Received: 5 December 2019 | Revised: 15 May 2020 | Accepted: 23 June 2020

DOI: 10.1002/sys.21551

WILEY

REGULAR PAPER

A network perspective on assessing system architectures: Robustness to cascading failure

Matthew W. Potts[1,*] | Pia A. Sartor[1] | Angus Johnson[2] | Seth Bullock[3]

[1] Aerospace Department, The University of Bristol, Bristol, UK

[2] Thales Research, Technology and Innovation, Reading, UK

[3] Department of Computer Science, The University of Bristol, Bristol, UK

Correspondence
Matthew Potts, Aerospace Department, The University of Bristol, Bristol, Avonmouth, BS8 1TB, UK.
Email: matt.potts@bristol.ac.uk

*Matthew Potts, Pia Sartor, Angus Johnson and Seth Bullock contributed equally to this article.

Funding information
Engineering and Physical Sciences Research Council, Grant/Award Number: iCASE 16000139

Abstract

Despite a wealth of system architecture frameworks and methodologies available, approaches to evaluate the robustness and resiliency of architectures for complex systems or systems of systems are few in number. As a result, system architects may turn to graph-theoretic methods to assess architecture robustness and vulnerability to cascading failure. Here, we explore the application of such methods to the analysis of two real-world system architectures (a military communications system and a search and rescue system). Both architectures are found to be relatively robust to random vertex removal but more vulnerable to targeted vertex removal. Hardening strategies for limiting the extent of cascading failure are demonstrated to have varying degrees of effectiveness. However, in taking a network perspective on architecture robustness and susceptibility to cascade failure, we find several significant challenges that impede the straightforward use of graph-theoretic methods. Most fundamentally, the conceptualization of failure dynamics across heterogeneous architectural entities requires considerable further investigation.

KEYWORDS
SEE02 Complexity Science

1 | INTRODUCTION

Robustness and resilience are key considerations for system architects as they are often critical stakeholder concerns.[1–5] However, system architects seeking to ensure that their systems demonstrate robustness and resilience face an increasingly difficult task when those systems operate in the context of a System of Systems (SoS), given the potential for emergent behavior in diverse and often autonomous systems which may be interconnected and interdependent.[6–8] Furthermore, engineering in an SoS context may include conflicting or opaque SoS "authorities" which creates a challenge in identifying and managing influence in an SoS.[9–11]

While the resilience of a system is centered on the anticipation, survival, and recovery from internal and external threats,[12] robustness is the constituent term relating to the system's ability to remain unchanged in the face of some assault or change. The identification of system robustness can be critical, and the failure to do so can have serious, life-threatening consequences. High-profile system failures have promoted the importance of a system's robustness. For instance, the partial collapse of the Ronan Point apartment tower in 1968 in London, had consequences that far outweigh the initiating damage sustained. A gas explosion caused damage to load-bearing walls, which resulted in the collapse of one entire corner of the building, killing four people and injuring 17.[13] While building codes have since been updated to ensure improved *robustness* of built structures themselves,[13] the robustness and resilience of the wider systems within which such built structures

Abbreviations: SoS, System of Systems; SoI, System of Interest

are located is less well assured, as exemplified by the direct and indirect impact of the 2017 Grenfell Tower fire in London. The resulting Public Inquiry "Phase 1 Report" details that the principal reason the fire spread was the aluminium composite cladding filled with plastic used on the building exterior, and that the London Fire Brigade suffered "...significant systemic and operational failings revealed by the evidence."[14] A further example that suggests further work is required to promote consideration of system robustness within the systems engineering community can be found from the recent (2017) crash of a Watchkeeper Unmanned Air Vehicle into the sea in Wales, UK. The subsequent Service Inquiry by the UK Defence Safety Authority concluded that the incident (an aerodynamic stall) was a result of pitot tube blockages leading to inaccurate air speed reporting, citing a "lack of system robustness testing" as an organizational influence that was a contributing factor.[15]

While systems can be designed with resilience engineering principles to try to minimize the likelihood and severity of such incidents, eg, physical or functional redundancy, fail-safe principles, layered defense, and conducting extensive failure effects mode analysis, it seems that for complex systems or SoS, further effort is still required to ensure system resilience.[12,16] Despite several architecture frameworks utilized by industry seeking to establish common practices for analyzing architectures within a particular domain,[17–21] there is only a relatively sparse literature concerning approaches to evaluating architecture robustness and resilience.[16,22–26]

Several previous studies have analyzed network representations of system architectures, sometimes described as *social network analysis*, to support their evaluation.[27–32] This paper builds on previous work,[33–35] in seeking to explore the robustness and susceptibility to cascading failure of candidate system architectures from a network perspective, in order to assess the extent to which taking a network perspective on SoS architectures can assist architecture evaluation by helping determine whether one candidate architecture is more robust or resilient than another. Such an approach may address notions of dynamic complexity, such as: "what is the effect of the removal of some of these entities, whether in the form of a single failure or a cascade of failures?"

In this study, we make use of two real-world enterprise architectures originally created and validated by Thales, and chosen as representative of real-world SoS architectures featuring a diversity of entities and relationships. For further details about the use cases, the interested reader is directed to Ref. 33. The first use case is a Search and Rescue (SAR) NATO Architecture Framework (NAF)-based architecture, developed by Thales in order to inform systems architecture training and help the development of NAF v4.[36] The SAR architecture was produced in a common commercial enterprise architecture software package, created by following the NAF v4 eight architecting stages. The architecture includes the architecture products corresponding to NAF viewpoints described in NAF v4 (corresponding to the Subjects of Concern and Aspects of Concern in NAF v4 parlance)[36] and also includes representations of the high-level operational concept and the use case scenario.

The second use case is a tactical military communications enterprise architecture (MComms), created in accordance with the Ministry of Defence Architecture Framework (MODAF), to enable Thales and the customer to have a shared understanding of the complex environment within which a tactical military communications solution would have to interoperate. This architecture was created with the same commonly used commercial enterprise architecture software package as the SAR architecture. The architecture was created as part of concept development work while developing a bid for a particular client organization. The architecture includes the architecture products corresponding to MODAF viewpoints for the "as-is" architecture of the client's solutions, along with architecture products corresponding to the "to-be" architecture of the proposed solution. As is common practice, not all architecture viewpoints in MODAF had an architecture product created as the selection of viewpoints is tailored to support specific organizational objectives (in this case, supporting concept development and the refinement of bid response activity).

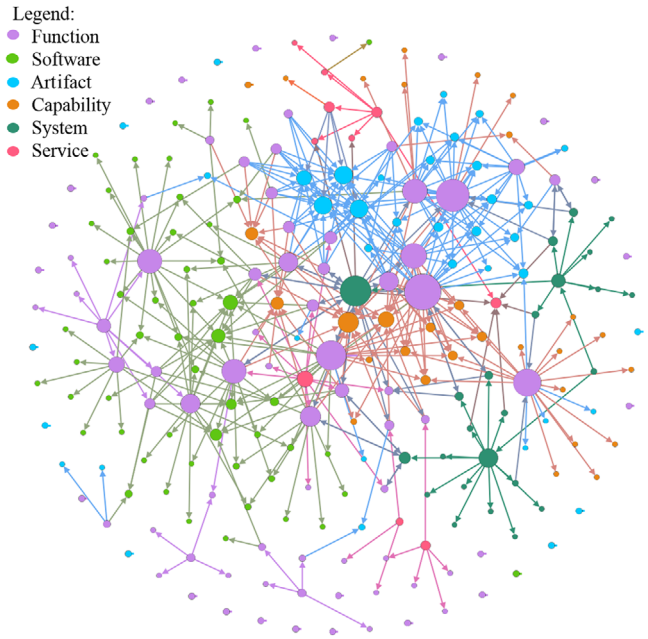


FIGURE 1 Network diagram representing the Military Communications (MComms) use case. A vertex's size is proportional to its degree. A vertex's color represents entity type. Networks visualized using Gephi, an open source software for exploring and manipulating networks. Figure reproduced from Ref. 33

Here, we follow previous studies[33–35] in representing each architecture as a network of vertices (representing Capabilities, Services, and Logical Nodes for the SAR use case, and representing Systems, Services, Functions, Components, Software. and Capability Configurations for the MComms use case) connected by edges (representing dependencies and relationships between vertices). A network diagram of each architecture is shown in Figures 1 and 2.

This paper makes three primary contributions to the emerging literature describing the application of networks science tools to system architecting. First, we assess the robustness of two real-world architectures to vertex removal using two standard centrality metrics as measures of network viability. Second, we evaluate each architecture's susceptibility to cascading failure using a simple threshold model,

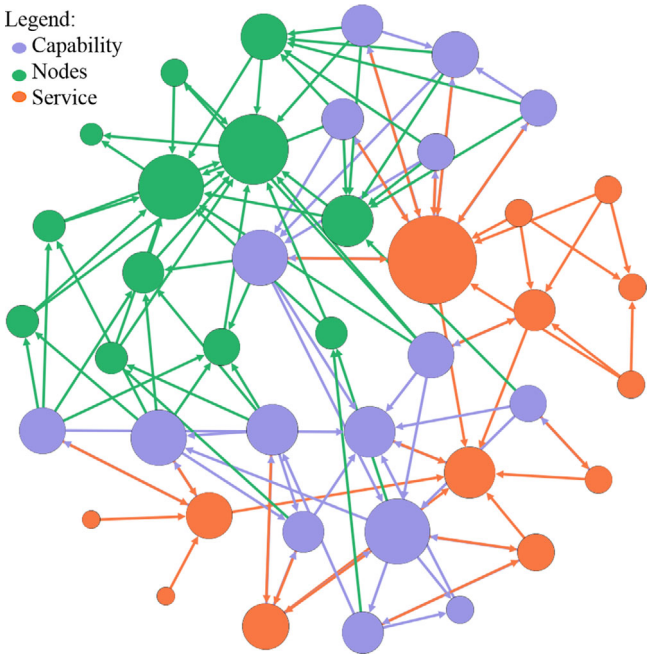


FIGURE 2 Network diagram representing the Search and Rescue (SAR) use case. A vertex's size is proportional to its degree. A vertex's color represents entity type. Networks visualized using Gephi, an open source software for exploring and manipulating networks. Figure reproduced from Ref. 33

before exploring the effects of two hardening strategies for limiting the extent of this cascading failure. Finally, we assess the strengths and weaknesses of networks analysis techniques for the evaluation of system architectures in general, highlighting the challenges that currently limit their utility, and recommend guiding principles for architecture robustness evaluation in light of these challenges.

The next section briefly reviews relevant literature and defines the terms robustness and resilience for the purposes of this paper. A synthetic motivating example is then introduced, and revisited later, to provide context for the approach taken in this paper. The theory supporting the assessment of network robustness is presented. The methodology used to assess architecture response to perturbation and susceptibility to cascades is then detailed before the results of the assessments are presented. The discussion then turns to the challenges of adopting a network perspective to evaluate the robustness of a complex SoS architecture, highlighting the areas where care must be taken, before conclusions are drawn.

1.1 | Literature review

Some authors argue that an SoS is not fit for purpose when it cannot transfer materials, energy, or information in a timely, correct, cost-effective way and encourage designers to consider these transfers as areas to focus on, whether in terms of opportunity enhancement or risk reduction.[22] However, for a complex SoS with considerable scope, how can an organization refine their focus to something tractable? Furthermore, what makes one SoS configuration more robust or resilient than another?

Designers are concerned with the ability of their designed systems to cope with perturbation and change. Several related concepts are often invoked in discussing this issue. System robustness, for example, can be characterized as the ability of a system to withstand perturbation—its ability to remain unchanged in the face of some assault or change. Resilience, by contrast, is the ability of a system to recover from some degree of failure. However, this relatively clean distinction is difficult to maintain in practice,[23,26] where robustly resisting perturbation may always involve some microscale reorganization or refreshment and recovering from a failure may or may not return the system to precisely its original functional state. To some extent the character of a system's response to perturbation is observer-relative, depending on subjective factors such as the timescale of interest, the granularity of the analysis, etc.[37,38]

Here, we will use the term *resiliency* to include the three most important aspects of a system's ability to cope with perturbation; robustness, how much damage a system can withstand before it fails to function; recovery, the ability of the system to repair or recuperate within some resource constraints; and adaptability, the ability of the system to effectively change over time so that its likelihood of success does not decrease.[37] Similarly, resilience has been defined as "the ability to prepare and plan for, *absorb* or mitigate, *recover* from, or more successfully *adapt* to actual or potential adverse events" (emphasis added).[9,12,39] The same aspects of resilience are presented by Madni and Jackson, and Goerger et al, suggesting notions of anticipation (prepare and plan for, avoid), resistance (absorb, withstand), respond and adapt to, and recover from.[12,40] There are, however, a wealth of qualities with which to describe resiliency, and the interested reader is directed to Ref. 16 for a more detailed review. While other aspects of resilience may also include: quality, agility, repairability, extensibility, flexibility, and versatility, here we focus on the three aforementioned aspects of robustness, recovery, and adaptability.[16,41] For this paper, resiliency can be described as: "a system's ability to adjust its activity to retain its basic functionality when errors, failures, and environmental changes occur."[42]

While system resilience has been argued to be an emergent property that cannot be measured,[9] evaluating system robustness may provide a quantifiable means to support resilience engineering. The Engineering Systems community argues that the *architecture* of a system is a key contributor to functional behavior and desirable properties, such as resilience and robustness.[5,6] Approaches to evaluate the resiliency of a system, early in its lifecycle (ie, without probabilistic calculations of component reliability) are relatively sparse.

