Analyzing Resilience Properties of Different Topologies of Collective Adaptive Systems

Thomas J. Glazier, Javier Cámara, Bradley Schmerl, David Garlan
Institute for Software Research
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{tglazier, jcmoreno, schmerl, garlan}@cs.cmu.edu

***Abstract*—Modern software systems are often compositions of entities that increasingly use self-adaptive capabilities to improve their behavior to achieve systemic quality goals. Self-adaptive managers for each component system attempt to provide locally optimal results, but if they cooperated and potentially coordinated their efforts it might be possible to obtain more globally optimal results. The emergent properties that result from such composition and cooperation of self-adaptive systems are not well understood, difficult to reason about, and present a key challenge in the evolution of modern software systems. For example, the effects of coordination patterns and protocols on emergent properties, such as the resiliency of the collectives, need to be understood when designing these systems. In this paper we propose that probabilistic model checking of stochastic multiplayer games (SMG) provides a promising approach to analyze, understand, and reason about emergent properties in collectives of adaptive systems (CAS). Probabilistic Model Checking of SMGs is a technique particularly suited to analyzing emergent properties in CAS since SMG models capture: (i) the uncertainty and variability intrinsic to a CAS and its execution environment in the form of probabilistic and nondeterministic choices, and (ii) the competitive/cooperative aspects of the interplay among the constituent systems of the CAS. Analysis of SMGs allows us to reason about things like the worst case scenarios, which constitutes a new contribution to understanding emergent properties in CAS. We investigate the use of SMGs to show how they can be useful in analyzing the impact of communication topology for collections of fully cooperative systems defending against an external attack.**

I. Introduction

Modern software systems are often compositions of large and complex entities that increasingly use self-adaptive capabilities to improve their behavior against defined quality standards. For example, Netflix software is composed of separate deployments in multiple regions, each controlled by a self-adaptation manager, Scryer, to provision the resources required to handle changing customer traffic in an effort to maintain a scalable and resilient system [1], [2].

These individual deployments and self-adaptive managers attempt to provide locally optimal results. For example, the self-adaptive manager in each deployment improves the reliability and scalability of that deployment. However, local optimality does not guarantee that quality targets are achieved globally. In such cases, it would be advantageous for each self-adaptive system to cooperate, and potentially coordinate, to obtain more globally optimal results. For example, if Netflix undergoes a security attack, deployments that are under attack could communicate this fact to the collective so other members could be prepared in case the attack migrates to other deployments. In designing the collective, there are many choices about the protocol of communication between elements of the collective and the structural topology of the communication that need to be reasoned about.

In order to effectively reason about these choices it is important to account for the complex relationship between the environment and the CAS that is often nondeterministic along multiple dimensions. For example, Netflix was motivated to build Scryer and have multiple geographic deployments to maintain quality standards in an environment that is subject to redundant unpredictable spikes in traffic, which resulted in increased reliability and resiliency. Therefore, accounting for the nondeterminism along multiple dimensions in both the environment and the CAS is an important factor to consider in understanding the emergent properties of CASs.

In this paper we propose an approach to reasoning about emergent properties of collective adaptive systems that uses probabilistic model checking (PMC) of stochastic multiplayer games (SMG). Probabilistic Model Checking of SMGs is a technique particularly suited to analyzing emergent properties in CAS, since SMG models are expressive enough to capture: (i) the uncertainty and variability intrinsic to the CAS and its execution environment in the form of probabilistic and nondeterministic choices, and (ii) the competitive/cooperative aspects of the interplay among the constituent systems of the CAS (as well as of the CAS with its environment). These cooperative/competitive behaviors can be modeled as players in a game whose behavior is independent (i.e., not controlled by other entities).

