Modeling, Monitoring and Scheduling Techniques for Network Recovery from Massive Failures

Diman Zad Tootaghaj*, Thomas La Porta†, Ting He†
*Hewlett Packard Enterprise Labs (USA), †The Pennsylvania State University (USA),
{*tlp, tzh58*}@*cse.psu.edu*, {diman.zad-tootaghaj}@hpe.com

***Abstract*—Large-scale failures in communication networks due to natural disasters or malicious attacks can severely affect critical communications and threaten lives of people in the affected area. In the absence of a proper communication infrastructure, rescue operation becomes extremely difficult. Progressive and timely network recovery is, therefore, a key to minimizing losses and facilitating rescue missions. To this end, we focus on network recovery assuming partial and uncertain knowledge of the failure locations. We proposed a progressive multi-stage recovery approach that uses the incomplete knowledge of failure to find a feasible recovery schedule. Next, we focused on failure recovery of multiple interconnected networks. In particular, we focused on the interaction between a power grid and a communication network. Then, we focused on network monitoring techniques that can be used for diagnosing the performance of individual links for localizing *soft failures* (e.g. highly congested links) in a communication network. We studied the optimal selection of the monitoring paths to balance identifiability and probing cost. Finally, we addressed, a minimum disruptive routing framework in software defined networks. Extensive experimental and simulation results show that our proposed recovery approaches have a lower disruption cost compared to the state-of-the-art while we can configure our choice of trade-off between the identifiability, execution time, the repair/probing cost, congestion and the demand loss.**

***Index Terms*—Network Recovery, Massive Disruption, Optimization, Uncertainty, Cascading Failures, Interdependent Networks, Power Grid, Software-Defined Networking**

I. Introduction

Large-scale failures due to natural disasters or malicious attacks can severely affect operation of critical infrastructures and cause catastrophic economic and social disruptions. Communication networks and power grids are examples of such critical infrastructures that are highly vulnerable to such failures. In 2005, Hurricane Katrina led to outage of over 2.5 million lines in the BellSouth (now AT&T) network [1]. In 2003, a large cascading blackout, in northeast of the United States, led to over 50 million people losing power, some for several days. The overall cascade propagation lasted approximately four hours, during which a cascade prevention mechanism could have stopped further propagation of the failure and lowered cost of recovery.

The leading causes of these failures has been reported to be inadequate training, planning and operations studies to respond to the emergency situations [2, 3], which highlights the necessity for a holistic control and recovery approach that has the ability to use the real-time data taken from a monitoring network to predict and prevent possible failures. Furthermore, it is crucial to have a strategic recovery plan that effectively utilizes the available resources and maximizes the total operation of the disrupted services during the recovery time.

In this paper, we present the contribution of thesis [4]. We first study large-scale failures in (i) communication networks in Section II-A and (ii) an interdependent power grid and it's monitoring network in Section II-B. We then focus on network monitoring techniques that can be used for diagnosing individual links' performance or localizing the failures. In Section II-C, We study the optimal selection of the monitoring paths to balance identifiability and cost. In Section II-D, we study a minimum disruptive way of updating on flow rules for software defined networks.

978-3-903176-15-7 © 2019 IFIP

A. Motivation and Challenges

Despite considerable research in the past few decades leading to multi-fold improvements on large-scale failure detection and mitigation approaches, the problem has become more interesting and challenging for three main reasons: (1) Lack of complete knowledge, (2) Interdependency between multiple networks, and (3) Progressive recovery. In the following subsections, we summarize the three reasons.

1) Lack of Complete Knowledge: Almost all failure recovery and prevention algorithms assume complete information about the failures and ignore the uncertainty caused by failure of the dependent monitoring systems. In a real system, one needs to assume that only incomplete data may arrive at control centers due to failures in the underlying communication and monitoring network. Network recovery and failure prevention is a challenging problem under uncertainty of the exact location of the disrupted network components.