Design strategies for improved system robustness have been proposed, such as "Relax a constraint limit on an uncoupled control factor" or "Create two distinct operating modes for two different demand conditions" and may aid the design of predominately mechanical systems (ie, a paper feeder in a printer), but may not be straightforward to implement for complex engineered systems (ie, an air-traffic management system).[43] Similarly, "design principles" for resilient enterprise information systems have been proposed by Zhang and Lin, based on

derived axioms from literature in ecology, which may usefully prompt system designers, but whose utility for organizations involved in the design of complex SoS may be limited (eg, see "A resilient system should be designed to have a certain degree of redundancy, preferably functional redundancy. The more redundancy the system has, the higher the degree of resilience.").[44]

The "SoS Architecting with Ilities (SAI)" method encourages early phases of SoS design to explicitly include design options: so-called change options that change the design of the SoS in order to *respond* to a perturbation, or so-called resistance options that *resist* perturbation-imposed changes in the design of the SoS.[25] Deliberate design emphasis on ensuring robustness and resiliency seems appropriate. However, the qualitative generation, evaluation, analysis, and trade-off of a diverse and ever growing set of "ilities" and "options" for architecture alternatives places a considerable resource burden on design organizations. Similarly, the following techniques are encouraged to ensure a design organization arrives at a resilient design: "Developing applicable and realizable resilience heuristics—to inform and guide resilient system design" and "Developing appropriate resilience metrics—to evaluate candidate resilience strategies."[40] However, guidance for resilience metrics and resilience strategies is currently limited in practice.

Due to the increasing scale and complexity of modern engineered systems, some have turned to a network representation of a system architecture to gain insights into various aspects such as, inter alia, architecture complexity,[31,45–47] modularity or community structure,[24,31,33] important architectural entities,[24,28,29,31,33,48–50] architecture topology,[24,28,29,31,33,49,50] etc, although there is little consensus on the most suitable definitions or measures of these properties (eg, what makes degree a more suitable measure of influence than closeness, betweenness or eigenvector centrality, or characteristic path length?).[33] Within the engineering design community, a *robust physical design* can be interpreted as a design which minimizes interfaces between subsystems and maximizes interactions within subsystems, using a network representation of a physical system architecture (requirements decomposed into subsystems, decomposed into components) and clustering algorithms to determine the extent to which this design principle is applied.[51] Within the systems engineering community, such approaches are perhaps more likely to be termed modular designs as opposed to robust designs.[33]

Furthermore, authors use different approaches to model a system architecture as a network, with the resultant models sensitive to fine-grained modeling assumptions of what should be included and what should be omitted from a model and this sensitivity is a known challenge for those wishing to utilize a network perspective or a *social network analysis* to evaluate a system architecture (eg, some model a *physical* system architecture, or a *functional* architecture, or a *logical* architecture, while others model a design structure matrix, others model multiple domain matrices, while yet others select a subset of a system architecture).[33]

Some authors have used a network model of an SoS to examine failure propagation in order to evaluate the effects of disruptions, eg, the effect of local airport disruptions on national commercial air travel, or network models of SoS functions to examine critical functional dependencies between constituent systems.[16] Often these approaches require considerable complexity in the real SoS to be abstracted away in order to define a mathematical model of a network; eg, creating "scale-free" or "exponential" network topologies, upon which resiliency can be evaluated by considering the impact of vertex removal on message propagation through the network. Again, however, in real-world SoS architectures such topologies are not necessarily present and there is an assumption of architectural entity homogeneity (architectural entities are understood as differing only in so far as they occupy a different location in the network).[32] These network models can struggle to cope with the heterogeneity of a complex SoS, even when they employ more sophisticated "multilayer" network models, or adopt a "narrower" modeling choice in the case of examining functional dependencies.[16,32]

The current paper seeks to utilize a network perspective (or *social network analysis*) on failure propagation and the identification of influential entities in terms of their contribution to robustness and failure cascades, but applied to complex SoS *enterprise architectures* modeled as networks. In a departure from other methods, we start with a network model of a complex real-world SoS created directly from the enterprise architecture.[33] Furthermore, we evaluate the robustness of an architecture in terms of its susceptibility to vertex removal measured in terms of the impact on average network centrality as a proxy for architecture viability. We also evaluate architecture robustness in terms of susceptibility to cascading failure using a load shedding model. This approach may be useful for system architects undertaking trade-off studies when evaluating competing architectures. Furthermore, the approach may help ensure subsequent designs are more robust and resilient by adopting protection strategies for the entities which, when removed, have the greatest effect on architecture viability or which are the greatest contributors to cascading failure. While this approach offers some interesting insights, there are several significant conceptual hurdles that we wish to highlight so that system architects attempting to make use of these techniques can do so with "eyes open" to the challenges they face.

1.2 | Motivating example

To help provide context for this work, consider a large commercial organization that operates primarily as a systems integrator and currently designs, develops, delivers, and operates Command, Control, Communications, Computers, and Intelligence (C4I) systems in a maritime SAR setting. The organization has recently developed a novel, proprietary sensor that provides a new capability to an existing customer, meeting a stated need expressed in this customer's Invitation to Tender (ITT). The customer has requested a new C4I system with the new sensor integrated, and the organization has completed an initial design to meet the customer requirements. The systems engineering manager notes that their system operates within a broader, complex, maritime SAR SoS with challenges of autonomy, diversity, operational, and managerial independence, among other

challenges, and instigates the creation of an Enterprise Architecture, using a common Architecture Framework to guide their activity.

The customer and supplying organization may both be concerned with the robustness of the architecture; is one candidate architecture more robust than another, are some architectural entities more important in the architecture in the sense that their removal impacts overall SoS effectiveness (whether the loss of a single entity is considered or the triggering of a cascade through the architecture), can hardening or protection strategies be put in place to improve architecture robustness, and to aid an understanding of the overall SoS by considering *where responsibility* for these important entities resides. While organizations likely have processes and guidelines to support architecture evaluation,[3] and they may have approaches to promote design principles that encourage resilience,[25,40,43] they may not have enough information early in the system lifecycle to effectively utilize these (eg, a lack of data to support Failure Mode and Effects Analysis (FMEA)), or they may have an enterprise architecture that is more diverse than a *product* architecture (eg, approaches that are based on design structure matrices[24,48]), and more diverse than approaches that treat an entire SoS as a network.[28,29,31]

System architects within an organization can instead represent their *enterprise architecture* as a network, and ask if one architecture is more robust to architectural entity changes or removal than another, and is therefore more desirable than another? For one particular architecture, which architectural entities are important in the sense that their removal affects network viability, or triggers cascading failure, ergo affecting SoS architecture effectiveness? In particular for a complex SoS, where do these entities exist: are they part of a prime organization's remit of control or are they external and thus a source of technical or operational risk?[11,46] This research attempts to provide system architects with an approach to support such evaluations.

2 | APPROACHES

This section introduces and defines graph-theoretic approaches to exploring network robustness. The selected approaches are later applied to evaluate the robustness of network representations of complex system architectures.

2.1 | Network perturbation

Large complex networks, such as the World Wide Web, the Internet, metabolic networks, etc, can be considered to rely on their continued connectivity for their effective operation. In such networks, the removal of vertices (nodes) interferes with paths between pairs of vertices until the network becomes largely disconnected and unable to effectively function.[52,53] Some of these networks have strongly skewed degree distributions, with many low-degree nodes and very few high-degree nodes, or even have degree distributions that are approximated by power laws, eg, some technological or social media networks. As a consequence, they tend to exhibit high robustness to the random removal of vertices but have significant vulnerability to the targeted removal of the highest degree vertices.[54,55] Similar results have been shown for social and biological networks, eg, email networks[56] and metabolic networks.[57] If systems architects could utilize similar analyses they could provide an evaluation of a complex SoS architecture's robustness and consider potential recovery and adaption strategies to ensure overall resiliency, considerations which may influence the selection of a candidate architecture or inform design decisions.

There are some challenges in applying such approaches to complex SoS architectures, however, which are introduced here but revisited in more detail in Section 5. The first is that in complex network research the removal of a vertex has a very natural meaning in context. For computer networks, the loss of a vertex models the failure of a system component. In epidemiology, the loss of a vertex models the death of an individual or them gaining immunity to the disease being modeled. In graphical models of complex SoS architecture, however, finding a meaningful abstraction that corresponds to vertex removal is difficult, given the heterogeneity of system entities modeled: people, teams, services, physical systems, software, and communications carriers. Solutions to this challenge include considering more homogenous architectures as the starting point: for example, to only model constituent architecture viewpoints, such as parts of a logical system architectures, where vertex removal would correspond to physical component failure. However, here we are interested in evaluations of enterprise architectures, not evaluations of only physical or logical architectures. Similarly, another solution would be to only consider removing vertices that correspond more clearly to system entities failing; eg, only allowing physical entities represented in an enterprise to be removed, to correspond to a less diverse notion of "failure." A potential solution, and the one used in this manuscript, is that the removal of a vertex is taken to correspond to the failure of a system entity to cope with some change. As the graphical model represents a diverse range of system entities and their interactions, the notion of change employed here must also capture some diversity: in the context of a complex SoS architecture, a relevant change might correspond to an alteration to the operational status of a system (a system becomes unserviceable for example), a change in environmental conditions, a change in doctrine or concepts for systems or the overall SoS, etc. In this way, the removal of a vertex corresponds to a change, internal or external, that effects the constituent entities of the SoS to such an extent that the entity is no longer effective.

One approach to characterizing robustness, is to consider the fraction of vertices in the largest component of the network after it has suffered a perturbation of some kind.[38,58] The larger the size of this component, the more robust to perturbation the system has proved to be. This makes sense for, eg, a communication network, where vertices within the same connected component can be assumed to communicate with one another. Again, however, the interpretation for a complex SoS architecture is not always clear. See Section 5.

Relatedly, determining a suitable metric for evaluating system effectiveness is also conceptually challenging. For instance, while, for some systems (eg, software systems modeled as networks of subroutines and relationships between them, or product development

processes modeled as networks of tasks and information flows between them),[31] changes in the average path length (characteristic path length) of a graph representation may make sense as a proxy for system connectivity and thus effectiveness, it is not clear that this metric would be a suitable measure of effectiveness for the architectures being considered here. Similarly, in a social network of relationships between terrorists in "dark networks," different centrality measures identify different individuals as most influential as different centrality measures make different assumptions about what constitutes influence or importance.[59,60] There is an inherent challenge in finding a suitable network measure to represent architecture effectiveness despite a range of arguably suitable candidates (eg, average characteristic path length, average degree centrality, average betweenness or closeness centrality, etc)[31,33] and we return to this point in Section 5.

If an acceptable effectiveness measure can be identified, a further challenge involves determining how to represent the regime of perturbations that the architecture is subjected to during resiliency analyses. One simple aspect of this challenge is determining whether vertices affected by a perturbation should be selected at random or in a targeted manner, eg, targeting high-degree vertices first. In the latter case, is the targeted property (eg, degree) recalculated after each vertex is removed or not? While some studies have recalculated the target measure,[54] while others have not ([56,57],) the effects of these relatively fine-grained decisions regarding methods are not known fully at this stage. More problematically, how can a perturbation regime be constructed for a model that reflects the fact that challenges to a real-world system are typically neither random nor targeted, but, rather, tend to be semistructured as a result of common cause influences such as spatial or environmental effects, or resourcing problems, and as a consequence have a somewhat correlated impact on architecture components.

2.2 | Cascading failure

Given the interconnectedness of a complex SoS it is appropriate to assess the potential for "avalanches" of failure that spread across an architecture when perturbed. Models from epidemiology (eg, Susceptible Infectious Removed (SIR) or Susceptible Infectious Susceptible (SIS) models[52,61]) are used to assess the spread of disease over a network (or the spread of computer viruses over computer networks[56]). Although such models have been widely studied, they become less tractable, and hence less suitable, for highly heterogeneous systems such as the SoS architectures being considered here.