We illustrate our approach on a model of a CAS with identical fully cooperative systems that are attempting to defend against an external attack. The model enables us to explore the emergent properties of resiliency and reliability that result from the selection of different communication topologies (line, ring, mesh, tree, star, and full) to disseminate security information for preemptive adaptation. In the scenario, an external attacker uses a defined amount of available resources to attempt to breach members of the CAS. Each member of the CAS has the ability to detect the attack, defend itself against it by employing a fixed set of defense resources, and has the ability to notify other members of the CAS of the attack. Once a CAS member is notified, it will adapt and become invulnerable to the attempted breach. The metric for evaluation

is the percentage of members of CAS that survive the attack. Performing a PMC analysis of this model allows us to reason about questions such as topologies that are most appropriate for a given CAS, and what is a worst case scenario for each topology under consideration.

Our results show distinct differences in the resiliency and reliability properties of the network topologies, demonstrating that stochastic multiplayer games and probabilistic model checking can enable developers to reason about emergent properties in representative scenarios utilizing cooperative CAS.

The remainder of this paper is organized as follows: Section 2 highlights background and related work, Section 3 discusses a motivating scenario, Section 4 details the model and analysis technique employed in our study, Section 5 presents the results, and Section 6 addresses conclusions and future work.

II. Background & Related Work

Network topologies or the "geometrical arrangement of computer resources, remote devices, and communication facilities" [3] are well studied with defined advantages and disadvantages. Figure 1 illustrates the six different possible communication topologies. While it is possible to compose more topologies by mixing these six topologies, in this work we consider only these "pure" topologies. Two critical properties that are often in conflict based upon the scenario, and therefore traded off, are the performance and reliability of the messages sent across a topology. For example, in disaster scenarios the reliability of the messages is a critical concern which has made "mesh" topologies a popular choice [4]. However, in other scenarios, like security, performance and reliability might be equally important leading to the choice of a different network topology, like a "star" or "full".

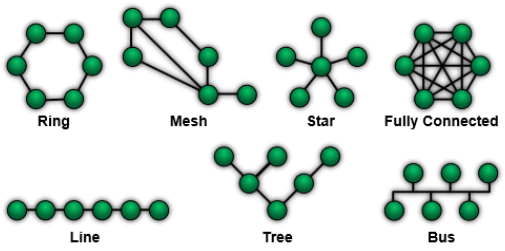

Fig. 1. Common Network Topologies [5]

Our approach to analyzing emergent properties in CAS builds upon a recent technique for modeling and analyzing SMGs [6]. In this approach, systems are modeled as turn-based SMGs, meaning that in each state of the model only one player can choose between several actions, the outcome of which can be probabilistic. Players can either cooperate to achieve the same goal, or compete to achieve their own goals.

The approach includes a logic called rPATL for expressing quantitative properties of stochastic multiplayer games, which extends the probabilistic logic PATL [7]. PATL is itself an extension of ATL [8], a logic extensively used in multiplayer games and multiagent systems to reason about the ability of a set of players to collectively achieve a particular goal. Properties written in rPATL can state that a coalition of players has a strategy which can ensure that either the probability of an event's occurrence or an expected reward measure meets some threshold.

rPATL is a CTL-style branching-time temporal logic that incorporates the coalition operator $\langle\langle C \rangle\rangle$ of ATL, combining it with the probabilistic operator $\mathsf{P}_{\bowtie q}$ and path formulae from PCTL [9]. Moreover, rPATL includes a generalization of the reward operator $\mathsf{R}^{\mathsf{r}}_{\bowtie x}$ from [10] to reason about goals related to rewards. An extended version of the rPATL reward operator $\langle\langle C \rangle\rangle \mathsf{R}^{\mathsf{r}}_{\mathsf{max}=?}[\mathsf{F}\ \phi]$ enables the quantification of the maximum accrued reward r along paths that lead to states satisfying state formula ϕ that can be guaranteed by players in coalition C, independently of the strategies followed by the rest of players. An example of typical usage combining the coalition and reward maximization operators is $\langle\langle \mathsf{car} \rangle\rangle \mathsf{R}^{\mathsf{distance}}_{\mathsf{max}=?}[\mathsf{F}\ \mathsf{fuel\_level} = 0]$, meaning "value of the maximum $\mathsf{distance}$ that a car can guarantee to have travelled before its fuel tank is empty." Notice that in this example, the rPATL expression is capturing the maximum distance that the car can travel in the worst case, regardless of disturbances or the action of other agents present in its environment.

