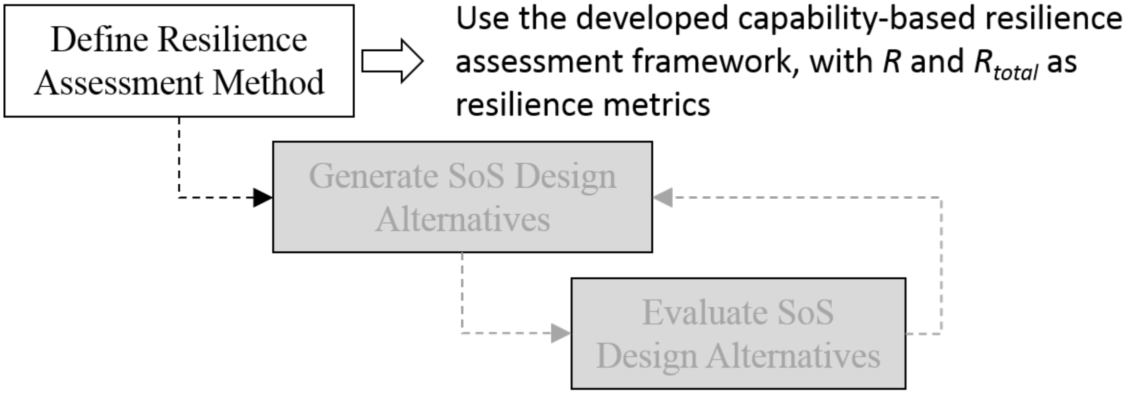

Define Resilience Assessment Method
Use the developed capability-based resilience assessment framework, with R and R_{total} as resilience metrics
Generate SoS Design Alternatives
Evaluate SoS Design Alternatives

Figure 28: Updated overview of the ReSSNET methodology with the selected resilience assessment method.

- RQ 3.1: What method should be used to smooth system performance data?

- Response to RQ 3.1: Savitzky-Golay filters are used to smooth performance data, due to their ability to capture desired trends in noisy and peaked data, while still removing undesired noise.

- Research Question 3.2: What method should be used to estimate the steady-state time and value of system performance data?

- Response to RQ 3.2: The MSER method is used to estimate steady-state time and value of system performance data, due to its accuracy, robustness, and ability to be automated for large-scale simulation studies.

- HYP 3: The resilience metric R provides a quantitative metric for assessing SoS resilience that captures the desired aspects of resilience better than an integration-based metric commonly used in the literature.

- Outcome of EXP 1: Hypothesis 3 is confirmed since the resilience metric R better captures desired aspects of resilience than the integration metric I.

CHAPTER V

A COMPLEX NETWORKS APPROACH TO DEFINING SOS ALTERNATIVES

The second step in the ReSSNET methodology requires a method for generating SoS design alternatives. An SoS design (e.g., an IE network) is defined by it's initial network topology and adaptation method, due to the networked nature of SoS and a focus on resilience achieved through adaptation for this thesis. Therefore, methods for defining potential network topologies and adaptation methods are needed to generate design alternatives. Since resilience depends on the specific threat faced, a method for defining potential network threats is also required. These needs are reflected in research question four (repeated here for convenience):

> **Research Question 4:** How should we define potential SoS network topologies, threats, and adaptation methods?

The initial network topology, adaptation method, and threat type are referred to as network design variables, forming a 3-dimensional network design space (notionally shown in fig. 29). Threat type is included in the design space despite being out of the control of SoS designers because the performance of SoS networks likely depends on the threat faced. The axes of this design space can be categorical or continuous, depending on the topologies, threats, and adaptation methods considered.

A complex networks approach is used to generate SoS design alternatives, due to the coevolutionary nature of network resilience (see section 2.3.2). Additionally, the focus on real world networks within the complex networks domain likely results in methods that are more applicable to real SoS problems than many graph-theoretic

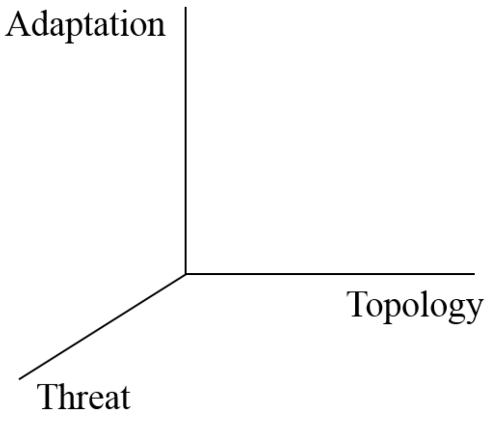


Figure 29: Notional network design space.

methods, which often focus on abstract graphs. Many complex networks also represent systems that have naturally evolved into their current state, with no clear central control figure and little external guidance. And yet despite the lack of a central designer, these systems often display a surprising level of efficiency and inherent robustness. Examples of such systems include the US economy, communication networks, transportation networks, societies, market systems, organisms, ant colonies, and ecosystems [76]. This observation further motivates a complex networks approach to generating SoS designs, and is summarized by the following quote:

> ...can something be learned from [complex systems and networks] that would help us build not only better airplanes and computers, but also smarter robots, safer buildings, more effective disaster response systems, and better planetary probes? - Minai et al. [76]

This chapter describes complex network methods selected to define potential network topologies, threats, and adaptation methods, as well as an experiment performed to validate the IE network model for the selected network designs.

5.1 Defining Network Topologies

Scale-free and random networks are used to represent two potential classes of network topologies for this thesis (see section 2.3.2 for a discussion of scale-free and random networks). Scale-free networks represent the class of networks with a heterogeneous, hub and spoke-like structure. These network hubs often improve the efficiency of networks (e.g., enabling fast message delivery in an IE network). Random networks represent the class of networks with a homogeneous, random structure. Scale-free networks are generated with the BA model. These networks are relevant to SoS designs because of their common occurrence in many naturally occurring and engineered networks. Random networks are generated with the ER model. Comparing scale-free networks to random networks provides an understanding of the importance of defining features in scale-free networks, since many of those features are not present in random networks (e.g., network hubs). Scale-free and random networks also show different behaviors when subjected to random and targeted node removals (see section 2.3.3), further supporting their use for comparisons of potential SoS network designs.

5.2 Defining Network Threats

The typical method for modeling complex network threats is targeted and random node removal. Targeted removals represent attacks where an adversary has knowledge of the network topology and uses that knowledge to intentionally damage a network. Varying the targeting strategy enables representation of a variety of threats or uncertainty in potential threats. Potential targeting strategies include targeting by node degree [10, 59], various centrality measures [111], geographic location [103], and cascading neighbors of a randomly selected node [38]. Random removals represent random failures of network nodes or unintentional network damage.

Targeted and random node removal is used to define potential SoS threats for this thesis because of the networked nature of SoS. Targeting is done by recalculated

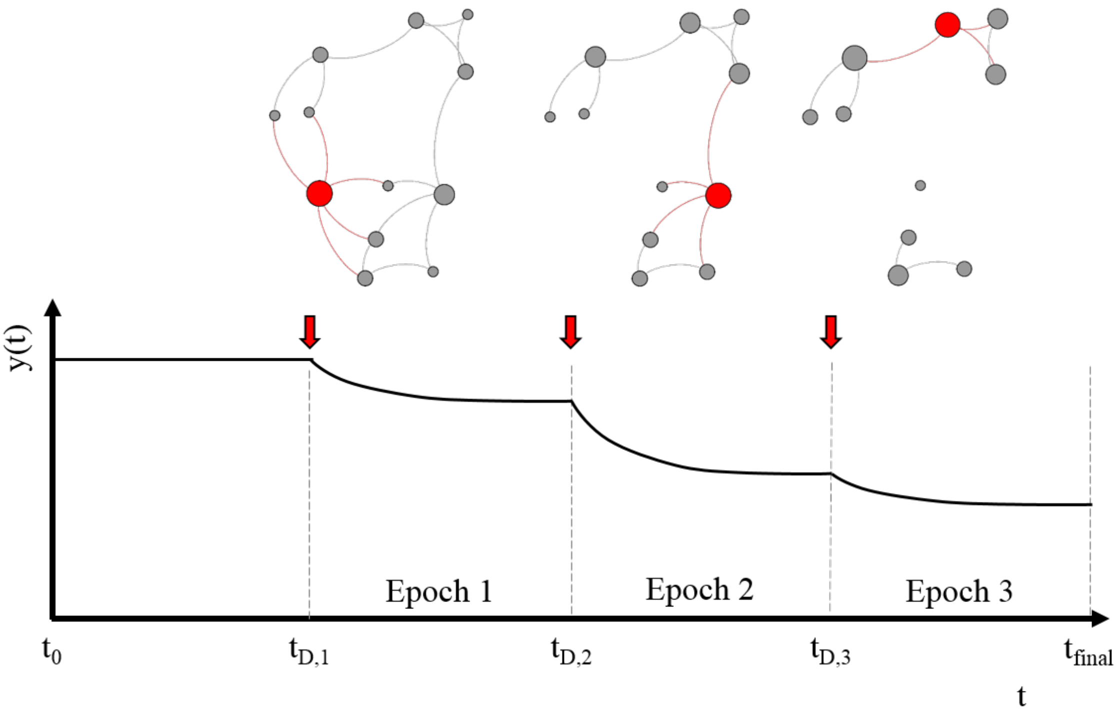


Figure 30: Notional example of performance data for an SoS network subjected to repeated, targeted node removals ($S_{threat} = 1, N_{threats} = 3$). The highlighted node and its incident links are removed at each attack, using the RD removal strategy. Node sizes are scaled by recalculated node degree, showing that the most connected node (largest in size) is removed at each attack.

node degree (RD), where the most connected node is removed at each attack, with node degrees being recalculated, or updated, following any change to the network topology. If there are multiple nodes with the maximum degree in the network, one of those nodes is randomly removed. RD targeting is used because it focuses on node connectivity, which is expected to be critical to the performance of SoS networks. Random removals (R) remove nodes uniformly at random. Random removals are used because they represents the opposite case of RD targeting, providing an understanding of the range of impacts threat types can have on SoS network resilience.

Removing a node models an attack on or failure of a system in an SoS. Once a node is removed from the network, all incident links are also removed (i.e., all links connected to that node are removed). A limitation of this method is that removals only allow modeling of threats that completely disable nodes. However, this approach can be extended to include partial node failures by having threats alter node properties, rather than removing nodes from a network.

Threat size, S_{threat}, is defined to specify the number of nodes removed in a single removal event (i.e., within a single time step). RD threats with $S_{threat} > 1$ remove nodes according to their degree at the beginning of a removal event, such that node degrees are only recalculated after all S_{threat} nodes are removed.

Repeated node removals are also considered, to represent scenarios in which an SoS is subjected to multiple network attacks over time. For threats with $N_{threats} > 1$ repeated removals, the same targeting strategy is used for each removal. Removal events are defined to occur at equal intervals in time within a given scenario, such that $t_{D,1} - t_0 = t_{D,2} - t_{D_1} = t_{D,3} - t_{D_2} = \ldots = t_{final} - t_{D,N_{threats}}$. Figure 30 shows a notional example of a repeated threat scenario.

5.3 Defining Network Adaptation Methods

Network adaptation to node removals is modeled by allowing neighbors of removed nodes to rewire lost links following threat, or node removal, events. More precisely, if node i is removed from a network, all neighbors of node i rewire their previous link with node i to another node in the network that they are not currently linked with (according to the network topology at the beginning of an adaptation event). Since nodes choose new neighbors based on the topology at the beginning of an adaptation event, there is a possibility for two rewiring nodes to choose to rewire links to each other, in which case one of those links does not get rewired and is lost. However, in the absence of such duplicate rewirings, $L\left(t_{D,i}\right) = L\left(t_{A,i}\right)$, where $L\left(t_{D,i}\right)$ is the number of links in the network before the ith removal event, $L\left(t_{A,i}\right)$ is the number of links in the network after the corresponding adaptation event, and $t_{D,i} < t_{A,i}$. If a rewiring node is already linked with all remaining nodes in the network, no rewiring

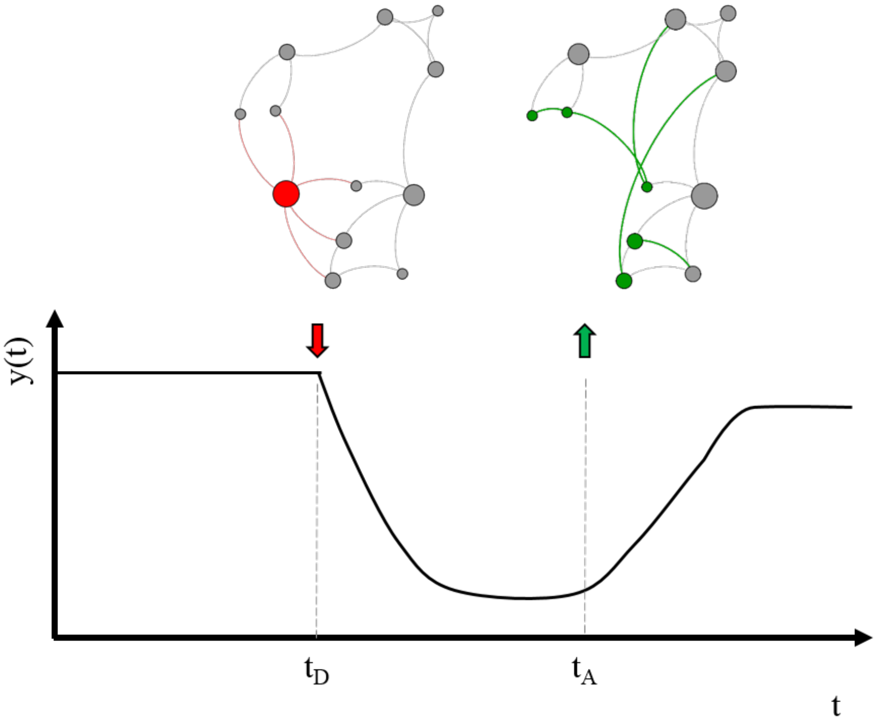


Figure 31: Notional example of performance data for an SoS network that adapts to a targeted node removal through random rewiring. The red highlighted node and its incident links are removed at time t_D. The network is then allowed to rewire lost links (highlighted in green) at time t_A. Node sizes are scaled by recalculated node degree.

action is taken by that node.

A delay time, t_{adapt}, is incorporated into the adaptation model, such that $t_{A,i} = t_{D,i} + t_{adapt}$, where $t_{A,i}$ is the adaptation time and $t_{D,i}$ is the threat event time. This time delay represents the time required for network nodes to rewire links following a threat event. Figure 31 shows a notional example of network adaptation following a threat event.

Recalculated node degree, preferential attachment, and random rewiring are used to define how nodes choose to rewire links for potential adaptation methods. Similar to the use of scale-free and random networks as potential topologies, using these adaptation methods provide an understanding of the range of effects adaptation methods can have on SoS network resilience.

Recalculated degree adaptation (RDA) is based on recalculated degree-based threats (RD threats). This adaptation method requires rewiring nodes to choose the most connected nodes in the network as new neighbors. Therefore, a node rewiring $n_{rewires}$ links in an adaptation event (i.e., within a given time step) will rewire to the $n_{rewires}$ most connected nodes it is not already linked with. Node degrees are recalculated at the start of every adaptation event. This adaptation method should create network hubs over time, since rewiring nodes always link with the most highly connected nodes. For the IE network model, these hubs should improve the ability of a network to quickly share information and make the network more resilient to future random removals. However, these hubs may also make a network more susceptible to targeted attacks [10].

Preferential adaptation (PA) is based on the BA preferential attachment model and defines the probability, $\prod(k_i)$, that a rewiring node chooses to rewire a link to node i, n_i, as

$$\prod(k_i) = \begin{cases} k_i / \sum_{j=1}^{|\mathcal{V}|} k_j & \text{if } n_i \in \mathcal{V} \\ 0 & \text{otherwise,} \end{cases} \tag{52}$$

where k_i is the degree of node i and $\mathcal{V}$ is the set of all nodes the rewiring node is not currently linked with. This adaptation should also create network hubs over time, though the connectivity of those hubs should be less than with RDA because there is a non-zero probability of choosing nodes with low connectivity.

Random rewiring is modeled by allowing rewiring nodes to choose new neighbors uniformly at random from the set of possible neighbors, $\mathcal{V}$. This method defines the probability, $\prod(k_i)$, that a rewiring node chooses to rewire a link to node i, n_i, as

$$\prod(k_i) = \begin{cases} 1/|\mathcal{V}| & \text{if } n_i \in \mathcal{V} \\ 0 & \text{otherwise,} \end{cases} \tag{53}$$

Random rewiring is considered because similar random adaptation methods have been shown to improve network robustness [97, 71, 111]. However, this implementation of random adaptation differs from previous ones in the literature by focusing on rewiring existing links, rather than link re-addition or link swaps (see section 2.3.3 for further discussion of previous work in the literature). This adaptation method should limit the presence of network hubs since node degrees are not considered when nodes choose new neighbors. A lack of network hubs should provide less network efficiency than RDA and PA, but more resilience to targeted attacks.

These methods range from being deterministically defined by node degree (RDA), to being probabilistically defined by node degree (PA), to having no consideration of node degree (random rewiring). Progressing from RDA to PA to random rewiring can also be viewed as increasing the randomness allowed in network adaptation.

The selected methods for defining potential network topologies, threats, and adaptation methods are summarized in the following answer to research question four:

> **Response to RQ 4:** A complex networks approach is taken towards generating SoS design alternatives, because it provides methods that consider the coevolutionary nature of resilience, as well as real world network topologies and evolutionary processes. The methods used to define network initial topologies, threats, and adaptation methods are shown in table 5.

5.4 EXP 2: IE Network Model Validation

Purpose of the experiment: Validate that the IE network model matches results from the complex networks literature for scenarios using generated SoS design alternatives.

Table 5: Selected methods for generating SoS design alternatives

Network design variable	Selected methods for defining the variable
Initial topology	Scale-free networks (BA model) Random networks (ER model)
Threat	Targeted node removals (recalculated node degree) Random node removals
Adaptation method	Recalculated degree adaptation Preferential adaptation Random rewiring

Experimental Setup

Having selected network topologies and threats to focus on for this thesis, experiment two (EXP 2) is performed to validate the IE network model for relevant scenarios. The model is considered validated if it matches results from the literature for changes to the largest connected component (LCC) and inverse average path length of scale-free and random networks, following RD and random node removals. This validation does not use the capability defined for IE networks (i.e., the ability to receive messages) because there are no such results in the literature to validate against. Instead, the size of the LCC and inverse average path length are used because they are commonly used in the literature to measure network resilience. Additionally, these properties are expected to be critical to the ability of an IE network to provide its desired capability.

These validation results also do not consider network adaptation; however, the omission of network adaptation is accepted because the algorithms used to model adaptation are similar to those used to generate network topologies and model node removal threats. Additionally, no results exist in the literature with network adaptation implemented in a manner similar to this thesis.

Validation results shown are averages from 100 simulation replications of each scenario, where a scenario is defined by the network topology and threat. Since the BA and ER models used to generate network topologies are stochastic, each replication for a given scenario is not guaranteed (or even likely) to begin with the same network topology. However, each replication for a given scenario begins with a topology generated by the same algorithm, and uses the same algorithm to perform node removals. Random threats are also stochastic, further introducing variability into results.

Experiment Results

Albert et al. show that scale-free and random networks have similar robustness to random node removals, based on changes to the size of the LCC; however, scale-free networks are more susceptible to RD node removals than random networks [10, 11]. Figure 32 shows results from the IE model for scale-free and random networks subjected to RD and random node removals. The size, S, of the LCC is normalized by the initial network size N, and plotted against the fraction, f, of nodes removed from the network. The simulated scenarios use the same topologies and threats as those from [10, 11], except that the IE model uses networks of size $N = 100$, while Albert et al. use networks of size $N = 10,000$ (all networks have $L = 2N$ links). However, Albert et al. note that their results are independent of network size, enabling the use of smaller networks for the IE model to reduce computation time.

IE model results show that S linearly decreases at similar rates for scale-free and random networks facing random threats. However, both networks show a critical point at which the LCC size non-linearly drops to near-zero values for RD targeted threats. This critical failure point occurs sooner for scale-free networks than it does for random networks. These results match those from [10, 11].

Holme et al. show results for inverse average path length in addition to S/N, in their study of complex network robustness [59]. Figure 33 shows results from the IE model for scale-free and random networks subjected to initial degree (D) and RD node removals. Initial degree threats are considered to further validate the model for other potential threats. Results for network robustness measured by $\langle d \rangle'$ are similar

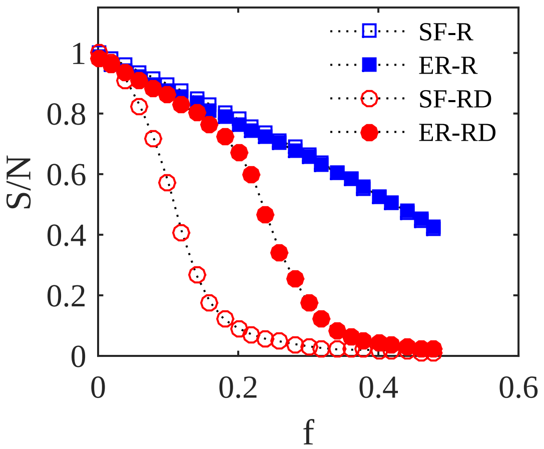


Figure 32: Comparison of the robustness of scale-free (SF) and random (ER) networks, subjected to random (R) and recalculated degree (RD) threats. Scale-free networks are generated using the BA model with $m_0 = 5$ and $m = 2$. Random networks are generated with the ER model with $L = 200$. Networks have $N = 100$ nodes. Threats remove two nodes at each removal event, such that $N_{threat} = 2$.

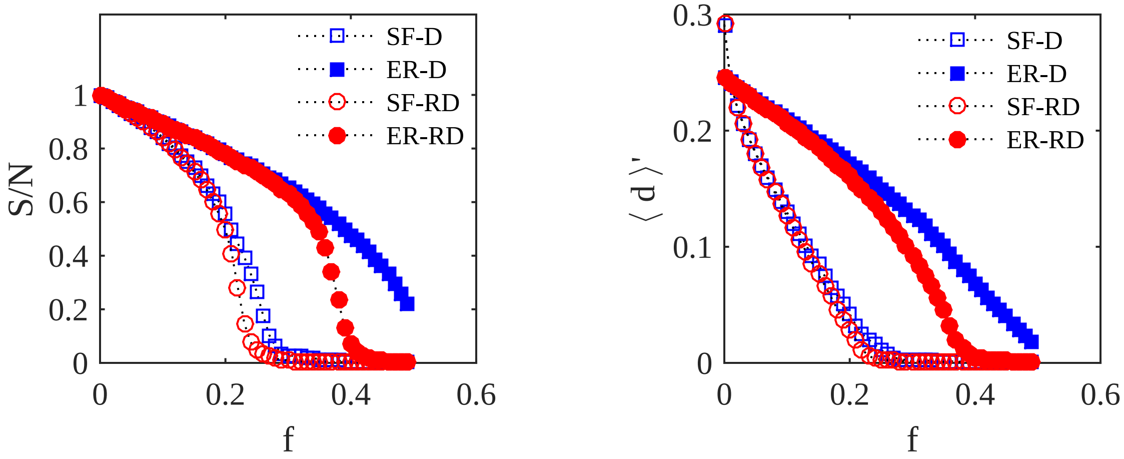
to those measured by S/N. Initial degree threats are less damaging than recalculated degree threats. These results match those from [59].

> **Outcome of EXP 2:** The IE network model is considered validated for this thesis, because it matches trends from the literature for the robustness of scale-free and random networks to RD and random node removals, measured by changes to the size of the LCC and inverse average path length.

5.5 *Summary of Methods for Generating SoS Alternatives and Validation Results*

Complex network methods are used to perform the second step of the ReSSNET methodology (see fig. 34). The IE network model is then validated for relevant scenarios, based on methods selected to generate SoS network designs. The following summarizes research questions, answers, and experiments from this chapter:

- RQ 4: How should we define potential SoS network topologies, threats, and



(a)

(b)

Figure 33: Comparison of the robustness of scale-free (SF) and random (ER) networks, subjected to initial degree (D) and recalculated degree (RD) threats. Scale-free networks are generated using the BA model with $m_0 = 5$ and $m = 3$. Random networks are generated with the ER model with $L = 4500$. Networks have $N = 1500$ nodes. Threats remove one nodes at each removal event, such that $N_{threat} = 1$.

adaptation methods?

- Response to RQ 4: A complex networks approach is taken towards generating SoS design alternatives, because it provides methods that consider the co-evolutionary nature of resilience, as well as real world network topologies and evolutionary processes. The methods used to define network initial topologies, threats, and adaptation methods are shown in table 5.

- Outcome of EXP 2: The IE network model is considered validated for this thesis, because it matches trends from the literature for the robustness of scale-free and random networks to RD and random node removals, measured by changes to the size of the LCC and inverse average path length.

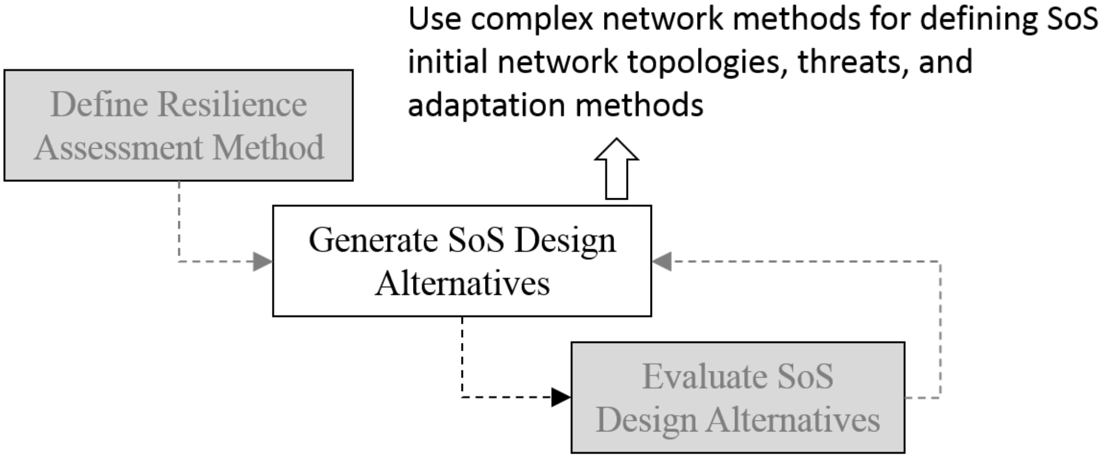

Use complex network methods for defining SoS initial network topologies, threats, and adaptation methods
Define Resilience Assessment Method
Generate SoS Design Alternatives
Evaluate SoS Design Alternatives

Figure 34: Updated overview of the ReSSNET methodology with the selected method for generating SoS design alternatives.

CHAPTER VI

EVALUATING SOS ALTERNATIVES WITH RSM

The third step in the ReSSNET methodology requires methods for evaluating the SoS design alternatives generated in step two. This evaluation should provide an understanding of the advantages and disadvantages of considered alternatives, such that an analyst can make informed decisions in the design of current or future SoS networks. This need is reflected in research question five (repeated here for convenience):

> **Research Question 5:** What methods should be used to compare SoS design alternatives and understand their advantages and disadvantages?

The process of evaluating SoS network designs is separated into two phases: (1) an *exploration* study of the resilience of generated network designs within the network design space and (2) an *optimization* study of the most resilient network designs as the network threat is varied. The purpose of the exploration study is to develop a general understanding of how SoS resilience changes as network design variables (initial topology, adaptation method, and threat) are changed. This phase compares the resilience of potential network designs to potential threats, as specified by table 5. The design space considered in this exploration study is a categorical design space (as opposed to a continuous space), since only two or three settings are considered for each of the design variables, with each setting being defined by a different model. These models, as currently defined, do not enable a continuous representation of the design space. This exploration study directly compares designs of interest, but provides limited insights into the behaviors of potential designs existing in the space between the specified variable settings.

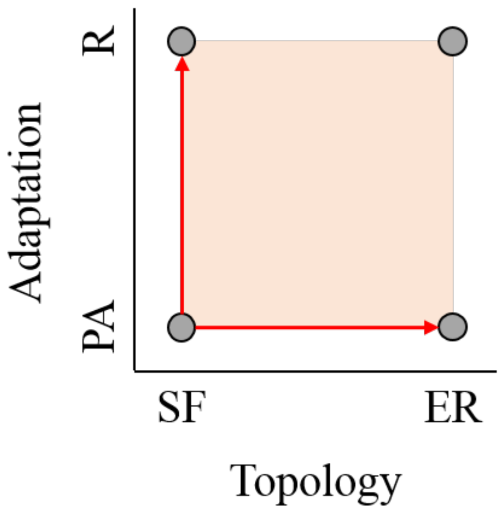


Figure 35: Notional view of the network design space, where categorical network designs from table 5 are displayed as circles. Red arrows and the shaded region represent the consideration of the continuous space between categorically defined network designs. The threat variable is not shown in this 2-dimensional view.

The purpose of the optimization study is to expand on the exploration study and identify how the optimal network design (with respect to resilience) changes as the threat variable changes. This study uses a continuous representation of the design space specified by table 5, to allow the optimizer to consider network designs with topologies between scale-free and random, and adaptation methods between RDA, PA, and random (see fig. 35). A continuous representation of potential threats enables the threat to be transitioned from fully targeted to fully random. This study should provide an understanding of how the optimal design transitions from one optimal to another, as threat randomness is increased. For example, the optimal topology may show a smooth transition, where the randomness in the optimal topology gradually increases or decreases as the threat randomness increases. Alternatively, the optimal topology may show a sharp transition, with a critical point at which the optimal shifts from fully scale-free to fully random, or vice versa. Characterizing the behavior of the optimal design throughout the range of potential threats improves the ability of SoS network designers to design for resilience.

Methods for evaluating design alternatives, specifically those from the field of statistical experimental design, are reviewed for use in this step of the methodology. Following this review, results are discussed from experiment three, which identifies methods that can be used to generate a continuous network design space for the optimization phase of the evaluation process.

6.1 Statistical Experimental Design for Resilient SoS

Method Requirements and Alternatives

The method used to evaluate SoS design alternatives should provide a thorough understanding of advantages and disadvantages of considered alternatives, such that a designer can identify what types of network designs are best suited for certain scenarios. The method should also enable optimization of a continuous design space, to provide an in-depth understanding of SoS network resilience. Given the large scale and complexity of most SoS, the method should be computationally feasible for simulation studies of SoS. Based on these observations, a set of requirements are derived for selecting a method to evaluate SoS alternatives. The selected method should:

- be rigorous, with quantified comparisons of design alternatives,
- enable optimization of a continuous design space,
- and be computationally feasible for SoS simulation studies.

One approach to evaluating SoS network designs is to perform a one-factor-at-a-time (OFAT) study. OFAT studies measure the effects of system factors by performing simulation experiments where one factor is varied and all other factors held constant. This approach requires a large number of simulation runs to understand the effects of all system factors. Additionally, an OFAT approach does not consider potential interactions, since all other factor levels are held constant. Since network topologies, threats, and adaptation methods are expected to show strong interactions, OFAT is not used for this thesis.

Another approach is to simulate and analyze every possible case of interest. Running every case of interest enables direct comparisons of generated design alternatives, including the use of some statistical design methods to analyze simulated scenarios. This approach can also be referred to as using a full-factorial experimental design, since every combination of design settings is considered. However, for large, multi-dimensional design spaces, as are common with SoS designs, this approach can require significant computational time due to the large number of possible combinatorial designs. Given the scale of many SoS networks, SoS simulations may be too complex and have too large of a computational requirement to consider the full design space within a reasonable period of time. Additionally, determining the optimal design from a full-factorial design limits the resolution with which the optimal can be determined, since a discrete number of design settings is used to define the design space. For example, if five settings are considered for the initial network topology and adaptation method, then the resolution of optimal variable settings is limited to those five levels.

A third approach is using statistical experiment design methods. The field of statistical experimental design provides methods that can be used to analyze, design, and optimize new systems or products. This approach attempts to more intelligently explore a design space than through a simple full-factorial experiment, significantly reducing computational expenses. Statistical experimental design also provides methods for modeling a design space with a continuous function, which can be used by a continuous function optimizer to determine the best design alternatives. A review is given of statistical experiment design methods, due to their potential to efficiently analyze and optimize SoS design alternatives.

Background on Statistical Experimental Design

Statistical experimental design, or Design of Experiments (DoE), uses statistical methods to determine experimental settings, or levels, and interpret data collected from those experiments. Sir Ronald Fisher, a pioneering researcher in the field of experiment design, describes an example experiment in which the hypothesis that a lady can determine whether milk or tea was first added to a cup of tea can be tested, and acceptance or rejection of the hypothesis substantiated [47]. Motivation for much of Fisher's work came from the realization that "statistical analysis of data could be informative only if the data themselves were informative, and that informative data could best be assured by applying statistical ideas to the way in which the data were collected in the first place" [102]. The methods developed by Fisher and others in the field are used in many research domains, including biology, sociology, and engineering design. Many books [26, 68, 78, 77] and review papers [118, 102] provide a thorough discussion of this topic, including a discussion of experimental design specifically for simulation studies [69].

Many experimental design studies begin with a screening experiment, which identifies factors that most strongly affect the response. 2-level factorial designs are often used for screening experiments and the identification of main effects of factors and interactions between factors. Following the identification of factor and interaction effects, a response surface (also referred to as a metamodel or surrogate model) can be created from current or additional experimental data to predict the outcome of the response at different factor levels. Response surfaces can be used to efficiently optimize a system design for simulation studies.

This section provides a brief overview of statistical experiment design from the perspective of a simulation design study, focusing on aspects relevant to this thesis. The overview begins with basic terminology, then describes 2-level factorial designs and response surface methodology.

Basic Terminology

Experimental design attempts to develop an understanding of the effects independent and dependent variables have on a specified performance measure for a system or process. The independent and dependent variables are referred to as factors. The performance measure is referred to as a response. A design point is a specific combination of factor levels.

Factors can be qualitative (also referred to as categorical) or quantitative. Qualitative factors have distinctly defined levels that are typically not numerical. Quantitative factors can be represented numerically, often in a continuous manner. The factors considered in this thesis are the network initial topology, threat, and adaptation method. These factors are qualitatively defined for the exploration phase of evaluating SoS alternatives, with each level defined in table 5. These factors are quantitatively defined for the optimization phase of evaluating SoS alternatives. The response considered in this thesis is the resilience of an SoS network, measured by R_{total}.

Factors can also be defined as controllable or uncontrollable. For a system designer, a controllable factor is one that the designer can specify in the design process. An uncontrollable factor is one that is not within the control of the designer, such as noise factors or uncertainty regarding the environment within which the system operates. Most simulation experiments focus on controllable factors, since those are the ones that can be set in the design process. However, uncontrollable factors are often included in a simulation experiment to provide information regarding the behavior of a system as those factors are changed. For this thesis, the initial network topology and adaptation method are controllable factors, while the threat is uncontrollable.

One of the key principles of experimental design is replication. Replication is the process of making multiple runs at each design point within an experiment. For example, if an experimental design for a simulation study has m replications, m

simulation runs will be performed at each combination of factor levels, resulting in m observations of the system response for each design point. The concept of a replication is different from that of a repetition. Repetition is the process of making multiple measurements from a single run of a design point. Consider a manufacturing study that aims to improve the tolerance of a specified dimension in a component. An experimental design with m repetitions would measure the dimension of the same component m times. An experimental design with m replications would measure the dimension of m different components once, where each component is produced under the same conditions (i.e., factor levels). For simulation studies, replication is used to account for randomness in stochastic simulations, since the response, y_i, of a stochastic simulation at design point i is a random variable.