Instead, an approach that explores cascading changes in terms of "load shedding" could be more suitable. In electrical power network models, the removal of a vertex could correspond to the failure of a component causing its electrical load to be passed on to its neighbors. The same idea can be considered for a complex SoS architecture, where the removal of a vertex corresponds to some change that has rendered that entity ineffective. The neighboring entities (in the graphical model) are then required to cope with the demand originally placed on the entity, or the lack of support from that entity in carrying out their own function.

The simplest model of such dynamics is provided by Watts;[62] a vertex, i, fails if a given fraction, ϕ_i, of its neighbors have failed. By allowing different nodes to be assigned different threshold values, the fact that different entities in an SoS will have different inherent capacity to cope with change can be reflected. Despite its simplicity, the model captures important features of complex systems: the role of local dependencies, structure, connectivity, and heterogeneity (of elements and of failure thresholds). This model can be used to explore the effects of an abstract notion of "change" on the SoS architecture in order to assess its vulnerability to change cascades.

Again, when trying to exploit network science insights to assess complex SoS architecture resiliency there are significant challenges which are introduced here but revisited in Section 5. Constructing accurate models of cascades may require knowledge of the failure cascade processes that would take place in the actual SoS. For example, would the failure of a communications bearer result in the failure of the systems that rely on that bearer, and would this lead to the failure of entire services and capabilities or would the systems simply switch to another communication bearer to halt the cascade. How long would such a switch take to occur? Models of how cascades propagate through a complex SoS are difficult to formulate unless a posteriori knowledge of the process on the network is provided. Given that the architecture assessment processes that are the focus of this paper occur in the early design phases of system lifecycles, possession of this knowledge is unlikely, although we can imagine the evolving nature of real-world SoS deployments and the increased focus on Model-Based Systems Engineering meaning that in some scenarios such knowledge is available.[63] Consequently, since developing detailed models of dynamic change processes for an SoS architecture may not be feasible at an early or predesign stage, the focus for a system architect may perhaps best be placed on understanding the general susceptibility and impact of cascading changes for a class of architecture, in order to support architecture selection decisions or inform system designs with mitigation strategies in mind.

3 | METHODOLOGY

For each of the two real-world architectures, an MComms and a SAR, a subset of the entire architecture is modeled as a directed, unweighted graph where different types of architecture entities are represented as nodes (termed vertices) connected by links (termed edges) representing different types of interdependencies or relationships. A network diagram of each architecture is shown in Figures 1 and 2. Architecture views used to create the models were selected as representative of the primary areas of importance for the architectures. The MComms architecture describes the challenge of enabling effective tactical communications between soldiers, where the overall ability for a soldier to be able to communicate effectively with a range of other actors in adverse environmental conditions over a large and contested geographical region depends on the ability of the diverse systems

and agents that make up the SoS. The SAR use case is concerned with the overall ability to rapidly locate and recover distressed vessels in changing and adverse environmental conditions and depends on the ability of the diverse constituent systems, acting as autonomous agents, to coordinate in order to fulfill the capability demand that none can achieve alone. For more detail about the use case networks, refer to Refs. 33–35.

We assess the *robustness* of each of two use case architectures to successive vertex removal in three different ways (random, targeted, and live targeted). To measure network viability as vertices are removed, we use two different centrality metrics (the average closeness centrality and the average betweenness centrality). We also employ closeness centrality and betweenness centrality as measures of which architectural entities are important in the architecture. Closeness centrality identifies those entities that are (geometrically) closest to other architecture entities and betweenness centrality likewise identifies those entities that enable communication between many entities. The *average* centrality measure of the overall architecture is a proxy for the overall cohesiveness or effectiveness of the architecture. Changes in average centrality is therefore a reflection of the architecture's robustness as a result of some change. There are, however, challenges and limitations in using these metrics, eg, underpinning assumptions of network flows occurring over shortest paths for betweenness centrality or the handling of paths existing in *directed* graphs as considered here for closeness centrality, and we will return to these challenges in Section 5.[33] There is no consensus on the most suitable network measure of importance or influence in a network; alternative approaches include eigenvector centrality and characteristic path length.[31,33] However, we make use of closeness centrality and betweenness centrality as they are commonly used in the networks science literature, and a central aim of this paper is to consider the extent to which standard networks science techniques can be usefully employed in the analysis of systems architectures.

We perturb each directed graph to an increasing extent by repeatedly removing a chosen vertex and its associated edges and then recalculating the average centrality of all remaining nodes in the perturbed network. Vertex removal is repeated N times, until no vertices remain. Rather than removing vertices at random, a variant of this procedure selects the highest-degree vertex for removal at each step, either using the original degree values associated with the unperturbed network (targeted perturbation), or using the live degree values associated with the increasingly perturbed network (live targeted perturbation).

We evaluate the susceptibility to *cascading failure* of each architecture using a simple threshold model originally presented by Watts.[62] Each vertex, i, in the directed graph, G, is initially labeled "effective" and is given a threshold value ϕ_i, specifying the fraction of that vertex's neighbors that must be in state "ineffective" before it is rendered ineffective itself. This threshold value is either a fixed uniform value for all nodes in G, or is drawn from a uniform distribution over a given range, or a Poisson distribution with a given mean (divided by 10 to provide a value between 0.1 and 1.0, rejecting any threshold values greater than 1.0). The cascade is triggered by changing the state of a vertex from effective to ineffective. Each vertex then updates their own state, changing from effective to ineffective if sufficient neighbors have become ineffective. The cascade runs until no further changes in state occur. One replicate of the cascading failure algorithm is shown as pseudo-code in Figure 3.

```
Input: Direct Graph (G), Threshold Function (F), Initial Target Vertex (v)
Output: Total No. of Vertices Rendered Ineffective (K)
for each vertex, i, in G do
    i_state ← 'Effective'
    φ_i ← F(i)
end for
v_state ← 'Ineffective'
K ← 1
Cascading ← True
while Cascading == True do
    Cascading ← False
    for each vertex, i, in G do
        R = ineffective_neighbors(i)/total_neighbors(i)
        if i_state == 'Effective' and R > φ_i then
            i_state ← 'Ineffective'
            K ← K+1
            Cascading ← True
        end if
    end for
end while
return K
```

FIGURE 3 One replicate of the cascading failure algorithm shown as pseudo-code

For each allocation of threshold values to a network, N cascades are modeled, each triggered by initially perturbing one of the network's N vertices. For each network and each method of randomly allocating threshold values, 24 independent replicates are simulated. This allows analysis of the degree to which a network is vulnerable to cascades in general, but also the degree to which particular nodes in the network are more liable to generate significant cascades.

4 | RESULTS

4.1 | Perturbations

Removing the most connected vertices from a network representing a complex SoS architecture has a significant impact on centrality measures, as shown in Figures 4 and 5. For both architectures, the effect of removing even just a small fraction of the highest degree vertices (recalculating degree as the networks become increasingly perturbed) has a significant effect on the average Closeness Centrality and average Betweenness Centrality. Removing vertices at random produces a more gradual reduction in average centrality measures. These results are to be expected, however, since removing higher-degree vertices will tend to increase the distance between the remaining vertices. The result implies that system architects concerned with the robustness of

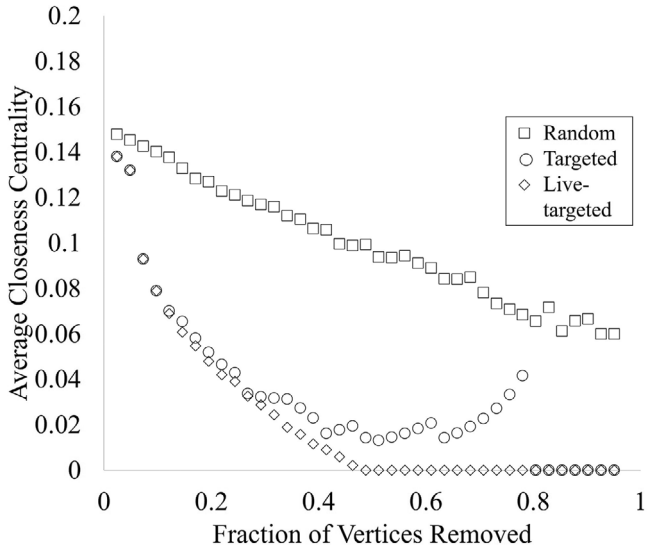


FIGURE 4 Average Closeness Centrality versus the fraction of SAR use case vertices removed at random (squares), removed in order of original degree value (circles), and removed in order of current perturbed degree value (diamonds)

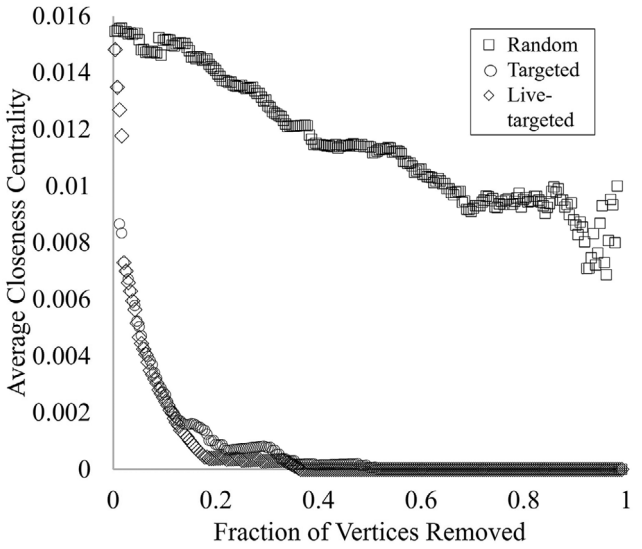


FIGURE 5 Average Closeness Centrality versus the fraction of MComms use case vertices removed at random (squares), removed in order of original degree value (circles), and removed in order of current perturbed degree value (diamonds)

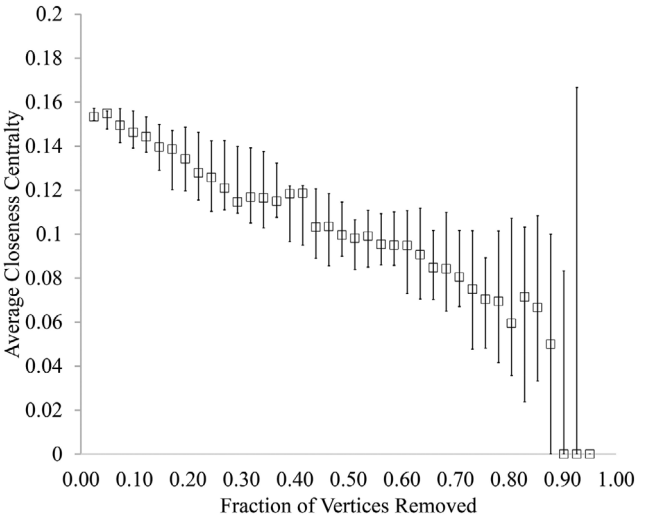


FIGURE 6 Variance in average Closeness Centrality across 25 simulation runs (shown as median and interquartile range (IQR) as vertices are removed at random from the SAR network

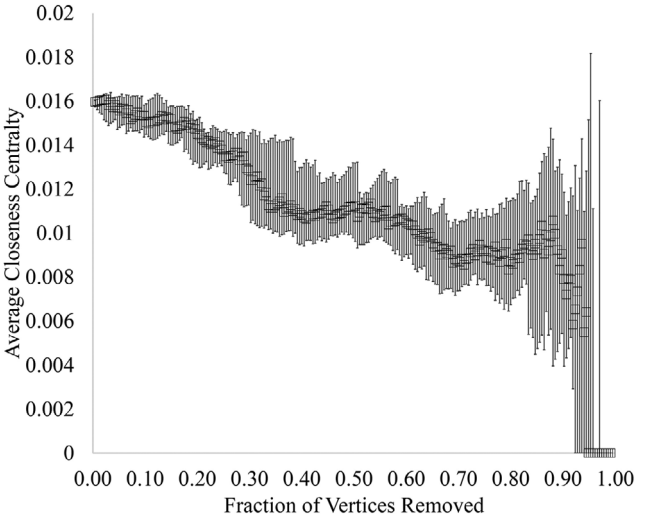


FIGURE 7 Variance in average Closeness Centrality across 25 simulation runs (shown as median and interquartile range [IQR]) as vertices are removed at random from the MComms network

these architectures should pay particular attention to the architectural entities that are particularly well connected, as their removal has a significant impact on the overall connectivity of the architecture, which under the modeling assumptions here relates to the effectiveness of the SoS. A more robust architectural pattern for both use cases would be a less hierarchical architecture with a greater overall density. However, the suitability of such a design change for the architectures would require further, contextually grounded, analysis.