Reasoning about strategies is a fundamental aspect of model checking SMGs, which enables checking for the existence of a strategy that is able to optimize an objective expressed as a property including an extended version of the rPATL reward operator. The checking of such properties also supports *strategy synthesis*, enabling us to obtain the corresponding optimal strategy. An SMG strategy resolves the nondeterministic choices in each state, selecting actions for a player based on the current state and a set of memory elements.[1]

Probabilistic model checking of SMGs has been applied to a variety of analysis and synthesis problems [11], [12]. In the context of self-adaptive systems, we presented in previous work [13] an analysis technique based on model checking of SMGs to quantify the potential benefits of employing different types of algorithms for self-adaptation. Specifically, the paper shows how the technique enables the comparison of alternatives that consider tactic latency information for proactive adaptation with those that are not latency-aware. We have also applied this analysis technique to reason about human-in-the-loop adaptation [14], extending SMG models with elements that encode an extended version of Stitch adaptation models [15] with constructs that capture information about humans interacting with the system. In all of this prior work we were concerned with a single adaptive system. In this paper we show how the same techniques can be used to reason about CASs.

In [16], the authors describe a similar approach in which they introduce a four step process that includes modeling, simulation, formal verification, and tuning in an effort to "drive design choices until the required quality attributes are obtained." While similar in mechanics, our approach is more focused on the use of these techniques to improve run-time adaptation of the CASs to improve both local and global quality attributes.

[1] See [6] for more details on SMG strategy synthesis.

III. Example Scenario

To illustrate our approach, consider a global enterprise, similar to Netflix, that has deployed multiple customer and transaction management systems, one in each of their distinct operating geographies (North America, South America, Europe, and Asia-Pacific). Each of these deployments is similar in its functional and non-functional requirements as well as its physical deployments, with some minor variations in software packages and versions. To ensure quality standards are met, each deployment has a self-adaptation manager with a locally defined set of tactics and strategies specific for that deployment.

Due to the sensitive nature of the data contained within the systems and the global profile of the enterprise, these systems are under constant attack by external parties. As such, the self-adaptive mechanisms are being modified to include the ability to detect and identify potential malicious behavior and appropriately adapt the system to mitigate the threat. For example, if the self-adaptation manager in one deployment determines that the deployment is compromised, it can adapt the deployment to a new, more secure configuration (e.g. blocking traffic from a malicious IP). However, the engineers implementing the system are concerned that an attacker could potentially attempt to breach a system, trigger an adaptation, and in the process gain enough information to make an attack on one of the other systems more effective. Therefore, the engineers want the self-adaptation managers to share information about current breach attempts and mitigation strategies to promote preemptive adaptation increasing resiliency and reliability.

In this scenario nondeterminism arises from the uncertainty relating to (i) which of the component systems an attacker will attempt to breach, (ii) the likelihood that an adaptation manager will detect an attack, (iii) whether a message sent by an adaptation manager will successfully reach other adaptation managers, and (iv) the resources that an attacker can deploy to breach the system. This scenario also presents multi-dimensional variations in the form of the number of adaption managers in the CAS, the reliability of the communication channels, and the amount of resources available to the attacker.

The run-time selection of the appropriate communication topology for the sharing of information becomes an important decision as it needs to balance the performance and reliability of information dissemination assuming the system could be compromised at any moment. For example, a system that detects a breach could select a "star" pattern in which it quickly notifies all other members, but if it is compromised and the system is unable to send further messages, a subset of systems will remain permanently unaware of the breach. However, a "mesh" topology could improve reliability of the messages, but might be slower to achieve complete dissemination. Further, the preferred option could potentially vary with several of the scenario factors including the number of adaptation managers in the collective and the reliability of the communication channels. We therefore require an approach that allows us to explore and examine the resilience properties of each of these possibilities.