To clarify the discussion, consider Deltacom topology taken from the Internet Topology Zoo shown in Figure 1 [5, 6]. After a large-scale failure occurs in the network, the state of the entire network is not visible to the network manager or control centers. Instead, the network manager knows that some nodes and links have failed, some continue to work and the fate of others is uncertain. The working nodes and links are shown with green color in Figure 1, the broken nodes and links are

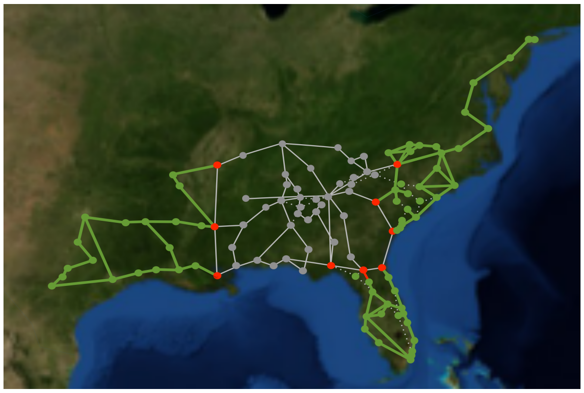


Fig. 1: An example of a failure in a real network topology from the internet topology zoo [6].

shown in red and the uncertain nodes and links are shown in grey. The main challenge is that the status of grey nodes and links is unknown to the network manager and therefore, current recovery techniques that assume complete and accurate information is accessible, might not work as they should. To this end, we propose an itrative stochastic recovery approach (ISR) in [7], that runs a multi-stage stochastic optimization algorithm. At each iteration step, ISR finds a feasible solution set and selects a candidate node to repair and exploits it as a monitor to discover the surrounding network. The procedure is repeated until all critical services are restored.

2) Inter-dependency between Multiple Networks: Most of the research on large-scale failure management has concentrated on the recovery of a single network [8, 9, 10]. Many man-made or natural systems can be modeled as an interconnection of multiple networks, where the nodes are the system components and the edges show the interaction or dependency between different components. Because of the dependency between different components in multiple networks, perturbations caused by physical attacks or natural disasters in one node can cascade and affect other nodes in the system. The cascaded failure can repeat multiple times, feeding on itself and accelerating, eventually resulting in a total failure of the whole system.

Today, critical infrastructures are becoming increasingly correlated and interdependent. Therefore, modeling and understanding the interactions between multiple networks and designing failure resilient infrastructures is crucial for the reliability and availability of many applications and services. In particular we study the inder-dependency between a power grid and a communication network. The communication network provides monitoring and controllability to the power grid and the power grid provides power to the communication network. We tackle the problem of mitigating the ongoing cascading failure and providing a recovery strategy. We propose a failure mitigation strategy in two steps: 1) Once a cascading failure is detected, we limit further propagation by re-distributing the generator and load's power. 2) We formulate a recovery plan to maximize the total amount of power delivered to the demand loads during the recovery intervention.

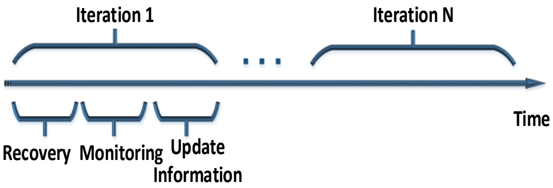


Fig. 2: Different steps of our progressive recovery approach.

3) Progressive Recovery: Restoring critical services after a large-scale disruption or a cascaded failure is not a one shot operation. The amount of repair resources required to restore damaged network elements may vary over time. Also, each network element might require a different amount of resources to be restored. Therefore, finding the optimal assignment of resources to maximize the recovery over time is a challenging problem. To the best of our knowledge, the proposed progressive recovery approach is the first work that studies progressive recovery of a disrupted network under uncertainty. Figure 2 shows different steps of the proposed recovery approach. At each iteration step, based on the available resources, the proposed algorithm repairs some of the damaged network elements, performs a monitoring step and gains more information and iterates this procedure until all critical services are restored.

II. Our Contribution

This paper aims to summarize the main contributions in [4] where we provide comprehensive solutions for accurately modeling, monitoring and scheduling the recovery of the network from large-scale failures under uncertain knowledge of failures. Specifically, we focus on four main goals: (1) minimizing the number of repaired elements, (2) minimizing the amount of demand loss, (3) minimizing the recovery time, and (4) minimizing the cost of monitoring probes. We briefly explain these goals in the following four subsections.