For experiments with replications, the sample mean and variance of observed system responses are often used to characterize system performance. Let $y_{i1}, y_{i2}, \ldots, y_{im}$ be observations for the response of a system at design point i, where m replications are performed. The sample mean, $\bar{y}_i$, is used to estimate the population mean or true mean, μ_i, of the system response at design point i, and calculated as

$$\bar{y}_i = \frac{1}{m} \sum_{j=1}^{m} y_{ij}. \tag{54}$$

Since these observations are independently and identically distributed (IID) random variables, $\bar{y}_i$ is an unbiased estimator of μ_i. The sample variance is similarly an unbiased estimator of the population variance, σ_i^2, and is calculated as

$$S_i^2 = \frac{1}{m-1} \sum_{j=1}^{m} \left(y_{ij} - \bar{y}_i\right)^2. \tag{55}$$

A confidence interval for μ_i can be used to measure the precision with which $\bar{y}_i$ estimates μ_i. A $100(1-\alpha)$ percent confidence interval (specifically a t confidence interval) for μ_i is given by

$$\bar{y}_i \pm t_{m-1,1-\alpha/2}\sqrt{S_i^2/m}, \tag{56}$$

where $t_{m-1,1-\alpha}$ is the standard t-statistic from the t-distribution with $m-1$ degrees of freedom. This confidence interval should be interpreted as an interval that may vary from sample to sample for a given population or experiment; the proportion of times that independently calculated intervals contains μ_i is $1-\alpha$. This proportion is called the coverage of the confidence interval, where a confidence interval is said to cover μ_i if it contains μ_i. This representation of a confidence interval for μ_i assumes that the y_i random variables are normally distributed. Non-normal responses will decrease the correctness of the confidence interval. Law provides a description of the effect the distribution of y_i has on the coverage of the t confidence interval [69].

2-Level Factorial Designs

2-level factorial designs, or 2^k factorial designs, consider k factors with two levels for each factor. Design points can be displayed as a design matrix, where levels are represented by a "+" or "−" in a table. Table 6 shows an example design matrix for a 2^3 factorial design. For experiments where m replications are performed at each design point, the response, y_i, for design point i is defined as the sample mean of responses from all replications of that design point [calculated with eq. (54)]. 2-level factorial designs are often used to determine factor main effects and interactions.

The main effect, e, of a factor is defined as the change in the response as that factor is changed from its "−" level to its "+" level, averaged over all levels of other factors. A positive main effect (i.e., $e > 0$) means that on average, changing a factor from its "−" level to its "+" level will increase the response. A negative main effect means that on average, changing a factor from its "−" level to its "+" level will decrease the response. The magnitude of e determines how much of an effect, on average, a change in the factor level has on the response value. However, if interactions exist

Table 6: Example design matrix for a 2^3 factorial design

Design point	Factor A	Factor B	Factor C	Response
1	$-$	$-$	$-$	y_1
2	$+$	$-$	$-$	y_2
3	$-$	$+$	$-$	y_3
4	$+$	$+$	$-$	y_4
5	$-$	$-$	$+$	y_5
6	$+$	$-$	$+$	y_6
7	$-$	$+$	$+$	y_7
8	$+$	$+$	$+$	y_8

between factors, interpretations of main effects must consider other factor levels.

Interaction effects account for the presence of interactions between two factors, where the effect of one factor may depend on the level of another factor. The interaction effect, e_{AB}, between two factors A and B, is defined as the difference between the average effect of factor A with factor B at its "+" level, and the average effect of factor A with factor B at its "−" level. By convention, the interaction effect is calculated as half of this difference. The interaction, e_{ABC}, between factors A, B, and C is half the difference between the average AB interaction with factor C at its "+" level, and the average AB interaction with factor C at its "−" level. See appendix A for calculations of main effects and interactions.

For experiments using stochastic simulations, it may be necessary to determine if the observed effects are statistically significant, rather than effects due to randomness in the sampled data. Analysis of Variance (ANOVA) can be used to determine statistical significant of effects [78]. ANOVA calculates a sum of squares for each effect, a total sum of squares, and an error sum of squares. These sum of squares are used to calculate mean squares and corresponding F-test statistics, which are then compared to the F-distribution to determine p-values. A p-value is the probability that a value drawn from the F-distribution is greater than or equal to the F-test statistic. If the p-value for an effect is less than or equal to a specified significance level α (e.g., $p \leq 0.05$ for a 5 % significance level), the null hypothesis that the effect is not significant is rejected.

Response Surface Methodology

Response surface methodology (RSM) provides techniques for constructing a model of the response of a system based on observed data. A response surface, or regression model, attempts to explain the relationship between the response of a system, y, and a vector of explanatory input variables (or factors), $\boldsymbol{x}$, such that

$$y = f\left(\boldsymbol{x}\right) + \varepsilon, \tag{57}$$

where ε is an error term that represents variability not described by f. Multiple linear regression represents the response as a polynomial model in the following form,

$$y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \varepsilon = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon, \tag{58}$$

where $x_1, x_2, \ldots, x_k$ are regressor variables and $\beta_0, \beta_1, \ldots, \beta_k$ are regression coefficients. Equation (58) is a first-order model. Interaction and higher order terms can be added to the model to handle more complex systems that are not well-represented by a first-order model. For example, a second-order model with interaction terms and two input variables, x_1 and x_2, could take the following form,

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon. \tag{59}$$

The method of least squares is usually used to estimate regression coefficients, based on a set of data containing $n > k$ observations of the response at n design points or combinations of input variables [78]. The data used to fit the model is referred to as fit or training data. Validation data refers to data used to validate the model, or check its adequacy. Validation data can be separate from or part of the fit data. Once regression coefficients are estimated, a regression model is obtained, which can be used to predict a response given a set of input variable settings or values. Since this regression model is a simple polynomial model of the data, potential system designs can be quickly evaluated within an optimization process.

Model fit checks must be performed to determine if a regression model is accurate enough for use in a simulation study. Hypothesis testing with ANOVA can be used to test for the significance of the regression, with regards to explaining the variability in the observed data [78]. The hypotheses used in this test are

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0 \tag{60}$$

$$H_1 : \beta_j \neq 0 \text{ for at least one } j. \tag{61}$$

If the p-value calculated from ANOVA is less than the specified significance level (i.e., $p < \alpha$), then we can reject the null hypothesis that none of the regressor variables contributes significantly to the model. Rejection of the null hypothesis provides statistical support for using the regression model.

The coefficient of multiple determination, R^2, provides another goodness of fit measure for the model. R^2 measures how much of the variability in the data is explained by the model, as opposed to being explained by residuals. Low R^2 values indicate a poor model fit, where $0 \leq R \leq 1$. One issue with using R^2 to check model adequacy is that R^2 always increases as the number of regressors increases, regardless of whether or not additional regressors are statistically significant. The adjusted R^2 statistic, R^2_{adj}, can be used to account for the inclusion of unnecessary regressors, since R^2_{adj} will not always increase as regressors are added; R^2_{adj} can actually decrease as unnecessary regressors are added.

A plot of predicted versus actual responses provides a visual test for model adequacy. A good model fit should show minimal separation between plotted data points and a 1:1 reference line representing a perfect fit to the training data.

Residual analysis can also be used for goodness of fit checks. A residual, e_i, for the ith observation is calculated as

$$e_i = y_i - \hat{y}_i, \tag{62}$$

where y_i is the observed response for the ith observation and $\hat{y}_i$ is the predicted response for that observation. A residual versus predicted plot shows the magnitude of residuals from the model. A random scattering of points in this plot is desired, since that suggests normality in the residuals. An underlying assumption in linear regression is that model errors, or residuals, are normally distributed. Patterns in the residual plot often suggest the need for a transformation in the fitted data. A normal probability plot of the residuals provides another visual representation of how close residuals are to being normally distributed. Histogram plots of model percent error from fit data (model fit error) and validation data (model representation error) give further insight into the normality and magnitude of residuals. Model percent error for the ith observation is calculated as

$$\%\ \text{error}_i = \frac{y_i - \hat{y}_i}{y_i}. \tag{63}$$

A lack of fit test can be used for regression models fit to experimental designs with replications [78]. This test determines if there is strong evidence that the underlying function, f, for the data is in fact linear, using a hypothesis test with the F-test statistic. If there is no statistical evidence that underlying function is linear, a linear regression model is deemed unfit for the data.

RSM for Designing Resilient SoS Networks

Statistical experimental design, in particular RSM, provides methods for comparing SoS design alternatives, identifying factor main effects and interactions, and optimizing network design settings with linear regression models. A disadvantage of using linear regression is that there is some level of error in the model. However, the efficient use of computational time offered by RSM experimental designs outweighs the potential for model inaccuracies expected from using linear regression. Therefore, an RSM approach is used to evaluate SoS design alternatives, answering research question five as follows:

> **Response to RQ 5:** An RSM approach with multiple linear regression is used to explore and optimize SoS design alternatives, providing insight into the advantages and disadvantages of potential SoS networks.

6.2 EXP 3: Interpolating the Network Design Space

Purpose of the experiment: Answer research question 5.1 by defining models that can be used to interpolate between scale-free and random network topologies, as well as RD and random network threats.

The optimization phase of evaluating network designs represents the network design space as a continuous design space. In comparison, the design space generated in chapter 5 (which is used for the exploration phase) only considers categorically defined network designs. A categorical design space limits the ability to gain insights into the performance of intermediately defined networks. A continuous representation of the network design space enables a more thorough investigation of how network adaptation can be used to provide resilience to network threats. From an optimization perspective, a continuous design space provides an optimizer with an exhaustive set of potential designs to consider, giving a designer confidence that they are sufficiently exploring the best set of potential alternatives.

An approach to generating a continuous network design space is to define a model for each design variable that can be used to interpolate from one setting of the variable to another. If this model is parameterized by a single model parameter, then design variable settings can be continuously changed by altering the value of that parameter. An example of this approach is the small-world network model from Watts and Strogatz [113]. Their simple model interpolates between regular and random networks by varying a single model parameter, p. Their model begin with a regular network, defined as a ring lattice with N nodes, each with a degree of k. Each link is then randomly rewired with probability p. Therefore, if $p = 0$, a regular network is generated. If $p = 1$, a random network is generated. Values of $0 < p < 1$ generate intermediate network topologies in the space between regular and random.

Defining a model similar to the small-world network model for the network topologies, adaptation methods, and threats defined in chapter 5 would enable an exploration of a continuous network design space. Based on generated network designs, the network topology interpolation model should be able to continuously transition from scale-free to random networks, and the network threat model from RD to random node removals. Since the adaptation methods defined for this thesis are either degree-based or random, the model used to interpolate between RD and random threats can also be used for interpolating between adaptation methods. The need for topology and threat interpolation models is reflected in the following research question:

> **Research Question 5.1:** What models should be used to interpolate between scale-free and random networks, and RD and random threats?

This research question is defined as a sub-question of research question five because it further answers the question of what methods should be used to compare and evaluate SoS design alternatives.

Model Requirements and Alternatives

The interpolation models selected should be able to generate topologies that structurally match scale-free and random networks, and generate threats able to replicate the effects of RD and random threats on the structure of scale-free and random networks. This analysis focuses on structural network properties because at their most fundamental level, networks are defined by their structure. The importance of network structural properties is reflected in their widespread use in the complex networks literature.

Complex network topologies are often represented by their degree distribution, because these distributions provide a quantitative way to capture and visualize the overall connectivity of network nodes. Considering the topologies used in this thesis, scale-free and random networks have significantly different degree distributions (see section 2.3.2). Therefore, one requirement for a topology interpolation model is the ability to generate networks with degree distributions matching those of scale-free and random networks. The second requirement is the ability to generate intermediate network topologies that smoothly transition from scale-free to random, again measured by changes to network degree distributions.

The effect of RD and random threats on scale-free and random networks is often characterized by changes to the size of the LCC and inverse average path length [10, 59]. Therefore, the primary requirement for the threat interpolation model is the ability to generate threats that replicate the structural effects of RD and random threats on scale-free and random networks. Structural effects are measured by changes to the LCC and inverse average path length. A second requirement is the ability to generate intermediate threats that smoothly transition between RD and random. The following requirements are set for selecting network topology and threat interpolation models:

- The topology model should be able to generate networks with degree distributions matching scale-free and random networks.

- The threat model should be able to generate threats that affect the structure (measured by size of the LCC and inverse average path length) of scale-free and random networks similar to the way RD and random threats affect the structure of those networks.

- The topology and threat models should be able to generate intermediate topologies that smoothly transition between scale-free and random topologies, and intermediate threats that smoothly transition between RD and random threats.

A potential model for interpolating between scale-free and random networks is the Degree and Locality-based Attachment (DLA) model [119]. The DLA model extends the BA model for scale-free networks by incorporating a degree preference parameter, u, and considering node distances, in addition to node degrees, when forming new links. The model begins with a small number of unlinked nodes, N_0, and grows the network by adding a new node to the network at each time step. Each new node is added with m links. Two attachment rules are used for determining which existing nodes are connected to by those m links.

The first link of a new node connects with an existing node j with degree k_j, with probability

$$\prod(k_j) = \frac{k_j^u}{\sum_{i=1}^{N} k_i^u}, \tag{64}$$

where u is the degree preference parameter and N is the current size of the network. The degree preference parameter determines how much preference to give to highly connected nodes, with $0 \leq u \leq 1$. A preference of $u = 0$ does not consider the degree of a node when choosing neighbors, resulting in links being randomly connected among existing nodes. A preference of $u = 1$ is equivalent to preferential attachment used in the BA model, and therefore results in scale-free networks. For $m > 1$, the remaining link(s) are added such that the probability an existing node j with geodesic distance from the new node, d_j, is linked with is

$$\prod(d_j) = \frac{d_j^\gamma}{\sum_{i=1}^N d_i^\gamma}, \tag{65}$$

where γ is the locality preference parameter, $0 \leq \gamma \leq 1$. High values of γ give preference to nodes in the neighborhood of the newly selected neighbor.

The DLA model can be adapted to interpolate between scale-free and random topologies by using eq. (64) to define preferential attachment for all links. Therefore, by changing the degree preference parameter from zero to one, network topologies ranging from random ($u = 0$) to scale-free ($u = 1$) can be generated.

Another potential model for interpolating between scale-free and random networks is a model proposed by Gómez-Gardeñes and Moreno [49] (referred to as the GM model within this thesis). The GM model is defined by a single parameter, α, which determines the node heterogeneity of generated networks, where $0 \leq \alpha \leq 1$. The model begins with a fully connected network of m_0 nodes, similar to the BA model. However, unlike the BA model, the network also begins with a set $\mathcal{U}$ of unconnected nodes, where $|\mathcal{U}| = N - m_0$ and N is the total network size. The model then steps through $N - m_0$ steps, where the ith step adds m links for the ith node in the set $\mathcal{U}$ of initially unconnected nodes. Each of the m links added at a time step are added using one of two attachment models. With probability α, a node links to any other node in the total set of N nodes with uniform probability, such that the probability a node with degree k_j is linked with is

$$\prod(k_j) = \frac{1}{N}. \tag{66}$$

With probability $1-\alpha$, a node links to another node using some form of preferential attachment based on node degrees. The authors define two potential functions to use for modeling preferential attachment. However, since this thesis seeks a model that can replicate scale-free networks created by the BA model, preferential attachment is defined using the BA model. Therefore, the probability, $\prod(k_j)$, that an existing node with degree k_j is linked with is

$$\prod(k_j) = \frac{k_j}{\sum_{i=1}^{N} k_i}. \tag{67}$$

Using the GM model, scale-free networks can be generated with $\alpha = 0$ and random networks with $\alpha = 1$.

The DLA and GM models are defined as models for interpolating between scale-free and random network topologies. These models perform this interpolation by adjusting the level of randomness allowed in the model, relative to degree-based preferential attachment. Since the network threats considered in this thesis also range from being degree-based to random, these models can be modified to interpolate between RD and random threats.

For the DLA model, eq. (64) is used to define the probability that a node is removed, rather than defining how new nodes add links. Therefore, $u = 1$ defines preferential targeted threats and $u = 0$ defines random threats. Similarly, the GM model can be used to define a network threat by having each node removed randomly using eq. (66) with probability α, or in a targeted manner with probability $1-\alpha$. Two types of targeting can be used with the GM model: one that preferentially targets by node degree using eq. (67) (referred to as GM1) and another that always uses recalculated node degree as described in section 5.2 (referred to as GM2).

Experimental Setup

Experiment three is performed to compare the ability of the DLA model and GM model to interpolate the network design space generated in chapter 5, with respect to the defined requirements for interpolating the space. The models are first tested for use in interpolating between scale-free and random network topologies by comparing degree distributions generated by each model for scale-free and random networks, to those generated by the BA model (for scale-free topologies) and the ER model (for

Table 7: IE network model parameters used for EXP 3

Parameter	Settings used	Description
N	100	Initial network size
L	200	Initial number of links in a network
$[t_0\ t_{final}]$	N/A	Simulated scenario time interval (in seconds)
S_{threat}	2	Number of nodes removed per threat event
$N_{threats}$	24	Number of threat events within a scenario
t_{adapt}	N/A	Adaptation delay time following a threat event (in seconds)
μ	0	Message generation rate
Δ	N/A	Time sensitivity

random topologies). The models are then tested for use in interpolating between RD and random threats by simulating changes to the LCC and inverse average path of scale-free and random networks subjected to targeted and random threats generated by the DLA and GM model, and comparing those to trends from networks subjected to RD and random threats as defined in chapter 5.

Since topology comparisons only consider network degree distributions, no simulation is needed. Instead, scale-free networks are generated using the BA, DLA, and GM models, and random networks generating using the ER, DLA, and GM models. 1000 network replications are generated using each model, since they are stochastic models. Networks are generated with $N = 1000$ nodes and $L = 2000$ links. Larger networks are used for this comparison than others in this thesis to more clearly show trends with respect to degree distributions.

Network simulations are required for the comparison of threat interpolation models. The IE network model is used for these comparisons, with simulation parameters shown in table 7. Simulations are run with no message generation, since this experiment only considers changes to the structure of networks as threats remove nodes. Since no resilience calculations are performed, the simulated time scenario is also not applicable; instead, enough time steps are used to simulate every threat event desired.

Table 8: Experimental design matrix used for EXP 3 (50 replications at each point)

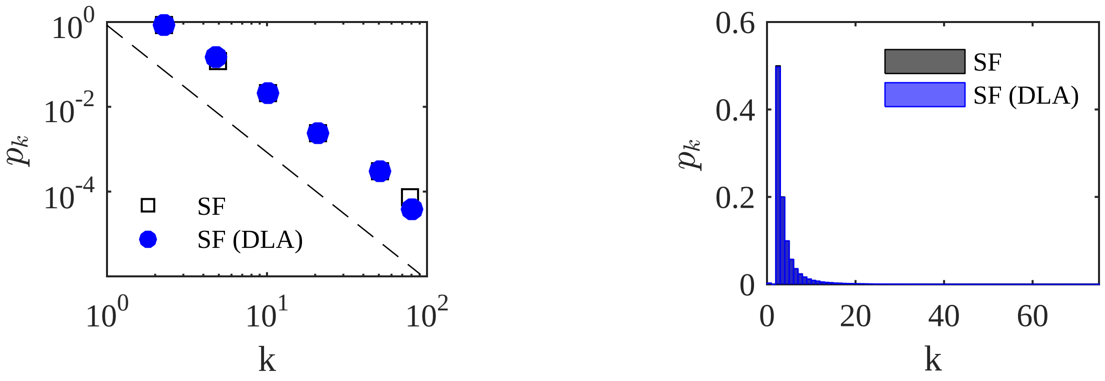
Design point	Topology	Threat
1	Scale-free (interpolated)	RD
2	Scale-free (interpolated)	RD (DLA)
3	Scale-free (interpolated)	RD (GM)
4	Random (interpolated)	RD
5	Random (interpolated)	RD (DLA)
6	Random (interpolated)	RD (GM)
7	Scale-free (interpolated)	Random
8	Scale-free (interpolated)	Random (DLA)
9	Scale-free (interpolated)	Random (GM)
10	Random (interpolated)	Random
11	Random (interpolated)	Random (DLA)
12	Random (interpolated)	Random (GM)

Simulation design points for threat interpolation comparisons are shown in table 8. Topologies for these simulations are defined using the model selected from the topology interpolation comparison. 50 replications are run at each design point. Threats without a specified model (i.e., those specified as “RD” or “Random”) are modeled as described in chapter 5.

Experiment Results (Topology Interpolation)

Figure 36 shows that scale-free networks generated using the BA and DLA models have similar degree distributions. This result is expected, since for $u = 1$, the only difference between the DLA model and the BA model is that the DLA model begins with m_0 disconnected nodes, while the BA model begins with m_0 fully connected nodes. Since the initial m_0 nodes are disconnected in the DLA model, this model also shows a non-zero probability of nodes with $k < 2$ links. In comparison, the BA model ensures that all nodes have $k \geq 2$ links. Despite this difference, the overall structure of these networks is similar. Therefore, scale-free networks generated by the DLA model are determined to sufficiently match those generated by the BA model.

Figure 37 shows that random networks generated using the ER and DLA models



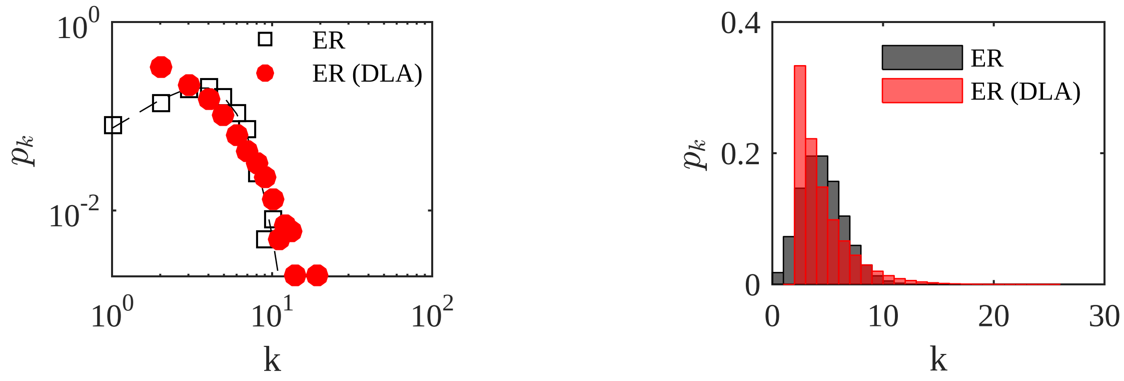
(a)

(b)

Figure 36: Degree distributions (plotted with exponentially increasing degree bin sizes) for scale-free networks generated using BA and DLA models with $N = 1000$ nodes, $L = 2000$ links, $m_0 = 5$, $m = 2$. The DLA model uses $u = 1$. Degree distributions from two network instances are plotted on a log-log scale in (a). Degree distributions from all 1000 replications are shown as a histogram in (b). p_k is the fraction of nodes in a network with degree k [i.e., $p_k = \Pr(K = k)$]. The dashed line in (a) shows a linear power law, calculated using eq. (24) with $\gamma = 3$.

do not have similar degree distributions. Random networks generated with the DLA model show less of an exponentially decaying tail at the high end of their degree distributions than those from the ER model. Additionally, there are more low degree nodes ($k < 4$) for random networks from the DLA model than the ER model.

These differences are due to the inclusion of network growth in the DLA model, which is not included in the ER model. Including network growth exposes early network nodes (i.e., those added to the network early in the growth process) to more linking opportunities with new nodes than late network nodes (i.e., those added to the network late in the growth process). Therefore, despite links being randomly added, early network nodes have a higher average node degree than late network nodes, creating the presence of more high degree nodes in the DLA model. In comparison, the ER model exposes all nodes to the same number of linking opportunities, limiting the presence of high degree nodes. Network growth also creates fewer low degree nodes, because every new node added to the network is added with $m = 2$ links. Therefore, few nodes have $k < 2$ links for the DLA model. These differences result in



(a)

(b)

Figure 37: Degree distributions for random networks generated using ER and DLA models with $N = 1000$ nodes and $L = 2000$ links. The DLA model uses $m_0 = 5$, $m = 2$, and $u = 0$. Degree distributions from two network instances are plotted on a log-log scale in (a). Degree distributions from all 1000 replications are shown as a histogram in (b). The dashed line in (a) shows the expected Poisson distribution of ER random networks, calculated using eq. (26) with $\langle k \rangle = 4$.

a shift in the distribution peak for random networks generated with the DLA model (near $k = 2$) compared to those from the ER model (near $k = 4$). Random networks from the DLA model are therefore pseudo-random, since they do not fully match ER random networks.

Figure 38 shows that the DLA model smoothly interpolates between DLA scale-free networks ($u = 1$) and DLA random networks ($u = 0$), since degree distributions gradually change from being linear to showing the curvature of ER networks. However, since DLA random networks are pseudo-random, the DLA model does not meet the requirements established for a topology interpolation model.

The BA model and GM model (with $\alpha = 0$) are identical algorithms for creating scale-free networks, since the GM model always uses preferential attachment to add the m links of each new node when $\alpha = 0$. Therefore, only random networks are tested for the GM interpolation model. Figure 39 shows that random networks generated by the ER and GM model have similar degree distributions. Unlike those from the DLA model, GM random networks show a drop in p_k below the peak degree in the

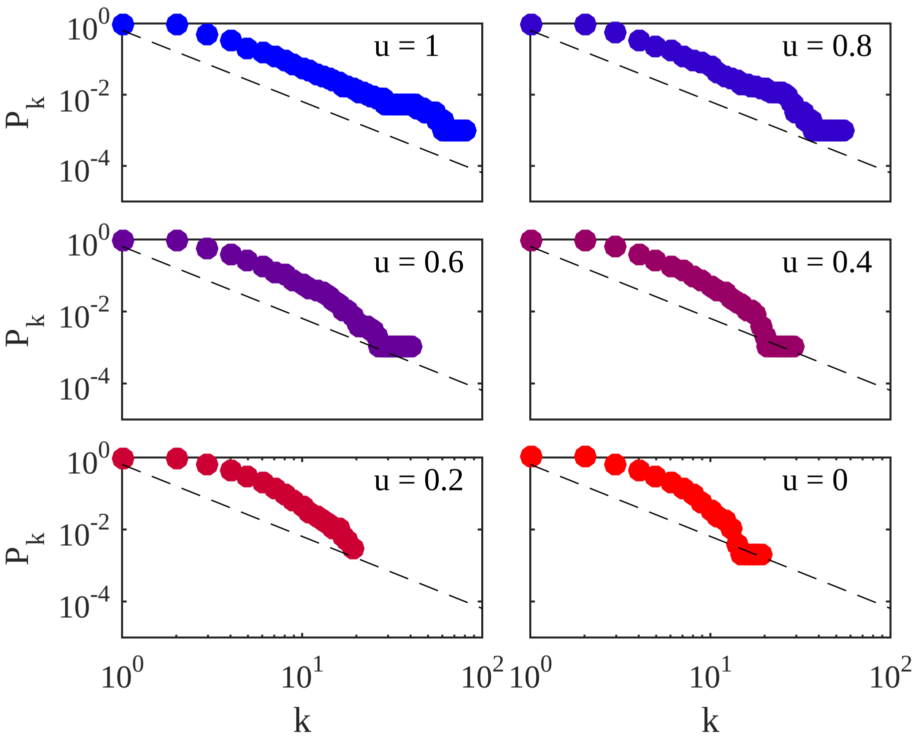
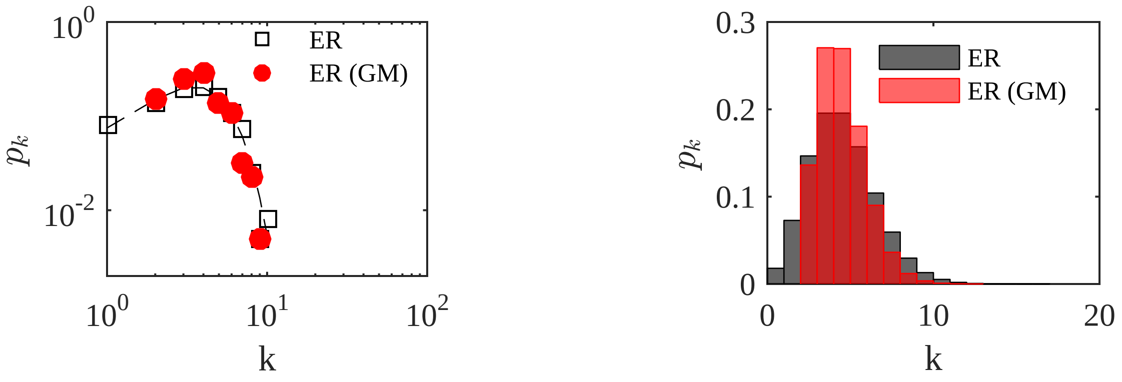


Figure 38: Degree distributions for single instances of networks generated using the DLA model as the interpolation parameter u is varied ($N = 1000$ nodes, $L = 2000$ links, $m_0 = 5$, and $m = 2$). These plots show cumulative degree distributions, where $P_k = \Pr(K \geq k)$, to enable better comparisons between distributions.

distribution. This peak is also at a similar location to ER random networks. GM random networks differ from DLA random networks because the GM model does not include network growth. For the DLA model, all added nodes are required to connect to the currently existing connected component (i.e., the ith new node can only link with $i - 1$ nodes). For the GM model, nodes in the set $\mathcal{U}$ of initially unconnected nodes can link with any other node in the network, whether they are in the current connected component or not. The ability to link with any node better replicates the randomness of ER networks.

One difference between GM random networks and ER networks is that all nodes have $k \geq 2$ links in the GM model, which slightly reduces the number of low degree nodes and increases the magnitude of the degree distribution peak. However, this



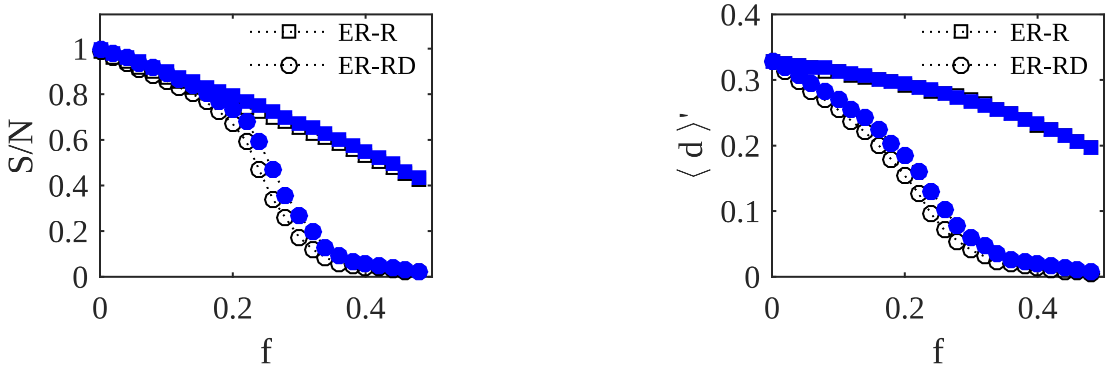
(a)

(b)

Figure 39: Degree distributions for random networks generated using ER and GM models with $N = 1000$ nodes, $L = 2000$ links. The GM model uses $m_0 = 5$, $m = 2$, and $\alpha = 1$. Degree distributions from two network instances are plotted on a log-log scale in (a). Degree distributions from all 1000 replications are shown as a histogram in (b).

difference has a small effect on the general shape of GM random network degree distributions. To further verify that this difference minimally affects the behavior of random networks, changes to the LCC and inverse average path length are compared for GM and ER random networks facing RD and random threats. Results for the GM random networks should match those from experiment two, since experiment two used ER networks and non-interpolated RD and random threats. Figure 40 shows that changing from ER to GM random networks has little effect on the general robustness of random networks to RD and random threats.

Figure 41 shows that the GM model smoothly interpolates between GM scale-free networks ($\alpha = 0$) and GM random networks ($\alpha = 1$). Therefore, the GM model meets established requirements for a network topology interpolation model and is selected for use in this thesis.



(a)

(b)

Figure 40: Changes to the size of the LCC normalized by N (a) and inverse average path length (b) with a fraction f of removed nodes for random networks facing targeted (RD) and random (R) node removals. Networks have $N = 100$ nodes and $L = 200$ links and are generated using the ER model (unfilled black symbols) and GM model (filled blue symbols). The GM model uses $m_0 = 5$, $m = 2$, and $\alpha = 1$.

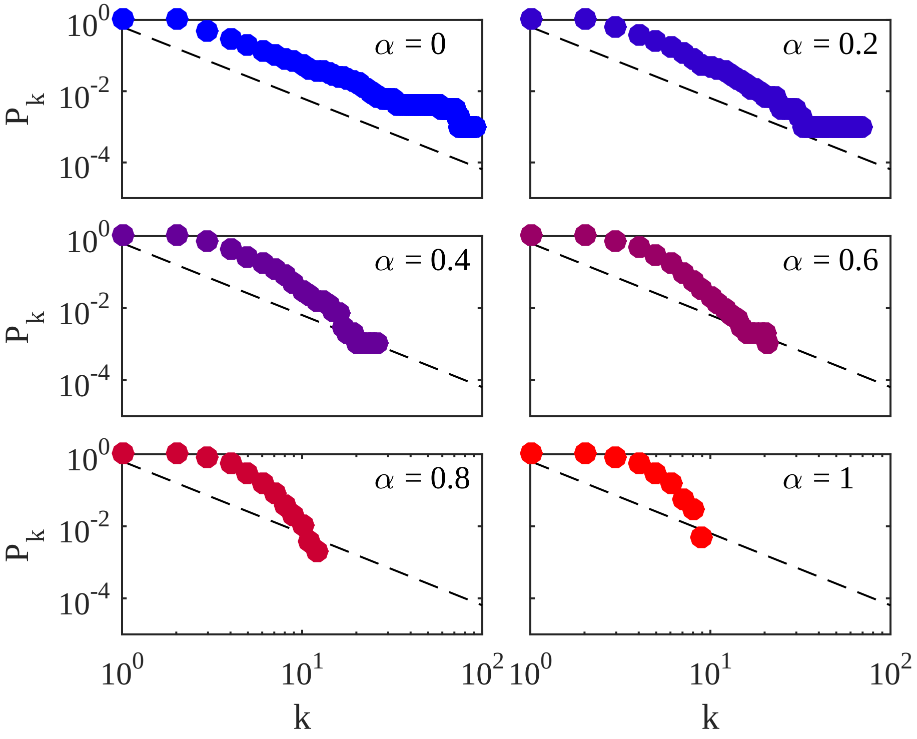


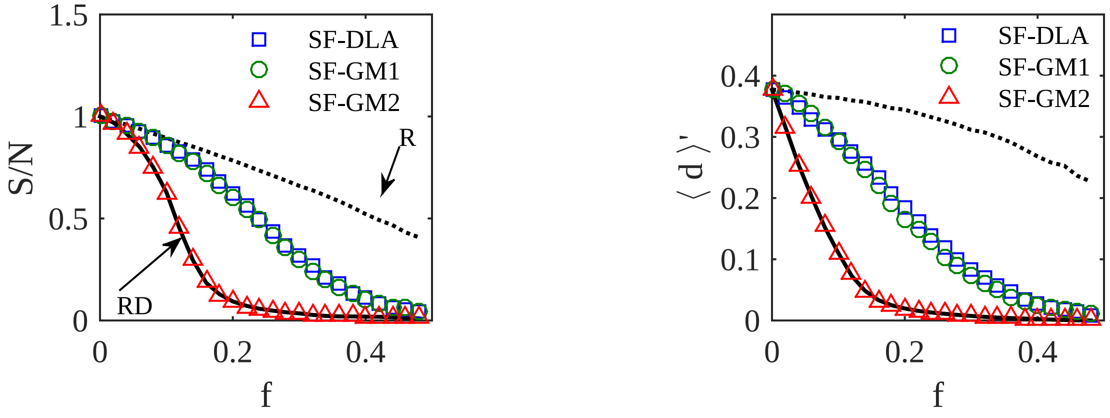
Figure 41: Degree distributions for single instances of networks generated using the GM model as the interpolation parameter α is varied ($N = 100$ nodes, $L = 200$ links, $m_0 = 5$, and $m = 2$). These plots show cumulative degree distributions, where $P_k = \Pr(K \geq k)$.

Experiment Results (Threat Interpolation)

Algorithmically, random threats defined using the DLA model (with $u = 0$) or the GM model (with $\alpha = 1$) are identical to those described in section 5.2. Therefore, either the DLA model or GM model can be used to represent random threats. However, targeted threats are defined differently between the DLA model (with $u = 1$) and the GM model (with $\alpha = 0$).