IV. Model & Analysis

The approach that we use to examine the resilience properties of different communication patterns is stochastic multiplayer games. Specifically, we model all self-adaptive managers as one player and attackers as a separate player in a turn-based game. Although self-adaptive managers could have been modeled as individual players, we chose to model them as a coalition of different stochastic processes under the control of a single player, which is enough to capture the description of a CAS with fully cooperative behavior. CASs in which self-adaptive managers have to compete for shared resources demand modeling of self-adaptive managers as separate players. This section describes the mechanics of the model including the players and the turn-based semantics, as well as the analysis methods used to produce the reported results.

A. Game Model

The game includes two players: the attacker (ATT) and the defenders (CAS). The game alternates between the defenders and the attacker until all of the members of the CAS are in either a "Compromised" or in "Adapted" state. How these states are achieved is discussed below, along with the turn-based semantics of the attacker and defenders. Furthermore, the communication topology for the defenders is a global property of the model and is encoded by setting which members of the CAS an individual member can communicate with. The reliability of the communication channels is also a variable of the model which is an important factor with large-scale globally distributed systems.

1) Defender: The defenders, a CAS, are modeled as a collection of individual entities with a fixed amount of defense resources available to repel attacks. Each of the adaptive systems also detects attack attempts that are dependent upon the amount of resources the attacker has deployed; the more resources deployed as a percentage of the total attack resources available the greater the likelihood of detection. In the event that the attack is detected, the system will attempt to alert other members in the collective in accordance with the defined topology. The success of notification is controlled by the reliability of the communication channel.

For example, in a "star" topology the detecting system will attempt to alert all of the other members in the network. In a "mesh" topology the system will alert a predefined subset of partner systems which will then adapt and alert additional members of the network. If a system is notified of an attack, it must wait until its next turn to start the adaptation and the adaptation itself takes one turn. If the attack resources deployed against a particular adaptive system exceed the defense resources available that system will be considered compromised at which point the system has one last opportunity to send notification messages with a lower probability of success.

The defender's mechanics model the scenario in which individual members of the CAS have some resources available to detect and repel attacks as well as notify other members, all of which are nondeterministic in nature.

2) Attacker: The attacker is modeled as a single entity with a fixed amount of attack resources available per turn to breach

target systems. These resources can be deployed against any number of targets, as the attacker sees fit. The selection of the amount of resources employed and the system targeted for the attack during a turn are specified as non-deterministic choices in the model. This is accomplished by establishing the set of candidate target systems and cycling through the list assigning a block of available attack resources to the selected target system. There are only two conditions under which the attacker would not choose to target a specific system: (i) the system has already been compromised, or (ii) the system has already successfully adapted.

The attacker's mechanics model the scenario in which an intelligent attacker will not target adapted or compromised systems allowing themselves to efficiently deploy available resources, attack any system on any turn and do so with varying levels of resources. These varying levels of resources will also influence the defender's ability to detect the attack and the likelihood of the attack being successful.

The stop condition for the game is given when all the defenders are either adapted or compromised, or alternatively, the attacker's resources are exhausted. To enable the quantification of the outcome of the game, we explicitly label end states of the game as stop, and define a reward structure compromised that maps each end state to the number of compromised defenders in that state.

B. Analysis

The established model was analyzed both as an SMG and as a Discrete Time Markov Chain (DTMC), each yielding different, but complementary, results. The SMG analysis is used to perform a comprehensive model check to provide a *worst case* scenario analysis, but suffers from an explosion in the state-space for large networks, greater than 6 members in the CAS for this particular model. The DTMC analysis uses statistical model checking to analyze larger network sizes, the results of which correspond to the *average* behavior of the network.

The SMG analysis evaluated each of the six network topologies with a CAS composed of 6 members and varied the amount of attacker resources. Similarly, the DTMC analysis evaluated each of the network topologies with CASs composed of 6, 12, and 24 members. Two parameters were varied: the amount of attacker resources available and the notification probability, to determine the percentage of compromised systems.