Targeting nodes in order of their original degree in the unperturbed network may sometimes increase average Closeness Centrality in the perturbed network for both architectures. Positive assortativity ensures that removing high degree nodes tends to reduce the degree of other high degree nodes that become effectively isolated as the original highest degree vertices are removed, thus resulting in their removal having a less significant negative effect (or even a positive effect) on average closeness of each vertex.

Care should be taken when using an averaged topological property to make assertions about the robustness of an architecture, however, as the average value over a number of simulations may hide a large variance in individual simulation results, as shown in Figures 6 and 7. This variation in average Closeness Centrality as an architecture is perturbed can be seen in individual simulation runs (Figures 8 and 9). Snapshots of individual simulation runs show that as vertices

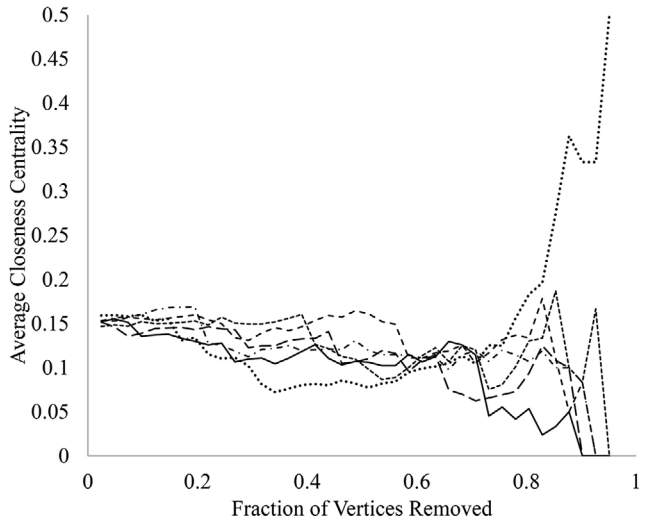


FIGURE 8 Variation in average Closeness Centrality across six simulation runs as vertices are removed at random from the SAR network

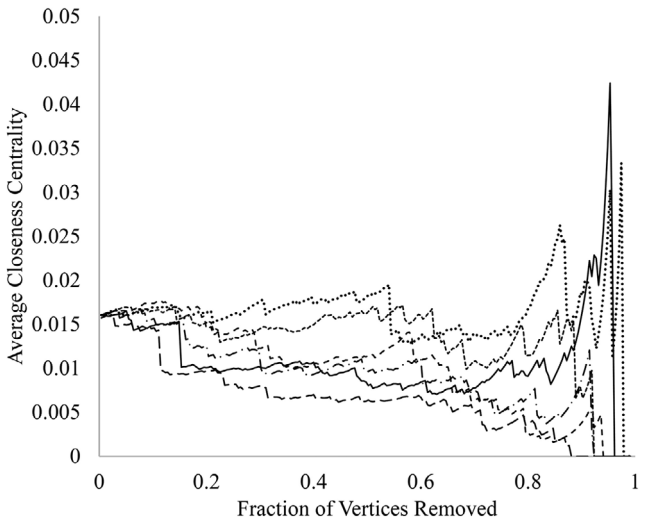


FIGURE 9 Variation in average Closeness Centrality across six simulation runs as vertices are removed at random from the MComms network

are removed average Closeness Centrality can increase, potentially counterindicating the suitability of topological measures of significance as proxies for architecture effectiveness. Increases in *average* closeness centrality correspond to the removal of vertices that are not able to be reached in both directions by all vertices, demonstrating some of the pitfalls of using network science measures of vertex significance such as closeness centrality or averages of these measures over the entire network. For the kind of enterprise architectures considered here which are directed and largely hierarchical and acyclic networks, there is no guarantee that each vertex can reach every other in both directions. While some approaches examined in the literature get around this issue by developing more abstract network models (ie, modeling away the *directionality* of relationships or dependencies), for those architectures where directionality is important, additional care must be taken in selecting suitable measures of architecture effectiveness. An alternative solution would be to use the average harmonic closeness centrality as a proxy instead of closeness centrality.[33] However, as we will see in Section 5, despite the choice of network measures used, there remain significant hurdles to overcome.

Correlations between the significance of a vertex and the impact on the network of its removal were calculated using various measures of significance and impact were examined to see if it is possible to predict which vertices cause large changes in architecture topology. These correlations were examined for both the original measures calculated from the unperturbed network (Table 1) and for the "live" measures recalculated as the network was perturbed (Table 2).

The lack of significant correlations shown in Table 1 suggests that there is little ability to predict network-level changes in topological measures of significance by considering vertex-level measures of significance calculated from the unperturbed architecture. Such correlations are significant for predicting Harmonic Closeness Centrality from the recalculated topological measures of vertex significance, but not for average Closeness Centrality or average Betweenness Centrality (Table 2). The implication of this result is that as a system architecture is perturbed, system architects cannot look to individual architectural entities to try and predict the impact of further degradation. Thus, while the *initial* connectivity of an architectural entity can serve as an identifier of importance to overall architecture effectiveness, this is not the case as the architecture is perturbed.

4.2 | Cascades

The cascading failure model seeks to determine the role of local dependencies and the structure of the architecture in the potential for a failure to cascade throughout the architecture, triggered by the failure of a single entity. The simulation explores the extent to which an SoS architecture is robust to cascading failures in a stochastic manner.

For both use cases, the results of cascading failure simulations are shown as histograms of cascade size (the percentage of ineffective vertices at the end of a cascade) alongside cumulative distributions of cascade size (Figures 10, 11, 12).

Snapshots of individual results of cascading failure simulation across the networks are shown in Figures 13 and 14 for the SAR use case and MComms use case, respectively, where vertices that have failed are shown in black and edges are colored by the color of their source vertex, based on initial conditions in which a threshold value of 0.2 is applied to all vertices in each network. For the SAR use case, the original vertex that fails corresponds to a "Search Node" (an operational maritime search asset) in the architecture and the failure cascade halts after two iterations with six vertices failed in total, affecting the search service and search capability. For the MComms use case, the original vertex that fails corresponds to a "system" (a military communications system deployed in the wider SoS) in the architecture and the failure cascade halts after eight iterations with 27 vertices failed in total, affecting other systems, services, one function and one operational capability.

TABLE 1 Correlations (R^2) between the original property of a vertex and the change in an architecture-level topological property caused by the vertex's removal

Network	Property of vertex removed	Change in average Closeness Centrality	Change in average Betweenness Centrality	Change in average Harmonic Centrality
SAR	Degree	0.084***	0.002	0.370***
MComms	Degree	0.007***	0.012***	0.393***
SAR	Closeness Centrality	0.003	0.002	0.074***
MComms	Closeness Centrality	0.005***	0.018***	0.398***
SAR	Betweenness Centrality	0.084***	0.010**	0.361***
MComms	Betweenness Centrality	0.004***	0.028***	0.512***
SAR	Harmonic Centrality	0.035***	0.002	0.095***
MComms	Harmonic Centrality	0.002**	0.002**	0.055***

*$P < .05$
**$P < .01$ and
***$P < .001$, otherwise R^2 is not significant.

TABLE 2 Correlations (R^2) between the recalculated property of a vertex and the change in an architecture-level topological property caused by the vertex's removal

Network	Property of vertex removed	Change in average Closeness Centrality	Change in average Betweenness Centrality	Change in average Harmonic Centrality
SAR	Degree	0.064***	0.016***	0.549***
MComms	Degree	0.075***	0.102***	0.467***
SAR	Closeness Centrality	0.161***	0.108***	0.349***
MComms	Closeness Centrality	0.267***	0.291***	0.596***
SAR	Betweenness Centrality	0.097***	0.122***	0.715***
MComms	Betweenness Centrality	0.121***	0.553***	0.808***
SAR	Harmonic Centrality	0.016***	0.002	0.213***
MComms	Harmonic Centrality	0.012***	0.016***	0.092***

*$P < .05$
**$P < .01$ and
***$P < .001$, otherwise R^2 is not significant.

The results demonstrate that the extent to which a failure cascades through an architecture depends on both the topology of the architecture and the thresholds assigned, with the distribution of cascade sizes highly sensitive to the initial setting of threshold values. The dependence of failure cascade size on architecture topology is shown for example in Figure 10, where a threshold value of 0.3 for all vertices in the MComms use case results in cascade sizes of up to 5% of the architecture, whereas for the SAR use case cascades may effect 35% of the network.

The results of this kind of analysis provides a perspective on architecture robustness when comparing candidate architectures: under the same modeling assumptions one architecture may be interconnected and interdependent in such a way that it is particularly vulnerable, while the converse may also be true. Furthermore, the analysis reveals the criticality of each vertex, in terms of the size of the cascade that they trigger when they fail. We explored this further by correlating cascade size with topological measures of vertex significance. A sample of correlations (R^2) for both architectures is shown in Table 3. The most significant correlation found for both use cases is between the degree of the failing vertex and the resulting cascade size. Again, this suggests to a system architect that particular focus and emphasis should be placed on the protection of the most connected entities in their architecture. Especially in enterprise architectures where the full extent of an architecture's connectivity may not be present without making a deliberate inquiry as relationships and dependencies may be fragmented across multiple different viewpoints. However, even here there are some threshold profiles for both architectures where degree is not strongly correlated with cascade size. The lack of significant correlations for these threshold profiles could indicate that the threshold values assigned to vertices are more

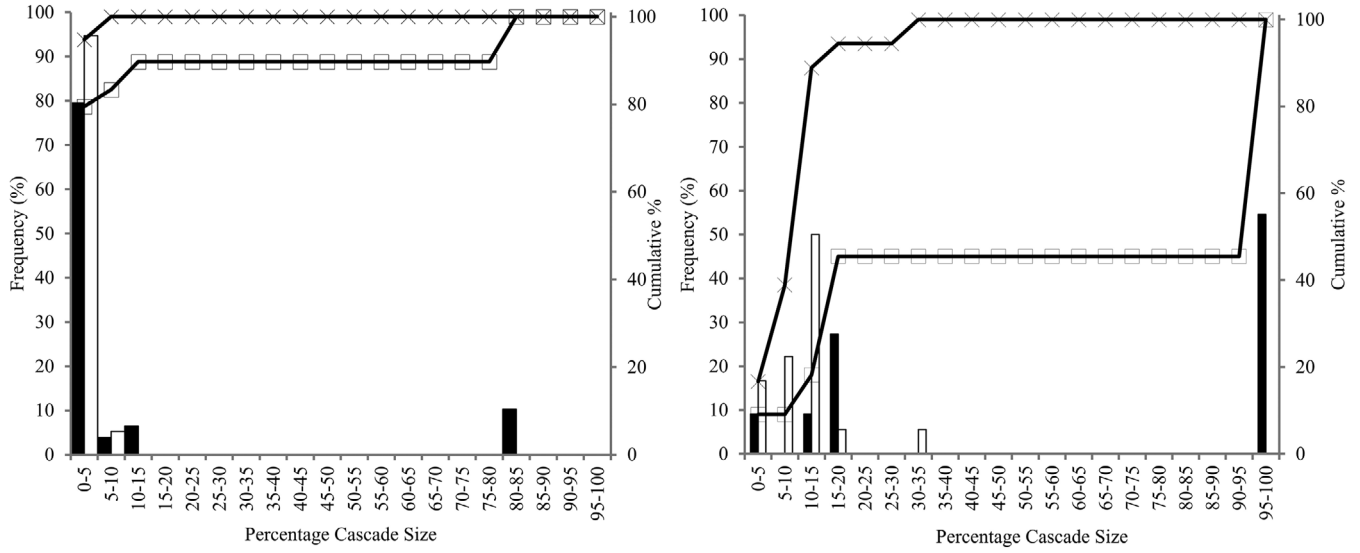


FIGURE 10 Plots show cascade size distributions for MComms (left) and SAR (right). Columns show frequency distributions for $\phi = 0.2$ (solid) and $\phi = 0.3$ (open). Lines show cumulative frequency distributions for $\phi = 0.3$ (squares) and $\phi = 0.2$ (crosses). For the SAR network, no cascades were triggered in 46% of cases for $\phi = 0.2$ and 56% of cases for $\phi = 0.3$. For the MComms networks, the equivalent figures were 67% and 76%