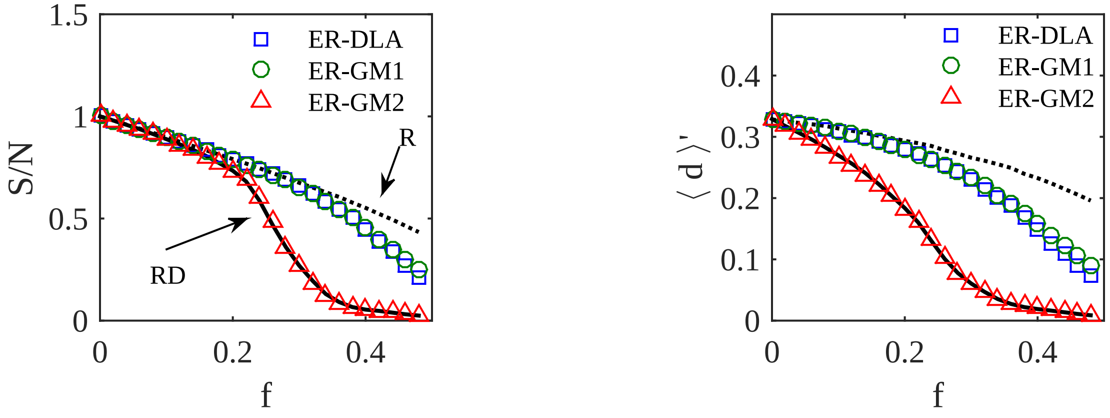
Figures 42 and 43 show the effects of DLA and GM defined targeted threats on scale-free and random networks. Targeting by the DLA model and GM1 model show reduced impacts on the normalized size of the LCC, S/N, and the inverse average path length, $\langle d \rangle'$, compared to RD threats. For random networks, DLA and GM1 targeted threats actually more closely match a random threat than RD threat. However, GM2 targeting is able to match the effects of RD threats on scale-free and random networks. DLA and GM1 differ from GM2 and RD threats because DLA and GM1 use preferential targeting, which defines a non-zero probability of nodes without the highest degrees being removed. Allowing low degree nodes to be removed results in more threat randomness than GM2 and RD threats.

Figures 44 and 45 shows that the GM2 model smoothly interpolates between RD threats ($\alpha = 0$) and random threats ($\alpha = 1$). Therefore, the GM2 model meets established requirements for a network threat interpolation model and is selected for use in this thesis.



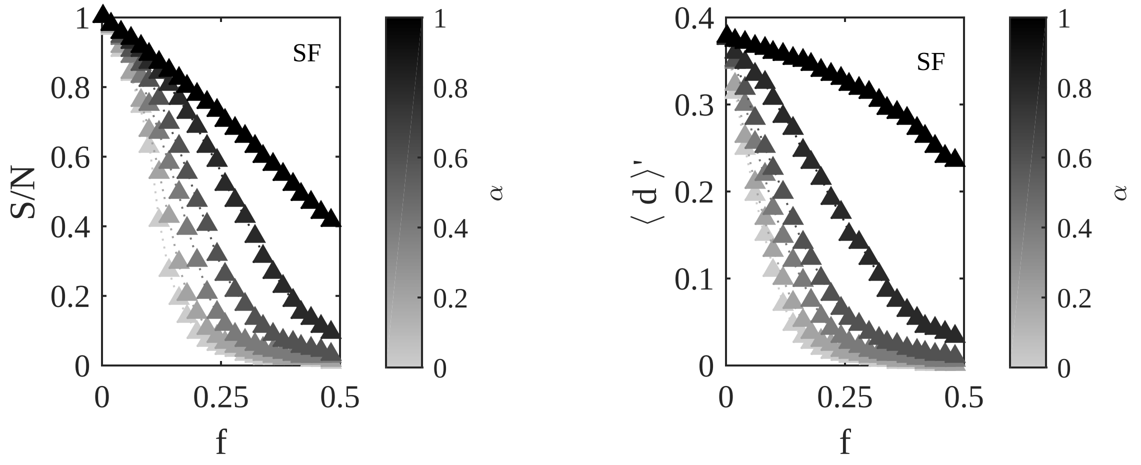
(a) (b)

Figure 42: Changes to the size of the LCC (a) and inverse average path length (b) with a fraction f of removed nodes for scale-free networks facing targeted threats generated using the DLA model and two GM model variants. Scale-free topologies are generated using the GM model with $N = 100$ nodes, $L = 200$ links, $m_0 = 5$, $m = 2$, and $\alpha = 0$. Solid lines show data using RD threats as defined in section 5.2. Dashed lines show data from random threats as defined in section 5.2.



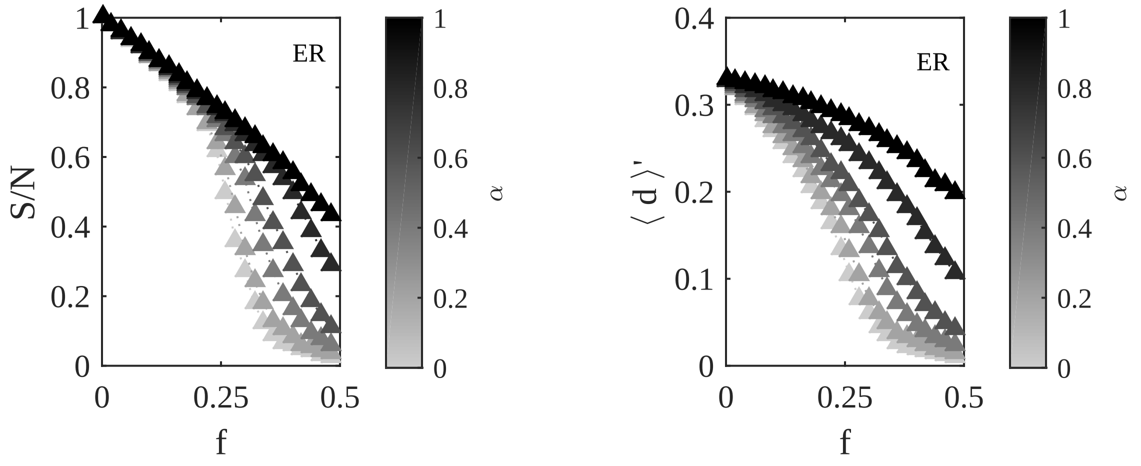
(a) (b)

Figure 43: Changes to the size of the LCC (a) and inverse average path length (b) with a fraction f of removed nodes for random networks facing targeted threats generated using the DLA model and two GM model variants. Random topologies are generated using the GM model with $N = 100$ nodes, $L = 200$ links, $m_0 = 5$, $m = 2$, and $\alpha = 1$. Solid lines show data using RD threats as defined in section 5.2. Dashed lines show data from random threats as defined in section 5.2.



(a) (b)

Figure 44: Changes to the size of the LCC (a) and inverse average path length (b) for scale-free networks facing GM2 threats with α varied. Scale-free topologies are generated using the GM model with $N = 100$ nodes, $L = 200$ links, $m_0 = 5$, $m = 2$, and $\alpha = 0$.



(a) (b)

Figure 45: Changes to the size of the LCC (a) and inverse average path length (b) for random networks facing GM2 threats with α varied. Scale-free topologies are generated using the GM model with $N = 100$ nodes, $L = 200$ links, $m_0 = 5$, $m = 2$, and $\alpha = 1$.

Discussion of Results

Scale-free networks generated by the DLA and GM models closely match degree distributions of those generated by the BA model commonly used in the literature. However, only the GM model is able to sufficiently match degree distributions of random networks generated by the ER model. The DLA model does not match ER random networks well because it includes network growth, which indirectly includes preferential attachment for early network nodes. Network growth in the DLA model results in the creation of pseudo-random networks, which approach a scale-free topology rather than purely random one. These results support the claim by Barabási and Albert that network growth is an important mechanism in the formation of scale-free networks [22]. The GM model is also shown to smoothly interpolate between scale-free and random networks, and is therefore selected as the topology interpolation model.

The GM model (specifically GM2) is used to interpolate between network threats, because it better captures the effects of RD threats on network robustness than GM1 and DLA threats. These results show that threats defined by preferential node removal include too much randomness to replicate the damage inflicted by RD threats on scale-free and random networks, despite their preference for removing high degree nodes. Therefore, preferential threats can be viewed as an intermediate threat between RD and random ones. GM2 threats are also shown to smoothly interpolate between RD and random threats. These results substantiate the following response to research question 5.1:

> **Response to RQ 5.1:** The GM model is used to interpolate between scale-free and random networks, as well as RD and random threats using GM2 targeting.

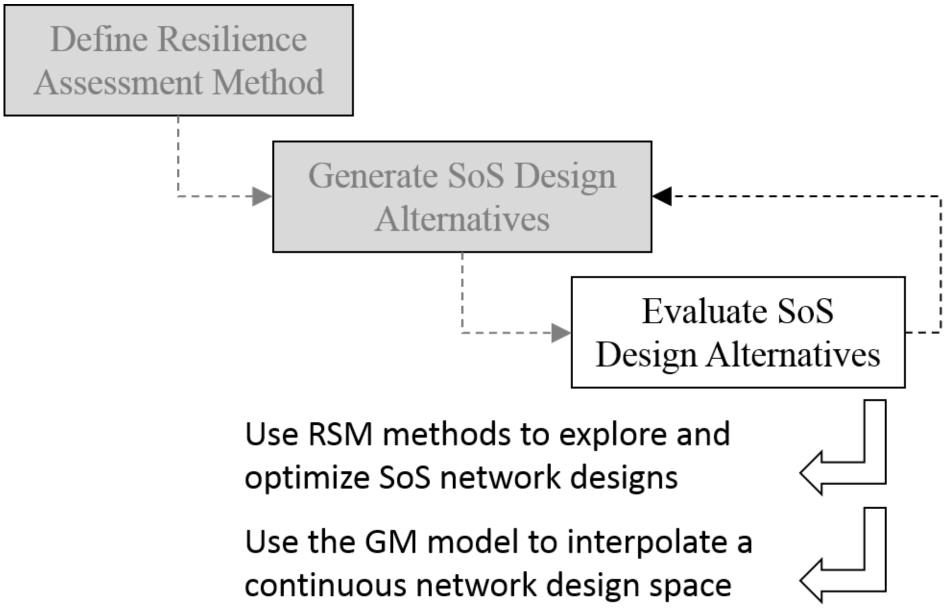

Define Resilience Assessment Method
Generate SoS Design Alternatives
Evaluate SoS Design Alternatives
Use RSM methods to explore and optimize SoS network designs
Use the GM model to interpolate a continuous network design space

Figure 46: Updated overview of the ReSSNET methodology with the selected SoS network evaluation methods.

6.3 Summary of Methods for Evaluating SoS Alternatives and Results

RSM provides methods for performing the third step of the ReSSNET methodology (see fig. 46). This evaluation step is split into two phases, an exploration phase and an optimization phase. A continuous network design space for the optimization phase of the study is defined using the GM model to interpolate between network topologies, threats, and adaptation methods. The following summarizes research questions, responses, and experiments from this chapter:

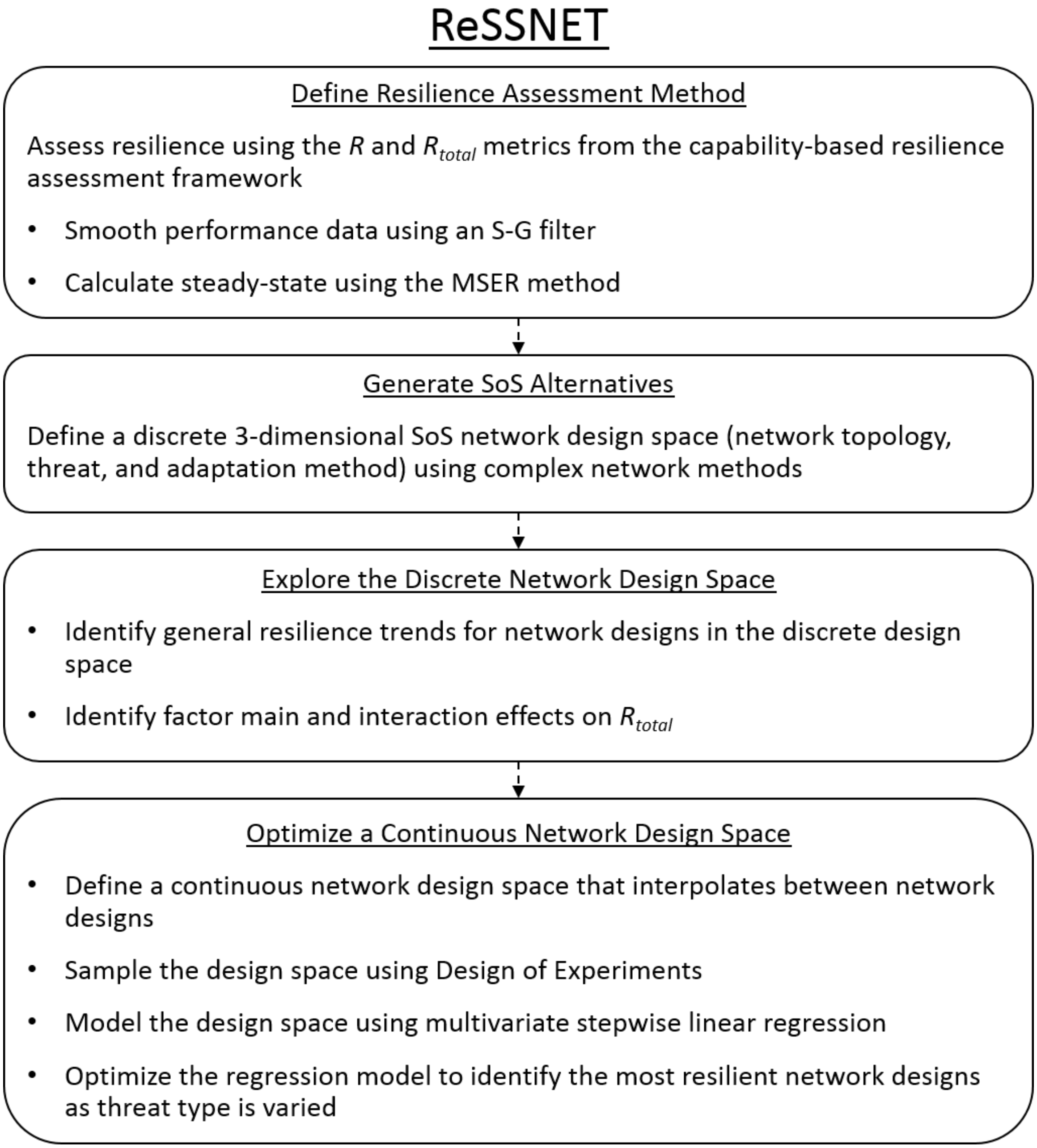
- RQ 5: What methods should be used to compare SoS design alternatives and understand their advantages and disadvantages?
- Response to RQ 5: An RSM approach with multiple linear regression is used to explore and optimize SoS design alternatives, providing insight into the advantages and disadvantages of potential SoS networks.
- RQ 5.1: What models should be used to interpolate between scale-free and

random networks, and RD and random threats?

- Response to RQ 5.1: The GM model is used to interpolate between scale-free and random networks, as well as RD and random threats using GM2 targeting.

6.4 Summary of the ReSSNET Methodology

Chapters 3 to 6 develop the ReSSNET methodology for designing resilient SoS networks, satisfying research objective one for this thesis. The methodology is composed of three primary steps: quantifying resilience, generating an SoS network design space, and evaluating those SoS network designs. Figure 47 summarizes the methods used to perform each step.



ReSSNET

Define Resilience Assessment Method

Assess resilience using the R and R_{total} metrics from the capability-based resilience assessment framework

- Smooth performance data using an S-G filter
- Calculate steady-state using the MSER method

Generate SoS Alternatives

Define a discrete 3-dimensional SoS network design space (network topology, threat, and adaptation method) using complex network methods

Explore the Discrete Network Design Space

- Identify general resilience trends for network designs in the discrete design space
- Identify factor main and interaction effects on R_{total}

Optimize a Continuous Network Design Space

- Define a continuous network design space that interpolates between network designs
- Sample the design space using Design of Experiments
- Model the design space using multivariate stepwise linear regression
- Optimize the regression model to identify the most resilient network designs as threat type is varied

Figure 47: Overview of the ReSSNET methodology for designing resilient SoS networks.

CHAPTER VII

DESIGNING RESILIENT IE NETWORKS

The second research objective for this thesis is to use the developed methodology to investigate the first overarching research question (RQ 1). This research question asks what happens to an SoS network when nodes fail or are attacked, and seeks to identify methods for mitigating the effects of potential node losses. The IE network model is used as an application problem for the methodology. The capability-based resilience assessment framework is used to quantify the ability of IE networks to maintain and recover lost capabilities, where IE performance is defined from the number of received messages. IE network alternatives are generated by specifying initial network topologies, adaptation methods, and threats. Design alternatives are evaluated with statistical experimental design methods. The evaluation step is split into two phases. The first phase explores the resilience of generated categorical network designs and is discussed in section 7.1. The second phase optimizes a continuous representation of the network design space and is discussed in section 7.2.

This chapter describes experiments four and five, which explore and optimize the IE network design space, satisfying the second research objective for this thesis.

7.1 EXP 4: Exploring the IE Network Design Space

Experiment four (EXP 4) explores a categorical IE network design space generated using methods from chapter 5. This experiment investigates two questions derived from research question one; one focusing on the resilience of networks without adaptation to node losses, the other focusing on the resilience gained by incorporating adaptation to SoS networks. These derived research questions are formalized as follows, with respect to resilience as measured by R_{total}:

> **Research Question 1.1:** How resilient are scale-free and random IE networks to targeted attacks and random node failures, with no adaptation considered and resilience measured by R_{total}?
>
> **Research Question 1.2:** How is the resilience gained by adding network adaptation to IE networks affected by threat type and initial network topology?

This experiment is split into two sub-experiments, experiment 4.1 and 4.2. Experiment 4.1 focuses on answering research question 1.1 by comparing network designs without adaptation. Experiment 4.2 focuses on answering research question 1.2 by adding adaptation to SoS networks.

Experimental Setup

Experiments 4.1 and 4.2 use the IE network model as an application problem with model input parameters defined by table 9. A maximum simulation time of 1000 seconds with nine threats per scenario results in threat events occurring every 200 seconds in a simulated scenario. For network designs with adaptation, adaptation occurs 50 seconds after each threat event. A message generation probability, or rate, of 0.25 means that the expected number of new messages generated by an entire network per time step is 25.

The resilience assessment framework developed in chapter 4 is used to quantify the resilience of design alternatives. Figure 48 shows how the assessment framework is applied to these experiments. Performance data is smoothed using an S-G filter $(n = 3, m = 5)$. Two sets of performance data are used; one set with $\Delta = 1$, the other set with $\Delta = 0.8$. Results using performance data with $\Delta = 1$ represent scenarios where message time sensitivity is low. Results using performance data with $\Delta = 0.8$ represent scenarios where time sensitivity is high, such that a long message travel time degrades the value of a message.

Table 9: IE network model parameters used for EXP 4

Parameter	Settings used	Description
N	100	Initial network size
L	200	Initial number of links in a network
$[t_0\ t_{final}]$	[1 1000 s]	Simulated scenario time interval (in seconds)
S_{threat}	5	Number of nodes removed per threat event
$N_{threats}$	9	Number of threat events within a scenario
t_{adapt}	50 s	Adaptation delay time following a threat event (in seconds)
μ	0.25	Message generation rate
Δ	1 (EXP 4.1) 0.8, 1 (EXP 4.2)	Time sensitivity

The desired performance level of a network, y_D, used in resilience calculations [e.g., in eq. (32)] is defined as the steady-state mean network performance [where network performance is calculated using eq. (6)] within the initial time period before the first threat event (i.e., the steady-state mean network performance for the time period specified by $t \leq 199$). Thus, y_D is calculated using eq. (40) with $N = t_{final} = 199$. The steady-state mean of the initial time period is used because it represents normal network operating conditions. The first threat event is defined to provide enough time for the system performance to reach steady-state.

This definition for y_D results in resilience calculations relative to the initial performance level of a network. Therefore, analysis with R and R_{total} is focused on network resilience, rather than overall network performance. Conclusions drawn from this analysis can be supplemented with performance-based analysis to provide a more complete understanding of IE network performance.

Design alternatives considered in experiment four are shown in table 10. 100 replications are run for each design point. Experiment 4.1 analyzes the first four design points, which can be represented as a 2^2 factorial design. These designs are viewed as baseline designs, since they do not include adaptation. Experiment 4.2 focuses on the remaining design points, with comparisons to the baseline to determine

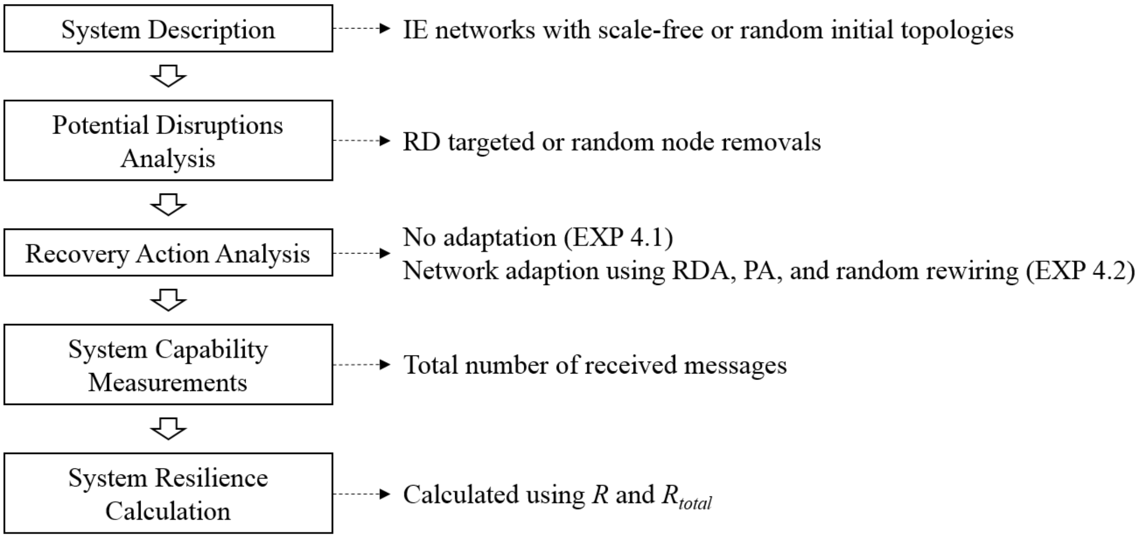

System Description
IE networks with scale-free or random initial topologies
Potential Disruptions Analysis
RD targeted or random node removals
Recovery Action Analysis
No adaptation (EXP 4.1)
Network adaption using RDA, PA, and random rewiring (EXP 4.2)
System Capability Measurements
Total number of received messages
System Resilience Calculation
Calculated using R and R_{total}

Figure 48: Application of the capability-based resilience assessment framework to the scenarios considered in EXP 4.

the benefits of adaptation.

7.1.1 EXP 4.1: IE Network Resilience without Adaptation

Purpose of the experiment: Answer research question 1.1 by exploring the resilience of IE networks without adaptation.

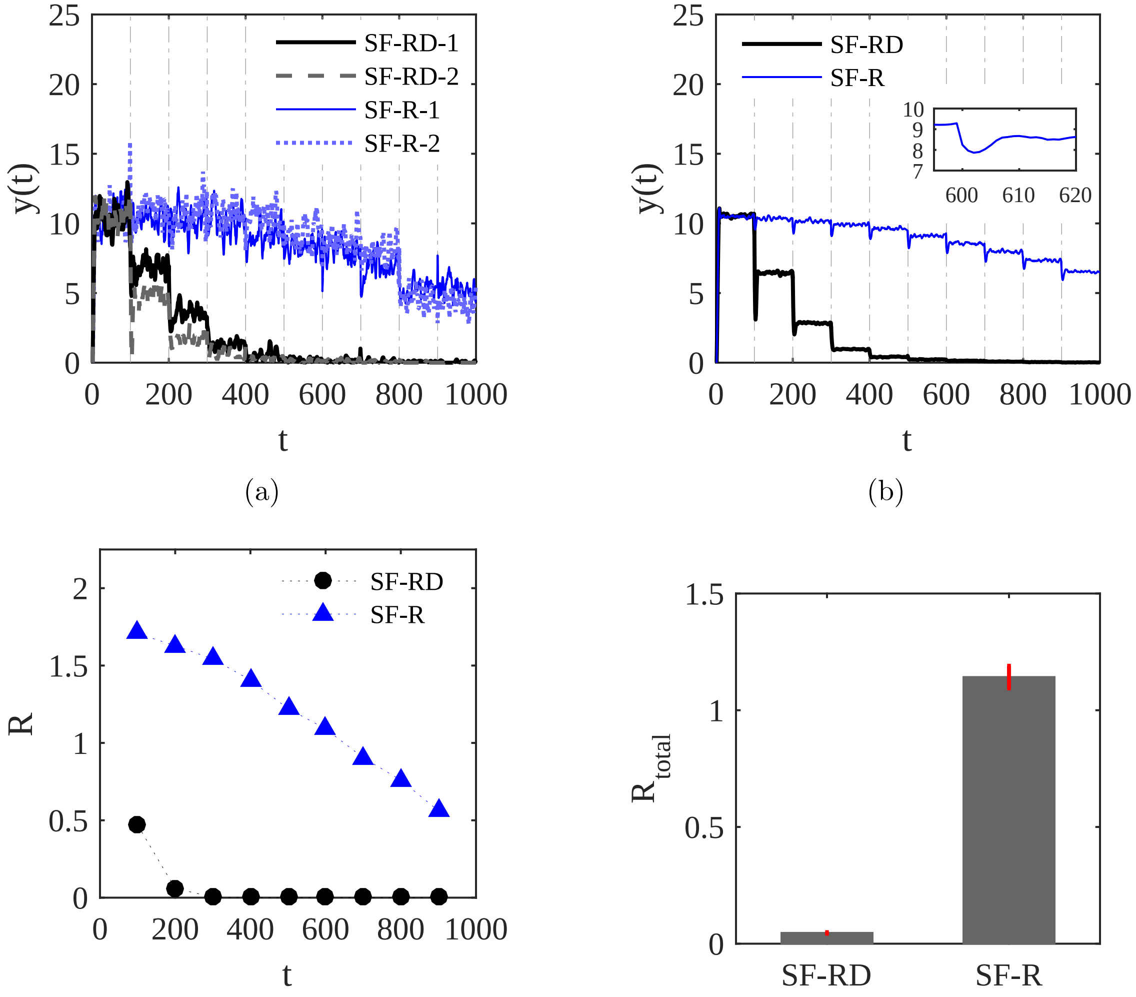
Experiment 4.1 investigates research question 1.1 by comparing the resilience of designs with scale-free and random topologies facing RD targeted and random threats. All network performance levels for experiment 4.1 are calculated using eq. (6) with $\Delta = 0.8$ to represent networks exchanging time sensitive information.

Figure 49 shows results for the resilience of scale-free networks to RD and random threats. Figure 49a shows performance data from two replications for these design points, or scenarios (the first two replications are arbitrarily used). Figure 49b shows mean performance data for these design points, where the mean is calculated from all 100 replications of each design point, using eq. (54) for data at each time step t with $m = 100$. Figure 49c shows mean R results calculated from performance data; note that R is not calculated from the mean performance data. Instead, a set of nine R values (one for each epoch) is calculated for each replicated simulation run.

Table 10: Experimental design matrix used for EXP 4 (100 replications at each point)

Design point	Topology	Adaptation	Threat
1	Scale-free	None	RD
2	Random	None	RD
3	Scale-free	None	Random
4	Random	None	Random
5	Scale-free	RDA	RD
6	Random	RDA	RD
7	Scale-free	PA	RD
8	Random	PA	RD
9	Scale-free	Random rewiring	RD
10	Random	Random rewiring	RD
11	Scale-free	RDA	Random
12	Random	RDA	Random
13	Scale-free	PA	Random
14	Random	PA	Random
15	Scale-free	Random rewiring	Random
16	Random	Random rewiring	Random

Since 100 replications are run for each design point, 100 sets of nine R values are calculated for each scenario, where each set of R values is calculated from a single replicated simulation run. These data sets of R are then averaged over all replications using eq. (54) with $m = 100$ for each epoch, and plotted in fig. 49c. The R values plotted at time $t = 100$ correspond to the resilience of these design points in the first epoch, which spans the time interval $100 \leq t \leq 199$. A similar process is used to calculate R_{total} for each design point, where a single R_{total} value is calculated for each replicated simulation run. These R_{total} values are then averaged over all replications using eq. (54) with $m = 100$. Figure 49d shows the mean R_{total} for each design point, with 90 percent confidence intervals calculated using eq. (56).



(a)

(b)

(c)

(d)

Figure 49: Results from design points one and three, with scale-free topologies subjected to RD threats (SF-RD) and random threats (SF-R). Capability data from replications one (SF-RD-1, SF-R-1) and two (SF-RD-2, SF-R-2) are shown in (a). Mean performance data is shown in (b). Mean R results are shown in (c). Mean R_{total} results are shown in (d), with 90% confidence intervals for R_{total} marked by red error bars. Vertical dashed lines show threat event times. Capability is calculated with $\Delta = 0.8$.

These results show that scale-free networks are more resilient to random threats than RD targeted threats, as measured by R_{total}. The performance and epoch-based resilience, R, of these networks significantly decreases after the first RD threat event, at which point only five percent of the most connected network nodes have been removed. This point at which a significant decrease in performance occurs can be viewed as a critical point in the performance of the network. In comparison, performance and R results show a linear decrease over time for random threats.

The small size of 90 percent confidence intervals on R_{total} suggests that the differences seen among the resilience of considered designs is due to changes in the designs themselves, rather than randomness in the model.

Figure 50 shows a probabilistic visualization of the data, focusing on median values, 25 and 75 percent quartiles, minimum and maximum ranges, and probability density functions (PDFs) and cumulative density functions (CDFs). These results show that more variability is seen in results with random threats, than those with RD targeted threats. However, the range in R_{total} for scenarios with random threats is always higher than the range for scenarios with RD threats. This result suggests that a scale-free network facing an RD threat will nearly always have lower resilience than one facing a random threat, even with variability due to randomness in the model.

Figures 51 and 52 show results for random networks subjected to RD and random threats. Compared to scale-free networks, random networks show similar resilience to random threats, but higher resilience to RD threats. This increased resilience is seen in a delay in the critical point, or time at which the performance of the network quickly degrades to low levels, compared to results for scale-free networks. The range of R_{total} seen for scenarios facing random threats still shows no overlap with the range of R_{total} for scenarios facing RD threats.

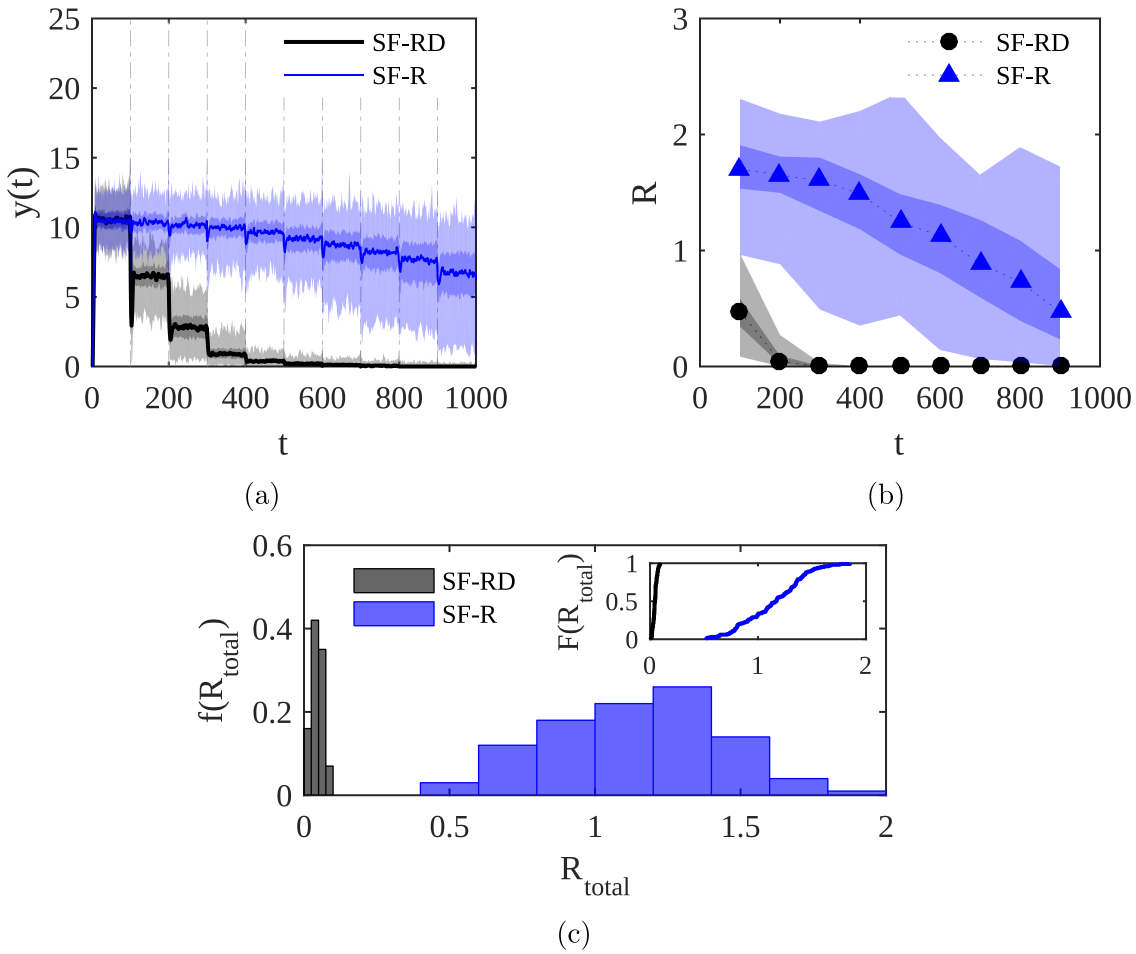
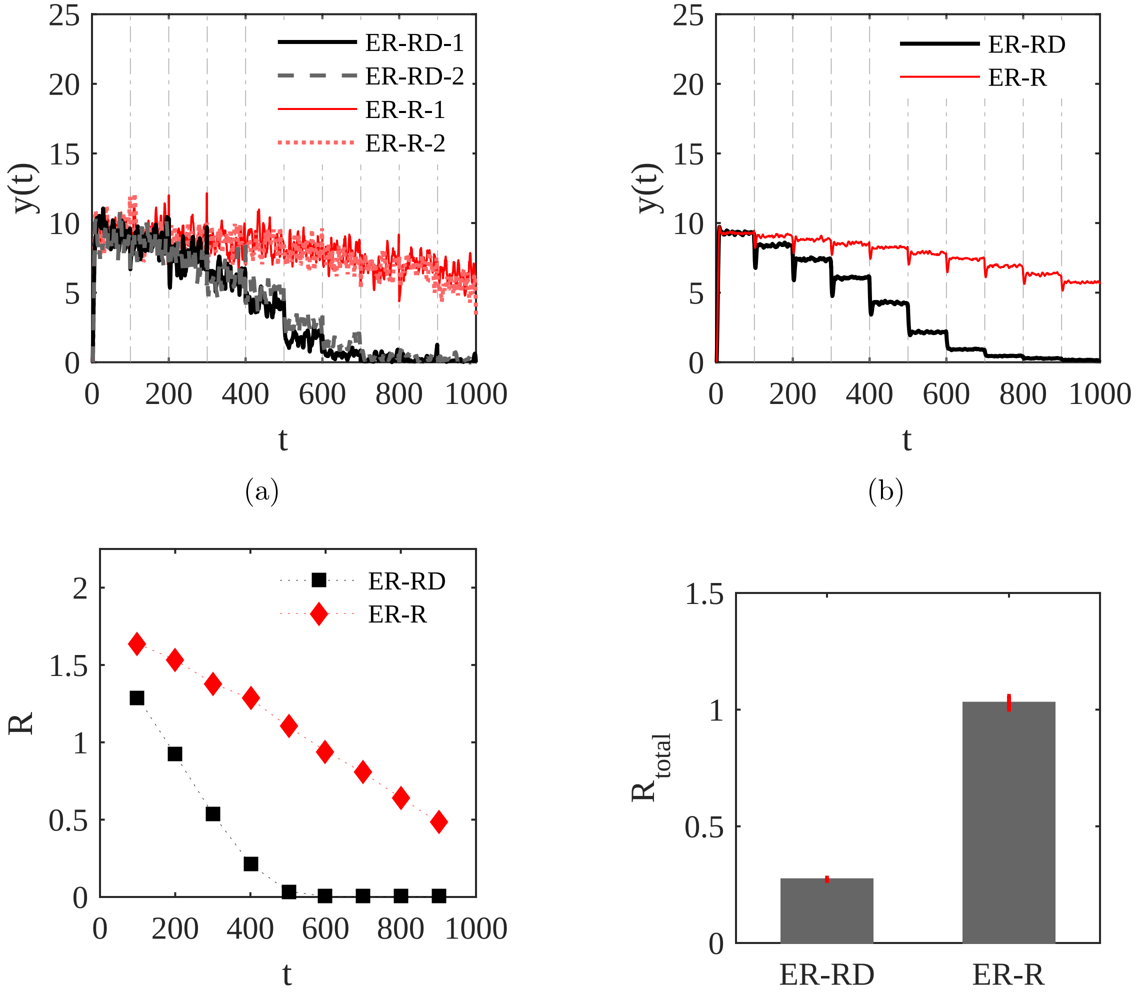


Figure 50: Probabilistic results from design points one and three. Capability quantiles are shown in (a), where solid lines show median data, dark shaded regions show 25 and 75% quartiles, and light shaded regions show the full range of data from all 100 replications. R quantiles are similarly shown in (b). PDFs for R_{total} are shown in (c), with corresponding CDFs shown in the inset of (c).