1) Stochastic Multiplayer Game (SMG): To analyze the worst case scenario, we model checked the following rPATL property on the SMG version of our models employing PRISM-games [17]:

$$\langle\langle \mathsf{ATT} \rangle\rangle \mathsf{R}^{\mathsf{compromised}}_{\mathsf{max}=?}[\mathsf{F\ stop}]$$

The property quantifies the maximum number of compromised systems in stop states that the attacker can guarantee, regardless of the strategy followed by the defenders in the CAS.

2) Discrete Time Markov Chain (DTMC): To analyze the average behavior, we model checked the following PCTL property on the DTMC version of our models employing the statistical model checking engine of PRISM [18]:

$$\mathsf{R}^{\mathsf{compromised}}_{=?}[\mathsf{F\ stop}]$$

In this case, the PCTL property above quantifies a probabilistic estimate of the number of compromised systems in stop states. Note that in the DTMC version of the models, the behavior of the attacker and the CAS is specified in a fully probabilistic fashion, therefore player strategies are not considered for this kind of analysis.

V. Results

Both the SMG and DTMC analyses demonstrated clear differences in the reliability and resiliency properties of the various network topologies when applied to a CAS of systems defending against and cooperating to share information about security attacks.

A. SMG Analysis

The results of the SMG analysis (Figure 2) show the worst case scenario for each of the network topologies, with a 6 member CAS, given a varying level of attacker resources. As depicted, the "line" topology has a bleak worst case scenario with 100% (6/6) of the systems being compromised (the first system compromised preventing the rest from being notified) resulting in poor resiliency within the CAS. However, the "full" network topology has much better resiliency with only 38.7% (2.32/6)[2] of the members compromised. The other topologies have varying degrees of resiliency between these two extremes.

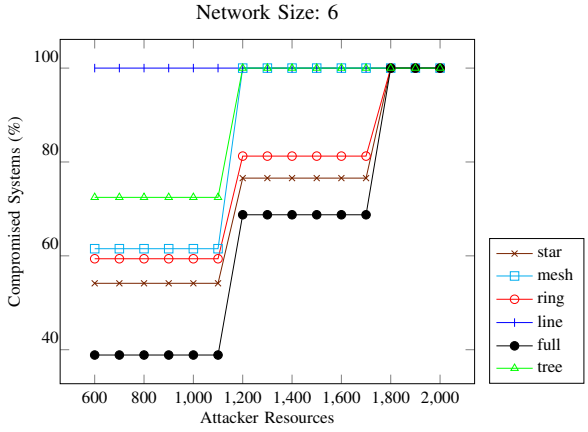

Fig. 2. Compromised systems for different topologies

Another interesting pattern develops when the attacker's resources cross a specific threshold, specifically 1100. At this point it is possible for the attacker to compromise more than one member of the CAS during the initial attack. This leads to a much higher number of compromised systems but the general pattern of resiliency of the network topologies remaining consistent. Similarly, at 1800, the attacker has enough resources to compromise enough of the CAS to be able to compromise the complete CAS.

[2]Note that a worst case can result in fractional number of servers down due to probabilistic choices that are not under the control of any of the players in the game (e.g., successful notification based on channel reliability).