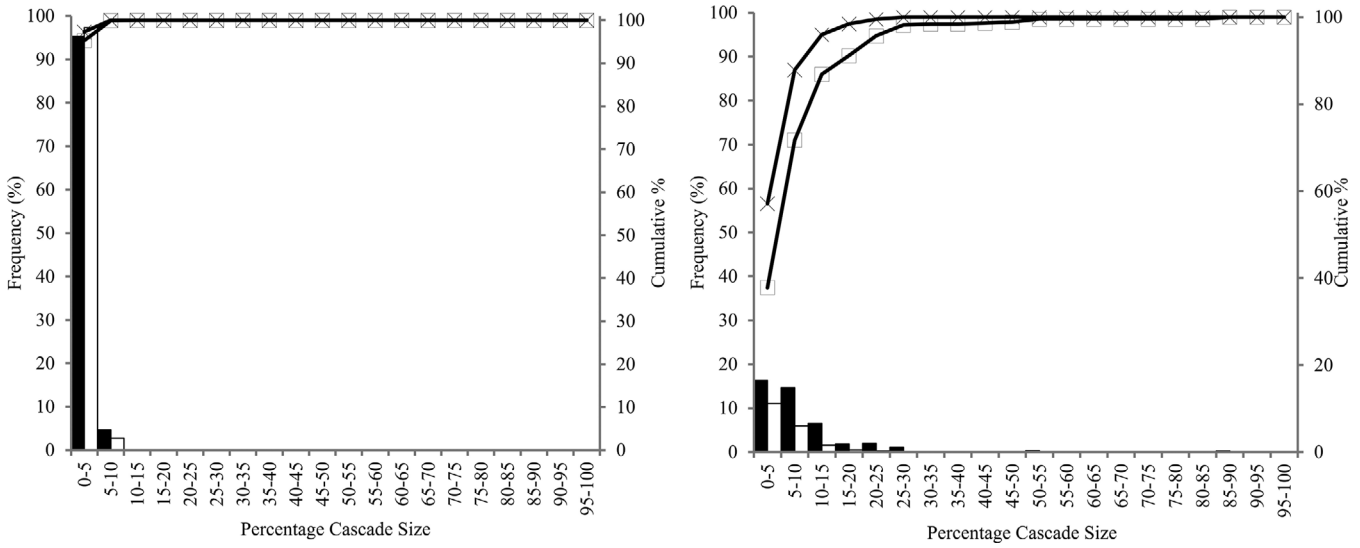


FIGURE 11 Plots show cascade size distributions for MComms (left) and SAR (right). Columns show frequency distributions for $\phi_i \in [0.1, 0.9]$ (solid) and $\phi_i \in [0.25, 0.75]$ (open). Lines show cumulative frequency distributions for $\phi_i \in [0.1, 0.9]$ (squares) and $\phi_i \in$ (crosses). For the SAR network, no cascades were triggered in 57% of cases for $\phi_i \in [0.1, 0.9]$ and 81% of cases for $\phi_i \in [0.25, 0.75]$. For the MComms networks, the equivalent figures were 77% and 83%

significant in determining cascade size than the topological properties of the failing vertices.

Vertex degree makes sense as a predictor of cascade size to some extent, but the degree of a failing vertex only accounts for the first step of the subsequent cascade and does not capture the onward connections of the vertex's neighbors. Although other topological measures of significance based on eigenvector centrality do reflect these higher-order properties of a vertex's neighborhood, in our results neither the eigenvector centrality or Katz centrality correlated more significantly with cascade size than vertex degree. The lack of strong correlations between other topological measures of importance and triggered cascade size could be due to the particular topology of the two architectures considered here, a feature of the model dynamics and threshold values used to simulate cascades, or an inherent limitation of the topological measures of significance considered, or some combination of these factors.

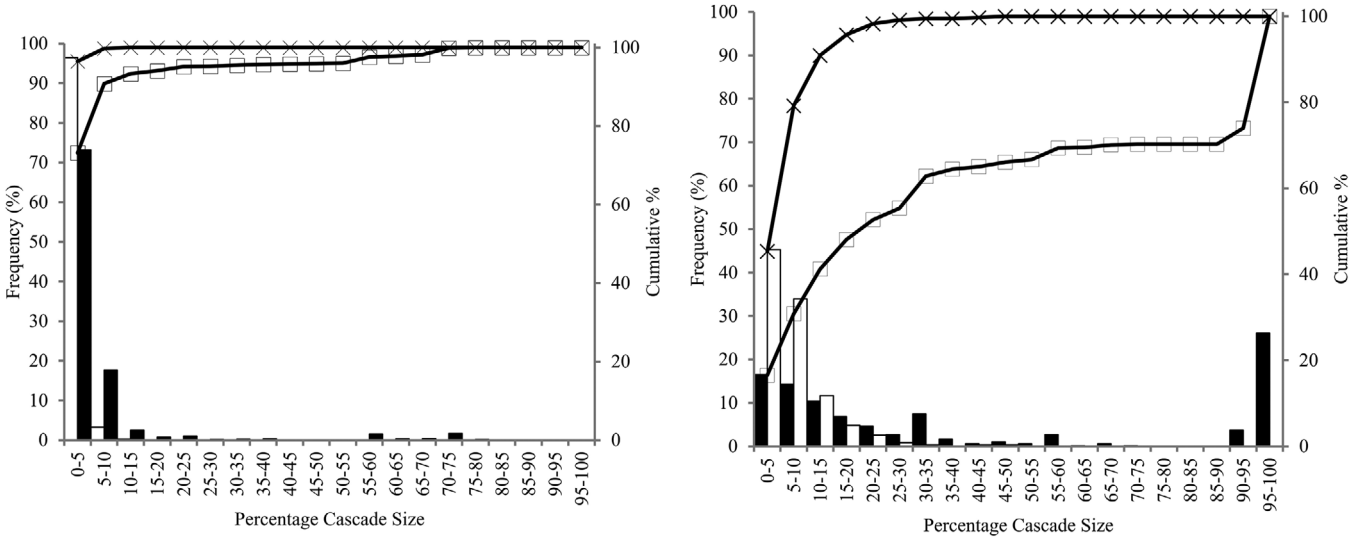


FIGURE 12 Plots show cascade size distributions for MComms (left) and SAR (right). Columns show frequency distributions for $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$ (solid) and $\phi_i \sim \frac{\text{Pois}(\lambda=5)}{10} \in [0.1, 1.0]$ (open). Lines show cumulative frequency distributions for $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$ (squares) and $\phi_i \sim \frac{\text{Pois}(\lambda=5)}{10} \in [0.1, 1.0]$ (crosses). For the SAR network, no cascades were triggered in 34% of cases for $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$ and 66% of cases for $\phi_i \sim \frac{\text{Pois}(\lambda=5)}{10} \in [0.1, 1.0]$. For the MComms networks, the equivalent figures were 68% and 79%

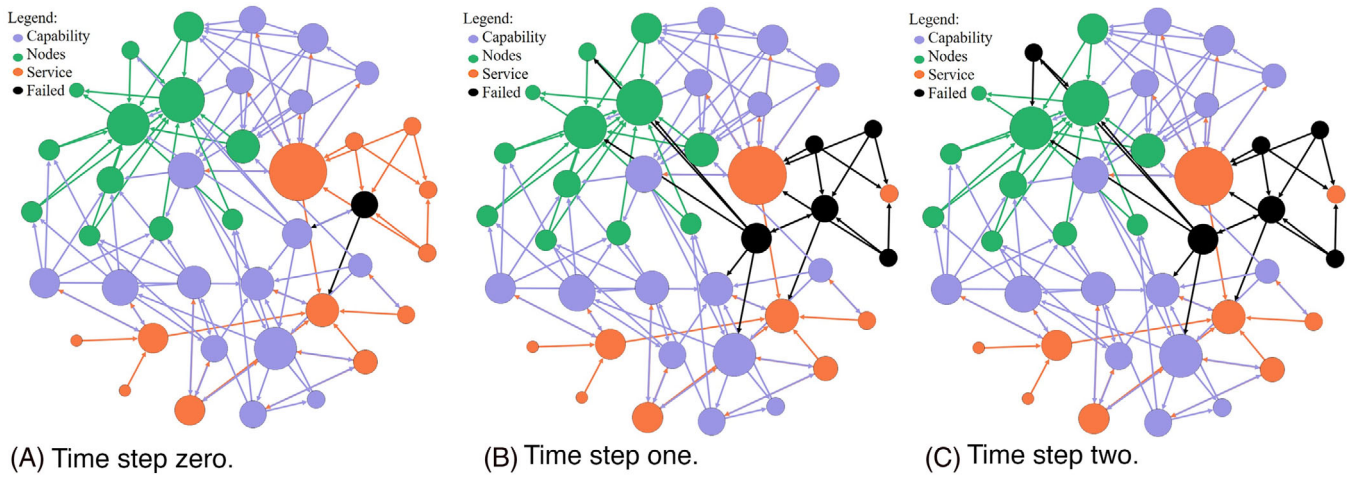


FIGURE 13 SAR network diagram visualized in Gephi with vertices sized by their degree. Vertices that have failed are shown in black and edges are colored by the color of their source vertex. For initial conditions of a threshold value of 0.2 applied to all vertices in each the network, the original vertex that fails corresponds to a "Search Node" (an operational maritime search asset) in the architecture and the failure cascade halts after two iterations with six vertices failed in total, affecting the search service and search capability

4.3 | Hardening against cascades

To what extent can an architect protect their architecture from the cascading failures explored above? Here, a simple hardening strategy, inspired by other research,[38] was adopted that swapped the thresholds of the 10 most critical vertices (as identified above) with those of the 10 vertices with the highest assigned thresholds in the network. This ensures that the most critical vertices are hardened without changing the "hardness" of the network as a whole, and can be thought as equivalent to a reallocation of resource to the most critical parts of the network. The most critical vertices were determined as those associated with the largest mean cascade size over 25 simulations employing a particular threshold profile. The identity of the most critical vertices depended on which threshold profile was used. For example, the most critical vertices under a Poisson distribution of thresholds were not necessarily the same as the most critical vertices identified when a uniform distribution of threshold values was employed. To control for the stochastic nature of the threshold assignment, a hardened

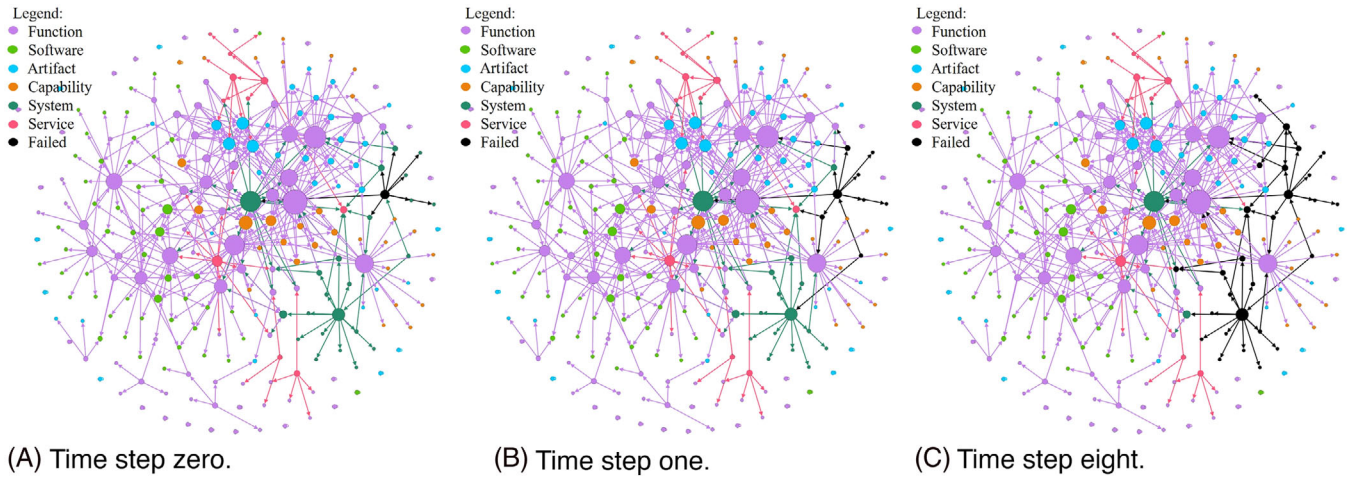


FIGURE 14 MComms network diagram visualized in Gephi with vertices sized by their degree. Vertices that have failed are shown in black and edges are colored by the color of their source vertex. For initial conditions of a threshold value of 0.2 applied to all vertices in each the network, the original vertex that fails corresponds to a "system" (a military communications system deployed in the wider SoS) in the architecture and the failure cascade halts after eight iterations with 27 vertices failed in total, affecting other systems, services, a function, and an operational capability

TABLE 3 Correlations (R^2) between vertex measures of significance and average triggered cascade size for both use cases using different threshold profiles