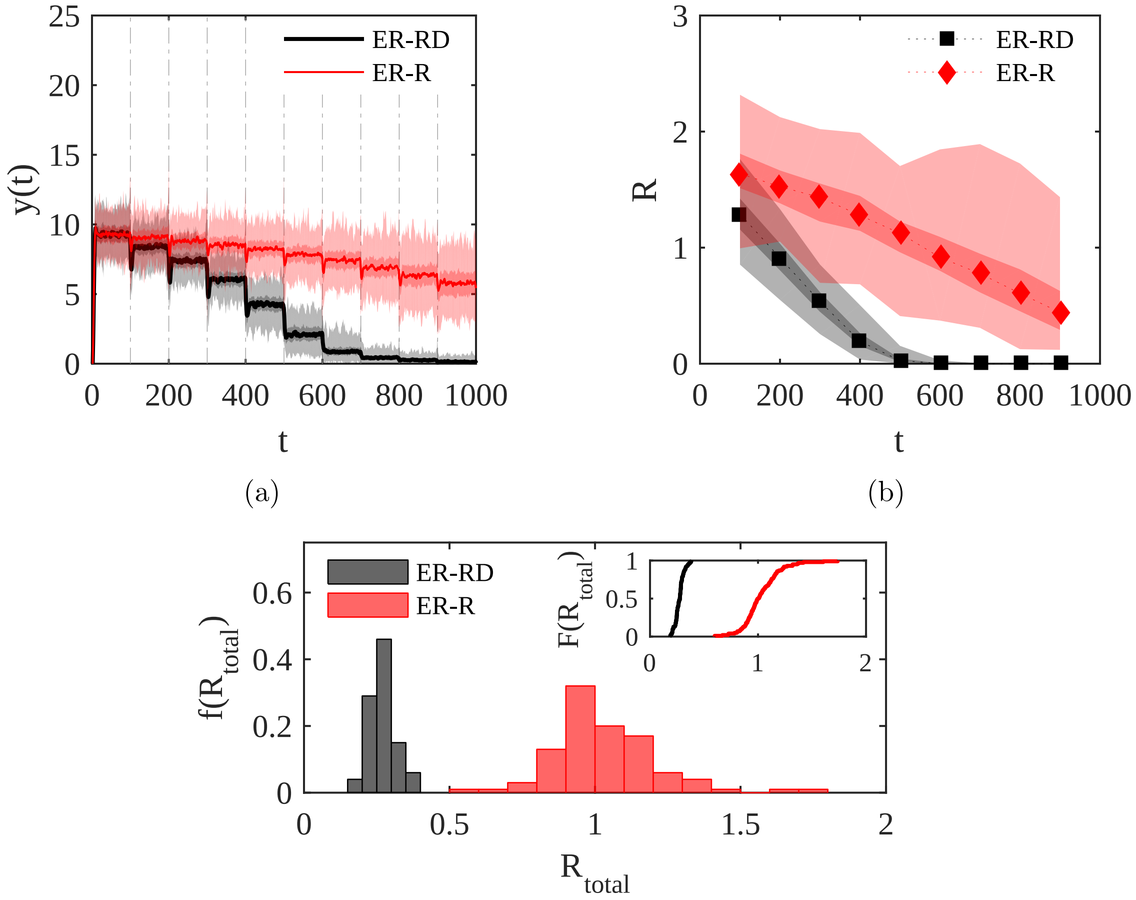
(a)

(b)

(c)

(d)

Figure 51: Results from design points two and four, with random topologies subjected to RD threats (ER-RD) and random threats (ER-R). Capability data from replications one (ER-RD-1, ER-R-1) and two (ER-RD-2, ER-R-2) are shown in (a). Mean performance data is shown in (b). Mean R results are shown in (c). Mean R_{total} results are shown in (d), with 90% confidence intervals for R_{total} marked by red error bars.



(a)

(b)

(c)

Figure 52: Probabilistic results from design points two and four. Capability quantiles are shown in (a), where solid lines show median data, dark shaded regions show 25 and 75% quartiles, and light shaded regions show the full range of data from all 100 replications. R quantiles are similarly shown in (b). PDFs for R_{total} are shown in (c), with corresponding CDFs shown in the inset of (c).

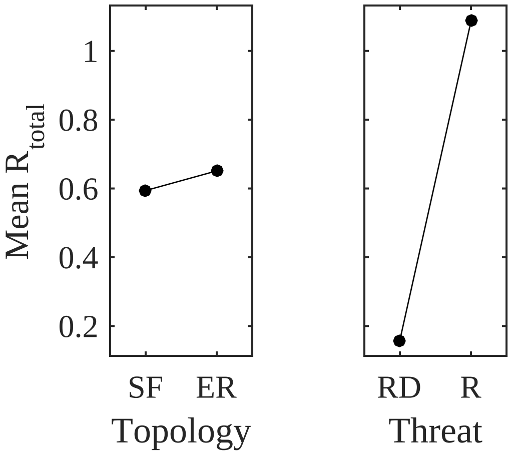


Figure 53: Main effects of network topology and threat type on R_{total}.

Figure 53 shows main effects of the network topology and threat type on R_{total}. Changing threat type has over eight times the effect on mean R_{total} as changing network topology. This analysis also shows that resilience is highest, on average and without consideration of interactions, for scenarios with random topologies and random threats. However, the potential for an interaction between topology and threat type limits conclusions that can be drawn from these main effects. ANOVA shows that calculated main effects are statistically significant, where $p_{topology} \approx 0.0031$ and $p_{threat} \approx 0$.

Figure 54 show that an interaction exists between network topology and threat type, with respect to effects on R_{total}. The effect of changing from a scale-free to random network depends on the threat faced, where scale-free topologies are more resilient than random topologies when facing random threats, but random topologies are more resilient than scale-free topologies when facing RD threats. This result agrees with those seen in figs. 49 and 51. Scenarios facing RD threats have lower resilience, on average, than those facing random threats; however, the effect of changing from an RD to random threat depends on the network topology. Changing threats has a larger effect on R_{total} for designs with scale-free topologies than those with random topologies. This analysis shows the importance of considering interaction effects

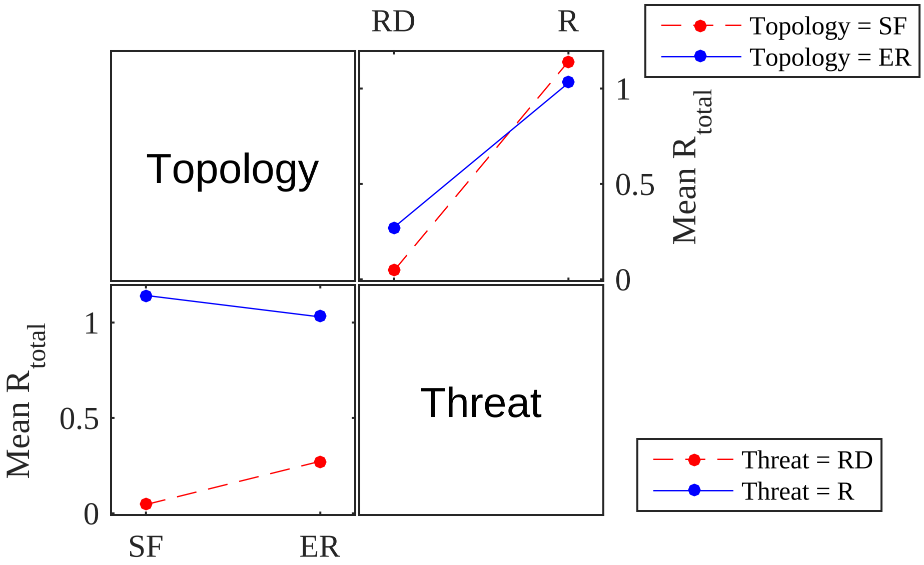


Figure 54: Interactions effects of network topology and threat type on R_{total}.

in addition to main effects, since only analyzing main effects may have led to the conclusion that random topologies are always preferred.

Discussion of Results

Experiment 4.1 shows that scale-free networks are more resilient to random node failures than random networks. This resilience is due to scale-free networks having many nodes with low connectivity. Therefore, when nodes are randomly removed, the average connectivity of removed nodes is low. Scale-free networks also have network hubs, which provide short path lengths between nodes. Since this experiment includes time sensitivity in performance calculations (i.e., $\Delta < 1$), these short path lengths reduce message travel times and increase network performance, relative to random networks which generally have longer average path lengths. However, network hubs also make scale-free networks less resilient to RD targeted attacks than random networks, since removing the most connected nodes significantly decreases connectivity for scale-free networks.

The observed resilience of scale-free networks to random threats, but susceptibility to targeted threats, agrees with results from the complex networks literature, where resilience is measured by changes to network structural properties (see section 2.3.3). This agreement suggests that network structural properties are critical to the ability of IE networks to effectively distribute information, and have the potential to be used as a surrogate for network performance or R for the model considered. The following answer is provided to research question 1.1:

> **Response to RQ 1.1:** Scale-free SoS networks are more resilient to random failures than random SoS networks but less resilient to RD targeted attacks, with no adaptation considered and resilience measured by R_{total}. These results agree with those from the complex networks literature where resilience is measured by network structural properties, suggesting the potential to use those properties as surrogates for network performance or R in the IE network model.

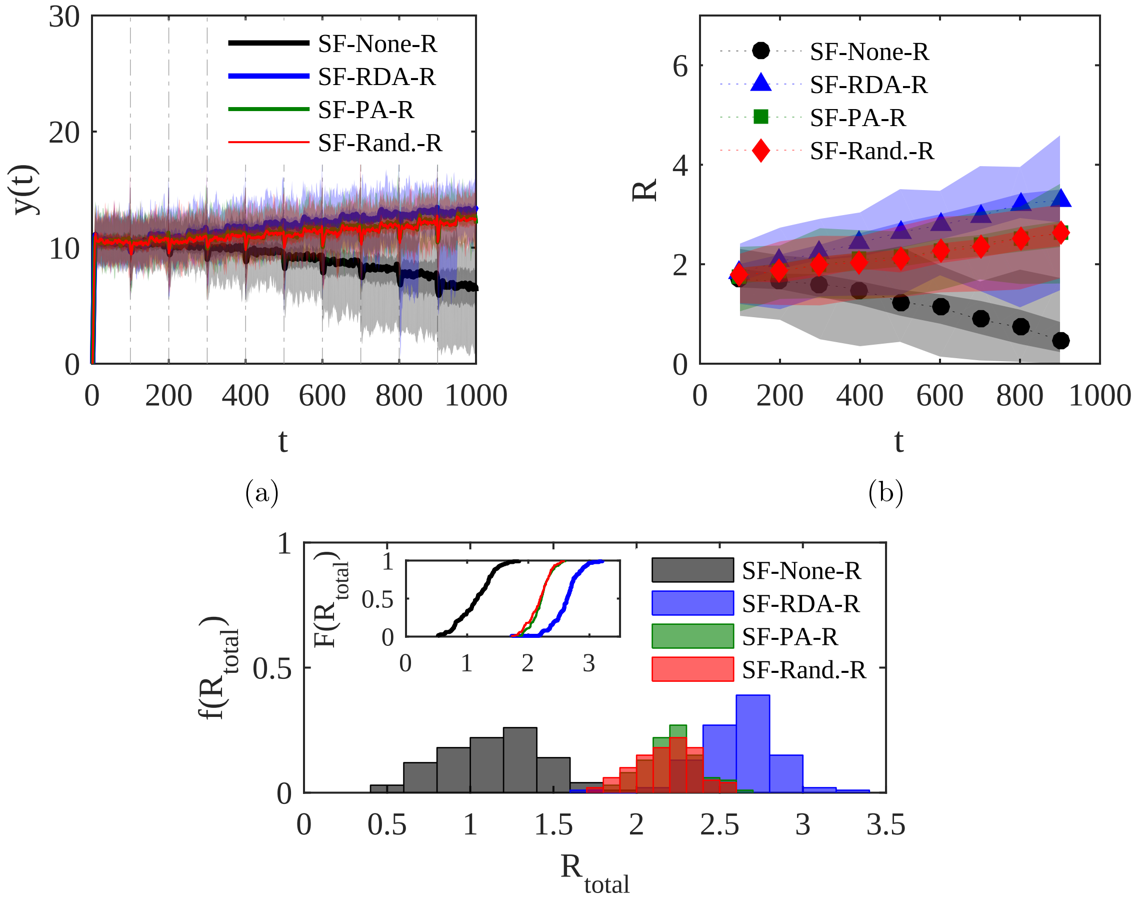
7.1.2 EXP 4.2: IE Network Resilience with Adaptation

Purpose of the experiment: Answer research question 1.2 by exploring the resilience of IE networks with adaptation.

Experiment 4.2 investigates research question 1.2 by adding potential adaptation methods to IE network designs. Network performance is calculated with $\Delta = 0.8$ and 1. Results for $\Delta = 0.8$ are discussed first.

Figure 55 shows that for scale-free topologies facing random threats, adding adaptation allows networks to recover lost capabilities and become more resilient to node losses over time. All three adaptation methods show an increase in network performance and resilience measured with R over time. These metrics increase because as nodes are removed and links rewired, network density increases and network size decreases. Since nodes are allowed to rewire any lost links, the overall number of links in a network stays relatively constant over time. Therefore, as nodes are removed, network density increases, which makes it easier for source nodes to find short paths to target nodes. Smaller network sizes also decrease average distances between nodes. Figure 56 shows how mean network density and inverse average path length increase throughout simulated scenarios. Since $\Delta = 0.8$, this decrease in path lengths (and therefore message travel times) improves network performance and R. Figure 56b also shows that designs with RDA-based adaptation achieve higher inverse average path length than those with PA or random rewiring. These shorter path lengths translate to slightly higher network capabilities, R, and R_{total} values for designs with RDA, and suggest a positive correlation between inverse average path length and R.

This correlation between inverse average path length, IE network performance, and R supports the observation from experiment 4.1 that network structural properties have the potential to be used as a surrogate for network performance or R. However, similar correlations are not seen between the size of the LCC and IE network performance or R. Figure 57 shows that changes to LCC size throughout simulated



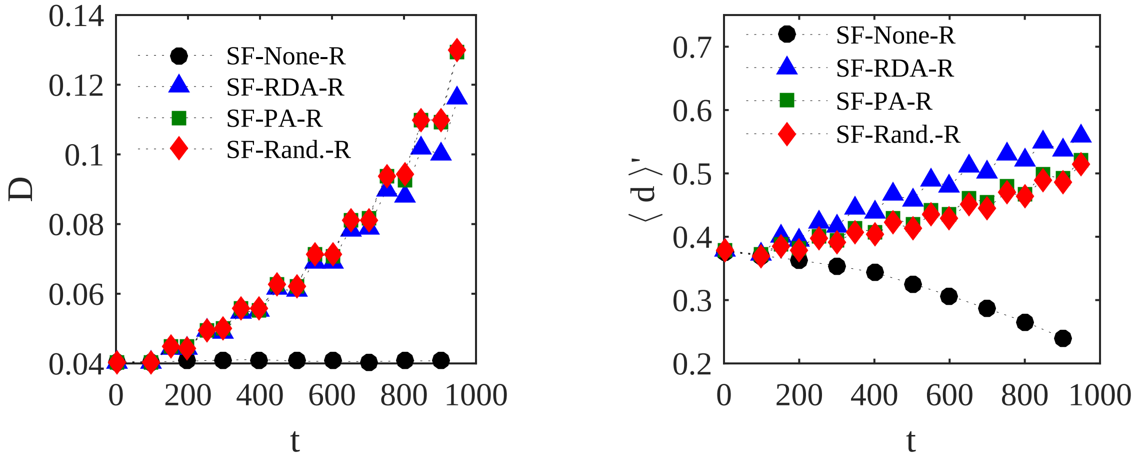
(a)

(b)

(c)

Figure 55: Probabilistic results for scale-free topologies (SF) subjected to random threats (R) with no adaptation (SF-None-R), recalculated degree-based adaptation (SF-RDA-R), and random rewiring (SF-Rand.-R). Capability quantiles are shown in (a), R quantiles are shown in (b), and PDFs for R_{total} with corresponding CDFs are shown in (c).

scenarios are nearly identical for all three adaptation methods, and relatively close in value to networks without adaptation. Therefore, LCC size should not be used as a surrogate for IE network performance or R.



(a)

(b)

Figure 56: Mean (a) network density and (b) inverse average path length throughout simulated scenarios, as nodes are removed and links rewired.

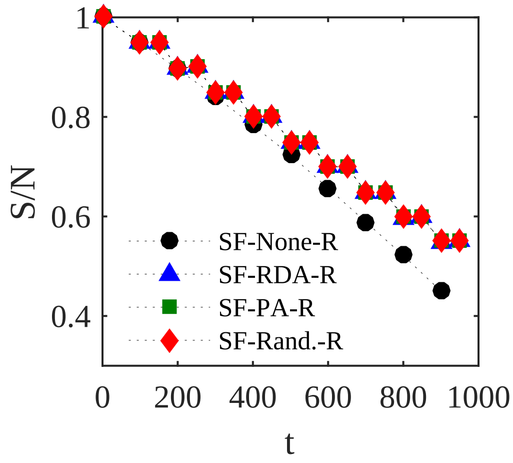
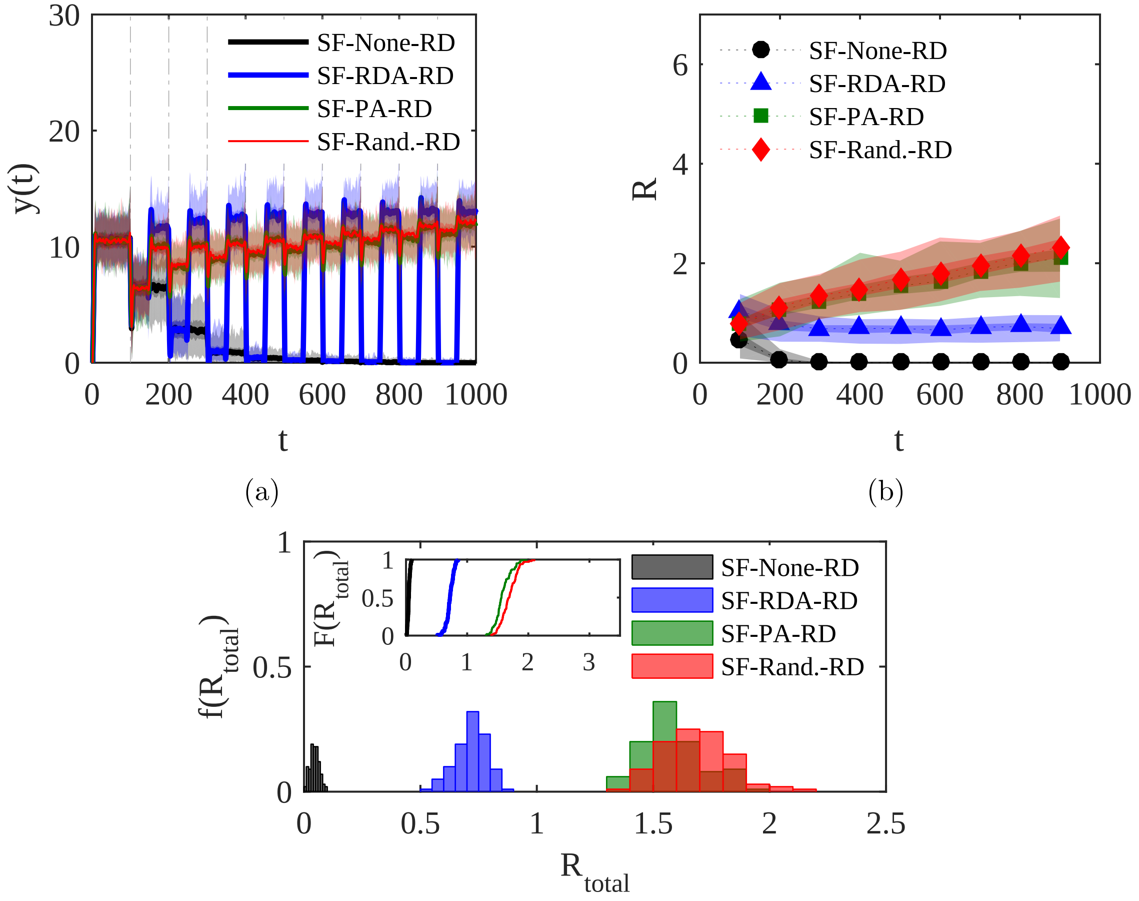


Figure 57: Mean size of the LCC (normalized by initial network size N) throughout simulated scenarios, as nodes are removed and links rewired.

Though all three adaptation methods improve the resilience of designs without any adaptation, distributions on R_{total} show that given the stochastic nature of the simulation, there is some overlap between the resilience of designs with and without adaptation. This overlap suggests the possibility for a scale-free network without adaptation to actually have higher resilience than one with adaptation, depending on the actual nodes removed by random threats. Therefore, the cost of incorporating adaptation should be considered when random threats are anticipated, due to the potential for no benefits from adaptation depending on removed nodes.

Figure 58 shows that similar to scenarios with random threats, adding adaptation to scale-free topologies facing RD targeted threats increases network performance and R over time. However, RDA-based adaptation shows lower resilience than PA and random rewiring-based adaptation for these scenarios. This decrease in resilience is due to large drops in network performance following node removals, as shown by fig. 58a. These large performance degradations occur because rewiring by recalculated node degree creates highly connected network hubs following adaptation events. These hubs provide short path lengths throughout the network, which result in high performance levels following adaptation. However, these hubs also make networks vulnerable to RD node removals, which result in large drops in performance following removals. For these scenarios, the benefits of node hubs are outweighed by the vulnerabilities they create, which results in lower R and R_{total} values for designs with RDA, compared to those with other adaptation methods.

Probabilistic results for random topologies show similar trends to those for scale-free topologies, with slightly higher increases in resilience due to adaptation (not shown here). Therefore, the remaining analysis focuses on mean R_{total}. Measuring resilience with mean R_{total} provides a single resilience metric for simpler comparisons between designs. Capability plots, R results, and distributions of R_{total} are only discussed when they provide additional insights to analysis of mean R_{total}.



(a)

(b)

(c)

Figure 58: Probabilistic results for scale-free topologies (SF) subjected to RD threats (RD) with no adaptation (SF-None-RD), recalculated degree-based adaptation (SF-RDA-RD), and random rewiring (SF-Rand.-RD). Capability quantiles are shown in (a), R quantiles are shown in (b), and PDFs for R_{total} with corresponding CDFs are shown in (c).

Figure 59 compares the mean total resilience of all 16 design points considered in experiment four. As suggested by probabilistic results for scale-free network designs, adding adaptation improves resilience, on average, for all topology and threat combinations considered. Results also show that the resilience gained by PA and random rewiring-based adaptation are very similar for considered combinations of topology and threat type. However, the resilience gained by RDA-based adaptation differs from that of PA and random rewiring. For RD threats, designs with PA or random rewiring show nearly twice the resilience of those with RDA-based adaptation.

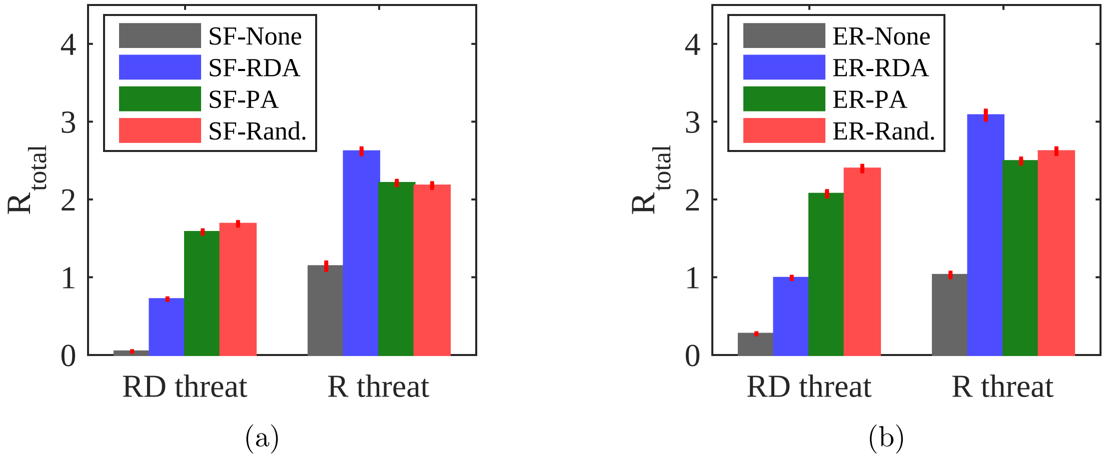


Figure 59: Mean R_{total} for all designs considered in EXP 4, where designs with scale-free topologies (SF) are shown in (a) and designs with random topologies (ER) in (b). Adaptation methods are specified as none, recalculated degree adaptation (RDA), preferential adaptation (PA), and random rewiring adaptation (Rand.). 90% confidence intervals for mean R_{total} are shown by red intervals on each bar.

Yet for random threats, designs with RDA show higher resilience than those with PA or random rewiring. Therefore, anticipated threat types should be considered when deciding what type of adaptation to incorporate, in addition to whether or not adaptation should be incorporated in the first place.

Main effects analysis in fig. 60 shows that threat type has the largest main effect on R_{total}, where changing from RD to random threats increases the mean R_{total} by nearly one. This analysis considers effects of network topology, adaptation method, and threat type on R_{total}, excluding designs with no adaptation (i.e., not including design points 1-4). Network topology and adaptation method show similar main effects on R_{total}, when considering the effect of changing adaptation method from RDA to random rewiring. The effect of changing adaptation method from PA to random rewiring is smaller than all others. Only considering main effects suggests that increasing randomness in topology, adaptation method, and threat generally improves resilience, when averaged over all designs considered. However, inspection of interaction effects is required to make more conclusive observations. Calculated main

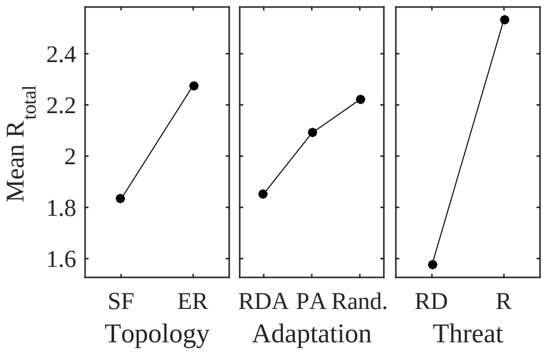


Figure 60: Main effects of network topology, adaptation method, and threat type on R_{total}.

effects are accepted to be statistically significant due to low p-values for those effects from ANOVA ($p_{topology} \approx 7.6 \times 10^{-65}, p_{adaptation} \approx 3.2 \times 10^{-32}, p_{threat} \approx 2.3 \times 10^{-216}$).

Analysis of interaction effects (shown in fig. 61) shows that interactions between network topology and adaptation method, as well as network topology and threat type, are small, with respect to R_{total}. Low interactions are evidenced by nearly parallel interaction effects among factors. These results suggest that as long as some form of adaptation is included, random topologies show higher mean resilience than scale-free topologies regardless of adaptation method and threat type.

However, there is a strong interaction between adaptation method and threat type. For RD targeted threats, changing adaptation method from RDA to PA or random rewiring has a large positive effect on resilience, increasing mean R_{total} by approximately one. However, for random threats, changing adaptation method from RDA to PA or random rewiring has a large negative effect on resilience, decreasing mean R_{total} by approximately 0.5. This interaction between adaptation method and threat type agrees with results from bar charts in fig. 59 showing that benefits from adaptation methods depend on threat type. This interaction is also seen when comparing the effects of changing threat type, for different adaptation methods. Changing threat type from RD to random has nearly twice the effect on mean R_{total} when RDA is used rather than PA or random rewiring. This result suggests that PA and random

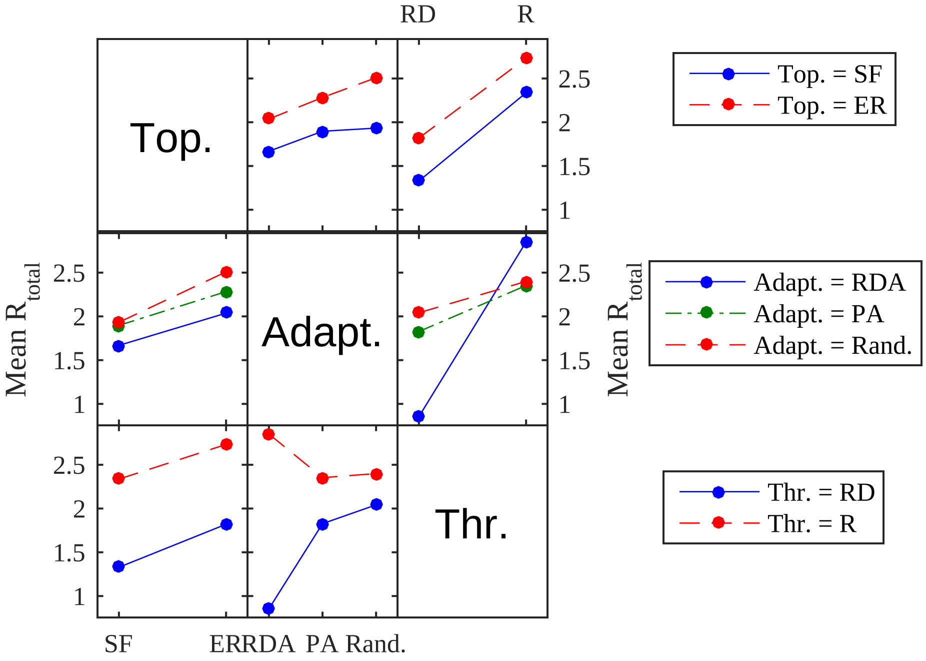
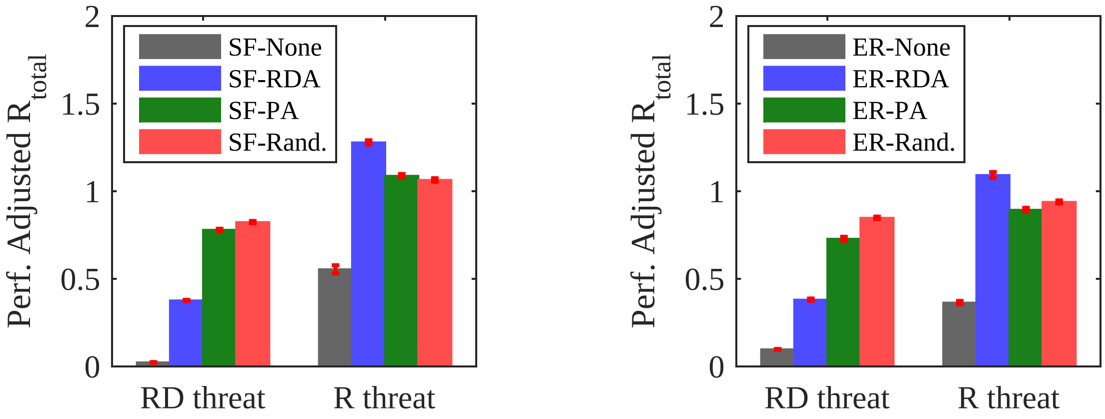


Figure 61: Interaction effects of network topology, adaptation method, and threat type on R_{total}.

rewiring are more robust to threat type than RDA.

Note that since y_D is defined from the initial capability of a network, this analysis focuses on network resilience rather than network performance. To more explicitly consider network performance in resilience calculations, fig. 62 shows results for performance adjusted R_{total} calculations, where the same y_D is used as the desired capability for all networks. The desired capability for this analysis is defined as $y_D = \mu \times N \times \Delta^d$, where $d = 2.6483$. This calculation normalizes all network performance data to the same initial network capability. d is defined as the mean average path length of scale-free networks created for this experiment, because this value represents the best travel times expected from considered networks.

Performance adjusted results show that when the same y_D is used for all networks, adaptive scale-free networks are able to achieve nearly the same resilience as random networks when facing targeted threats. Therefore, while random topologies



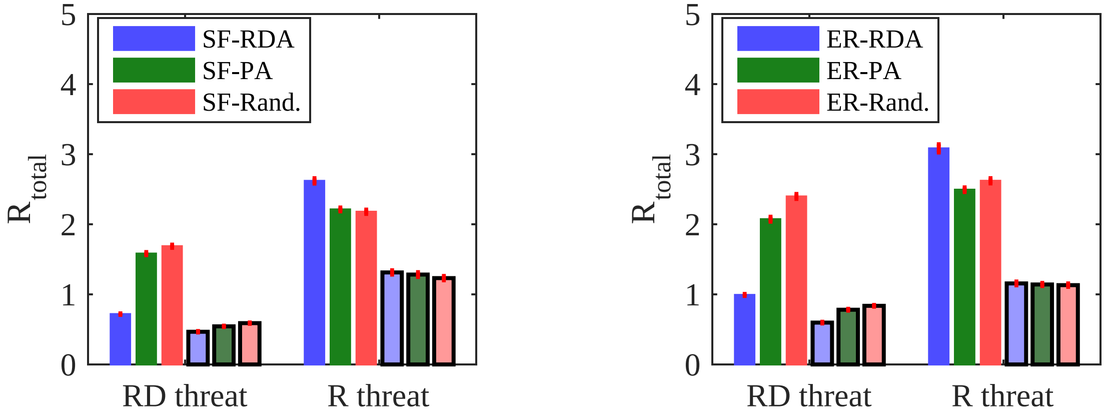
(a)

(b)

Figure 62: Mean performance adjusted R_{total} for all designs considered in EXP 4, where R_{total} is adjusted to more explicitly account for network performance by using the same desired performance level, y_D, for all networks (regardless of their initial performance). 90% confidence intervals for mean R_{total} are shown by red intervals on each bar.

provide more resilience than scale-free topologies, they do not always provide better performance. However, since this thesis focuses on network resilience, remaining calculations use the unadjusted R_{total} calculations, where y_D is defined relative to the initial capability of a network.

As seen in fig. 56a, adaptation methods implemented do not restrict network density, allowing it to increase as nodes are removed and networks adapt through link rewiring. Rather, the primary restrictions on adaptation are that only previously existing links are rewired and no new links added. Network adaptation can be further constrained to require that network density never increases relative to the initial density at time $t = 0$. This type of constant density adaptation represents scenarios in which the number of links a network can sustain is defined relative to its current network size, rather than initial network size. An SoS network with this type of constraint might be one where links are powered by nodes, therefore requiring that the ratio of links to nodes stays relatively constant over time.