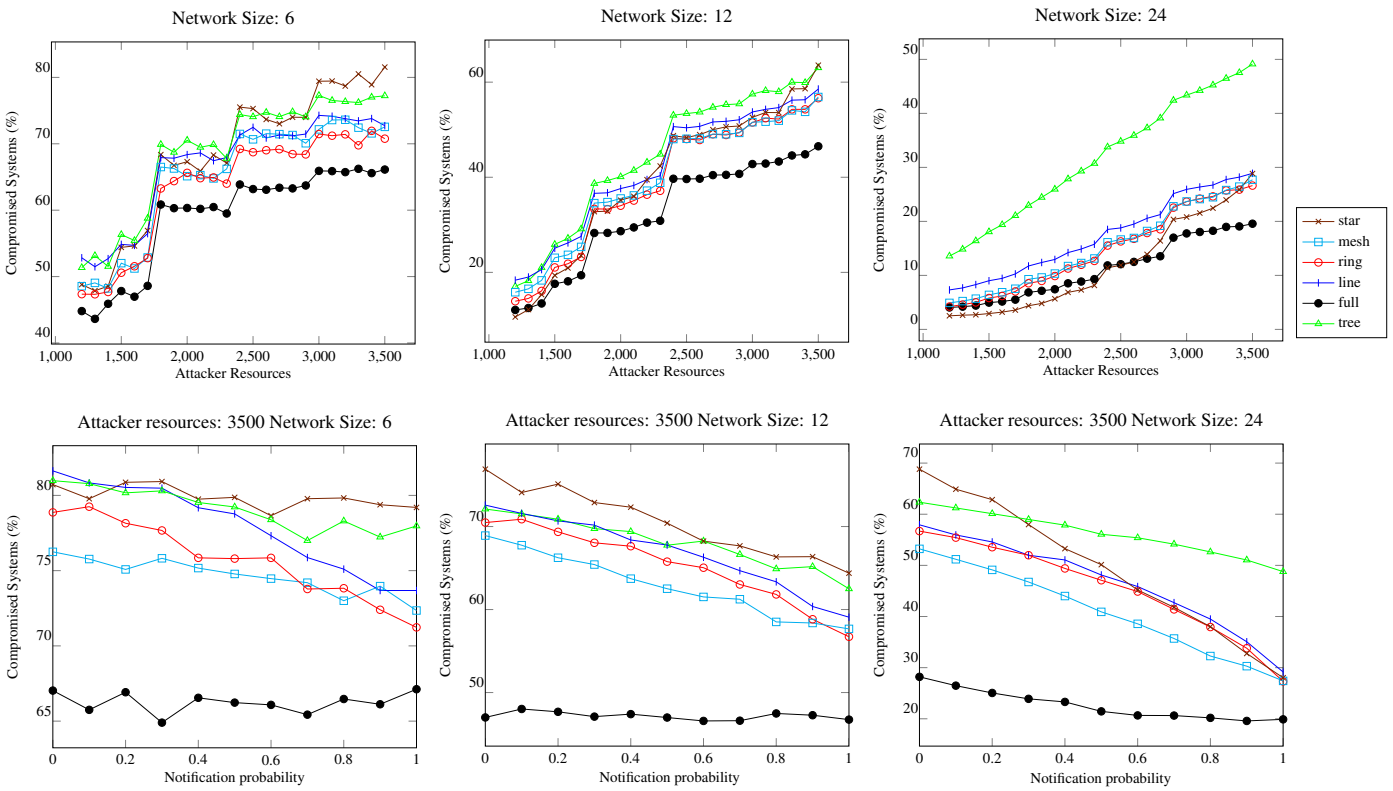

Fig. 3. Compromised systems for different topologies (average)

B. DTMC Analysis

The results of the DTMC analysis (Figure 3) show the average case scenario for each of the network topologies for CASs with 6, 12, and 24 members. The first set of analyses (the top row of Figure 3) varies the amount of attacker resources in order to determine the survival rate of the CAS. The second set of analyses (bottom row of Figure 3) varies the probability of successful notification to examine the relationship between the reliability of the communication channel and the survival rate of the CAS.

Varying the attacker resources and the number of members in the the network yielded interesting results. Specifically, while the "full" topology generally presents the best survival rate, there are places in which it might not be preferred, such as when attacker resources are low for large network sizes, in which case the "star" topology is preferred. Additionally, the "tree" topology is generally equivalent to a "star" topology in networks of 6 nodes, but becomes dramatically less preferred as the size of the networks grow.