	Degree	Closeness Centrality	Betweenness Centrality	Harmonic Closeness Centrality
SAR $\phi = 0.2$	0.349^{***}	0.052^{***}	0.255^{***}	0.175^{***}
MComms $\phi = 0.2$	0.281^{***}	0.211^{***}	0.190^{***}	0.000
SAR $\phi = 0.3$	0.199^{***}	0.095^{***}	0.008^{***}	0.001
MComms $\phi = 0.3$	0.490^{***}	0.380^{***}	0.159^{***}	0.006^{***}
SAR $\phi_i \in [0.1, 0.9]$	0.755^{***}	0.058	0.437^{***}	0.189^{**}
MComms $\phi_i \in [0.1, 0.9]$	0.639^{***}	0.388^{***}	0.120^{***}	0.000
SAR $\phi_i \in [0.251, 0.75]$	0.304^{***}	0.052	0.179^{**}	0.007
MComms $\phi_i \in [0.25, 0.75]$	0.384^{***}	0.282^{***}	0.042^{**}	0.009
SAR $\phi_i \sim \frac{\text{Pois}(\lambda=5)}{10} \in [0.1, 1.0]$	0.584^{***}	0.080	0.163^{**}	0.043
MComms $\phi_i \sim \frac{\text{Pois}(\lambda=5)}{10} \in [0.1, 1.0]$	0.532^{***}	0.359^{***}	0.107^{***}	0.001
SAR $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$	0.814^{***}	0.049	0.218^{**}	0.251^{***}
MComms $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$	0.861^{***}	0.419^{***}	0.337^{***}	0.052^{***}

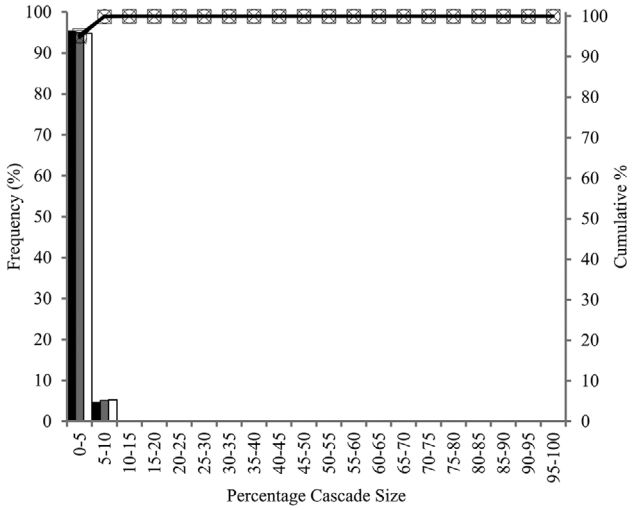
$^{*}P < .05$
$^{**}P < .01$ and
$^{***}P < .001$, otherwise R^2 is not significant.

network was compared with an unhardened network featuring the same threshold values. It is worth noting that a direct comparison of the results between the two SoS architectures is not enabled by this approach as each architecture contains a different number of vertices.

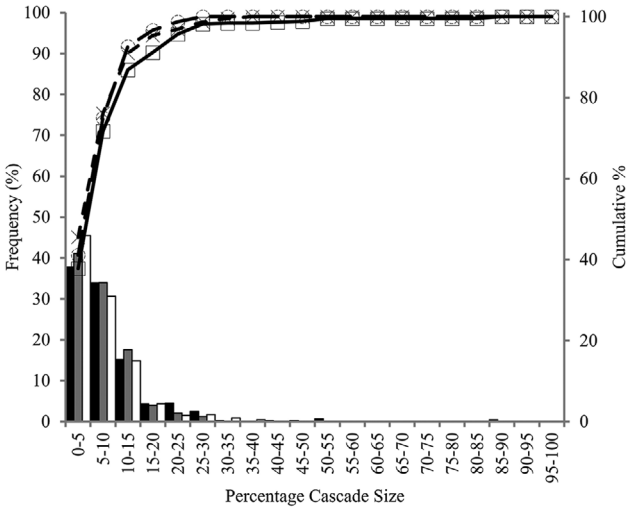
In order to harden the most critical vertices, this hardening strategy relies on a posteriori knowledge of the architecture and its vulnerability to cascade. An alternative hardening strategy was also considered. Noting the correlation uncovered earlier between degree and criticality, the 10 highest-degree vertices had their thresholds swapped with the highest thresholds in the network. Unlike the hardening procedure outlined above, this hardening strategy relies only on knowing, and protecting, the most connected entities without any further knowledge of how susceptible the vertices are to cascades.

The results of these simple hardening strategies are shown in Figures 15 and 16 and Table 4. Outcomes of the hardening strategies were compared with the unhardened outcomes using a Mann-Whitney two-sample rank-sum U test to determine if the distributions differed significantly, with the resulting U test statistic and *P* values provided in Table 4.

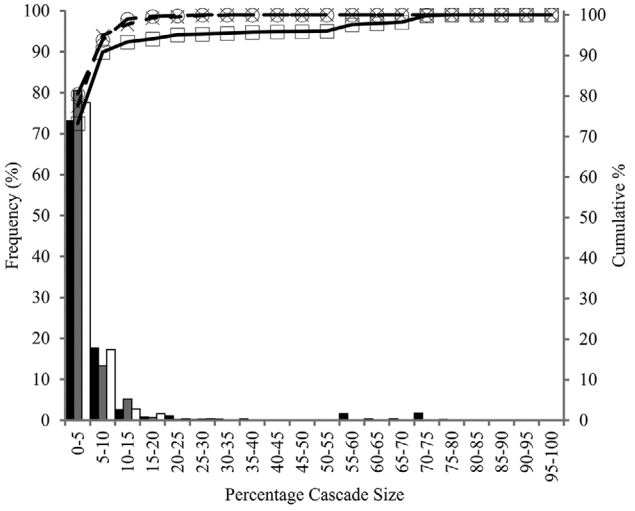
The vulnerability of the MComms network to cascading failure is relatively unaffected by either of the hardening strategies when vertex thresholds are drawn from a uniform distribution between 0.1 and 0.9.



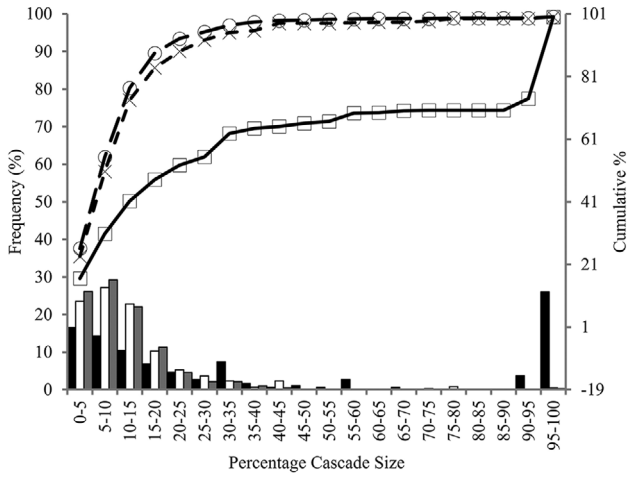
(A) MComms network $\phi_i \in [0.1, 0.9]$. Hardening the 10 highest degree vertices or the 10 most critical vertices has no effort on the number of events which do not cause a cascade (78%), which remains the same as for the unhardened network.



(B) SAR network $\phi_i \in [0.1, 0.9]$. Hardening the 10 highest degree vertices reduces the number of events which do not cause a cascade to 53%, compared to 57% for the unhardened network. Hardening the 10 most critical vertices reduces the number of cascades to 55%.



(C) MComms network $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$. Hardening the 10 highest degree vertices or the 10 most critical vertices has no effort on the number of events which do not cause a cascade (68%), which remains the same as for the unhardened network.



(D) SAR network $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$. Hardening the 10 highest degree vertices increases the number of events which do not cause a cascade to 40%, compared to 34% for the unhardened network. Hardening the 10 most critical vertices increases the number of cascades to 41%.

FIGURE 15 Impact of hardening strategies on the MComms network (left) and SAR network (right) with $\phi_i \in [0.1, 0.9]$ (top) and $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$ (bottom). Hardening the 10 highest degree vertices displayed with half-tone bars, circles, hardening the 10 most critical vertices displayed with open bars, crosses, and the unhardened network is displayed with filled bars, squares

This is perhaps because the architecture is already robust to cascades for this threshold profile, as shown in Figure 15A and in the first row of Table 4. Consistent with this thinking, both hardening strategies have a more significant impact for the same network when vertex thresholds are drawn from a uniform distribution between 0.0 and 1.0, which is associated with more significant cascades in the unhardened network (Figure 16).

The results suggest that a simple hardening strategy targeted at only the 10 most connected entities can reduce the number of cascades triggered to a similar extent as hardening with prior knowledge

TABLE 4 Summary statistics showing the effect of different hardening strategies on cascade size compared with the original unhardened network cascade size for both use cases ($U < U_{Crit}$ when

Profile	IQR	Max	Median	Mean	% of events trigger change	*U*
MComms $\phi_i \in [0.1, 0.9]$ Original	1.28	11.49	1.28	1.89	22%	
Hardened by most vulnerable	1.28	8.09	1.28	1.90	22%	850,872.0*
Hardened by degree	1.70	8.51	1.28	1.91	22%	849,386.5
MComms $\phi_i \in [1.0, 1.0]$ Original	3.40	54.47	2.55	4.33	32%	
Hardened by most vulnerable	1.70	11.49	1.28	1.92	22%	820,662.0***
Hardened by degree	2.13	17.02	2.13	2.74	31%	1,535,740.5***
SAR $\phi_i \in [0.1, 0.9]$ Original	7.32	87.80	7.32	10.20	43%	
Hardened by most vulnerable	4.88	36.59	7.32	8.78	45%	91,258.5**
Hardened by degree	4.88	29.27	7.32	8.68	47%	99,020.0
MComms $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$ Original	3.83	79.15	2.55	6.27	32%	
Hardened by most vulnerable	2.92	37.45	2.13	3.57	32%	1,684,920.0**
Hardened by degree	2.55	28.51	2.13	3.28	32%	1,698,750.5***
SAR $\phi_i \sim \frac{\text{Pois}(\lambda=3.5)}{10} \in [0.1, 1.0]$ Original	87.80	100.00	21.95	41.90	66%	
Hardened by most vulnerable	9.76	100.00	9.76	14.05	59%	124,009.5***
Hardened by degree	9.76	100.00	9.76	12.18	60%	116,041.0***

*$P < .05$
**$P < .01$
***$P < .001$, otherwise not significant)

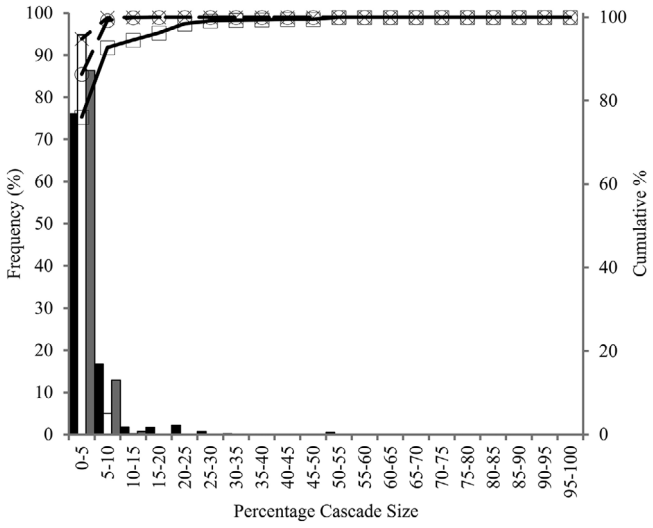


FIGURE 16 Impact of hardening strategies on the MComms network $\phi_i \in [0.0, 1.0]$. Hardening the 10 highest degree vertices (half-tone bars, circles) increases the number of events which do not cause a cascade to 69%, compared to 67% for the unhardened network (filled bars, squares). Hardening the 10 most critical vertices (open bars, crosses) increases the number of events which do not cause a cascade to 78%

of the most vulnerable vertices. Hardening the most connected entities was sometimes more effective than hardening what were identified as the most critical vertices, although it is not clear at this stage whether this is due to the sample size used to determine the most vulnerable vertices to cascading failure, or if the effect of using the average cascade size did not sufficiently account for extreme cascade sizes, or because degree is a more robust predictor of criticality than the results of the simulated cascades. What is clear, however, is that reducing vulnerability to cascading failure can be achieved by hardening the most connected entities in either architecture.