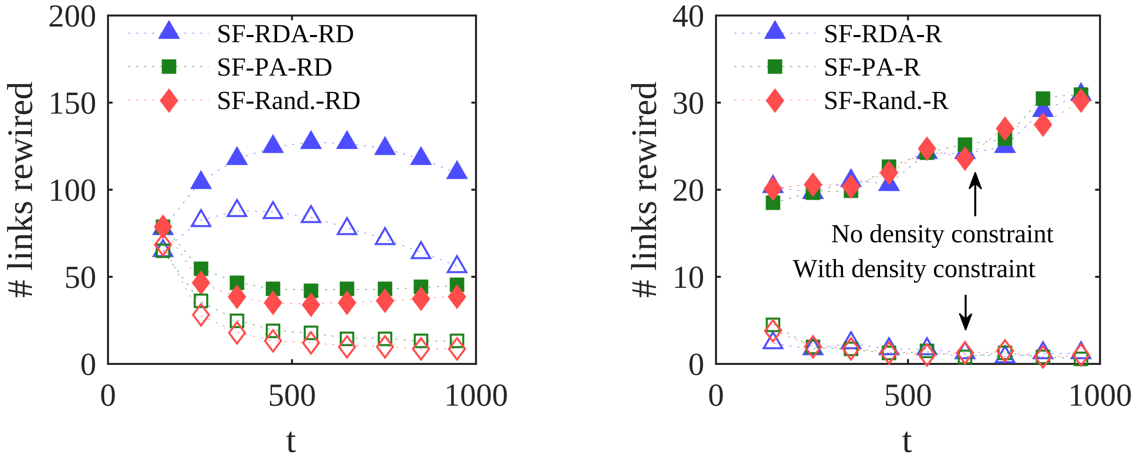


(a) (b)

Figure 63: Mean R_{total} for designs with scale-free topologies (SF) are shown in (a) and designs with random topologies (ER) in (b). Adaptation methods are specified as none, recalculated degree adaptation (RDA), preferential adaptation (PA), and random rewiring adaptation (Rand.). Adaptation methods without a constant density constraint are specified by bars without black borders. Adaptation methods with a constant density constraint are specified by bars with black borders. 90% confidence intervals for mean R_{total} are shown by red intervals on each bar.

Figure 63 shows that constraining network adaptation by initial network density collapses differences between considered adaptation methods, such that all three methods show similar resilience to specified threats. Little variation is seen among density constrained adaptation methods because few links are allowed to be rewired. However, density constrained adaptation methods show the same trends compared to each other as unconstrained methods do, i.e. RDA provides the most resilience to random threats, while random rewiring provides the most resilience to RD threats. Figure 64 shows the number of links rewired at each adaptation event for scenarios with and without density constraints. Similar results are seen for scenarios with random networks. Since little differences are seen between adaptation methods with density constraints, these methods are not considered for the remainder of this thesis. Only adaptation without density constraints are used.

All results shown have been processed with $\Delta = 0.8$, i.e. assuming time sensitivity for received messages. Scenarios with no time sensitivity (i.e., $\Delta = 1$) are



(a)

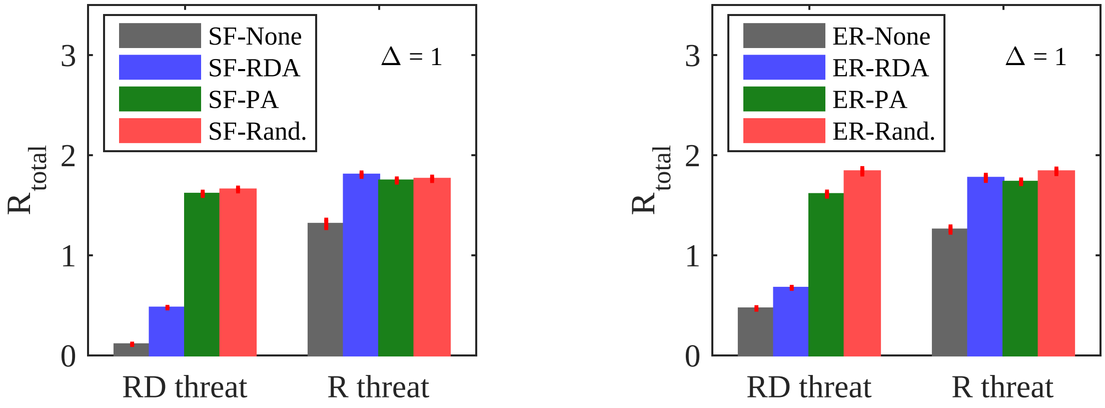
(b)

Figure 64: Mean number of links rewired at each adaptation event for designs with scale-free topologies facing (a) RD threats and (b) random threats. Adaptation methods are specified as recalculated degree adaptation (RDA), preferential adaptation (PA), and random rewiring adaptation (Rand.). Adaptation methods without a constant density constraint are specified by filled in symbols. Adaptation methods with a constant density constraint are specified by unfilled symbols.

also considered. Figure 65 shows that similar trends are seen between scenarios with $\Delta = 1$ and $\Delta = 0.8$, when comparing the resilience of designs with RDA, PA and random rewiring to RD threats (i.e., comparing fig. 65 to fig. 59). However, for random threats, changing Δ from 0.8 to 1 collapses differences in R_{total} between adaptation methods. Analysis of interaction effects also shows that benefits of RDA relative to PA and random rewiring are much smaller for scenarios with $\Delta = 1$ than $\Delta = 0.8$ (see fig. 66). For random threats, changing from RDA to PA or random rewiring no longer shows a large gain in mean resilience. However, the susceptibility of designs with RDA to targeted attacks still exists, since changing from PA to RDA shows a large decrease in mean resilience for designs facing RD threats.

Discussion of Results

Experiment 4.2 explores the resilience gained by incorporating network adaptation (through RDA, PA, or random rewiring) to SoS network designs. Results show that



(a) (b)

Figure 65: Mean R_{total} for all designs considered in EXP 4 with $\Delta = 1$. 90% confidence intervals for mean R_{total} are shown by red intervals on each bar.

on average, all three adaptation methods considered improve the resilience of network designs without adaptation. However, probabilistic analysis shows that for networks facing random threats, there is a potential for individual instances, or replications, of networks without adaptation to actually show higher resilience than those with adaptation. These results suggest that cost-benefit analysis for adding network adaptation is especially needed for networks facing random threats, particularly those with very few anticipated threat events. Simply analyzing mean results, rather than probabilistic results, may not have enabled this insight.

Analysis of main and interaction effects shows that threat type has the largest effect on mean R_{total}, and strong interactions exist between adaptation method and threat type. These effects further demonstrate the need to consider threat type when designing for resilience, as the best adaptation method for a network design strongly depends on the anticipated threat. For the considered network designs, adaptation using RDA provides the most resilience to random threats, while adaptation using random rewiring provides the most resilience to RD threats. RDA performs well against random threats because network hubs formed by this adaptation method decrease network path lengths, improving the ability of a network to quickly share

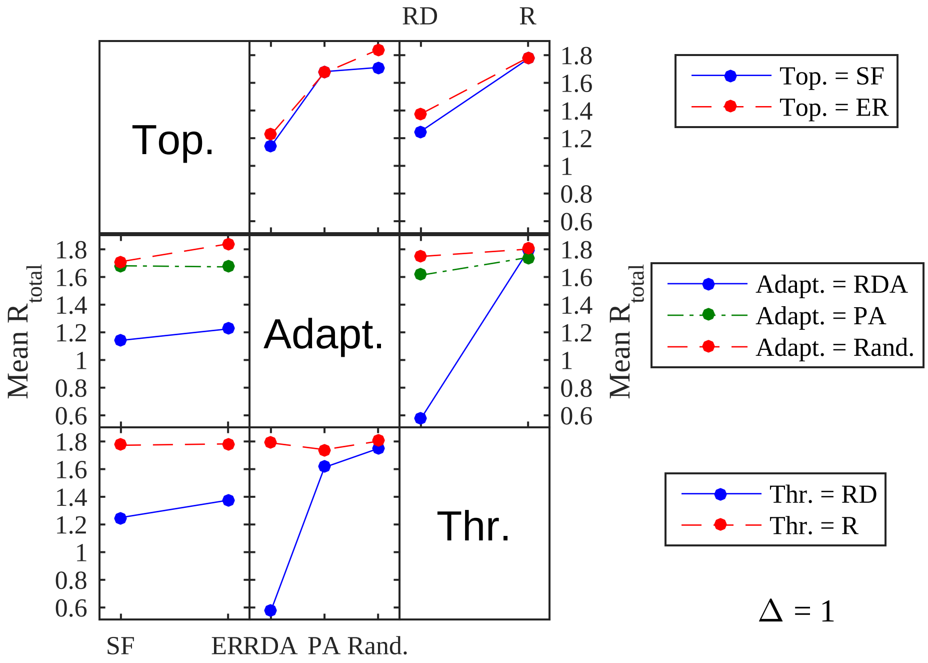


Figure 66: Interaction effects of network topology, adaptation method, and threat type on R_{total} for scenarios with $\Delta = 1$.

information. Random rewiring performs well against RD threats because it prevents the formation of network hubs, limiting the availability of highly connected nodes for targeted attacks to remove. Adaptation using PA shows similar resilience to adaptation using random rewiring, because node-degree preferences implemented in the PA model are not strong enough to provide significant structural differences to networks randomly rewiring links.

In comparison to the large effects and interactions seen between threat type and adaptation method, network topology shows a small main effect on R_{total} and small interactions with adaptation method and threat. In fact, interaction effects suggest that random topologies provide more resilience than scale-free topologies when adaptation is included, regardless of threat type and adaptation method. This is an unexpected result, since for designs without adaptation, scale-free topologies are more resilient than random topologies to random threats (see section 7.1.1). One would expect scale-free topologies to provide the same benefits when adaptation is added, and therefore expect network designs with scale-free topologies and RDA-based adaptation to be the most resilient to random threats. However, such designs are likely generating hubs that are too highly connected, such that even though random threats very rarely remove those hubs, when they do, the impact on network connectivity is large enough to outweigh benefits of short path lengths. In comparison, adding RDA to designs with random initial topologies provides a better balance between the generation of network hubs and the prevention of too much connectivity in those hubs. Additionally, this analysis focuses on resilience relative to initial capabilities, thereby limiting the benefits of scale-free initial topologies. While random topologies provide more resilience than scale-free topologies, they do not necessarily provide better overall performance, as shown by performance adjusted R_{total} results.

Network adaptation methods with density constraints are also considered. Enforcing density constraints on the number of links allowed to be rewired would provide more cost-effective network designs, assuming that there are costs incurred by an SoS network each time a link is rewired. However, results for these methods show small benefits to resilience and very little differences between designs with RDA, PA, and random rewiring. These small differences are due to the small number of links being rewired at each adaptation event when density is constrained. Therefore, adaptation without density constraints is used for the remainder of this thesis, to provide more insights into the benefits of network adaptation.

Results from scenarios with $\Delta = 1$ show that removing time sensitivity reduces benefits of RDA-based adaptation relative to PA and random rewiring. However, the disadvantages of RDA facing RD threats still exist, therefore suggesting that RDA-based adaptation should not be considered when no time sensitivity exists.

The following answer is given to research question 1.2, based on results from this experiment:

Response to RQ 1.2: The resilience gained by adding network adaptation strongly depends on threat type, due to interactions between threat type and adaptation method. For random threats, adaptation using RDA provides more resilience than PA or random rewiring. For RD threats, adaptation using random rewiring provides more resilience than RDA. Random network topologies provide more resilience (but not necessarily performance) than scale-free topologies for all scenarios considered, as long as adaptation is incorporated.

7.2 EXP 5: Optimizing the IE Network Design Space

Purpose of the experiment: Answer research question 1.3 by exploring the continuous IE network design space and optimizing network designs as threat type is varied.

Experiment five (EXP 5) optimizes a continuous IE network design space generated using methods from section 6.2. This experiment further investigates the overarching research question for this thesis (RQ 1), by extending the exploration performed in experiment four and developing an understanding of how the optimal network design changes as threat type changes. The primary research question for this experiment is derived from research question one as follows:

> **Research Question 1.3:** How does the optimally resilient IE network design (i.e., combination of topology and adaptation method) change as the threat type changes?

Experimental Setup

Experiment five uses the IE network model as an application problem with the same model input parameters from experiment four (table 9), except with $\Delta = 0.6, 0.7, 0.8, 0.9, 1$. This experiment varies Δ to further investigate the relationship between message time sensitivity and network resilience. The resilience assessment framework is applied as described for experiment four, shown in fig. 48.

An RSM process is used to optimize network designs, as described in section 6.1. This process consists of identifying experimental design points to simulate, fitting a linear regression model to the simulated response data, and using that regression model to optimize for R_{total} as the threat is varied. Results from linear regression-based optimization are compared to full-factorial simulation data, to characterize the effects of regression model error on observed trends.

The network design space is represented by three variables: the initial network topology, adaptation method, and threat. These variables are each defined by an α

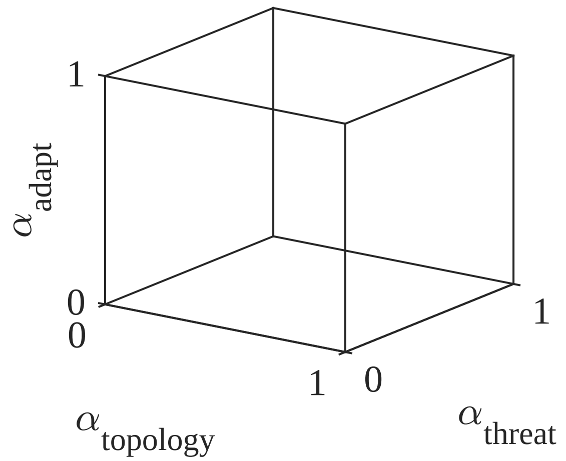


Figure 67: View of the continuous IE network design space, which includes all designs in the interior of the space. This representation includes threat type in the design space, with the understanding that threats are not typically within the control of the designer.

parameter, corresponding to the α parameter used by the GM model to interpolate between topologies, adaptation methods, and threats. Therefore, the design space is defined by $\alpha_{topology}$ (defining the initial topology), α_{adapt} (defining the adaptation method), and α_{threat} (defining the threat). These variables are defined with limits of $0 \leq \alpha_{topology} \leq 1$, $0 \leq \alpha_{adapt} \leq 1$, and $0 \leq \alpha_{threat} \leq 1$, as shown in fig. 67. Designs at the corners of this space can be described by topologies, adaptation methods, and threats from the categorical design space used in experiment four. Thus, a network with $\alpha_{topology} = 0$ has a scale-free topology, while one with $\alpha_{topology} = 1$ has a random topology. A network with $\alpha_{adapt} = 0$ uses RDA-based adaptation, while one with $\alpha_{adapt} = 1$ uses random rewiring. A network with $\alpha_{threat} = 0$ is subjected to RD threats, while one with $\alpha_{threat} = 1$ is subjected to random threats.

Experiment design methods are used to determine which design points to simulate from the space shown in fig. 67. A hybrid combination of experimental designs is used to fully capture the design space. A face-centered central composite design (CCD) is used to account for potential nonlinearities in the data and curvature with respect to the response, R_{total}. Face-centered CCDs fall within the class of second-order

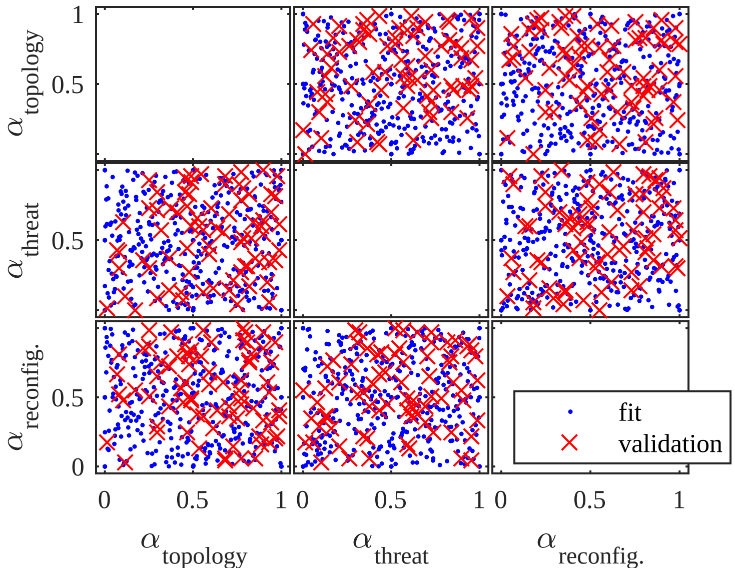


Figure 68: Scatterplot matrix of the experimental design used for experiment five. Fit points are used for the regression model. Random validation points are used for model fit checks.

experimental designs that enable fits to second-order regression models. These designs are often used for cuboidal design spaces where design points at the extremes of the space are of interest, since they include corners of the design space. A Box-Behnken design (BBD) is superimposed on top of a face-centered CCD to include sampling along the edges of the design space, not included by CCDs. Space-filling designs are often used for computer experiments because they nearly uniformly sample the interior of the design space, which is useful when the general form of the regression model is unknown. Since limited knowledge exists concerning the interior of the defined network design space, a latin hypercube design is also used to sample the interior of the space. A description of these experimental designs is given in [78].

Since this experiment contains three factors, or design variables, the CCD and BBD each include 15 design points. 350 latin hypercube points are run, resulting in a total of 380 experimental design points. An additional 76 random points in the space are run for model validation data (20 percent of 380 fit points). The experimental design is shown in fig. 68.

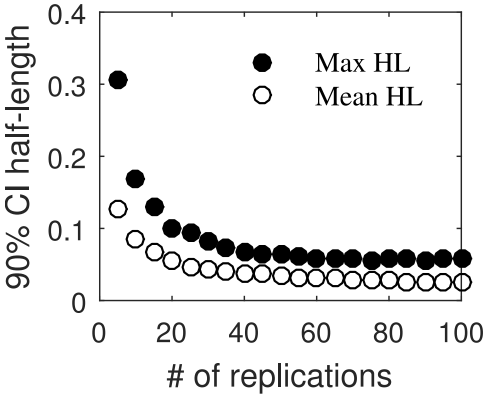


Figure 69: Changes to the maximum (Max HL) and mean (Mean HL) 90% t confidence interval half-lengths as the number of replications is varied for all designs points from experiment four. Maximum half-lengths are calculated as the maximum half-length of 90% confidence intervals among all designs points, where confidence intervals are calculated for the mean R_{total} at each design point. Mean half-lengths are similarly calculated as the mean half-length over all design points.

50 replications are run at each design point, based on analysis of 90 percent t confidence intervals for mean R_{total} data from experiment four. Figure 69 shows that the maximum 90 percent confidence interval half-length for mean R_{total} data from experiment four minimally decreases once 50 replications is used. The maximum half-length with 50 replication is 0.06 for that data. This half-length is 1.5 percent of the maximum R_{total} seen in that data. Since the design space used for this experiment is similar to the space from experiment four (just with the interior of the space considered), 50 replications is determined to be sufficient for this experiment.

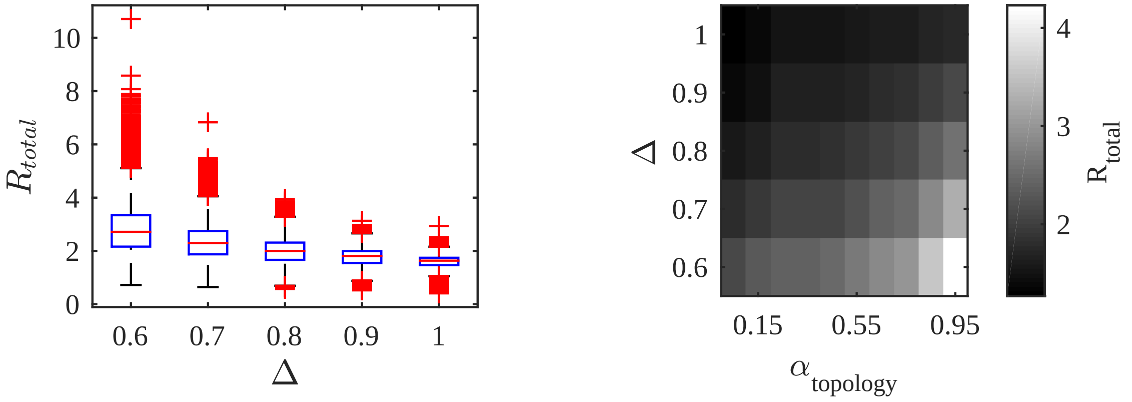
Stepwise linear regression is used to fit a surrogate model to the data from simulated experimental design points. Stepwise regression begins with a starting model (e.g., a second-order model with all interaction terms) and adds or subtracts terms one step at a time based on a defined criterion. MATLAB's stepwiselm function is used, with default model specifications (i.e., SSE criterion with a term enter threshold of 0.05 and exit threshold of 0.1 for the p-value of the term F-statistic). Stepwise regression is used to limit the inclusion of unnecessary model terms while allowing for the inclusion of important terms that may not be intuitively known to the analyst.

The factors for the regression model are $\alpha_{topology}$, α_{adapt}, and α_{threat}. Time sensitivity, Δ, is also considered as a factor for initial model fits; however, due to the large impact Δ has on the response, final model fits are performed separately, one for each Δ setting. The response for final regression models is the mean R_{total} at each design point, averaged over all simulated replications.

MATLAB's fmincon function is used to optimize regression models using the sequential quadratic programming (SQP) algorithm with default specifications. SQP outperforms the default interior point method for most models used in this thesis. Since regression models are polynomial functions, additional computational costs of SQP relative to an interior point method are insignificant.

Experiment Results (Trends in the Data)

Figure 70 shows general trends in the data for R_{total} with respect to Δ and $\alpha_{topology}$. This analysis is performed on all experimental data including replications. Δ and $\alpha_{topology}$ are viewed as scenario variables, since they define the scenario or task environment of a network and are likely out of the control of a network designer. The box plot shows that increasing Δ, or decreasing message time sensitivity, collapses differences between designs with respect to R_{total} (i.e., decreases the total range and range of the inner quartiles) and decreases median R_{total}. Increasing Δ decreases R_{total} ranges because when message time sensitivity is low, a network's performance does not depend on how long it takes messages to arrive at their target node. Therefore, the benefits of networks that promote short path lengths are reduced and most networks tend to show similar levels of resilience. Reducing the benefits of short path length networks limits the ability of networks to achieve high resilience, which reduces the median R_{total}. A similar result is seen in experiment four, where changing from $\Delta = 0.8$ to $\Delta = 1$ collapses differences between adaptation methods, particularly reducing the benefits of RDA-based adaptation (see fig. 65).



(a)

(b)

Figure 70: Box plot (a) and heat map (b) showing trends in the data as Δ and $\alpha_{topology}$ are varied. The box plot shows median (central red lines), 25% and 75% quartiles (blue box edges), minimum and maximum data points without outliers (black whiskers), and outliers (red symbols). The heat map shows the mean R_{total} within each bin of Δ and $\alpha_{topology}$.

The heat map shows that mean resilience is highest for scenarios with low Δ and high $\alpha_{topology}$. These scenarios have high time sensitivity and highly random threats. Networks facing random threats show more resilience than those facing more targeted threats because random threats are less likely to remove highly connected network nodes. This result supports those from experiment four, which show that RD threats are more damaging to network resilience than random threats, even with network adaptation included.

Figure 71 shows a parallel plot of data from this experiment with respect to network designs grouped by high, medium, and low resilience. Parallel plots are useful for identifying general trends in multi-variate data, since they show average variable settings for specified groups. These results show that designs with high resilience typically have topologies closer to random than scale-free, face threats that are more random than targeted, and use adaptation that is more degree-based than random. Highly resilient designs also show small variance with respect to the type of threat faced; most designs with high resilience face highly random threats. Designs

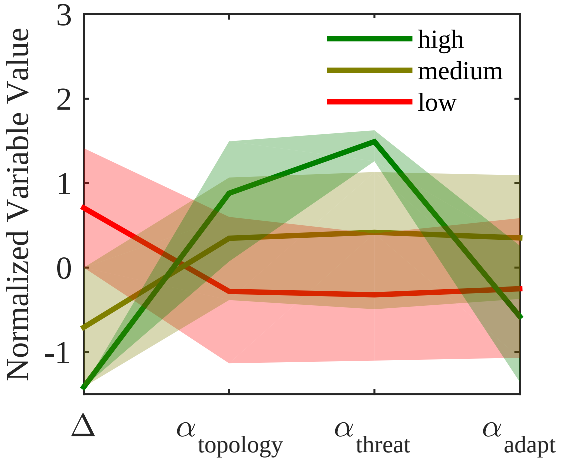


Figure 71: Parallel plot showing normalized values for each design variable, when grouped by high, medium, and low R_{total}. Solid lines show median normalized values with shaded lines capturing 25% and 75% quartiles. Statistics are calculated from all data used in this experiment, including replications. High resilience is defined as $R_{total} > 4$, medium resilience as $2 < R_{total} \leq 4$, and low resilience as $R_{total} \leq 2$. Variables values are normalized to have zero mean and unit standard deviation.

with low resilience have variable settings closer to the middle of the design space (i.e., a normalized value ≈ 0 or $\alpha \approx 0.5$).

Since threat type is identified to be important for network resilience, fig. 72 shows a scatterplot of mean R_{total} as threat type is varied, colored by initial topology. For low values of Δ, or high time sensitivity scenarios, increasing α_{threat} generally increases resilience. However, as Δ is increased, differences between the resilience of network designs collapses to the point where very few networks show resilience above two. These plots also shows that for low values of Δ, increasing $\alpha_{topology}$ increases resilience (seen by stacked color bands of data for $\Delta = 0.6, 0.7$). These results confirm the general trend that randomness in network topology and threat type increases resilience, while low time sensitivity collapses differences between network alternatives.

Figure 73a focuses on data for $\Delta = 0.6$ and shows a scatterplot of mean R_{total} as threat type is varied, this time colored by adaptation method. As shown by interaction effects analysis in experiment four (see fig. 61), the effect of adaptation

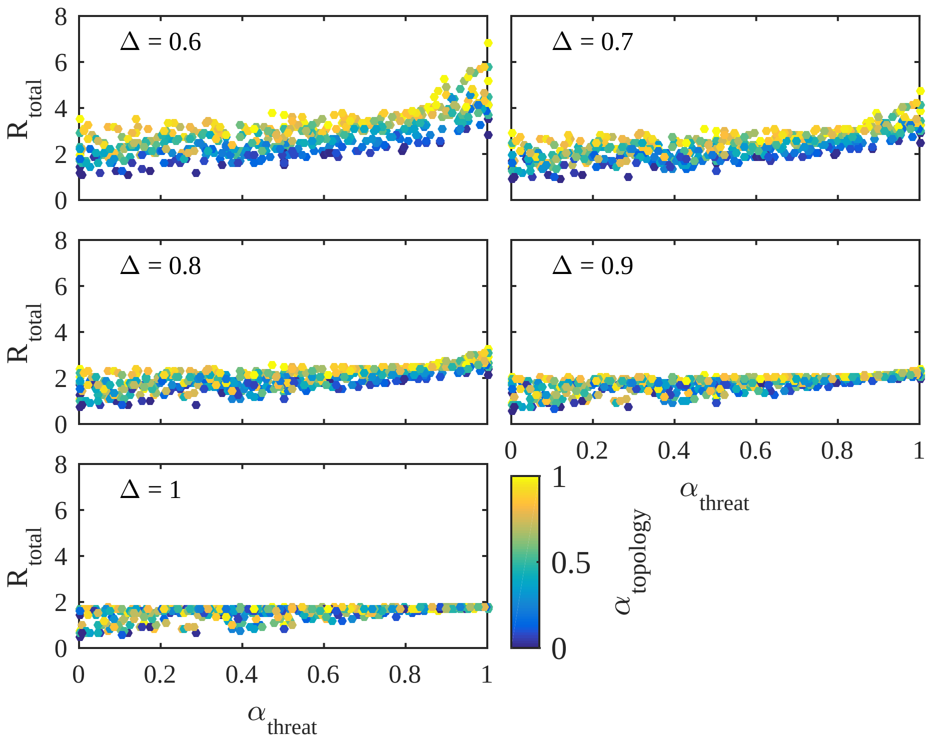


Figure 72: Scatterplots of mean R_{total} as threat type (α_{threat}) changes. Points are colored by initial topology ($\alpha_{topology}$). Each scatterplot shows results calculated for a different time sensitivity (Δ) and includes data from every replication.

method on R_{total} depends on the threat type. For low α_{threat} values, increasing α_{adapt} (i.e., increasing adaptation randomness) increases R_{total}. However, the opposite trend is seen for high α_{threat} values, where decreasing α_{adapt} increases R_{total}. These results also show that networks with degree-based adaptation (i.e., low α_{adapt}) show a wide range in resilience as the threat is varied. In comparison, networks with highly random adaptation (i.e., high α_{adapt}) show less variation in resilience as the threat is varied, suggesting that random adaptation is more robust to threat type than degree-based adaptation.

Figure 73b shows that resilience trends with α_{adapt} can be explained by trends with total network inverse average path length. Total inverse average path length, $\langle d \rangle'_{total}$, differs from the typical calculated inverse average path length, $\langle d \rangle'$, in that the total quantity is calculated for an entire scenario, which include multiple changes to the

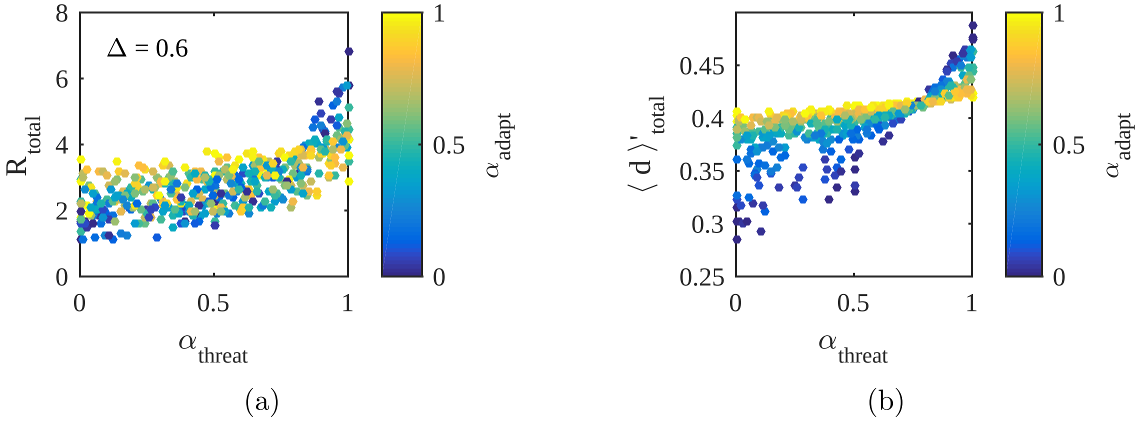


Figure 73: Scatterplots of (a) mean R_{total} and (b) total inverse average path length as threat type (α_{threat}) changes. Points are colored by adaptation method (α_{adapt}). R_{total} is calculated with $\Delta = 0.6$.

network structure over time for this data. In comparison, the typical inverse average path length quantity is calculated for a single network structure. The total quantity is used because each simulated scenario contains nine threat and nine adaptation events, resulting in 19 values of $\langle d \rangle'$ (one for the initial topology, and one following each threat or adaptation event). These 19 values are collapsed into a single value for the entire scenario using an exponentially weighted mean, such that

$$\langle d \rangle'_{total} = \sum_{i=1}^{N_{networks}} w_i \langle d \rangle'_i, \tag{68}$$

where $N_{networks}$ is the total number of network structures seen in the scenario, the weights w_i are calculated using eq. (48), and $\langle d \rangle'_i$ is the inverse average path length for the ith network structure seen, calculated with eq. (21). This calculation is similar to that of R_{total} from a set of R values (see section 4.2).

Trends for total inverse average path length suggest that for highly random threats, networks with low α_{adapt} have high resilience because they achieve high $\langle d \rangle'_{total}$, resulting in short message travel times. These networks have high $\langle d \rangle'_{total}$ because the degree-based adaptation creates network hubs. However, when threats become more targeted, these hubs make node removals more damaging to the network, resulting in lower $\langle d\rangle'_{total}$ and longer message travel times. Networks with random adaptation are more robust to threat type variations than those with degree-based adaptation because their path lengths show less sensitivity to threat type. Though $\langle d\rangle'_{total}$ can be used to explain trends with R_{total}, they are not directly correlated, since trends with respect to threat type are less clear in R_{total} data than $\langle d\rangle'_{total}$ data.

Experiment Results (Regression Models)

Linear regression models are created from the experimental data to enable further analysis and optimization of the network design space. Regression models are generated to predict the response, R_{total}, from the set of factors used to define a network design and scenario (i.e., Δ, $\alpha_{topology}$, α_{adapt}, and α_{threat}), such that

$$y = f\left(\Delta, \alpha_{topology}, \alpha_{adapt}, \alpha_{threat}\right) + \varepsilon. \tag{69}$$

A regression model is first fit to the entire set of experimental data, including individual replications. However, the large variance seen in simulation replications results in poor model fits using replicated data (artificial neural network fits are also attempted, which do not improve model fits). An attempt is then made to fit a regression model to mean data from the entire set of experimental data. Fitting to mean data uses the mean R_{total} at each design point (averaged over all 50 replications), and uses that data to fit a model to. Model fits are improved by fitting to mean data, compared to replicated data. However, regression models still show large residuals and model errors.

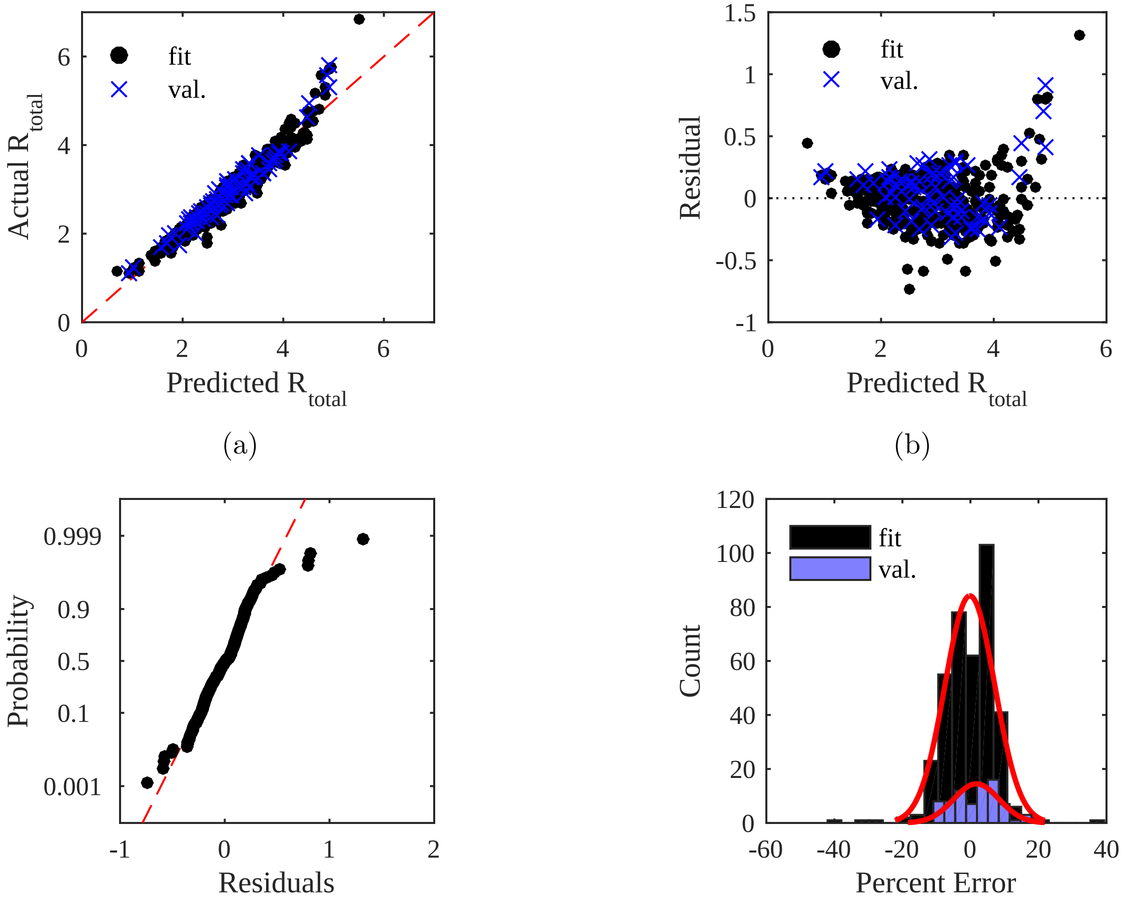
Since Δ has a large impact on network resilience, separate model fits are performed for each Δ setting. This process splits the experimental data into five datasets, one for each Δ setting, and creates a separate regression model for each of those datasets. Thus, five regression models are created where each model, f_i, is fit to mean data for a specific Δ (e.g., $f_{0.6}$ is the regression model for data with $\Delta = 0.6$) such that

$$y = f_i\left(\alpha_{topology}, \alpha_{adapt}, \alpha_{threat}\right) + \varepsilon. \tag{70}$$

Figure 74 shows model fit checks for a regression model of the dataset with $\Delta = 0.6$, generated using stepwise regression with a full second-order initial model (i.e., a second-order model with individual factors and all interactions). These model checks show that the generated regression model does not accurately represent the data, since actual by predicted results show large deviations from the 1:1 reference line, residuals do not appear to be randomly scattered (showing slight curvature in the positive range of residuals), and model errors are as large as 39.3%. Residuals are nearly normally distributed, but do show more high valued residuals than expected from normally distributed residuals. Similar fits are seen for second-order models of other Δ datasets.