Varying the notification probability produced equally interesting results. Specifically, in situations in which notification success is unreliable or uncertain, like with Internet of Things (IoT) use cases, a "star" topology clearly presents the best chance for survival in the CAS. This analysis once again shows a general pattern in which the "tree" topology becomes less preferable as the size of the members in the CAS grows.

The two analyses can be combined to understand the resiliency and reliability properties of the individual topologies. Figure 4 combines the results of the DTMC and SMG analysis,

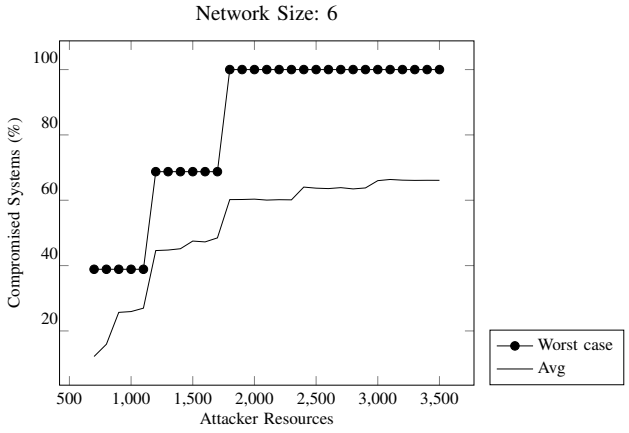

Fig. 4. Compromised systems for fully connected network: average vs. worst case scenario

for network size of 6, of the "full" topology to demonstrate the differences between the average and worst case scenario.

C. Reflection

While none of these analyses present conclusive results due to the simplifying assumptions in our models, they do demonstrate the potential for SMG and DTMC analysis to enable software engineers to discover, understand, and reason about emergent properties in CAS. For example, the results present the possible conclusion that a hierarchical or "tree" based decomposition of a CAS may be inappropriate, especially with increasing network size, in security based use cases.

Conversely, the "full" topology is potentially the most effective selection, despite the network size and attacker resources available.

VI. Conclusions & Future Work

In this paper we have shown how emergent resilience properties can be analyzed using SMGs and DTMCs. While the model contains many of the same nondeterministic factors inherent in realistic situations, the model currently contains several simplifying assumptions such as (1) the network distance between the components is ignored, leading to identical message transmission times, (2) the systems are nearly identical leading to identical adaptation times, (3) the defenders each have a pool of resources available instead of a pool of shared resources and (4) turn based semantics of the model limit evaluation of concurrent executions among CAS components. Relaxing each of these assumptions are targets for future work. However, the results of the SMG and DTMC analyses of this model clearly demonstrate differences in the resiliency and reliability properties of the various network topologies and, as a consequence, shows the potential of these methods to discover, understand, and reason about emergent properties in CAS.

We also intend to extend our work by studying the interplay among systems within collectives in which behavior is not fully cooperative. This includes collectives in which some systems give preference to their local goals over the global goals of the collective, as well as those in which systems compete for shared resources. A second research avenue is investigating the synthesis of topologies to satisfy emergent properties in the CAS. In the simplest case, synthesis can be employed to generate an optimal allocation of resources within a pre-established network with respect to a given property (e.g., resiliency), although we also plan to explore optimal topology synthesis. Another possible extension to this work is investigating different communication protocols between different adaptive systems, for example whether to adapt first and then share the information, or whether to share first and then adapt. In fact, we hypothesize that the approach could also be used to generate the optimal communication protocol for different topologies. Finally, we also intend to explore the use of formalisms that support the modeling of real-time behavior for investigating the impact of time (e.g., to adapt, notify) on the properties of the CAS.

Acknowledgment

This work is supported in part by awards N000141310401 and N000141310171 from the Office of Naval Research, CNS-0834701 from the National Science Foundation, and by the National Security Agency. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the U.S. government.

References

[1] D. Jacobson, D. Yuan, and N. Joshi, "Scryer: Netflix's predictive auto scaling engine," http://techblog.netflix.com/2013/11/scryer-netflixs-predictive-auto-scaling.html, Dec. 2013, [Online; accessed July, 2015].