5 | DISCUSSION

We return to the worked example to discuss how the findings can inform system architecture evaluation, where an organization examining architectural alternatives may consider the overall *robustness* of candidate architectures by comparing the impact of random and targeted vertex removal on network representations of their architectures. An architecture that experiences a more gradual reduction in average closeness centrality as vertices are removed could be considered to be more desirable from this perspective and would correspond with an architectural pattern that is more densely connected and less hierarchical. If an organization is not evaluating candidate architectures and is instead interested in understanding the robustness of one particular architecture, they can examine the vertices that, when removed, have the largest impact on network viability as a proxy for architecture effectiveness. While correlations of common network centrality measures were shown to be unreliable predictors of important vertices in terms on their impact on network viability if removed, organizations may wish to simulate the effect of vertex removal in order to identify vertices of particular importance in the architecture.

Similarly, organizations can compare the effect of cascading failures on their candidate architectures, potentially preferring an architecture less susceptible to cascading failure, or they can examine if some architectural entities are significant contributors to the triggering of substantial cascades. In such cases, organizations may wish to invest in protecting the most connected architectural entities in order to harden their architectures. While this finding may seem somewhat obvious, knowledge of the full extent of architectural entity connectivity may not be available without adopting a network perspective.

However, at several points throughout this paper, we have raised the issue that taking a network perspective on complex SoS architectures to support architectural robustness analysis confronts significant conceptual and technical challenges and these are considered in more detail here.

Perhaps the most fundamental observation relevant to this discussion is one that can be made whenever a network perspective is taken on any real-world system: "the map is not the territory." An SoS is not a graph, and neither is an SOS architecture. Rather, an SoS is a real-world system, actual or envisaged, and an SoS architecture is a rich set of models, documents, resources, and ideas related to this real-world system, whereas a network or graph is a mathematical object comprising only a set of vertices and a set of edges. Taking a network perspective on a real-world SoS, or SoS architecture, entails taking a particular stance on what is important and what is not, what should be represented in the graph and what can be omitted. The result is a network model that is an abstraction, representing parts of the actual real-world SoS in a particular way. Properties of this network model are not necessarily straightforward to interpret in terms of properties of the real-world system of interest. But this interpretation step is necessary before the implications of network analyses can be stated or understood.

In some domains, the network model of a real-world system provides considerable, and relatively direct, insight into this system of interest and dynamic processes. Consider, for example, social networks used to explain and describe complex human activity. Here, a network analysis might involve calculating the average shortest path length in a graph representing social interactions between pupils at a school. Many factors that are true of the real school will not be captured by such a network, such as which parents are friends with each other, or whether relationships between children have changed over time, or whether some children are relatives, or share the same favourite sports team, etc. Despite this, the path length metric can be interpreted as a measure that indicates useful things about how the population of pupils can be expected to behave: How quickly will gossip or communicable disease spread through the school? How likely are two pupils to share information? etc.

While the path length metric is only a *proxy* for the complicated set of processes and factors that influence the real dynamics of the school population, it has come to be regarded as a useful proxy. This is partly as a result of two factors: (a) the relatively straightforward relationship between path length in a graph and spreading processes in a social network, and (b) considerable effort by the social science community in coming to understand the relationship between social networks as mathematical objects and social networks as real-world systems.[52,53,64,65] Unfortunately, neither of these two factors are as applicable to network analysis of an SoS or SoS architecture.

Like schools, SoS architectures contain entities that are heterogenous and complicated, connected by relations that are also heterogeneous and complicated. In network models of complex SoS architectures, vertices may represent communication services, command and control capabilities, software running on geographically dispersed assets, the assets themselves, management and administrative functions, data flows, etc. Moreover, the same architecture may represent the same types of entity at differing levels of description: individual entities (eg, drones), component subsystems (eg, drone landing systems), and aggregate supersystems (eg, fleets of drones).

However, although schools are complex, social networks analysis typically tends to license a social network representation that idealizes away the diversity and heterogeneity inherent in a population of pupils: each pupil is represented by a node, each relationship by an edge, nodes are interchangeable objects, and edges are equivalent to one another. As a consequence, the meaning of a claim such as "the shortest path connecting vertex i to vertex j has length four" has a relatively straightforward meaning in the context of a social network representing a schoolyard: eg, in order for gossip to travel from i to j it will tend to have to pass through at least three intermediate children.

The same type of claim is much harder to interpret if i and j instead represent entities in a complex SoS architecture. What, if anything, passes along the edges in such a network? Is the answer to this question the same for every edge in the path? What is implied by the fact that a shortest path is relatively short or relatively long? And if assigning meaning to a simple measure like path length is challenging for a complex SoS architecture, this challenge is amplified for more sophisticated measures such as assortativity, eigenvector centrality, etc.

Furthermore, in order to explore issues around robustness and cascading failure, a conception of failure modes, dynamics and propagation is required, such as knowledge of partial failure propagation.[66] Again, such a conception may not always be present at an early design stage. For a communications network represented at the data link layer, a failure may equate to the loss of one or more comms devices, whereas for an air transport network a failure may be characterized as a cascading delay of flights arising as delays propagate through the system. For the SAR and MComms use cases considered here exactly what and how a failure occurs and propagates is more conceptually challenging, and may not even be equivalent across all parts of the network.

Moreover, in order to make direct claims about SoS resilience, the characterization of system failure discussed above must be extended to deal with recovery and repair processes. A binary switch between "effective" and "ineffective" states, as employed in our analyses here, may not be sufficient to capture the dynamics of failure and recovery in a real SoS. In particular, the timescales of recovery in different parts of the system and the mitigating interventions in place may have a significant influence on recovery.

What types of perturbation should an architecture's resilience be evaluated against? Here, we have tended to consider the impact of a sequence of nodes becoming ineffective, but a more realistic

characterization might allow for several, potentially correlated entities to become ineffective at the same time. Current studies tend to either consider independent random failures or targeted attacks. In real systems, however, scenarios in which multiple problems cooccur tend to arise as the result of common cause failures, perhaps because assets are colocated in the same geographical region or subject to the same latent vulnerability to an extreme environmental factor.[67] The correlated failure profiles that result are not addressed by modeling either random perturbation or attacks targeting vertices with particular network properties (eg, high degree). Potential solutions include providing a weighting to edges in the network,[33] or considering "interdependent" network models,[38] or adopting an approach cognisant of vertex connectivity (ie, approaches that consider the "fan-in" or "fan-out" between vertices).[31] However, the intelligence upon which to base more relevant or sophisticated models of perturbation may be difficult to collate and interpret, especially at an early design stage, or for SoS contexts that are innovative or first-of-kind in some respect.

Finally, in order to characterize the negative impact of perturbation or the positive impact of recovery, some notion of system viability or performance is required.[68,69] To what extent can an effected SoS be expected to continue to achieve its function? At what point is system effectiveness wholly compromised by structural changes to its graphical representation? Is it sufficient to assume that once the SoS architecture network is fragmented to a certain degree, the associated SoS architecture is ineffective? How would one determine such a threshold a priori? While the size of the largest surviving connected component may be used to characterize the viability of a postattack graph, and this may be a better measure than simply counting the number of surviving nodes, this approach may not translate smoothly to the context of a complex SoS architecture. It may be the case that the loss of a small number of subcomponents on the "periphery" of an architecture is likely to be less damaging to overall system viability than a loss of the same number of entities from the "core" of the architecture. However, we must be careful not to necessarily equate the *structural* core of a graph with the *functional* core of an SoS.

To a first approximation, the issues described above tend to stem from the fact that SoS architectures involve a high degree of heterogeneity in their component entities and the relationships among these entities (see "Guiding Principle 1"[33]). To the extent that an architecture is relatively homogeneous (eg, it comprises a set of similar components organized in some "flat" configuration), relatively standard network analysis approaches can be applied—although it is still the case that the interpretation of the results of such analyses may require careful thought. In the case of less homogeneous SoS architectures, in order for the desired insights into, say, system robustness to be more amenable to graph-theoretic analysis it may be appropriate to examine a more homogeneous subset of the architecture that can be more readily abstracted as a simple network, eg, the communications architecture within a complex SoS. However, carving up an SoS in this way is problematic as it precludes the kind of holistic analysis that should, ultimately, be the aim of a systems architect working in the context of a complex SoS project.[70]

6 | CONCLUSIONS

In this paper, we have taken a network perspective on SoS architecture analysis in order to assess robustness, by examining the effect of vertex removal on the overall architecture topology, and vulnerability to cascading failure. Both analyses may be useful to system architects looking to inform their evaluations of candidate architectures, by considering if one architecture is more robust to perturbation than another. Furthermore, system architects can identify which architectural entities are important, or influential, in the sense that their removal affects network viability either as a consequence of the direct impact of their absence, or the likelihood of their failure triggering a cascade. In particular, for a complex SoS, such an approach may support technical or operational risk identification by identifying particularly vulnerable architectural entities, and determining if they are within or outside of an organization's control.

Both of the architectures analyzed here were are found to be relatively robust to random vertex removal but more vulnerable to targeted vertex removal, implying that system architects should pay particular attention to system architectural entities with high connectivity, noting that without the adoption of a network perspective, the full extent of an architectural entity's connectivity may not be known. While hardening strategies for limiting the extent of cascading failure were shown to have varying degrees of effectiveness, results suggested that system architects should look to the most connected architectural entities as areas of particular concern when seeking to mitigate the potential impact of cascading failure.

The results presented here for the two real-world use case architectures suggest that SoS robustness and vulnerability to cascades depends on both the model imposed on the architecture and the topology of the architecture, and that improvements in robustness to cascades may be achieved by hardening the most connected entities. While the results of such analyses have the *potential* to inform architecture design and selection, we argue that a series of challenges need to be addressed if the approach is to be useful for system architects. These challenges include the conceptualization of failure dynamics across heterogeneous system architecture entities, the conceptualization of *resilience* aspects and approaches to assess network representations of architectures for both network perturbation and cascading failure analysis approaches.

The challenges described here are all compounded by the tendency for system architecting activity to correspond to early system lifecycle phases where information which would support such evaluations may be limited. Furthermore, the simple models used here neglect potentially more important contextual information that may be present in the full architecture, such as recovery strategies or criticality of system entities for overall functionality.

A network science perspective encourages us to ask important questions of our system architectures: how robust is our system to perturbations; how well protected are the most critical architectural entities; how extensive will the negative impact be when these protections fail; and how readily will our system be able to adapt and recover?

Given that networks science has delivered powerful insights into a wide range of social and technological systems, it likely also has the potential to inform our understanding of SoS robustness and resilience in order to help us answer these questions. However, we argue that the analyses presented here imply that this will only be possible if the numerical analyses that networks science enables are deployed and interpreted through the lens of an appropriately rich contextual understanding of the system architectures in question.

ACKNOWLEDGMENTS

This research was funded by the Engineering and Physical Science Research Council and Thales under ICASE 16000139. Supporting data are available from Ref. 71. The authors would like to thank Dave Harvey and Jean-Luc Garnier at Thales who provided valuable insights during this work. The authors also wish to thank the editor and anonymous reviewers for their detailed and insightful feedback which has significantly improved this paper.