Higher order terms are added to the initial regression model in an attempt create less patterned residuals and decrease model errors. Fifth-order models are found to provide the best model fits across all Δ datasets. Figure 75 shows model fit results for a fifth-order regression model with the $\Delta = 0.6$ dataset. This model shows an improved fit over the second-order model. Actual by predicted data is tightly bound to the 1:1 perfect fit line, indicating that the model accurately predicts fit and validation data. Compared to residuals from the second-order model, residual magnitudes are decreased, show a more random scattering, and are more normally distributed. Model fit errors are also decreased, with a maximum error of 6.5%.

Fifth-order model fits for other Δ datasets are similar to those for $\Delta = 0.6$ and shown in appendix B. Model fit statistics for all fifth-order models are shown in table 11, where f_i specifies the model for $\Delta = i$. ANOVA statistics are shown for the typical null hypothesis of a constant model form. High F-statistics, low p-values, and high R^2 and R^2_{adj} values provide statistical support for use of the generated models.



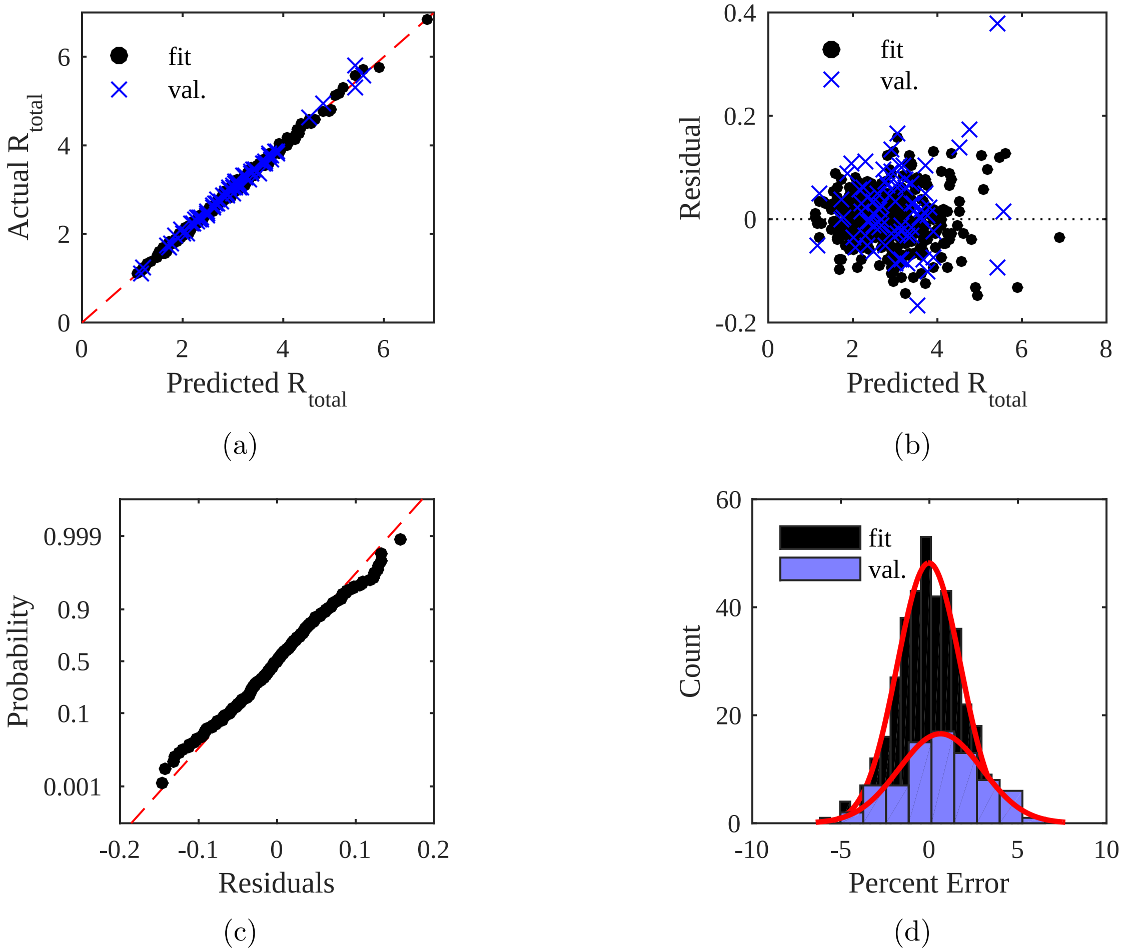
(a)

(b)

(c)

(d)

Figure 74: Model fit checks for a second-order regression model of data with $\Delta = 0.6$, grouped by calculations using fit data (fit) and validation data (val.). Actual by predicted responses are shown in (a), with the red dashed line showing a 1:1 perfect fit reference line. Residuals are shown in (b), a normal probability plot for fit data in (c), and model error distributions in (d). Normally distributed data should lie on or near the red dashed reference line in the normal probability plot shown in (c). Standard deviations for model error distributions in (d) are $\sigma_{fit} = 7.3$ and $\sigma_{val.} = 6.7$. Maximum model error is 39.3%.



(a)

(b)

(c)

(d)

Figure 75: Model fit checks for a fifth-order regression model of data with $\Delta = 0.6$. Actual by predicted responses are shown in (a), residuals are shown in (b), a normal probability plot for fit data in (c), and model error distributions in (d). Standard deviations for model error distributions are $\sigma_{fit} = 1.8$ and $\sigma_{val.} = 2.3$. Maximum model error is 6.5%.

Model fits are also attempted on natural log and exponential transformed responses to further reduce model errors. Models fit to transformed responses apply a natural log or exponential transformation to the actual response data such that the regression model is fit to $\ln(y)$ or $\exp(y)$, rather than the actual response y. The model then outputs a prediction for the transformed response. No significant improvements are seen to model fits with natural log or exponentially transformed responses. Some model fit checks show worse performance with transformed response data. Therefore, transformations are not included in generated regression models.

Table 11: Linear regression model fit statistics

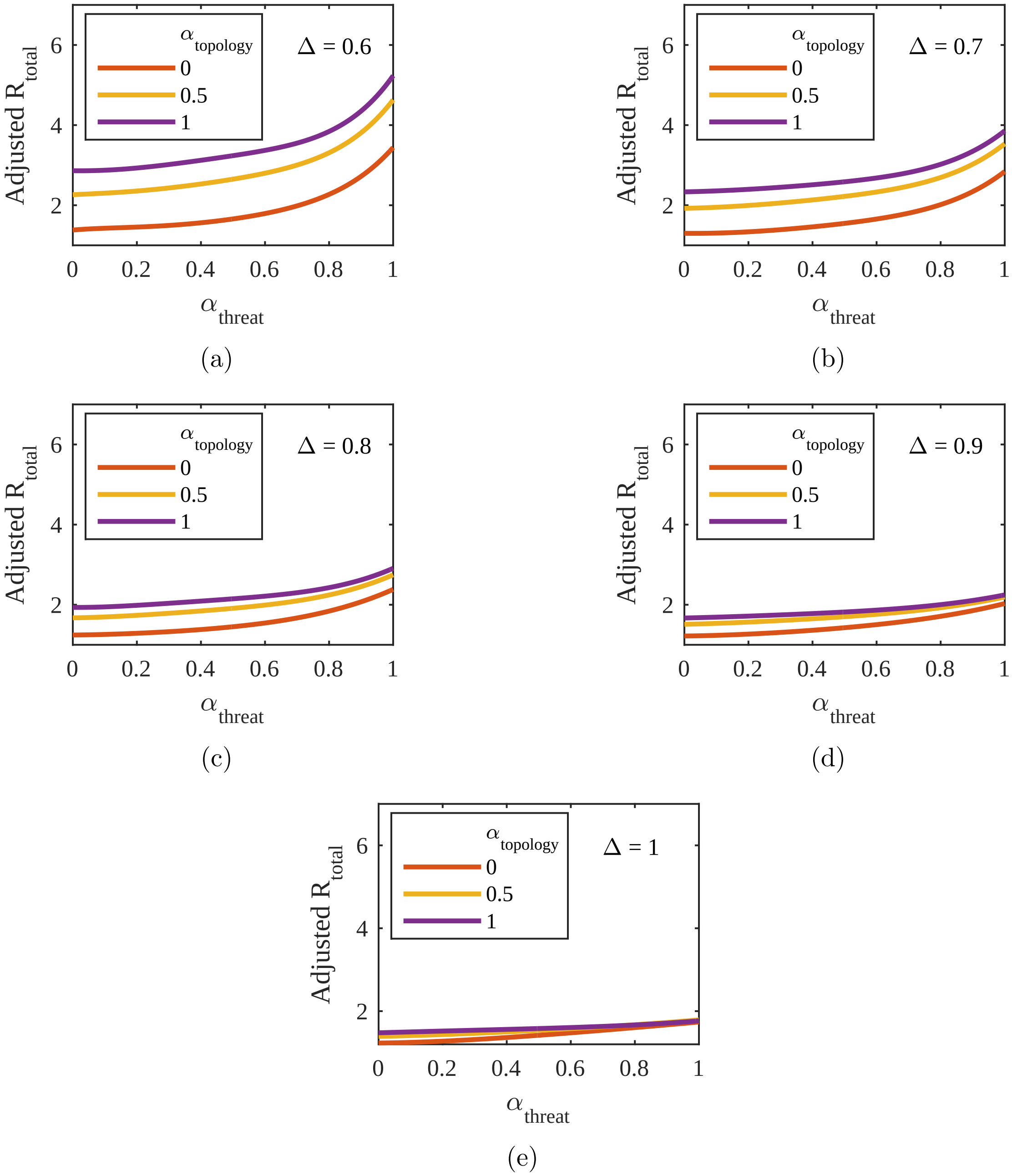
Regression model	ANOVA F-statistic	ANOVA p-value	R^2	R^2_{adj}
$f_{0.6}$	2.25×10^3	0	0.997	0.996
$f_{0.7}$	2.64×10^3	0	0.997	0.996
$f_{0.8}$	2.56×10^3	0	0.997	0.996
$f_{0.9}$	2.84×10^3	0	0.997	0.997
f_1	1.95×10^3	0	0.995	0.995

Fifth-order linear regression models with natural (i.e., untransformed) responses and factors are selected to further analyze and optimize network designs. Model errors are determined to be sufficiently low for this thesis, since this thesis focuses on the conceptual design of SoS networks. This experiment also focuses on understanding general trends for optimal network designs, with respect to abstracted, high-level information exchange processes, further reducing the need for high fidelity models.

Experiment Results (Optimal Network Designs)

The generated linear regression models are used to determine interaction effects among initial network topology, adaptation method, and threat type. Figure 76 shows the interaction between threat type and topology, and confirms previous findings that on average, the most resilience initial topology is a random one (i.e., $\alpha_{topology} = 1$), for networks with some form of adaptation, regardless of threat type. Since intermediate topologies are included in this analysis, results also show a smooth improvement in resilience as topology randomness is increased. Decreasing time sensitivity (i.e., increasing Δ) is shown to collapse differences between network designs, as seen in the analysis of general trends in the data from this experiment.

Figure 77 shows a strong interaction between threat type and adaptation method, as seen in analysis of the categorical design space in experiment four (see fig. 61). However, regression models enable a more detailed inspection of this interaction by providing data for intermediate networks in the design space. For example, experiment four shows that random rewiring provides more resilience to RD threats than RDA-based adaptation, while RDA provides more resilience to random threats. However, those results provide no insights into the possibility or location of a transition point at which RDA becomes a better option than random rewiring, as threat type is varied. Analysis of interactions using regression models suggests that such a transition point does exist, where the location of that point changes as time sensitivity changes. Interactions also show that on average, intermediate adaptation methods (i.e., those with $0 < \alpha_{adapt} < 1$) always perform worse than one of the extreme adaptation methods.



(a)

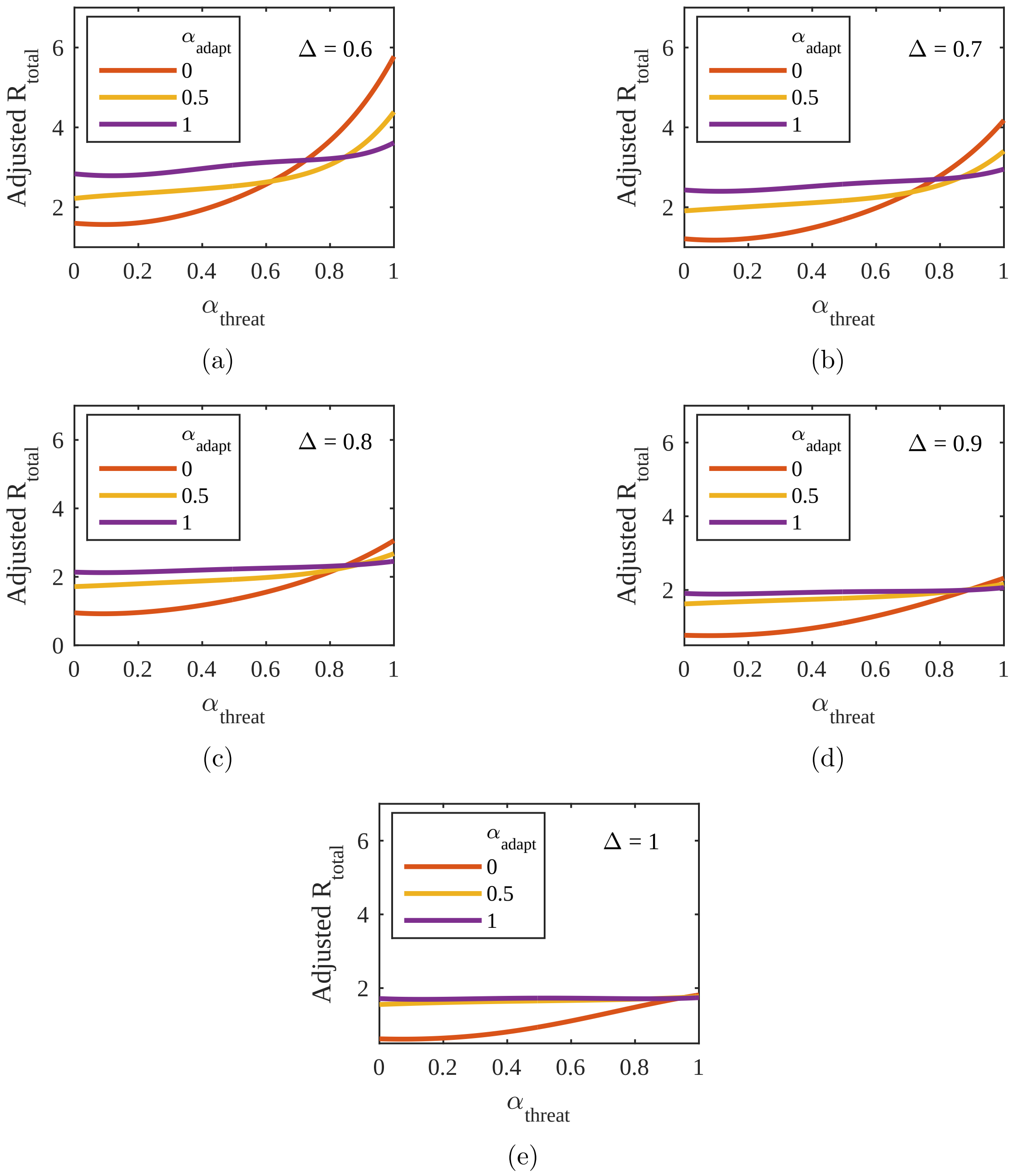
(b)

(c)

(d)

(e)

Figure 76: Interaction effects between α_{threat} and $\alpha_{topology}$. Interaction effects are shown as the adjusted R_{total} for specific topology settings as threat type is varied, averaged over all adaptation methods. Adjusted R_{total} is calculated as the sum of the residual and predicted R_{total} for each model fit data point.



(a) (b) (c) (d) (e)

Figure 77: Interaction effects between α_{threat} and α_{adapt}. Interaction effects are shown as the adjusted R_{total} for specific adaptation methods as threat type is varied, averaged over all initial topologies. Results suggest a transition point at which highly degree-based adaptation provides more resilience than highly random adaptation. This transition point is identified by the α_{threat} value at which the $\alpha_{adapt} = 0$ and $\alpha_{adapt} = 1$ lines intersect.

Analysis of interaction effects provides an understanding of general trends throughout the design space. However, conclusions drawn from this analysis are limited because interactions effects are averaged over all settings for the network variable not included in a particular interaction. Regression models are optimized with respect to resilience for various threat types to more deeply investigate network resilience, especially regarding potential transition points for the optimal adaptation method.

Since threat type shows a strong impact on network resilience, optimal network designs are determined with respect to R_{total} as threat type is changed. The optimizer is allowed to change the network design, specified by the initial topology, $\alpha_{topology}$, and adaptation method, α_{adapt}. Time sensitivity and threat type are assumed to be outside the control of the designer, and are therefore not changed by the optimizer. The optimization problem is formulated as follows:

$$\begin{aligned} \underset{\alpha_{topology},\alpha_{adapt}}{\text{maximize}} \quad & f_i(\alpha_{topology}, \alpha_{adapt}, \alpha^j_{threat}) \\ \text{subject to} \quad & 0 \leq \alpha_{topology} \leq 1 \\ & 0 \leq \alpha_{adapt} \leq 1, \end{aligned} \tag{71}$$

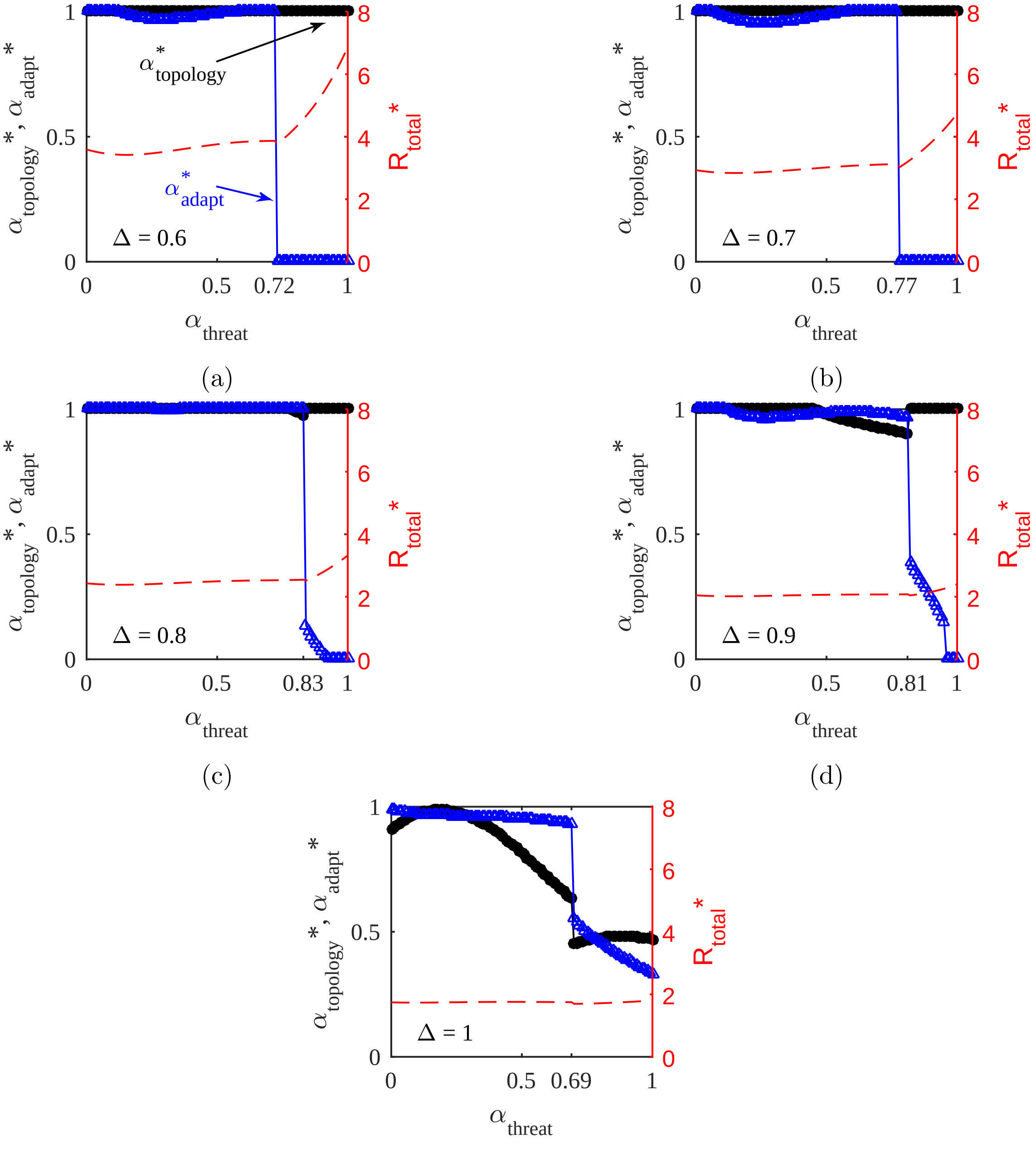
where f_i is the regression model for $\Delta = i$ and α^j_{threat} is the jth threat setting the optimal network design is being determined for.

Optimization results are shown in fig. 78. Focusing on the scenario with $\Delta = 0.6$ (i.e., high time sensitivity), the optimally resilient network topology is always fully random. This result confirms observations from experiment four and general trends in the data for this experiment that random topologies are more resilient than scale-free when adaptation is included. However, the optimally resilient adaptation method shows a sharp transition from being fully random to fully degree-based at $\alpha_{threat} = 0.72$. Above this transition point (i.e., for $\alpha_{adapt} > 0.72$), networks are able to gain resilience by switching from random rewiring to a degree-based adaptation method. As discussed in experiment four, random rewiring is preferred for targeted threats because of a lack of network hubs; degree-based adaptation is preferred for random threats because of the low probability of highly connected hubs being removed.

A small dip in the optimal adaptation method is seen around $\alpha_{threat} = 0.28$, where $\alpha^*_{adapt} = 0.97$ instead of 1. This dip is likely due to small errors in the regression model, amplified by small differences in R_{total} for networks in this part of the design space. For example, R_{total} for the optimal network at $\alpha_{threat} = 0.28$ (i.e., $\alpha^*_{topology} = 1$ and $\alpha^*_{adapt} = 0.97$) is 3.50. If α_{adapt} is changed to 1 instead of 0.97, $R_{total} = 3.49$, which is a 0.3% change in R_{total}. Since this difference is so small, model errors that may be causing this dip in the optimal adaptation method are assumed to be insignificant to this analysis.

Increasing Δ shifts the location of the optimal adaptation transition point, up until $\Delta = 0.9$, at which point fully degree-based adaptation methods (i.e., $\alpha_{adapt} = 0$) are only optimal for fully random threats. For $\Delta = 1$, the optimal adaptation method never goes below 0.33. Therefore, as time sensitivity decreases, the benefits of degree-based adaptation methods decrease, and increasing levels of randomness are preferred for network resilience.

Optimal network designs for $\Delta = 1$ show more disjointed trends than those for $\Delta < 1$. As with the dip seen in results for $\Delta = 0.6$, these behaviors are caused by small differences in the resilience of network designs. Since many networks are able to provide nearly optimal resilience, transition points and optimal trends become muddled for $\Delta = 1$.



(a)

(b)

(c)

(d)

(e)

Figure 78: Optimally resilient network designs as threat type, α_{threat}, and time sensitivity, Δ, are varied. Network designs are specified by their initial topology and adaptation method. Optimal combinations of those variables are shown on the left axis. Optimal topology settings, $\alpha^*_{topology}$, are shown by black circles. Optimal adaptation methods, α^*_{adapt}, are shown by blue triangles. Resilience, R_{total}, of optimal designs is shown by the dashed red line, corresponding to the right axis.

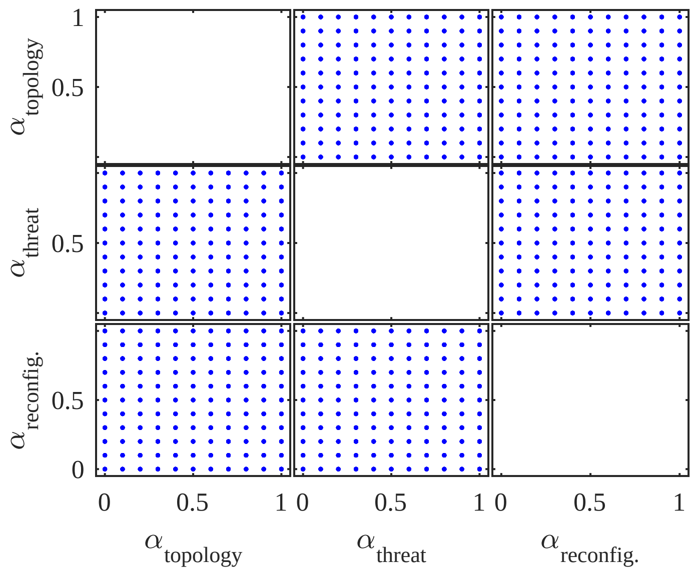


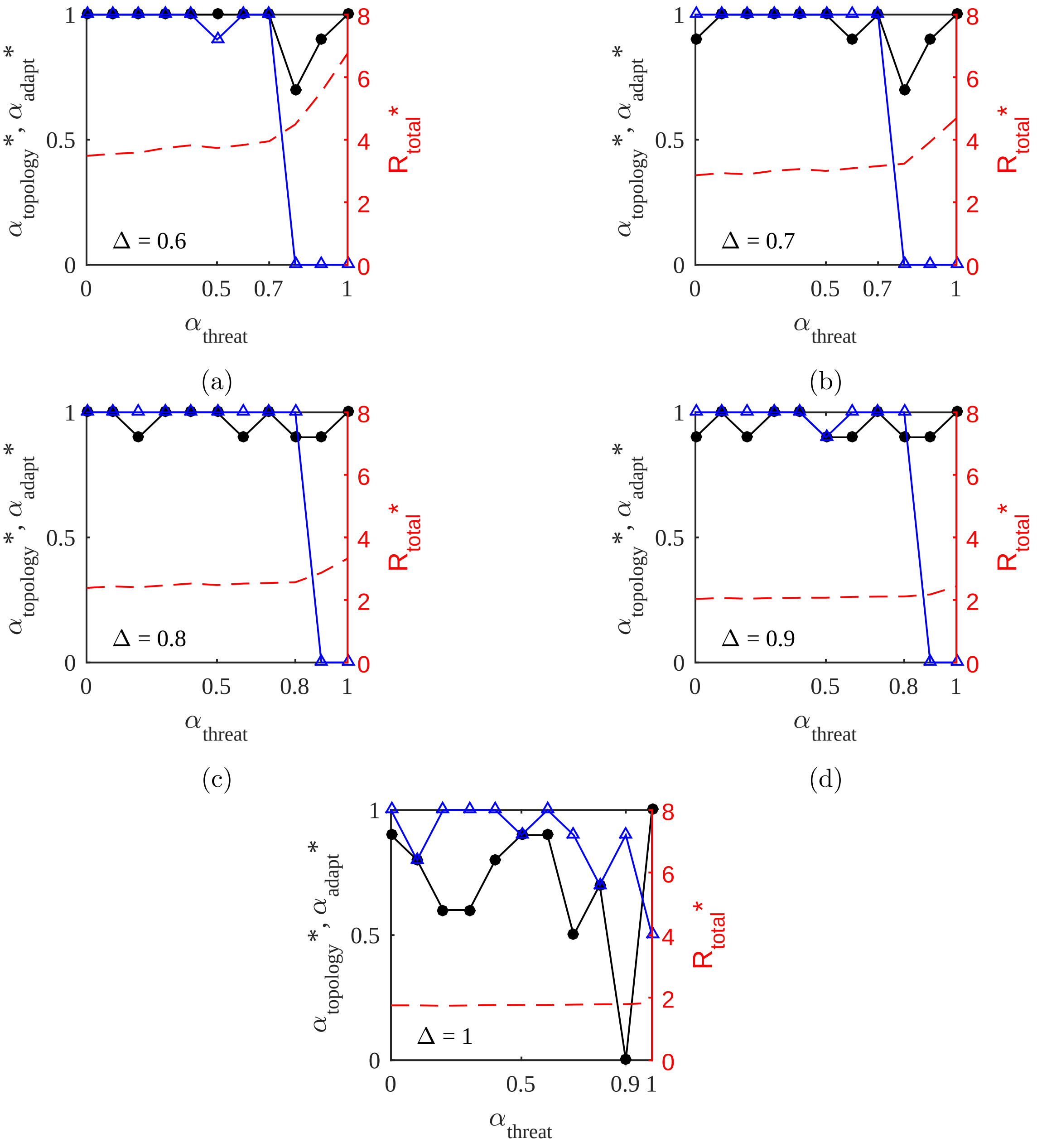
Figure 79: Scatterplot matrix of the full-factorial experimental design used to compare to regression model optimization results.

A full-factorial experimental design is run to ensure that optimal network trends determined from regression models are accurate. Figure 79 shows the full-factorial design used, where α settings are defined to be $0, 0.1, 0.2, \ldots, 1$. This full-factorial design contains 1331 design points, over three times as many points as used for regression model fit data. 50 replications are run at each design point.

Optimally resilient network designs are determined from the full-factorial experiment by selecting the design point that maximizes R_{total} at each level of α_{threat}. Optimal networks are therefore limited to those included in the full-factorial experiment design, resulting in a low resolution determination of optimal networks. However, these results can still be used to provide a general idea of optimization trends in the network design space, without model errors incurred by using regression models.

Figure 80 shows that optimization trends from the full-factorial experiment are similar to those from the regression models, where the optimal adaptation method shows a sharp transition point whose location changes with Δ. These similarities provide confidence that regression model errors do not influence observed optimization trends. Differences are seen between full-factorial results and regression models with

$\Delta = 1$; however, these differences are again attributed to the fact that many network designs show similar levels of resilience for $\Delta = 1$, and therefore determined to be insignificant to the analysis.



(a)

(b)

(c)

(d)

(e)

Figure 80: Optimally resilient network designs as threat type, α_{threat}, and time sensitivity, Δ, are varied. Optimal combinations of initial topology (shown by black circles) and adaptation method (shown by blue triangles) are shown on the left axis. Resilience of optimal designs is shown by the dashed red line, corresponding to the right axis.

Discussion of Results

Analysis of the continuous IE network design space shows that for adaptive IE networks operating with high message time sensitivity, the optimally resilient initial network topology is always random, regardless of threat type. Since random networks are characterized by node degree homogeneity, network designers aiming for resilience should strive to limit network hubs in initial topologies, when adaptation is included and multiple network threats are anticipated. Note that this analysis strictly optimizes for resilience, relative to the initial capability of a network; therefore, while random topologies provide optimal resilience, they do not necessarily provide optimal performance.

Network optimization for resilience confirms general adaptation trends observed from experiment four, where random rewiring gives more resilience against targeted threats than degree-based adaptation, and degree-based adaptation gives more resilience against random threats than random rewiring. However, this experiment extends those observations by showing that a sharp (rather than gradual, or smooth) transition exists for the optimal adaptation method from random rewiring to degree-based adaptation, as threat randomness is increased. This transition is also seen in interaction analysis (see fig. 77), and suggests that when resilience is the primary objective, network adaptation should be designed to be either fully random or fully degree-based. Therefore, there is no need to consider intermediately defined network adaptation methods when designing for resilience, regardless of the threat faced. Previous analysis of the discrete network design space may have lead one to consider intermediately defined network designs for threats that are not fully random or fully targeted.

The observed adaptation transition occurs at $\alpha_{threat} = 0.72$ for networks with high message time sensitivity. Therefore, in addition to showing a sharp transition, the optimally resilient adaptation method is random rewiring for 72% of the threats considered. Analysis of the discrete design space does not provide such insights into the relative percentage of threats considered for which each adaptation method is preferred.

The optimal adaptation transition is most clearly seen for scenarios with high time sensitivity. As message time sensitivity decreases, the sharpness of the transition decreases, since most network show similar levels of resilience. The impact of changing network initial topologies is also decreased as time sensitivity decreases. Therefore, investing resources to design optimally resilient networks is less important for networks operating with low time sensitivity constraints.

The following answer is given to research question 1.3, based on results from this experiment:

> **Response to RQ 1.3:** The optimally resilient IE network design is one with a random topology and random rewiring, for targeted threats. However, the optimally resilient design shows a sharp transition with respect to adaptation method as threat randomness increases; for highly random threats, the optimal design uses degree-based adaptation. Differences between optimal and sub-optimal designs are minimal for scenarios with low message time sensitivity.

7.3 Summary of Results

This chapter uses the methodology developed by chapters 3 to 6 to design resilient IE networks. A categorical network design space is explored, focusing on general trends and factor interaction effects. A continuous design space is then modeled with linear regression. Regression models are used to determine optimal network designs as threat type changes. The following summarizes research questions, responses, and experiments from this chapter:

- RQ 1.1: How resilient are scale-free and random SoS networks to targeted attacks and random node failures, with no adaptation considered and resilience measured by R_{total}?

- Response to RQ 1.1 (EXP 4.1): Scale-free SoS networks are more resilient to random failures than random SoS networks but less resilient to RD targeted attacks, with no adaptation considered and resilience measured by R_{total}. These results agree with those from the complex networks literature where resilience is measured by network structural properties, suggesting the ability to use those properties as surrogates for network performance in the IE network model.

- RQ 1.2: How is the resilience gained by adding network adaptation to SoS networks affected by threat type and initial network topology?

- Response to RQ 1.2 (EXP 4.2): The resilience gained by adding network adaptation strongly depends on threat type, due to interactions between threat type and adaptation method. For random threats, adaptation using RDA provides more resilience than PA or random rewiring. For RD threats, adaptation using random rewiring provides more resilience than RDA. Random network topologies provide more resilience (but not necessarily performance) than scale-free topologies for all scenarios considered, as long as adaptation is incorporated.

- RQ 1.3: How does the optimally resilient IE network design (i.e., combination of topology and adaptation method) change as the threat type changes?

- Response to RQ 1.3 (EXP 5) The optimally resilient IE network design is one with a random topology and random rewiring, for targeted threats. However, the optimally resilient design shows a sharp transition with respect to adaptation method as threat randomness increases; for highly random threats, the optimal design uses degree-based adaptation. Differences between optimal and sub-optimal designs are minimal for scenarios with low message time sensitivity.

CHAPTER VIII

COST-BENEFIT ANALYSIS OF RESILIENT AND ROBUST C2 NETWORKS

The third research objective for this thesis is to perform cost-benefit analysis on resilient and robust SoS networks to test the hypothesis that a resilience-based approach is better than a robustness-based approach for designing SoS networks (HYP 2). This analysis addresses the second overarching research question for this thesis (RQ 2), which asks what the most cost effective method to design SoS networks capable of mitigating network threats is. A military application problem is used for the cost-benefit analysis.

This chapter describes an experiment performed to substantiate hypothesis two, beginning with a description of the application problem, followed by results from the cost-benefit analysis.