[2] R. Meshenberg, N. Gopalani, and L. Kosewski, "Active-active for multi-regional resiliency," http://techblog.netflix.com/2013/12/active-active-for-multi-regional.html, Dec. 2013, [Online; accessed July 2015].

[3] A. Gokhale, *Introduction to Telecommunications*. Cengage Learning, 2004.

[4] N. Aschenbruck, C. de Waal, and P. Martini, "Distribution of nodes in disaster area scenarios and its impact on topology control strategies," in *INFOCOM Workshops 2008, IEEE*, April 2008, pp. 1–6.

[5] W. Commons. (2011) Diagram of different network topologies. [Online]. Available: https://en.wikipedia.org/wiki/Network_topology

[6] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "Automatic verification of competitive stochastic systems," *Formal Methods in System Design*, vol. 43, no. 1, pp. 61–92, 2013.

[7] T. Chen and J. Lu, "Probabilistic alternating-time temporal logic and model checking algorithm," in *Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on*, vol. 2, Aug 2007, pp. 35–39.

[8] R. Alur, T. A. Henzinger, and O. Kupferman, "Alternating-time temporal logic," in *Revised Lectures from the International Symposium on Compositionality: The Significant Difference*, ser. COMPOS'97. London, UK, UK: Springer-Verlag, 1998, pp. 23–60.

[9] A. Bianco and L. de Alfaro, "Model checking of probabilistic and nondeterministic systems," in *Foundations of Software Technology and Theoretical Computer Science*, ser. Lecture Notes in Computer Science, P. Thiagarajan, Ed. Springer Berlin Heidelberg, 1995, vol. 1026, pp. 499–513.

[10] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker, "Automated verification techniques for probabilistic systems," in *Formal Methods for Eternal Networked Software Systems*, ser. Lecture Notes in Computer Science, M. Bernardo and V. Issarny, Eds. Springer Berlin Heidelberg, 2011, vol. 6659, pp. 53–113.

[11] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Controller synthesis for autonomous systems interacting with human operators," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, ser. ICCPS '15. New York, NY, USA: ACM, 2015, pp. 70–79.

[12] T. Chen, M. Kwiatkowska, A. Simaitis, and C. Wiltsche, "Synthesis for multi-objective stochastic games: An application to autonomous urban driving," in *Quantitative Evaluation of Systems*, ser. Lecture Notes in Computer Science, K. Joshi, M. Siegle, M. Stoelinga, and P. DArgenio, Eds. Springer Berlin Heidelberg, 2013, vol. 8054, pp. 322–337.

[13] J. Cámara, G. A. Moreno, and D. Garlan, "Stochastic game analysis and latency awareness for proactive self-adaptation," in *9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, Hyderabad, India, 2-3 June 2014.

[14] ——, "Reasoning about human participation in self-adaptive systems," in *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2015)*, Florence, Italy, 18-19 May 2015.

[15] S.-W. Cheng and D. Garlan, "Stitch: A language for architecture-based self-adaptation," *Journal of Systems and Software, Special Issue on State of the Art in Self-Adaptive Systems*, vol. 85, no. 12, December 2012.

[16] L. Gardelli, M. Viroli, and A. Omicini, "Combining simulation and formal tools for developing self-organizing MAS," in *Multi-Agent Systems: Simulation and Applications*, ser. Computational Analysis, Synthesis, and Design of Dynamic Systems, A. M. Uhrmacher and D. Weyns, Eds. CRC Press, Jun. 2009, ch. 5, pp. 133–165.

[17] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis, "Prism-games: A model checker for stochastic multi-player games," in *Tools and Algorithms for the Construction and Analysis of Systems*, ser. Lecture Notes in Computer Science, N. Piterman and S. Smolka, Eds. Springer Berlin Heidelberg, 2013, vol. 7795, pp. 185–191.

[18] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Proceedings of the 23rd International Conference on Computer Aided Verification*, ser. CAV'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 585–591.