ORCID

Matthew W. Potts https://orcid.org/0000-0002-4266-0862

REFERENCES

1. ISO/IEC/IEEE 42020 International Standard. Software, systems and enterprise—architecture processes; 2019.
2. Martin JN. Overview of an emerging standard on architecture processes–ISO/IEC 42020. In: *INCOSE International Symposium*. Wiley Online Library; 2016;26:1431-1447.
3. ISO/IEC/IEEE 42030 International Standard. Software, systems and enterprise—architecture evaluation framework; 2019.
4. Crawley E, Cameron B, Selva D. *System Architecture: Strategy and Product Development for Complex Systems*. Boston: Prentice Hall Press; 2015.
5. Crawley E, De Weck O, Magee C, Moses J, Seering W, Schindall J, et al. *The Influence of Architecture in Engineering Systems*. Massachusetts Institute of Technology. Engineering Systems Division; 2004.
6. Maier MW. Architecting principles for systems-of-systems. *Syst Eng*. 1998;1(4):267-284.
7. Boardman J, Sauser B. System of Systems-the meaning of of. In: *IEEE/SMC International Conference on System of Systems Engineering*. IEEE Press; 2006:118-126.
8. ISO/IEC/IEEE 21839 International Standard. Systems and software engineering—system of systems (SoS) considerations in life cycle stages of a system; 2019.
9. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*. Wiley; 2015. Available from: https://books.google.co.uk/books?id=xSDyCQAAQBAJ
10. Dahmann J, Baldwin K. Implications of systems of systems on system design and engineering. In: *6th International Conference on System of Systems Engineering (SoSE)*. IEEE Press; 2011:131-136.
11. Dahmann J. 1.4.3 system of systems pain points. In: *INCOSE International Symposium*. Vol. 24. Wiley Online Library; 2014: 108-121.
12. Madni AM, Jackson S. Towards a conceptual framework for resilience engineering. *IEEE Syst J*. 2009;3(2):181–191.
13. Baker JW, Schubert M, Faber MH. On the assessment of robustness. *Struct Saf*. 2008;30(3):253-267.
14. The Rt Hon Sir Martin Moore-Bick. Grenfell Tower Inquiry: Phase 1 Report. Report of the public inquiry into the fire at Grenfell Tower on 14 June 2017. Grenfell Tower Inquiry, London, United Kingdom Government; 2019.
15. Defence Safety Authority. The Service Inquiry report into the loss of Watchkeeper (WK042) unmanned air vehicle over Cardigan Bay in West Wales on 3 February 2017, DSA/DAIB/17/006. The Ministry of Defence; 2019.
16. Uday P, Marais K. Designing resilient systems-of-systems: a survey of metrics, methods, and challenges. *Syst Eng*. 2015;18(5):491-510.
17. Zachman JA. A framework for information systems architecture. *IBM Syst J*. 1987;26(3):276–292.
18. Biggs B. *Ministry of Defence Architectural Framework (MODAF)*. Institution of Engineering and Technology; 2005:43-82. Available from: http://digital-library.theiet.org/content/conferences/10.1049/ic20050127
19. Architecture Capability Team, Consultation, Command & Control Board. NATO Architecture Framework Version 4. North Atlantic Treaty Organisation (NATO); 2018. Available from: https://www.nato.int/cps/en/natohq/topics57575.html
20. Lee S. *DoDAF V2.0 Overview. Department of Defense*, Washington DC Chief Information Officer; 2010.
21. ISO/IEC/IEEE 42010. Systems and software engineering—architecture Description; 2011.
22. Hinsley S, Henshaw M, Siemieniuch C. Maintaining system of systems that are fit for purpose: effectively and efficiently preparing for the unforeseeable. *IEEE Syst Man Cybern Mag*. 2017;3(1):23-29.
23. De Weck OL, Ross AM, Rhodes DH. *Investigating relationships and semantic sets amongst system lifecycle properties (Ilities)*. Massachusetts Institute of Technology Engineering Systems Division ESD Working Paper Series. 2012:1-13.
24. Bartolomei JE, Hastings DE, de Neufville R, Rhodes DH. Engineering systems multiple-domain matrix: an organizing framework for modeling large-scale complex systems. *Syst Eng*. 2012;15(1):41-61.
25. Nicola R, Fitzgerald ME, Ross AM, Rhodes DH. Architecting systems of systems with ilities: an overview of the SAI method. *Procedia Comput Sci*. 2014;28:322-331.
26. Urken AB, Schuck TM, Schuck TM. Designing evolvable systems in a framework of robust, resilient and sustainable engineering analysis. *Adv Eng Inform*. 2012;26(3):553-562.
27. Enos JR, Nilchiani R. Understanding how social network analysis can provide insights into emergent networks of systems. In: Madni AM, Boehm B, Ghanem RG, Erwin D, Wheaton MJ, eds. *Disciplinary Convergence in Systems Engineering Research*. Cham: Springer International Publishing; 2018:249-263.
28. Enos JR. Merging system architecture and social network analysis to better understand emergent networks of systems. In: *Proceedings of the International Annual Conference of the American Society for Engineering Management*. American Society for Engineering Management (ASEM). 2016:1-10.
29. Enos JR, Nilchiani R. Using social network analysis to identify systems of systems in a network of systems. In: *13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE; 2018:623-628.
30. Curry DM, Dagli CH. SoS explorer: a tool for system-of-systems architecting. In: Madni AM, Boehm B, Ghanem RG, Erwin D, Wheaton MJ, eds. *Disciplinary Convergence in Systems Engineering Research*. Cham: Springer International Publishing; 2018:187-196.
31. Braha D. The Complexity of Design Networks: Structure and Dynamics. In: Cash P, Stanković T, Štorga M, eds. *Experimental Design Research: Approaches, Perspectives, Applications*. Cham: Springer International Publishing; 2016:129-151.
32. Tran HT, Domerçant JC, Mavris DN. Designing resilient system-of-systems networks. In: *2017 Annual IEEE International Systems Conference (SysCon)*. 2017:1-6.
33. Potts MW, Sartor P, Johnson A, Bullock S. A network perspective on assessing system architectures: foundations and challenges. *Syst Eng*. 2019;22(6):485-501.
34. Potts M, Sartor P, Johnson A, Bullock S. Hidden structures: using graph theory to explore complex system of systems architectures.

In: Chapoutot A, Krob D, Roussel A, Stephan F, eds. *Complex Systems Design & Management*. Paris: CESAM Community; 2017:117-131.
35. Potts M, Sartor P, Johnson A, Bullock S. Through a glass, darkly? Taking a network perspective on system-of-systems architectures. In: Bonjour E, Krob D, Palladino L, Stephan F, eds. *Complex Systems Design & Management*. Cham: Springer International Publishing; 2019: 121-132.
36. Libert P, Garnier JL. *NAFv4 Chapter 2 Example.; Under Development. Libert,P (Airbus), Garnier, J-L (Thales)*. 2017:1-103
37. Khoury M, Bullock S. Multi-level resilience: reconciling robustness, recovery and adaptability from a network science perspective. *Int J Adapt Resilient Auton Syst*. 2014;5(4):34-45.
38. Fu G, Dawson R, Khoury M, Bullock S. Interdependent networks: vulnerability analysis and strategies to limit cascading failure. *Eur Phys J B*. 2014;87(7:148):1-10.
39. Haimes YY. Modeling complex systems of systems with phantom system models. *Syst Eng*. 2012;15(3):333-346.
40. Goerger SR, Madni AM, Eslinger OJ. Engineered Resilient Systems: A DoD Perspective. *Procedia Computer Science*. 2014;28:865–872. https://doi.org/10.1016/j.procs.2014.03.103
41. Enos JR. Using systems engineering ilities to better understand resiliency. In: *Systems Engineering in Context*. Springer; 2019:25-36.
42. Gao J, Barzel B, Barabási AL. Universal resilience patterns in complex networks. *Nature* 2016;530(7590):307-312.
43. Clausing D, Frey DD. Improving system reliability by failure-mode avoidance including four concept design strategies. *Syst Eng*. 2005;8(3):245-261.
44. Zhang WJ, Lin Y. On the principle of design of resilient systems—application to enterprise information systems. *Enterp Inf Syst*. 2010;4(2):99-110.
45. Aboutaleb H, Monsuez B. Quantifying system complexity in design phase using higraph-based models. In: *INCOSE International Symposium*. Vol. 26. Wiley Online Library; 2016:238-252.
46. Harrison WK. The role of graph theory in system of systems engineering. *IEEE Access*. 2016;4:1716-1742.
47. Braha D, Maimon O. The measurement of a design structural and functional complexity. In: *A Mathematical Theory of Design: Foundations, Algorithms and Applications*. Boston: Springer; 1998:241-277.
48. Okami S, Kohtake N. Transitional complexity of health information system of systems: managing by the engineering systems multiple-domain modeling approach. *IEEE Syst J*. 2019;13(1):952-963.
49. Braha D, Bar-Yam Y. Information flow structure in large-scale product development organizational networks. *J Inf Technol*. 2004;19(4):244-253.
50. Braha D. The topology and dynamics of complex man-made systems. In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE; 2007:4062-4068.
51. Zakarian A, Knight JW, Baghdasaryan L. Modelling and analysis of system robustness. *J Eng Des*. 2007;18(3):243-263.
52. Newman ME. The structure and function of complex networks. *SIAM Rev*. 2003;45(2):167-256.
53. Newman M, Barabási AL, Watts DJ. *The Structure and Dynamics of Networks. Princeton Studies in Complexity*. Princeton: Princeton University Press; 2011.
54. Albert R, Jeong H, Barabási AL. Error and attack tolerance of complex networks. *Nature*. 2000;406(6794):378-382.
55. Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, et al. Graph structure in the web. *Comput Netw*. 2000;33(1):309-320.
56. Newman MEJ, Forrest S, Balthrop J. Email networks and the spread of computer viruses. *Phys Rev E*. 2002;66:035101.
57. Jeong H, Mason SP, Barabási AL, Oltvai ZN. Lethality and centrality in protein networks. *Nature*. 2001;411(6833):41-42.
58. Buldyrev SV, Parshani R, Paul G, Stanley HE, Havlin S. Catastrophic cascade of failures in interdependent networks. *Nature*. 2010;464(7291):1025-1028.
59. Roberts N, Everton SF. Strategies for combating dark networks. 2011.
60. Everton SF. *Disrupting Dark Networks*. vol. 34. Cambridge, MA: Cambridge University Press; 2012.
61. Hethcote HW. The mathematics of infectious diseases. *SIAM Rev*. 2000;42(4):599-653.
62. Watts DJ. A simple model of global cascades on random networks. *Proc Natl Acad Sci*. 2002;99(9):5766-5771.
63. Abbott R. Open at the top; open at the bottom; and continually (but slowly) evolving. In: *IEEE/SMC International Conference on System of Systems Engineering*. IEEE Press; 2006:41-46.
64. Newman M. *Networks*. Oxford: Oxford University Press; 2018.
65. Barabási AL. *Network Science*. Cambridge, MA: Cambridge University Press; 2016.
66. Efatmaneshnik M, Ryan M. Failure propagation in SoS: why SoS should be loosely coupled. In: *9th International Conference on System of Systems Engineering (SOSE)*. IEEE; 2014:49-54.
67. Wisetjindawat W, Wilson RE, Bullock S, de Villafranca AEM. Modeling the impact of spatial correlations of road failures on travel times during adverse weather conditions. *Transp Res Rec*. 2019;2673(7):157-168.
68. Han SY, Marais K, DeLaurentis D. Evaluating system of systems resilience using interdependency analysis. In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2012:1251-1256.
69. Pan X, Yang Y, Jiang Z, Lin Y. Area method used in the evaluation of resilience implementation approaches in system of systems. In: *11th International Conference on Reliability, Maintainability and Safety (ICRMS)*. 2016:1-5.
70. Mansouri M, Sauser B, Boardman J. Applications of systems thinking for resilience study in maritime transportation system of systems. In: *3rd Annual IEEE Systems Conference*. 2009:211-217.
71. Potts M, Bullock S. 20191204_Data_Potts_PhD_Network_Perpsective_System_Architecture; 2019. https://doi.org/10.5523/bris.2a2yi8s0fv1q22ubogxb26y0an

AUTHOR BIOGRAPHIES

MATTHEW POTTS is a PhD Research Student at the University of Bristol, United Kingdom within the Aerospace Engineering Department. He received the BEng degree from the University of Southampton, United Kingdom in Electromechanical Engineering (2008-2011). He served in various roles in the Royal Air Force (2011-2016) as an Engineering Officer and has worked as an independent consultant. He is a registered Incorporated Engineer with the Institute of Engineering and Technology.

PIA SARTOR is a senior lecturer in the Department of Aerospace Engineering, University of Bristol, United Kingdom. She received a BEng (Hons) degree in Aerospace Engineering from Ryerson University, Toronto, Canada (2006) and a PhD degree in Mechanical Engineering from the University of Sheffield, United Kingdom (2011). She worked in industry for six years, including as a Senior Systems Engineer at Safran Landing Systems. She is a registered Chartered Engineer with the Institute of Mechanical Engineers.

ANGUS JOHNSON is the Thales Head of Systems Research and the Industry lead on the Thales-Bristol Partnership in Hybrid

Autonomous Systems Engineering. He has a background in radar spectrum sensing and multifunction RF systems engineering.

SETH BULLOCK is Toshiba Chair in Data Science and Simulation in the Department of Computer Science at the University of Bristol, United Kingdom. He holds a first degree in cognitive science (1993) and a doctorate in evolutionary simulation modeling (1997) from Sussex University, UK. His research expertise lies in complex systems simulation modeling. He has won over £28m in research and infrastructure funding in this area and has undertaken consultancy for the UK Government on complexity in ICT (2004) and financial systems (2012). He publishes in international peer-reviewed journals spanning health, social science, economics, biology, architecture, engineering, environmental science, computing, and physics. He was twice elected to the Board of Directors of the International Society for Artificial Life and serves on the editorial boards of the MIT Press journals *Adaptive Behavior*, and *Artificial Life*, and *Frontiers in Robotics & AI*. He has given invited keynote lectures in London, Paris, Athens, Melbourne, Madrid, Granada, and Tokyo.

How to cite this article: Potts MW, Sartor PA, Johnson A, Bullock S. A network perspective on assessing system architectures: Robustness to cascading failure. *Systems Engineering*. 2020;1–20. https://doi.org/10.1002/sys.21551