8.1 Application Problem: Military C2 Networks

A military command and control (C2) network application is used to test hypothesis two because there is a need for C2 networks able to operate in highly contested environments with uncertain and evolving threats [106]. C2 networks enable communications between military systems and are essential to the ability of military forces to share information and develop awareness of the battlespace. Therefore, C2 networks need to be able to mitigate the effects of potential threats to nodes or links. This need is commonly referred to as a need for C2 agility within the C2 community, where agility can be defined as the "capability to successfully cope with changes in circumstances," and includes responsiveness, versatility, flexibility, resilience, and adaptability [12].

C2 networks define connections between military systems operating across potentially large geographic spaces. Nodes represent individual systems (e.g., vehicles or command centers) and links represent communication links between those systems (e.g., secure data transfer links). A network approach to C2 aligns with recent military efforts to transition from platform-centric to network-centric forces, driven by the concepts of NCW and NEC (see section 1.1). C2 networks are representative of SoS networks because they are composed of independent systems, networked together to provide a greater capability than individually possible.

A military unmanned aerial vehicle (UAV) simulation is used to test the performance of potential C2 network designs, where UAVs (blue team) are tasked with maintaining surveillance over enemy and neutral agents in a defined battlefield. C2 networks define communication links between UAVs, enabling information sharing throughout a mission. Enemy agents (red team) are also connected by their own C2 network, accounting for the fact that modern adversaries are often technologically advanced and well-connected. Neutral agents (white team) are not connected to anyone, modeling the presence of people or systems not associated with the blue or red team. This scenario could represent a military force trying to track the actions of terrorists in a populated civilian area. The ability to share information is crucial to success in such a mission, due to large geographic distances that may need to be covered and the movements of people in the area.

The capability of a C2 network is defined as its ability to maintain surveillance of the battlefield. C2 performance is then measured with an awareness metric, defined to quantify the ability of blue agents to maintain awareness of other agent actions. C2 awareness, A, is calculated using Shannon's information entropy such that $0 \leq A \leq 1$. An awareness of $A = 0$ represents a scenario where UAVs have complete uncertainty regarding the locations of other agents. An awareness of $A = 1$ represents a scenario where UAVs have complete certainty regrading the locations of other agents. See appendix C or [107] for more details regarding calculation of the awareness metric.

NetLogo is used to create an agent-based model (ABM) of the C2 application problem. Agent-based modeling is often used to model complex systems composed of many interacting parts, typically networked together in a manner that results in emergent behaviors [25, 72]. NetLogo is a simulation environment commonly used to teach and create ABMs [117].

Three types of agents are defined for the model: blue UAV agents, red enemy agents, and white neutral agents. Agents are defined by a set of attributes and actions. UAVs perform search patterns throughout a simulation, sensing other agents as they enter their sensing radius. UAVs determine locations of other agents as they sense them, which translates to awareness for the blue team. UAVs share the location of other agents through their C2 network, making network connectivity important to their ability to maintain awareness of the battlefield. Red agents attempt to evade detection by moving away from UAVs that they are aware of.

Agents move throughout a square battlefield separated into 36 search grids, where each grid is a square area of equal size. These grids are used to define blue agent search areas and specify agent locations for awareness calculations. Figure 81 shows a screenshot of the ABM. See [107] for more details regarding the simulation.

8.2 EXP 6: Cost-benefit Analysis for Resilient and Robust C2 Networks

Purpose of the experiment: Answer research two and test hypothesis two by comparing the ability of resilient and robust C2 networks to mitigate the effects of network threats in a cost-effective manner.

Experiment six (EXP 6) performs a cost-benefit analysis of resilient and robust C2 networks to test the hypothesis that designing SoS networks for resilience is better than designing them for robustness. A resilient C2 network is defined as one with

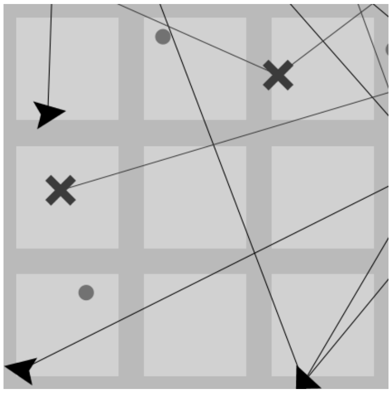


Figure 81: Screenshot of the NetLogo UAV model showing agents with their information links and square search grids in the battlefield (defined by darker shaded patches). Triangular agents are blue agents, “X”-shaped agents are red agents, and circular agents are white agents.

network adaptation, implemented as described in section 5.3. A robust C2 network is defined as one with a high number of initial links (i.e., high initial network density), but no adaptation. Highly dense networks are used to represent robust network designs because increasing network density adds redundant paths or connections between nodes. As described in section 1.5, traditional approaches to robust design include system or functional redundancies.

Experimental Setup

This experiment uses the UAV surveillance simulation with input parameters from [107]. Since C2 networks focus on the ability to share awareness, which can be modeled through message passing as done in the IE network model, results from experiments four and five are used to define scenarios of interest for this experiment.

Experiments four and five show that random topologies provide the most resilience for adaptive networks, regardless of threat type. However, adjusting resilience calculations to more explicitly consider network performance shows that scale-free topologies are able to achieve performance adjusted resilience levels similar to random topologies (when designed with an appropriate adaption method). Since network performance is important to consider in cost-benefit analysis of resilient and robust networks, this experiment focuses on networks with scale-free topologies. Additionally, many real military C2 networks already display scale-free features [61, 51], which may be hard to change due to the complexity and cost of military systems. Therefore, C2 networks are defined to have scale-free (SF) initial network topologies with $N = 20$ nodes (i.e., 20 UAVs or 20 enemy agents). Experiments four and five show that targeted threats are the most damaging to all networks (e.g., see fig. 61). Therefore, C2 threats are defined to be RD targeted node removals to simulate a worst-case scenario.

Multiple robust networks are generated with increasing initial network densities (ranging from 0.1 to 0.95) to account for the lack of adaptation. As a reminder, a network with a density $D = 1$ is a complete network (i.e., all possible links exist). An adjusted BA algorithm is used to generate initial network topologies because network densities that can be generated using the BA model are limited, due to the inclusion of network growth in the model. The number of links in a network from the BA model is defined by the combination of m_0 and m. Since m_0 and m are restricted to being integers, a limited set of network densities can be generated.

The adjusted BA model generates a starting scale-free topology using the BA model. For example, if $N = 20$, $m_0 = 2$, and $m = 1$, a scale-free network with $L = 19$ links (or density $D = 0.1$) is created. Additional links are then randomly added to the scale-free network to achieve the desired network density. Therefore, if a network with $L = 100$ links (or $D = 0.53$) is desired rather than $L = 19$, 81 links are randomly added to the BA scale-free network. This method provides a way to increase the density of scale-free networks. However, these networks are no longer purely scale-free since random link addition is used. Therefore, they are referred to as pseudo-scale-free networks (pseudo-SF), since they transition from being scale-free to complete as network density is increased to one.

Only one adaptive (i.e., resilient) C2 network design is considered. The BA model ($m_0 = 2$ and $m = 1$) is used to generate scale-free topologies for these networks

Table 12: Experimental design matrix used for EXP 6 (50 replications at each point)

Design point	Topology	Initial L	Initial D	Adaptation	Threat
1	Pseudo-SF	19	0.1	None	RD
2	Pseudo-SF	40	0.21	None	RD
3	Pseudo-SF	60	0.32	None	RD
4	Pseudo-SF	80	0.42	None	RD
5	Pseudo-SF	100	0.53	None	RD
6	Pseudo-SF	120	0.63	None	RD
7	Pseudo-SF	140	0.74	None	RD
8	Pseudo-SF	160	0.84	None	RD
9	Pseudo-SF	180	0.95	None	RD
10	SF	19	0.1	Random rewiring	RD

with 19 initial links for an initial network density of 0.1. Random rewiring is used for adaptation, since experiments four and five identify that as the best adaptation method for time sensitive networks facing targeted threats (C2 networks are time sensitive, due to movements within the battlefield). Additionally, optimization results from experiment five show that random rewiring provides the most resilience against a wide range of threats (nearly 72% of the considered threats), ranging from fully targeted to partially random (i.e., $\alpha_{threat} \leq 0.72$). Therefore, even if C2 threats are not fully targeted (e.g., C2 networks facing an adversary with incomplete information of the C2 network topology), the most resilient adaptation method is one that is fully random rewiring-based (i.e., there is no need to consider intermediately defined adaptation methods). Therefore, random rewiring is considered to be the most appropriate adaptation method for C2 networks and is used for this experiment.

Simulated design points for this experiment are shown in table 12, based on selected topologies, threats, and adaptation methods. Design points 1-9 are robust designs; design point 10 is a resilient design. Since the simulation is stochastic, 50 replications are run at each design point.

The cost of a C2 network, C, is represented by the number of links initially created or rewired throughout a simulated scenario, such that

$$C = (L \times C_{initial}) + (L_{rewired} \times C_{rewired}) \,, \tag{72}$$

where L is the number of initial links in the network, $C_{initial}$ is cost of creating an initial link, $L_{rewired}$ is the number of links rewired throughout a scenario, and $C_{rewired}$ is the cost of rewiring a link.

Network links are used as a proxy for cost because there is a cost associated with creating a new link or rewiring an existing one. For example, creating a link between two UAVs may require fitting each UAV with new equipment to enable the transmission, reception, and processing of information. Rewiring a link between UAVs is less easily defined. Certain types of links may simply require a redirection of an existing antenna to "rewire" a previous link to a new UAV. However, other links may require alterations to a UAV to be rewired. Therefore, a range of rewiring costs is considered. The cost of creating an initial link is arbitrarily set to one to simplify analysis.

The total resilience metric, R_{total}, is used to assess the ability of C2 networks to mitigate the effects of network threats. Though this metric is developed to assess resilience, robust and resilient designs share a common goal of preventing the disruption of necessary capabilities and/or recovering any lost capabilities. Therefore, the metric can be used to compare resilient and robust networks.

Network structural properties are also used to compare C2 networks. Total inverse average path length captures the ability of a network to maintain short path lengths throughout a scenario, and is calculated using eq. (68). Path lengths are important for UAV awareness because they determine how quickly information is shared. Since agents are constantly moving around the battlefield, receiving timely updates is important for developing awareness. Total LCC captures the ability of a network to maintain connectivity throughout a scenario and is calculated as

$$(S/N)_{total} = \sum_{i=1}^{N_{networks}} w_i \, (S/N)_i \, , \tag{73}$$

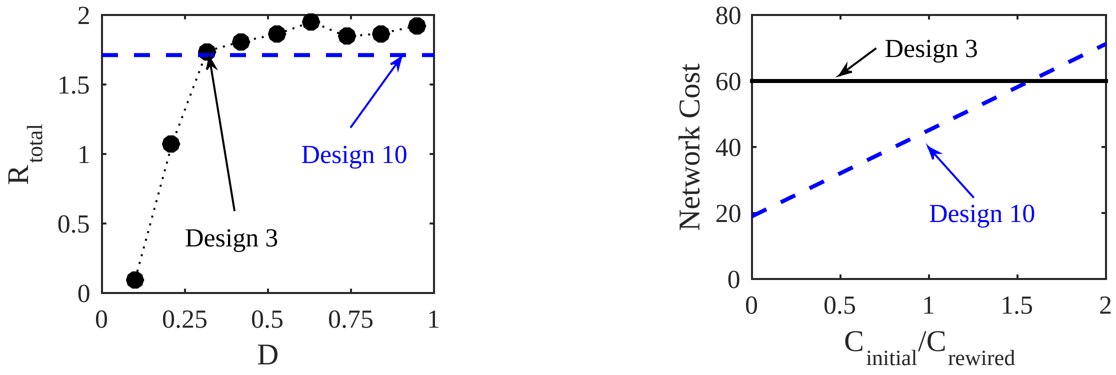
where $N_{networks}$ is the total number of network structures seen in the scenario, the weights w_i are calculated using eq. (48), and $(S/N)_i$ is the normalized size of the LCC for the ith network structure seen. These structural properties are commonly used, in some form or another, to assess network robustness and resilience (see section 2.3.3). Therefore, these metrics are suitable for comparing resilient and robust networks.

Experiment Results

Figure 82a compares the resilience and cost of resilient and robust C2 networks. Since only one resilient network is considered, a constant R_{total} line is shown for the resilient network (design point 10). Increasing network density for the robust designs increases R_{total}, though a threshold is seen at which point further increasing density provides small gains in R_{total}. The resilient network is able to achieve an R_{total} near the threshold of $R_{total} \approx 1.75$. The robust design with $D = 0.32$ (design point three), also achieves an R_{total} near the threshold value and therefore provides similar resilience to the resilient design.

Figure 82b compares the cost of the resilient design to the cost of the robust design specified by design point three, since these designs provide similar levels of resilience. Since the robust design does not include adaptation, its cost is constant and defined by the number of initial links, $C = L = 60$. The cost of the resilient design is shown for various ratios of $C_{intial}/C_{rewired}$. This plot shows that the cost of rewiring a link must be over 1.5 times that of creating a new link to make the resilient design less cost effective than the robust design (when comparing networks able to achieve similar R_{total} values).

Figure 83 shows cost-benefit analysis for resilient and robust designs where network performance is measured by structural properties, rather than R_{total}. These



(a)

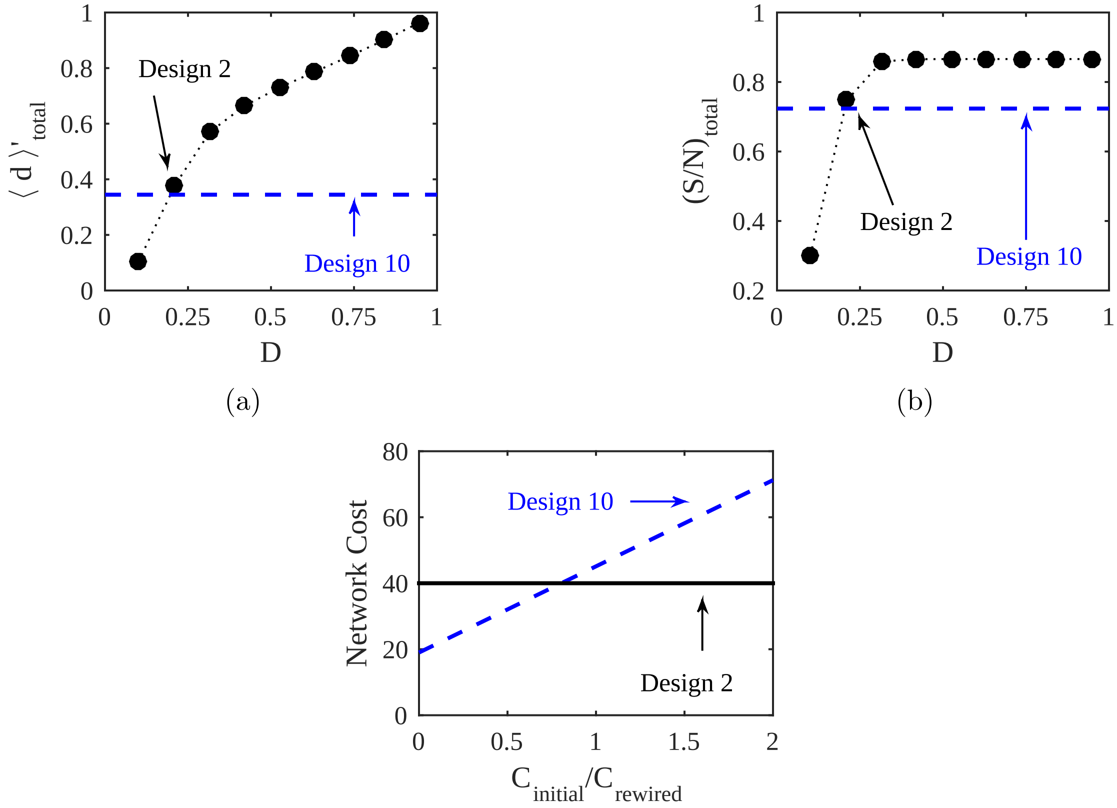
(b)

Figure 82: R_{total} for the resilient network design (design point 10, shown by the dashed blue line) and robust network designs with varied levels of initial network density (design points 1-9, shown by black circles) are shown in (a). Network cost, C, is shown in (b) for the resilient network where the cost of the resilient design is calculated for various link rewiring costs, $C_{rewired}$. Network cost is also shown for design point 3, the robust network showing the most comparable R_{total} to the resilient network.

results show slightly different trends, depending on the structural property used. Robust networks are able to provide higher total inverse average path lengths than the resilient network with no apparent threshold on their maximal value (within the scenarios considered). In comparison, a threshold is seen on the size of the total LCC robust networks can achieve. The resilient design is unable to match the performance of highly dense robust networks, when measured by these structural properties, though it is closer to matching maximum total LCC than maximum total inverse average path length.

However, comparison of network costs for similarly performing networks shows that depending on the cost of rewiring a link, a resilient design may still be more cost effective than a robust one. Comparing design points two and ten, which show similar total inverse average path lengths and LCCs, the resilient design is more cost effective if the cost of rewiring a link is less than 0.75 times that of creating a new link.

Since comparisons between resilient and robust designs using R_{total} and structural



(a)

(b)

(c)

Figure 83: Total inverse average path lengths for the resilient network design (design point 10, shown by the dashed blue line) and robust network designs with varied levels of initial network density (design points 1-9, shown by black circles) are shown in (a), with total size of the LCC shown in (b). Network cost, C, is shown in (c) for the resilient network as a function of the ratio $C_{initial}/C_{rewired}$. Network cost is also shown for design point 2, the robust network showing the most comparable structural properties to the resilient network.

properties show different trends, the actual performance of these networks is also analyzed. Analyzing the desired capability, awareness for this application, provides a sense of how well R_{total}, $\langle d\rangle'_{total}$, and $(S/N)_{total}$ are capturing differences between network designs. Figure 84 show awareness quantiles for robust networks specified by designs two and three, and the resilient network specified by design ten.

Recall that robust design three shows similar values for R_{total} to resilient design ten (see fig. 82a). However, comparison of awareness results shows that design three actually maintains higher awareness than design ten throughout the entire simulated

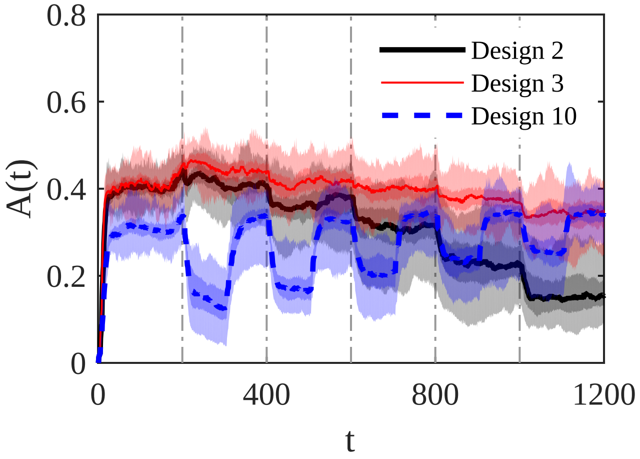


Figure 84: Quantile awareness results for the robust and resilient network designs showing similar performance when measured by R_{total}, $\langle d \rangle'_{total}$, and $(S/N)_{total}$. Solid and dashed lines show median awareness, dark shaded regions show 25 and 75% quartiles, and lightly shaded regions show minimum and maximum ranges. Dashed vertical lines show node removal events, where one blue UAV agent is removed at each event.

scenario, contradicting the R_{total} comparison. This contradiction is explained by the lower *initial* awareness of the resilient network [i.e., $A(t)$ for $0 \leq t < 200$]. Calculations of R (which are used for R_{total}) assume that the initial awareness is the desired performance level, y_D, resulting in R_{total} being resilience relative to the initial performance of a system. Therefore, despite design ten having lower awareness throughout the scenario, it shows similar levels of resilience to design three, because of its ability to recover performance levels equal to or greater than its initial performance level or capability. Resilience calculations can be adjusted to prevent this contradiction by specifying the same desired capability, y_D, for all designs being compared.

Recall that robust design two shows similar structural properties to resilient design ten (see figs. 83a and 83b). These two designs also show comparable awareness trends throughout the entire scenario, where the robust design is better able to absorb the effects of threats early in the scenario, but the resilient design is better able to recover from threats later in the scenario. Therefore, network structural properties may be better metrics for comparing robust and resilient designs than R_{total}, due to their inclusion of the initial network performance in calculated properties.

Discussion of Results

Cost-benefit analysis of resilient and robust C2 networks focuses on the cost of rewiring a link relative to creating a new one. When measuring the ability of a network to mitigate threats with R_{total}, the ratio $C_{initial}/C_{rewired}$ must be over 1.5 for robust networks to be more cost effective. However, when measuring threat mitigation ability with structural network properties, the point at which robust networks become more cost effective is lowered to $C_{initial}/C_{rewired} \approx 0.75$. Therefore, as long the cost of rewiring links is less than three-fourths the cost of creating a new link, a resilient C2 network design is more cost effective than a robust design (based on the provided definition of network cost). Note that this definition of network cost does not explicitly consider the over-head cost that may be associated with enabling link rewiring. However, this cost can be implicitly included in the cost of rewiring a link.

Current studies of network resilience are limited in their consideration of network costs, specifically with regards to the cost of achieving resilience through link rewiring rather than link redundancy. This analysis identifies a quantitative (though abstract) threshold for when adaptive networks are a more cost-effective solution than robust networks, and provides an analyst with a method for supporting one design approach over another (given estimations of link creation and rewiring costs). Since there are scenarios where robust networks are more cost effective than resilient networks, hypothesis two cannot be fully verified. The following answer is given to research question two, based on results from this experiment:

> **Response to RQ 2:** Resilient C2 network designs are generally more cost effective than robust designs, when the cost of rewiring a link is less than three-fourths the cost of creating a new one. Otherwise, robust networks may be more cost effective.

8.3 Summary of Results

This chapter presents a cost-benefit analysis for resilient and robust C2 network designs. A network UAV surveillance simulation is used to evaluate the performance of C2 networks facing targeted node removals. Network cost is represented by new and rewired link costs. A resilient C2 network is defined using random rewiring. Robust C2 networks are defined by increasing initial network densities. The following summarizes research questions, responses, and experiments from this chapter:

- RQ 2: What is the most cost effective method for designing SoS networks that can mitigate the effects of potential network threats?
- Response to RQ 2 (EXP 6): Resilient C2 network designs are generally more cost effective than robust designs, when the cost of rewiring a link is less than three-fourths the cost of creating a new one. Otherwise, robust networks may be more cost effective.

CHAPTER IX

SUMMARY

Increased connectivity among systems has created SoS dependent upon networks to provide desired capabilities and functionality. This connectivity introduces vulnerabilities to SoS networks that must be considered for the analysis and design of SoS. This thesis addresses two primary research questions related to this problem: (1) what happens to SoS networks when nodes fail or are attacked, and how can we mitigate the effects of those failures and (2) what is the most cost-effective method for designing SoS networks able to provide that mitigation ability? A review of the literature identifies designing for resilience as a promising approach to addressing SoS network vulnerabilities. This thesis hypothesizes that a resilience-based approach is more cost-effective than a robustness-based approach for designing SoS networks.

Three research objectives are identified for this thesis, in an effort to answer those research questions and test the formulated hypothesis. The first is to develop a resilience-based methodology for designing SoS networks able to mitigate potential threats. The second is to then use the developed methodology to design resilient SoS networks. The third objective is to perform a cost-benefit analysis of resilient and robust SoS networks to identify whether designing for resilience or robustness is better suited for SoS networks.

This chapter summarizes the developed ReSSNET methodology for designing resilient SoS networks, experimental results from applying the methodology to design resilient IE networks, as well as results from the cost-benefit analysis of resilient and robust C2 networks, main contributions from this thesis, and possible extensions for this work.

ReSSNET: A Methodology for Designing Resilient SoS Networks

Chapters 3 to 6 develop the ReSSNET methodology for designing resilient SoS networks, satisfying the first research objective for this thesis (summarized in fig. 47). The methodology includes a capability-based resilience assessment framework, which provides a set of quantitative metrics for assessing system, or SoS, resilience. The total resilience metric R_{total} provides a single metric for the resilience of a system facing multiple disruptions over time, enabling large simulation design studies, such as that performed in this thesis. This assessment framework builds upon and expands previously proposed frameworks in the resilience engineering community.

A complex networks approach is used to generate SoS design alternatives. This approach provides methods for defining network topologies, threats, and adaptation methods based on real world networks and processes occurring on those networks. The focus on adaptive networks within this thesis extends previous work on the robustness of statically defined complex networks.

Statistical experimental design methods, specifically RSM, are used to explore and optimize SoS design alternatives. Main and interaction effects identify overall trends in the data and provide a general idea of how particular factors affect network resilience. Linear regression provides polynomial functions representative of the continuous network design space, which are used to determine optimally resilient SoS networks. A network interpolation model is identified and modified to enable a continuous representation of the network design space, extending previous studies of discretely defined networks.

Experimental Results

The developed methodology is applied to an IE network model. Results for IE networks without adaptation confirm those from the complex networks literature, that scale-free networks are robust to random failures but susceptible to targeted attacks. Adding network adaptation to IE networks improves their resilience to all threat types considered. The most resilient network topology for IE networks with adaptation is found to random, rather than scale-free, regardless of threat type. The optimally resilient network adaptation method shows a sharp transition from being fully random to fully degree-based, as threat randomness increases. These results suggest that adaptive IE networks should have a random initial topology, with a fully random or fully degree-based adaptation method (depending on anticipated threat types) when multiple threats are anticipated throughout the life-cycle of a network.

Cost-benefit analysis is performed to compare resilient and robust C2 network designs. This analysis shows that when performance is measured by total resilience R_{total}, resilient C2 networks are more cost-effective than robust ones as long as the cost of rewiring a link is less than 1.5 times the cost of creating a new link. However, when performance is measured by network structural properties, the cost of rewiring a link must be less than 0.75 times the cost of a new link for resilient designs to be more cost effective.

Main Contributions

There are three main contributions from this thesis: the ReSSNET methodology for designing resilient SoS networks, the exploration and optimization of a continuous IE network design space, and the cost-benefit analysis of resilient and robust C2 networks.

The ReSSNET methodology provides a thorough process for a designer or analyst to generate potential network alternatives and quantitatively evaluate their resilience. Current resilience assessment methods are limited in their ability to account for the adaptive nature of desired resilient systems, as well as complexities related to actual data from SoS simulations. For example, many existing frameworks focus on notional data representative of SoS performance, rather than actual data characterized by performance volatility and noisy transitions from degraded to recovered states. The proposed resilience assessment framework provides easily automated methods for handling noisy data and quantifying resilience in a way that penalizes performance volatility. The total resilience metric, R_{total}, also provides a single value for the resilience of a system, or SoS, facing repeated threats. This metric enables large-scale simulation design studies that consider the overall life cycle of an SoS, such as the one performed in experiments four and five. Previous resilience metrics limit analysis of SoS resilience to a single threat event, require user inputs, or focus on qualitative comparisons of system performance. Experiment one also describes an analytic function (based on the logistic function) that can be used to generate notional system performance data for evaluating potential resilience metrics. This function enables more thorough comparisons of resilience assessment methods than those currently performed in the literature. By focusing on resilience, this methodology can be coupled with more traditional performance-based analysis to provide a complete understanding of how to design SoS networks.

The methodology also provides guidance for how to generate and analyze a continuous network design space using response surface methodology. Much of the work on complex network resilience focuses on discretely defined combinations of static networks and threat types. This thesis contributes to this area by considering several network adaptation methods as a response to node removals and exploring the full space of IE network designs, including intermediately defined networks, with design of experiments and linear regression models.

Exploration of a continuous network design space is important because network topologies and adaptation methods do not have to be designed as fully degree-based (e.g., scale-free topologies) or fully random, just as threats will not always be purely targeted or random. Therefore, network designers must understand the resilience of intermediately defined networks, and consider threats spanning the space between fully targeted and random. Without such analysis, one may assume that intermediately defined networks are best for intermediately defined threats (e.g., partially random networks provide the most resilience against partially targeted threats). In other words, one may assume that network resilience can be interpolated from analysis of the discrete design space.

However, optimization of the continuous network design space shows a sharp adaptation transition for time sensitive IE networks, where the optimally resilient adaptation method changes from fully random to fully degree-based as threat randomness increases. This transition does not occur until threats become highly random for time sensitive networks (i.e., fully random adaptation provides the most resilience against threats with up to 72% randomness). This result suggests that partially random adaptation should not be used; instead, fully random adaptation should be used unless threats are anticipated to be completely random, at which point fully degree-based adaptation should be used. In comparison, fully random topologies are found to provide the most resilience against all threat types considered. These results suggest that network designers should not consider intermediate network designs, and demonstrate that simply interpolating discrete network results may lead to poor network resilience.

The suggestion of fully random adaptation for targeted and partially random threats particularly impacts C2 network designers because C2 systems are often designed around hubs that control the overall C2 network. The presence of hubs in military networks is evidenced by the scale-free nature of many C2 networks. These results suggest that adaptive C2 networks should instead focus on degree homogeneity to achieve resilience, even for threats that contain some combination of targeting and randomness.

This thesis also addresses the relationship between resilient and robust design methods through cost-benefit analysis of adaptive and highly dense C2 networks. For network costs focusing on link creation and rewiring, results show that resilient C2 networks are more cost-effective than robust C2 networks as long as the cost of rewiring a link is less than three-fourths the cost of creating a link. This analysis provides quantitative support for choosing one design approach over another. For C2 systems with low rewiring costs (e.g., airborne networks that may only require a re-direction of existing antennae), this analysis suggests one should design adaptive, rather than dense C2 networks. However, for C2 networks with high rewiring costs (e.g., ground-based communications requiring landlines), this analysis suggests the design of dense C2 networks.

Future Work

There are limitations to this work, as well as many potential extensions, that if addressed can further develop our understanding of network resilience. One limitation is that the networks used are defined to be homogeneous with respect to individual node properties or capabilities. As discussed in section 1.2, many SoS are actually composed of heterogeneous systems, with varying capabilities and properties. Much of the work on complex network resilience also focuses on homogeneous networks. The inclusion of node heterogeneity to this work would advance research on network resilience and improve its applicability to the SoS design process.

The IE network model used also assumes that nodes have global knowledge of the network, i.e. nodes know the complete topology of the network at any given time. Due to the large scale and geographic dispersion of many SoS, this assumption may not always be accurate. Accounting for information locality constraints would extend this research and provide a more thorough consideration of adaptive network resilience.

Additionally, this thesis focuses on single-layered networks. Many network researchers have identified that networks are often characterized by interdependencies between network layers; i.e., networks are often multi-layered, such that failures in one layer affect the behaviors and processes occurring in another layer. Considering multi-layered networks and the interdependencies within them would extend the consideration of network resilience provided by this thesis.

This work can also be combined with big data analytics to provide real-time health monitoring and control of adaptive SoS networks and similar complex systems. There is a growing amount of data available that can be used to monitor and characterize the performance of modern systems. Classification of trends in this data, or perhaps more importantly anomalies in the data, can be combined with surrogate modeling-enabled decision support tools to identify potential responses and adaptive measures to use in response to future threats. This data can feed into network simulations and models to more accurately represent desired systems. In return, these simulations and models can predict system behaviors in potentially catastrophic conditions.

APPENDIX A

MAIN EFFECTS AND INTERACTIONS

Using the notation defined in table 6, the main effect of factor A, e_A, is calculated as

$$\begin{aligned} e_A &= \bar{y}_{A^+} - \bar{y}_{A^-} \\ &= \frac{y_2 + y_4 + y_6 + y_8}{4} - \frac{y_1 + y_3 + y_5 + y_7}{4} \\ &= \frac{1}{4}\left[y_2 + y_4 + y_6 + y_8 - y_1 - y_3 - y_5 - y_7\right], \end{aligned} \tag{74}$$

where $\bar{y}_{A^+}$ is the mean response for all design points where factor A is set to its "+" level. The main effects of factors B, e_B, and C, e_C, are similarly calculated as

$$\begin{aligned} e_B &= \bar{y}_{B^+} - \bar{y}_{B^-} \\ &= \frac{y_3 + y_4 + y_7 + y_8}{4} - \frac{y_1 + y_2 + y_5 + y_6}{4} \\ &= \frac{1}{4}\left[y_3 + y_4 + y_7 + y_8 - y_1 - y_2 - y_5 - y_6\right], \end{aligned} \tag{75}$$

$$\begin{aligned} e_C &= \bar{y}_{C^+} - \bar{y}_{C^-} \\ &= \frac{y_5 + y_6 + y_7 + y_8}{4} - \frac{y_1 + y_2 + y_3 + y_4}{4} \\ &= \frac{1}{4}\left[y_5 + y_6 + y_7 + y_8 - y_1 - y_2 - y_3 - y_4\right]. \end{aligned} \tag{76}$$

The interaction effect, e_{AB}, between two factors A and B is calculated as

$$\begin{aligned} e_{AB} &= \frac{1}{2}\left[\bar{e}_{A,B^+} - \bar{e}_{A,B^-}\right] \\ &= \frac{1}{2}\left[\frac{(y_4 - y_3) + (y_8 - y_7)}{2} - \frac{(y_2 - y_1) + (y_6 - y_5)}{2}\right] \\ &= \frac{1}{4}\left[(y_4 - y_3 + y_8 - y_7) - (y_2 - y_1 + y_6 - y_5)\right], \end{aligned} \tag{77}$$

where $\bar{e}_{A,B^+}$ is the average effect of factor A with factor B at its "+" level and $\bar{e}_{A,B^-}$ is the average effect of factor A with factor B at its "$-$" level. The interaction effects between A and C, e_{AC} and B and C, e_{BC}, are similarly calculated as

$$\begin{aligned} e_{AC} &= \frac{1}{2}\left[\bar{e}_{A,C^+} - \bar{e}_{A,C^-}\right] \qquad (78) \\ &= \frac{1}{2}\left[\frac{(y_6 - y_5) + (y_8 - y_7)}{2} - \frac{(y_2 - y_1) + (y_4 - y_3)}{2}\right] \\ &= \frac{1}{4}\left[(y_6 - y_5 + y_8 - y_7) - (y_2 - y_1 + y_4 - y_3)\right] \\ e_{BC} &= \frac{1}{2}\left[\bar{e}_{B,C^+} - \bar{e}_{B,C^-}\right] \qquad (79) \\ &= \frac{1}{2}\left[\frac{(y_7 - y_5) + (y_8 - y_6)}{2} - \frac{(y_3 - y_1) + (y_4 - y_2)}{2}\right] \\ &= \frac{1}{4}\left[(y_7 - y_5 + y_8 - y_6) - (y_3 - y_1 + y_4 - y_2)\right]. \end{aligned}$$

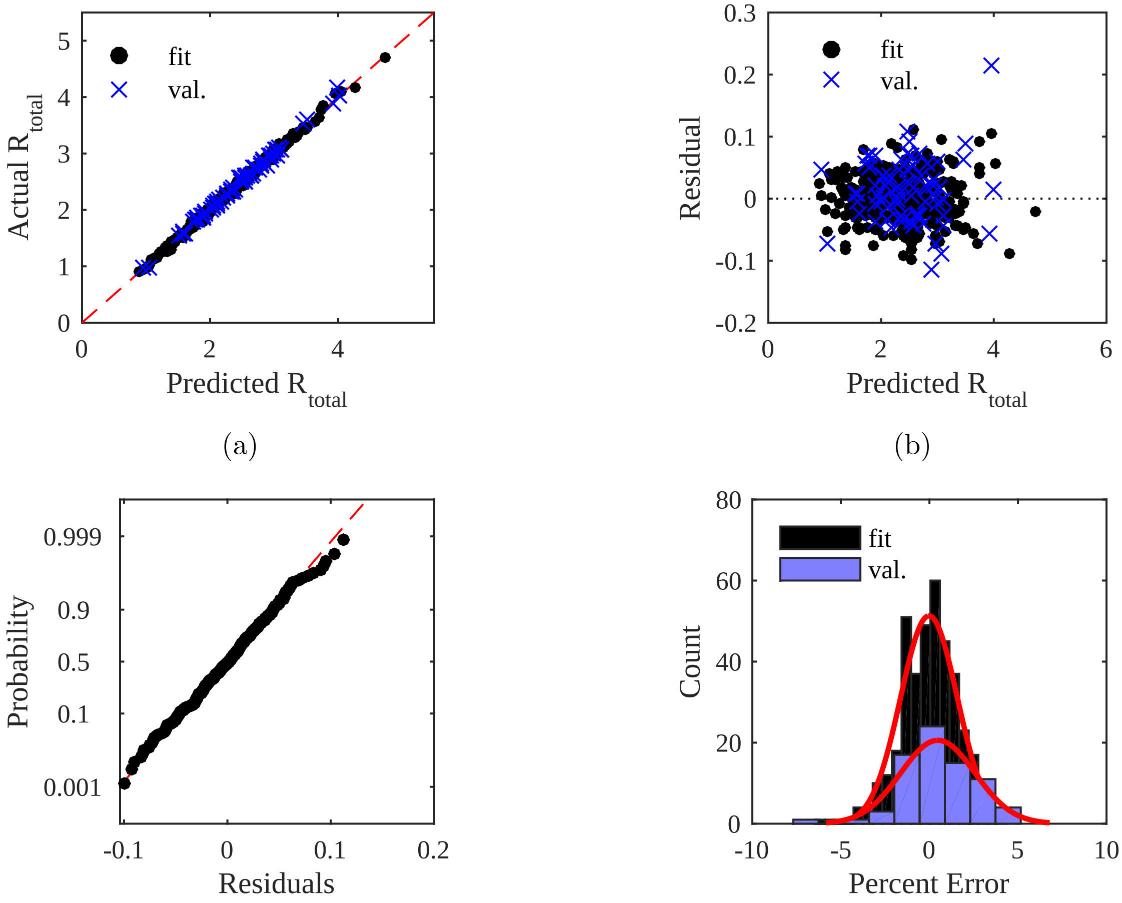
The interaction effect between factors A, B, and C is calculated as

$$\begin{aligned} e_{ABC} &= \frac{1}{2}\left[\bar{e}_{AB,C^+} - \bar{e}_{AB,C^-}\right] \qquad (80) \\ &= \frac{1}{2}\left[\frac{(y_8 - y_7) - (y_6 - y_5)}{2} - \frac{(y_4 - y_3) - (y_2 - y_1)}{2}\right] \\ &= \frac{1}{4}\left[(y_8 - y_7 - y_6 + y_5) - (y_4 - y_3 - y_2 + y_1)\right]. \end{aligned}$$

APPENDIX B

LINEAR REGRESSION MODELS

Figures 85 to 88 show model fit checks for linear regression models generated in experiment five.



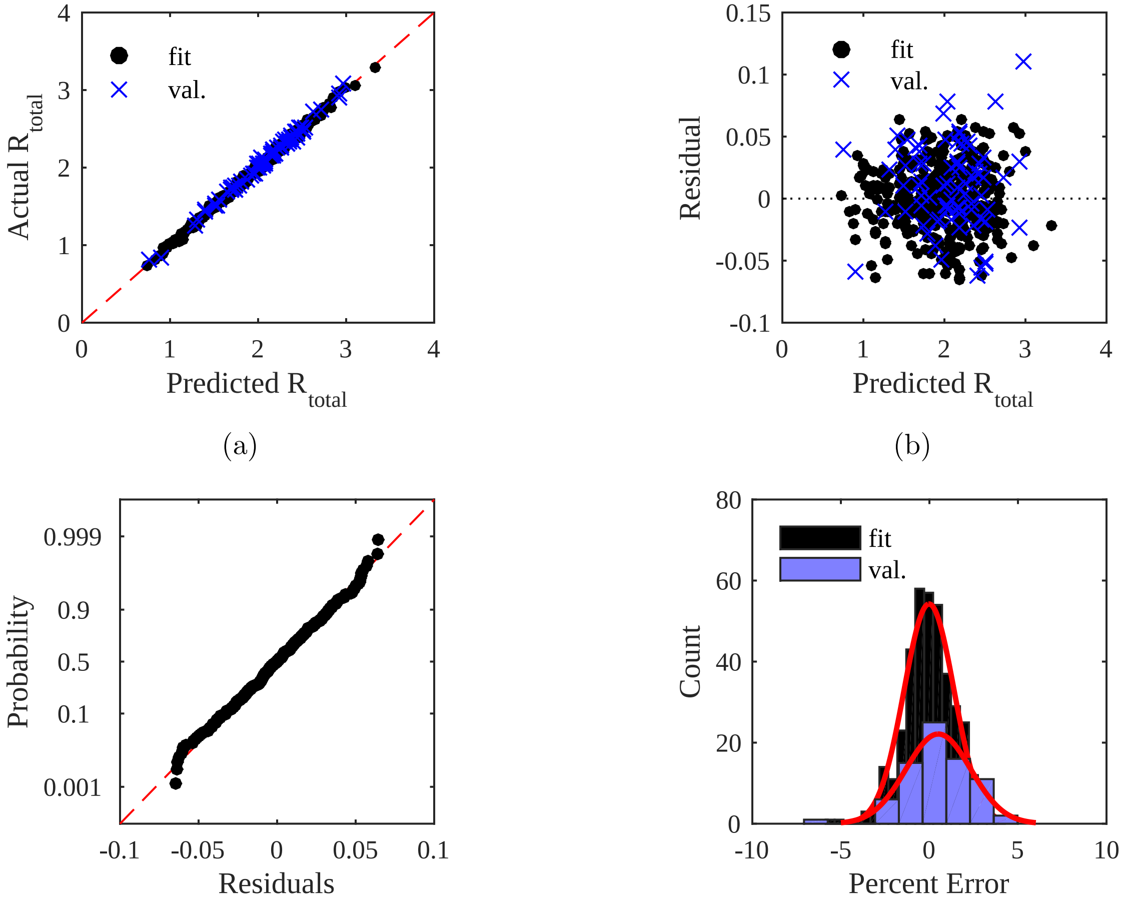
(a)

(b)

(c)

(d)

Figure 85: Model fit checks for a fifth-order regression model of data with $\Delta = 0.7$. Actual by predicted responses are shown in (a), residuals are shown in (b), a normal probability plot for fit data in (c), and model error distributions in (d). Standard deviations for model error distributions are $\sigma_{fit} = 1.6$ and $\sigma_{val.} = 2.1$. Maximum model error is 5.2%.



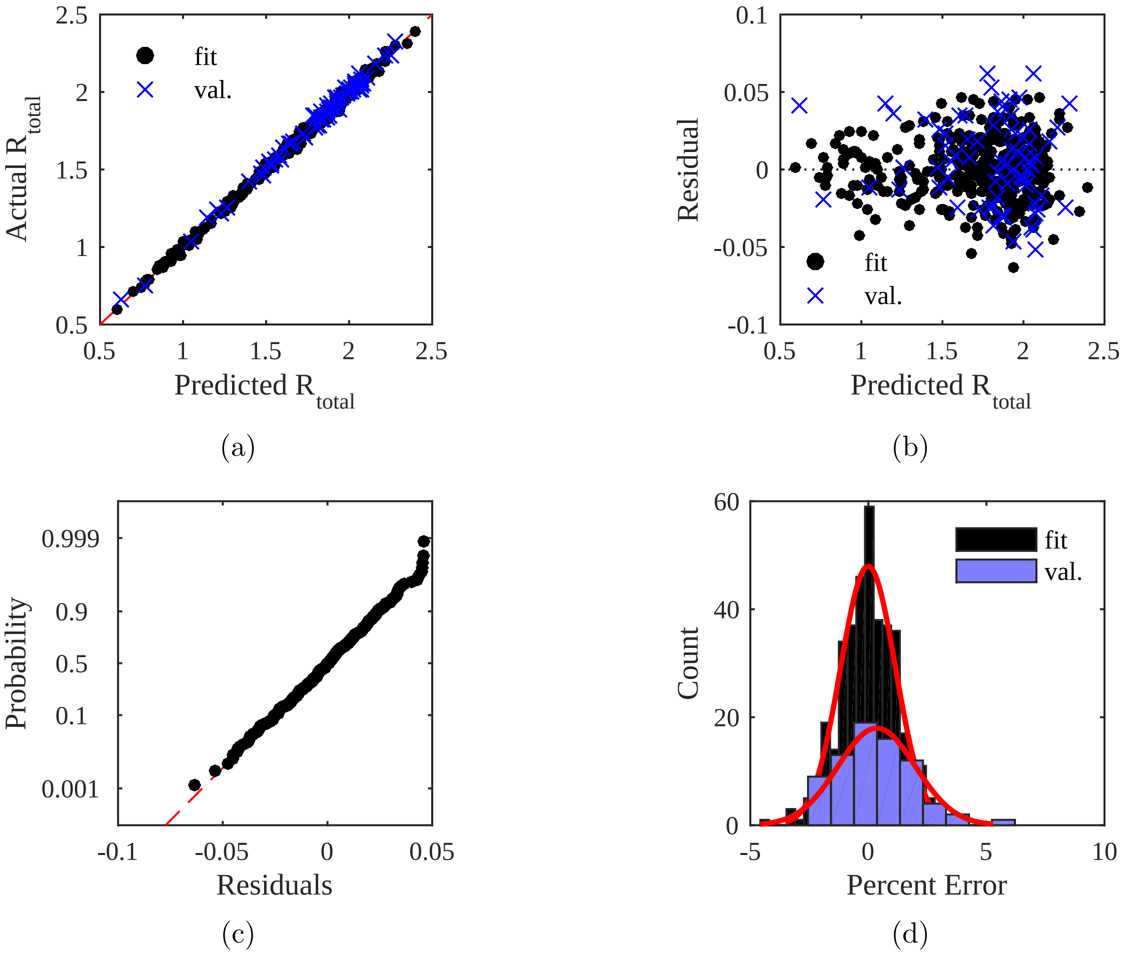
(a)

(b)

(c)

(d)

Figure 86: Model fit checks for a fifth-order regression model of data with $\Delta = 0.8$. Actual by predicted responses are shown in (a), residuals are shown in (b), a normal probability plot for fit data in (c), and model error distributions in (d). Standard deviations for model error distributions are $\sigma_{fit} = 1.4$ and $\sigma_{val.} = 1.8$. Maximum model error is 5.0%.



(a)

(b)

(c)

(d)

Figure 87: Model fit checks for a fifth-order regression model of data with $\Delta = 0.9$. Actual by predicted responses are shown in (a), residuals are shown in (b), a normal probability plot for fit data in (c), and model error distributions in (d). Standard deviations for model error distributions are $\sigma_{fit} = 1.2$ and $\sigma_{val.} = 1.6$. Maximum model error is 6.2%.

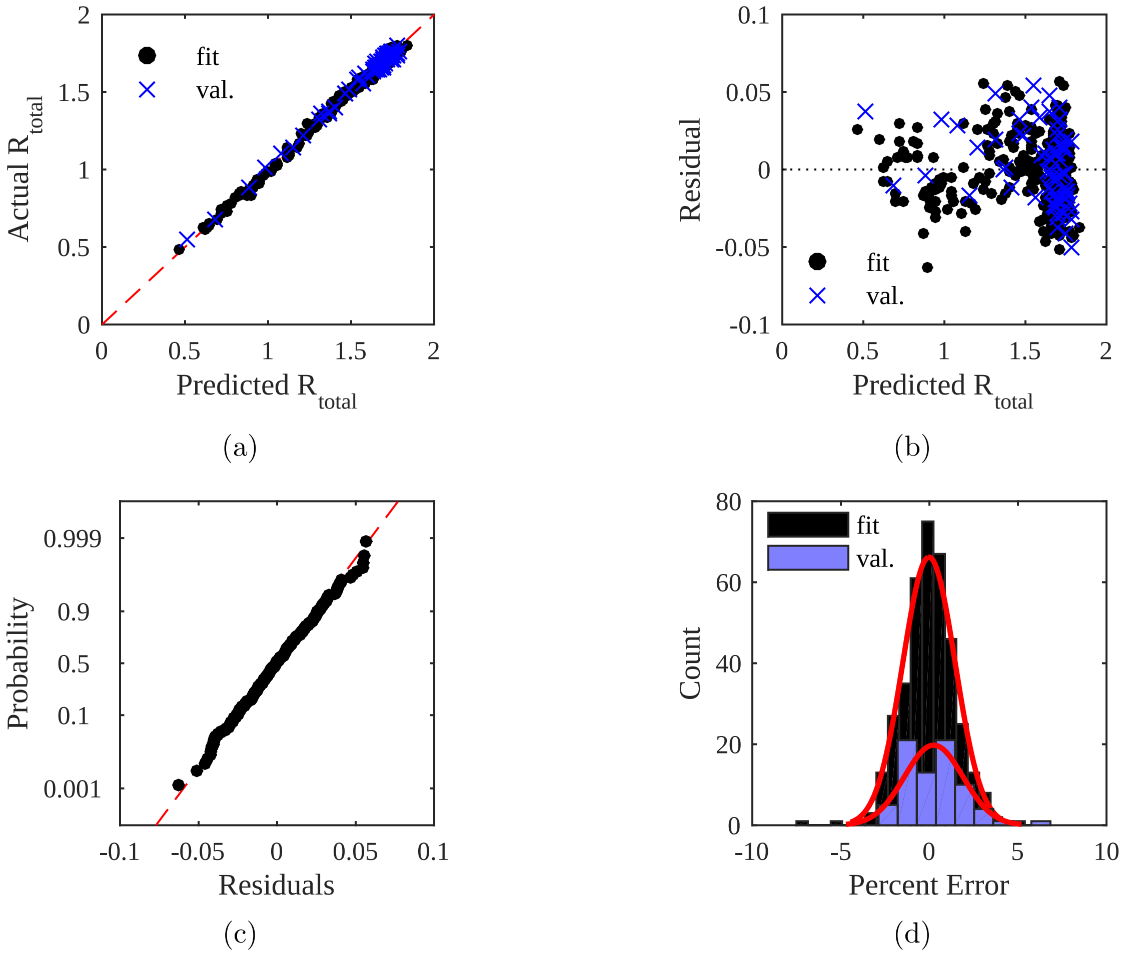


Figure 88: Model fit checks for a fifth-order regression model of data with $\Delta = 1$. Actual by predicted responses are shown in (a), residuals are shown in (b), a normal probability plot for fit data in (c), and model error distributions in (d). Standard deviations for model error distributions are $\sigma_{fit} = 1.5$ and $\sigma_{val.} = 1.7$. Maximum model error is 6.8%.

APPENDIX C

AWARENESS METRIC CALCULATIONS

Awareness is calculated using Shannon's information entropy as follows [39, 106]:

1. Discretize the battlespace into relevant features such as the ID, location, and team of friendly and enemy forces. These features are referred to as state properties of the battlespace, where each agent is defined by those state properties. Each state property is defined by a set of possible conditions or values. For example, the team state property consists of three possible conditions: blue, red, or white.

2. Model each state property as a random variable, X, with a discrete probability distribution. State probability distributions are derived from the performance of system functions (e.g., sensing, classification) corresponding to related mission tasks.

3. Use information entropy to determine the amount of maximum uncertainty, U, based on the maximum number of possible conditions or outcomes:

$$U = H(X)_{max} = log_b(n). \tag{81}$$

4. Use information entropy to determine the amount of uncertainty, $H(X)$, represented by a probability distribution, where entropy is calculated as

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_b p(x_i). \tag{82}$$

5. Transform $H(X)$ into a measure of awareness, A, using

$$A(t) = 1 - \frac{H(X)}{U}. \tag{83}$$

   Complete awareness of the battlespace means having absolute certainty of the condition of each state property or battlespace feature.

6. Calculate total awareness for a group of n agents (e.g., n blue team agents) considering m state properties (e.g., agent ID, location, team) as the mean awareness of all state properties over all active agents within that group (i.e., all agents that have not been removed from the simulation), using

$$A_{total}(t) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}(t). \tag{84}$$

7. Incorporate the awareness calculations into a warfare simulation.

8. Analyze awareness versus time results to determine the effectiveness of various C2 networks.

Table 13 shows example awareness calculations for the team state property of an un-identified agent, with three cases shown. The first case represents maximum uncertainty, as the unidentified agent is estimated to be a friendly agent (blue), enemy agent (red), or neutral agent (white) with equal probability. Case 3 represents the opposite situation, where the agent's team is identified with complete certainty, leading to an awareness value of one. Case 2 represents an intermediate situation, where the agent's team is believed to be red, but with some uncertainty. This type of probability distribution results in an intermediate value of awareness.

Table 13: Example entropy calculations for quantifying awareness of an agent's team (using a log base of 2)

	Team Probability Distribution			Awareness Calculation		
	Blue team	Red team	White team	U (bits)	$H(X)$ (bits)	$A(t)$
Case 1 (maximum uncertainty)	0.33	0.33	0.33	1.59	1.59	0
Case 2 (intermediate uncertainty)	0	0.75	0.25	1.59	0.81	0.49
Case 3 (no uncertainty)	0	1	0	1.59	0	1

REFERENCES

[1] "Numerical Analysis of Cyberattacks."

[2] "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems," 2000.

[3] "INCOSE Systems Engineering Handbook Version 2.0," 2000.

[4] "Maps," 2005.

[5] "SAS-065: NATO NEC C2 Maturity Model," 2008.

[6] "INCOSE Systems Engineering Handbook Version 3.2," 2011.

[7] "Transforming Transportation through Connectivity," tech. rep., US Department of Transportation, 2012.

[8] Albert, R. and Barabási, A.-L., "Statistical Mechanics of Complex Networks," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 47–97, 2002.

[9] Albert, R., Jeong, H., and Barabasi, A.-L., "Diameter of the World-Wide Web," *Nature*, vol. 401, pp. 130–131, 1999.

[10] Albert, R., Jeong, H., and Barabasi, A.-L., "Error and attack tolerance of complex networks," *Nature*, vol. 406, pp. 378–82, July 2000.

[11] Albert, R., Jeong, H., and Barabasi, A.-L., "correction: Error and attack tolerance of complex networks," *Nature*, vol. 409, no. January, pp. 542–542, 2001.

[12] Alberts, D. S., *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*. DoD CCRP, 2011.

[13] Alberts, D. S., Garstka, J. J., and Stein, F. P., *Network Centric Warfare: Developing and Leveraging Information Superiority.* DoD Command and Control Research Program, 2nd ed., 2000.

[14] Alderson, D. L., "Catching the Network Science Bug: Insight and Opportunity for the Operations Researcher," *Operations Research*, vol. 56, pp. 1047–1065, Oct. 2008.

[15] Alderson, D. L. and Doyle, J. C., "Contrasting Views of Complexity and Their Implications For Network-Centric Infrastructures," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, pp. 839–852, July 2010.

[16] Atzori, L., Iera, A., and Morabito, G., "The Internet of Things: A survey," *Computer Networks*, vol. 54, pp. 2787–2805, Oct. 2010.

[17] Bagler, G., "Analysis of the airport network of India as a complex weighted network," *Physica A: Statistical Mechanics and its Applications*, vol. 387, pp. 2972–2980, May 2008.

[18] Balchanos, M., Li, Y., and Mavris, D., "Towards a method for assessing resilience of complex dynamical systems," in *2012 5th International Symposium on Resilient Control Systems*, pp. 155–160, Ieee, Aug. 2012.

[19] Balchanos, M. G., Domercant, J. C., Tran, H. T., and Mavris, D. N., "Metrics-based Analysis and Evaluation Framework for Engineering Resilient Systems," in *2014 7th International Symposium on Resilient Control Systems*, (Denver, CO), IEEE, 2014.

[20] Balchanos, M. G., *A Probabilistic Technique for the Assessment of Complex Dynamic System Resilience.* Phd, Georgia Institute of Technology, 2012.

[21] Barabási, A.-L., *Network Science.* 2012.

[22] Barabási, A.-l. and Albert, R., "Emergence of Scaling in Random Networks," *Science*, vol. 286, pp. 509–512, Oct. 1999.

[23] Boardman, J. and Sauser, B., "System of Systems - the meaning of of," in *2006 IEEE/SMC International Conference on System of Systems Engineering*, (Los Angelos, CA), pp. 118–123, IEEE, 2006.

[24] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D., "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, pp. 175–308, Feb. 2006.

[25] Bonabeau, E., "Agent-based modeling: methods and techniques for simulating human systems.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 7280–7, May 2002.

[26] Box, G. E. P., Hunter, J. S., and Hunter, W. G., *Statistics for Experimenters: Design, Innovation, and Discovery.* Hoboken, NJ: Wiley, 2nd ed., 2005.

[27] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J., "Graph structure in the Web," *Computer Networks*, vol. 33, pp. 309–320, June 2000.

[28] Callaway, D. S., Newman, M. E., Strogatz, S. H., and Watts, D. J., "Network robustness and fragility: percolation on random graphs," *Physical review letters*, vol. 85, pp. 5468–71, Dec. 2000.

[29] Cebrowski, A. K. and Garstka, J. J., "Network-Centric Warfare: Its Origin and Future," in *Proceedings of the US Naval Institute*, 1998.

[30] Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S., "Resilience of the internet to random breakdowns," *Physical review letters*, vol. 85, pp. 4626–8, Nov. 2000.

[31] Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S., "Breakdown of the Internet under Intentional Attack," *Physical Review Letters*, vol. 86, pp. 3682–3685, Apr. 2001.

[32] Dahmann, J., Rebovich, G., Lane, J., Lowry, R., and Baldwin, K., "An implementers' view of systems engineering for systems of systems," *2011 IEEE International Systems Conference*, pp. 212–217, Apr. 2011.

[33] Davendralingam, N. and DeLaurentis, D., "A Robust Optimization Framework to Architecting System of Systems," *Procedia Computer Science*, vol. 16, pp. 255–264, Jan. 2013.

[34] DeLaurentis, D. A., "Understanding Transportation as System-of-Systems Design Problem," in *43rd AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, NV), AIAA, 2005.

[35] DiMario, M. J., "System of Systems Interoperability Types and Characteristics in Joint Command and Control," in *Proceedings of the 2006 IEEE/SMC International Conference on System of Systems Engineering*, (Los Angelos, CA), pp. 236–241, IEEE, 2006.

[36] Dimitrakopoulos, G. and Demestichas, P., "Intelligent Transportation Systems," *IEEE Vehicular Technology Magazine*, vol. 5, no. March 2010, pp. 77–84, 2010.

[37] Dixon, A. and Henning, J., "Nett Warrior gets new end-user device," 2013.

[38] Dodds, P. S., Watts, D. J., and Sabel, C. F., "Information exchange and the robustness of organizational networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, pp. 12516–21, Oct. 2003.

[39] Domercant, J. C. and Mavris, D., "Understanding and Evaluating Command & Control Effectiveness by Measuring Battlespace Awareness," in *19th International Command and Control Research and Technology Symposium*, (Alexandria, VA), 2014.

[40] Dunne, J. a., Williams, R. J., and Martinez, N. D., "Food-web structure and network theory: The role of connectance and size.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 12917–22, Oct. 2002.

[41] Dunne, J. a., Williams, R. J., and Martinez, N. D., "Network structure and biodiversity loss in food webs: robustness increases with connectance," *Ecology Letters*, vol. 5, pp. 558–567, July 2002.

[42] Eisner, H., Marciniak, J., and McMillan, R., "Computer-aided System of Systems (S2) Engineering," in *IEEE International Conference on Systems, Man, and Cybernetics*, (Charlottesville, VA), pp. 531–7, IEEE, 1991.

[43] Erdos, P. and Renyi, A., "On the Evolution of Random Graphs," *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, vol. 5, pp. 17–60, 1960.

[44] Eusgeld, I., Nan, C., and Dietz, S., "System-of-systems approach for interdependent critical infrastructures," *Reliability Engineering & System Safety*, vol. 96, pp. 679–686, June 2011.

[45] Faloutsos, M., Faloutsos, P., and Faloutsos, C., "On Power-Law Relationships of the Internet Topology," in *ACM SIGCOMM Computer Communication Review*, pp. 251–262, ACM, 1999.

[46] Filippini, R. and Silva, A., "A modeling framework for the resilience analysis of networked systems-of-systems based on functional dependencies," *Reliability Engineering & System Safety*, vol. 125, pp. 82–91, May 2014.

[47] Fisher, R., *The Design of Experiments*. Edinburgh: Oliver and Boyd, 6 ed., 1960.

[48] Francis, R. and Bekera, B., "A metric and frameworks for resilience analysis of engineered and infrastructure systems," *Reliability Engineering and System Safety*, vol. 121, pp. 90–103, 2014.

[49] Gómez-Gardeñes, J. and Moreno, Y., "From scale-free to Erdos-Rényi networks," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 73, no. 5, pp. 1–7, 2006.

[50] Gorod, A., Sauser, B., and Boardman, J., "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward," *IEEE Systems Journal*, vol. 2, pp. 484–499, Dec. 2008.

[51] Grant, T. J., Buizer, B. C., and Bertelink, R. J., "Vulnerability of C2 Networks to Attack: Measuring the topology of eleven Dutch Army C2 systems," in *16th International Command and Control Research and Technology Symposium*, (Quebec City, CA), 2011.

[52] Gross, T. and Blasius, B., "Adaptive coevolutionary networks: a review," *Journal of the Royal Society Interface*, vol. 5, pp. 259–71, Mar. 2008.

[53] Haimes, Y. Y., "On the definition of resilience in systems," *Risk Analysis*, vol. 29, no. 4, pp. 498–501, 2009.

[54] Heininger, C. and Walker, A., “New network provides ’digital guardian angel’ for Soldiers in Afghanistan,” 2013.

[55] Herrmann, H. J., Schneider, C. M., Moreira, A. a., Andrade Jr, J. S., and Havlin, S., “Onion-like network topology enhances robustness against malicious attacks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, pp. 1–27, Jan. 2011.

[56] Hoad, K., Robinson, S., and Davies, R., “Automating warm-up length estimation,” *Journal of the Operational Research Society*, vol. 61, pp. 1389–1403, 2010.

[57] Holland, J. P., “Engineered Resilient Systems (ERS) Overview,” 2013.

[58] Holme, P. and Kim, B., “Vertex overload breakdown in evolving networks,” *Physical Review E*, vol. 65, p. 066109, June 2002.

[59] Holme, P., Kim, B. J., Yoon, C. N., and Han, S. K., “Attack vulnerability of complex networks,” *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 65, p. 056109, May 2002.

[60] Jamshidi, M., “Introduction to system of systems,” in *Systems of Systems Engineering: Principles and Applications* (Jamshidi, M., ed.), ch. 1, Boca Raton, FL: Taylor & Francis, 2009.

[61] Jarvis, D. A., “A Methodology for Analyzing Complex Military Command and Control (C2) Networks,” in *10th International Command and Control Research and Technology Symposium*, 2005.

[62] Jeong, H., Mason, S. P., Barabási, a. L., and Oltvai, Z. N., “Lethality and centrality in protein networks.,” *Nature*, vol. 411, pp. 41–2, May 2001.

[63] Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N., and Barabási, a. L., “The large-scale organization of metabolic networks.,” *Nature*, vol. 407, pp. 651–4, Oct. 2000.

[64] Jönsson, P. and Eklundh, L., “TIMESAT - A program for analyzing time-series of satellite sensor data,” *Computers and Geosciences*, vol. 30, no. 8, pp. 833–845, 2004.

[65] Keating, C., Rogers, R., Unal, R., Dryer, D., Sousa-Poza, A., Safford, R., Peterson, W., and Rabadi, G., “System of Systems Engineering,” *Engineering Management Journal*, vol. 15, no. 3, pp. 36–45, 2003.

[66] Keating, C. B. and Katina, P. F., “Systems of systems engineering: prospects and challenges for the emerging field,” *International Journal of System of Systems Engineering*, vol. 2, no. 2/3, pp. 234–256, 2011.

[67] Kim, J. and Wilhelm, T., "What is a complex graph?," *Physica A: Statistical Mechanics and its Applications*, vol. 387, pp. 2637–2652, Apr. 2008.

[68] Kirk, R. E., "Experimental Design," in *The SAGE Handbook of Quantitative Methods in Psychology* (Millsap, R. E. and Maydeu-Olivares, A., eds.), ch. 2, pp. 23–45, SAGE, 2009.

[69] Law, A. M., *Simulation Modeling and Analysis.* New York, NY, USA: McGraw-Hill Education, 5th ed., 2015.

[70] Licouski, B. and Elliott, W. J., "Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and Recommendations," Tech. Rep. April, U.S.-Canada Power System Outage Task Force, 2004.

[71] Louzada, V. H. P., Daolio, F., Herrmann, H. J., and Tomassini, M., "Smart rewiring for network robustness," *Journal of Complex Networks*, vol. 1, pp. 150–159, Sept. 2013.

[72] Macal, C. M. and North, M. J., "Tutorial on agent-based modelling and simulation," *Journal of Simulation*, vol. 4, pp. 151–162, Sept. 2010.

[73] Madni, A. M. and Jackson, S., "Towards a Conceptual Framework for Resilience Engineering," *IEEE Systems Journal*, vol. 3, pp. 181–191, June 2009.

[74] Maier, M. W., "Architecting Principles for Systems-of-Systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.

[75] Marx, W. J., Mavris, D. N., and Schrage, D. P., "Integrating Design and Manufacturing for a High Speed Civil Transport Wing," in *ICAS/AIAA Aircraft Systems*, 1994.

[76] Minai, A. A., Braha, D., and Bar-Yam, Y., "Complex Engineered Systems: A New Paradigm," in *Complex Engineered Systems* (Braha, Dan and Minai, Ali A. and Bar-Yam, Y., ed.), ch. 1, Springer Berlin Heidelberg, 2006.

[77] Montgomery, D. C., *Design and Analysis of Experiments.* Wiley & Sons, Inc., 8 ed., 2013.

[78] Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M., *Response Surface Methodology.* Hoboken, NH: John Wiley & Sons, 3rd ed., 2009.

[79] National Research Council, "Network Science," tech. rep., National Academies Press, Washington, DC, 2005.

[80] Neches, R., "Engineered Resilient Systems (ERS)," 2011.

[81] Neches, R. and Madni, A. M., "Towards Affordably Adaptable and Effective Systems," *Systems Engineering*, vol. 16, no. 2, pp. 224–234, 2013.

[82] Newman, M., "Scientific collaboration networks.I. Network construction and fundamental results," *Physical Review E*, vol. 64, p. 016131, June 2001.

[83] Newman, M. E., "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, pp. 404–9, Jan. 2001.

[84] Newman, M. E. J., "Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality," *Physical Review E*, vol. 64, p. 016132, June 2001.

[85] Newman, M. E. J., "The structure and function of complex networks," *Society for Industrial and Applied Mathematics*, vol. 45, no. 2, pp. 167–256, 2003.

[86] Newman, M. E. J., "Coauthorship networks and patterns of scientific collaboration.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101 Suppl, pp. 5200–5, Apr. 2004.

[87] Newman, M. E. J., Forrest, S., and Balthrop, J., "Email networks and the spread of computer viruses," *Physical Review E*, vol. 66, p. 035101, Sept. 2002.

[88] Newman, M. E. J., Strogatz, S. H., and Watts, D. J., "Random graphs with arbitrary degree distributions and their applications," *Physical Review E*, vol. 64, p. 026118, July 2001.

[89] Newman, M., Barabási, A.-l., and Watts, D. J., *The Structure and Dynamics of Networks.* Princeton University Press, 2006.

[90] Office of the Deputy Under Secretary of Defense for Acquisition and Technology, S. and Engineering, S., *Systems Engineering Guide for Systems of Systems, Version 1.0.* Washington, DC: ODUSD(A&T)SSE, 2008.

[91] Orfanidis, S. J., *Introduction to Signal Processing*, vol. 20. Pearson Education, Inc., 2010.

[92] O'Rourke, T. D., "Critical Infrastructure, Interdependencies, and Resilience," *The Bridge: Linking Engineering and Society*, vol. 37, no. 1, pp. 22–9, 2007.

[93] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, vol. 1. New York, New York, USA: Cambridge University Press, 2nd ed., 1997.

[94] Reed, D. A., Kapur, K. C., and Christie, R. D., "Methodology for Assessing the Resilience of Networked Infrastructure," *IEEE Systems Journal*, vol. 3, no. 2, pp. 174–180, 2009.

[95] Savitzky, A. and Golay, M. J. E., "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[96] Schafer, R. W., "What is a savitzky-golay filter?," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[97] Schneider, C. M., Moreira, A. A., Andrade, J. S., Havlin, S., and Herrmann, H. J., "Mitigation of malicious attacks on networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 10, pp. 3838–3841, 2011.

[98] Schwartz, N., Cohen, R., Ben-Avraham, D., a. L. Barabási, and Havlin, S., "Percolation in directed scale-free networks," *Physical Review E*, vol. 66, p. 015104, July 2002.

[99] Schweitzer, F., Fagiolo, G., Sornette, D., Vega-Redondo, F., Vespignani, A., and White, D. R., "Economic networks: the new challenges.," *Science (New York, N.Y.)*, vol. 325, pp. 422–5, July 2009.

[100] Serrano, M. A. and Boguñá, M., "Percolation and Epidemic Thresholds in Clustered Networks," *Physical Review Letters*, vol. 97, p. 088701, Aug. 2006.

[101] Shenhar, A., "A new systems engineering taxonomy," in *Proceedings of the 4th International Symposium of the National Council on System Engineering*, National Council on System Engineering, 1994.

[102] Steinberg, D. M. and Hunter, W. G., "Experimental Design: review and comment," *Technometrics*, vol. 26, no. 2, pp. 71–97, 1984.

[103] Sterbenz, J. P. G., Çetinkaya, E. K., Hameed, M. a., Jabbar, A., Qian, S., and Rohrer, J. P., "Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation," *Telecommunication Systems*, vol. 52, pp. 705–736, Dec. 2013.

[104] Sterbenz, J. P., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., and Smith, P., "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, pp. 1245–1265, June 2010.

[105] Taleb, N. N. and Douady, R., "Mathematical definition, mapping, and detection of (anti)fragility," *Quantitative Finance*, vol. 13, no. 11, pp. 1677–1689, 2013.

[106] Tran, H. T., Domercant, J. C., and Mavris, D. N., "Evaluating the agility of adaptive command and control networks from a cyber complex adaptive systems perspective," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 2015.

[107] Tran, H. T., Domercant, J. C., and Mavris, D. N., "A System-of-Systems Approach for Assessing the Resilience of Reconfigurable Command and Control Networks," in *AIAA Infotech @ Aerospace*, (Kissimmee, FL), 2015.

[108] Turnquist, M. and Vugrin, E., "Design for resilience in infrastructure distribution networks," *Environment Systems & Decisions*, vol. 33, pp. 104–120, Jan. 2013.

[109] Uday, P. and Marais, K., "Exploiting Stand-in Redundancy to Improve Resilience in a System-of-Systems (SoS)," *Procedia Computer Science*, vol. 16, pp. 532–541, Jan. 2013.

[110] Vega-Redondo, F., "Network organizations," *Journal of Complex Networks*, vol. 1, pp. 72–82, Mar. 2013.

[111] Ventresca, M. and Aleman, D., "Network robustness versus multi-strategy sequential attack," *Journal of Complex Networks*, vol. 2, no. 2, 2014.

[112] Vugrin, E. D., Warren, D. E., Ehlen, M. A., and Camphouse, R. C., "A Framework for Assessing the Resilience of Infrastructure and Economic Systems," in *Sustainable and Resilient Critical Infrastructure Systems* (Gopalakrishnan, K. and Peeta, S., eds.), pp. 77–116, Springer Berlin Heidelberg, 2010.

[113] Watts, D. J. and Strogatz, S. H., "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–2, June 1998.

[114] White, K. P., "An Effective Truncation Heuristic for Bias Reduction in Simulation Output," *Simulation*, vol. 69, no. 6, pp. 323–334, 1997.

[115] White, K. P. and Robinson, S., "The problem of the initial transient (again), or why MSER works," *Journal of Simulation*, vol. 4, pp. 268–272, 2010.

[116] White, K. P. and Spratt, S. C., "A Comparison of Five Steady-state Truncation Heuristics for Simulation," in *Proceedings of the 2000 Winter Simulation Conference*, pp. 755–760, 2000.

[117] Wilensky, U., "NetLogo," 1999.

[118] Yates, F., "Sir Ronald Fisher and the design of experiments," *Biometrics*, vol. 20, no. 2, pp. 307–321, 1964.

[119] Zhao, K., Kumar, A., Harrison, T. P., and Yen, J., "Analyzing the Resilience of Complex Supply Network Topologies Against Random and Targeted Disruptions," *IEEE Systems Journal*, vol. 5, no. 1, pp. 28–39, 2011.

.