A. Network Recovery from Massive Failures under Uncertain Knowledge of Damages

In [7, 11], we tackle for the first time, the problem of network recovery after massive disruption under uncertainty of the exact location of the disrupted nodes/links. we formulate the *minimum expected recovery* (MINER) problem as a mixed integer linear programming and show that it is NP-Hard. MINER aims at satisfying the critical demand flows while minimizing the proposed expected recovery cost (*ERC*) function under network capacity constraints. The proposed iterative stochastic recovery (ISR) approach recovers the network in a progressive manner while satisfying the critical service demands [7]. ISR runs in three variants, namely, Iterative shortest path (ISR-SRT), Iterative branch and bound (ISR-BB),

and iterative multi-commodity (ISR-MULT). The skeleton of these versions follow the same structure and only differ in terms of the approximate algorithm they use. We summarize ISR algorithm in six main steps shown in Figure 3.
Given an undirected graph $G = (V, E)$ and a set of demand pairs $E_H = \{(s_1, t_1), ..., (s_k, t_k)\}$, where $E_H \subseteq V \times V$ and each demand pair $(s_h, t_h) \in E_H$ has a source s_h, a destination t_h and a positive demand flow d_h, the goal is to minimize the expected recovery cost (ERC) to satisfy the demands while having capacity constraint c_{ij} for every edge in the graph. The nodes and edges in the graph $G = (V, E)$ belong to three different categories:

1) the sets $E_B \subseteq E$ and $V_B \subseteq V$ are the set of **broken** edges and nodes in the red area which we know for sure have failed,
2) the sets $E_U \subseteq E$ and $V_U \subseteq V$ are the sets of edges and nodes in the gray area whose failure patterns is **unknown**,
3) the sets $E_W \subseteq E$ and $V_W \subseteq V$ are the sets of nodes and edges in the green area which are known to be **working** correctly in the system.

Initially, ISR starts by estimating the probability distribution of the network failure (Step 1). At each iteration, ISR uses an approximate algorithm to build a partial solution set of candidate network components to repair, $S_t = \{(i \in V_U \cup V_B \quad |\delta_i = 1), ((i, j) \in E_U \cup E_B \quad |\delta_{ij} = 1)\}$ (Step 2). Where the binary variables δ_{ij} and δ_i represent the decision to use link $(i, j) \in E$ and node $i \in V$ in the routing (when $\delta_{ij} = 1$, $\delta_i = 1$) or not (when $\delta_{ij} = 0$, $\delta_i = 0$).

In our evaluation, we do not consider infeasible problems, i.e., there exists at least one feasible solution which can satisfy all critical services.

We use three different optimization techniques explained in [7] to build the partial solution set. The partial solution minimizes the MINER problem based on the current estimated costs which can change as we gain more knowledge about the gray area. In step 3, the nodes in the partial solution set S_t, are ranked based on the amount of flow in critical services that they are likely to route, and a node with the maximum value is selected as a candidate node (Steps 3 and 4). We repair the candidate node, and use it to monitor (Step 5) the surrounding network and obtain more information about the status of the network. In step 6, the algorithm updates the previous estimate of the costs after the discovery. The procedure is repeated until all the demands are satisfied or no more repairs are possible although there is a demand loss. At each iteration step, ISR makes a decision to repair a part of the network and gathers more information by putting a monitor on the selected node. we propose several algorithms to find a feasible solution set at each iteration of the algorithm. Experimental results show that ISR outperforms the state-of-the-art ISP algorithm while having a configurable choice of trade-off between the execution time, number of repairs and demand loss.

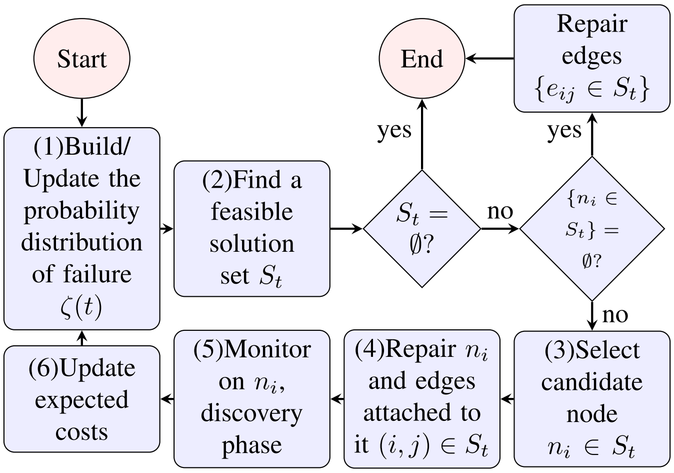

Start
End
Repair edges $\{e_{ij} \in S_t\}$
yes
yes
(1)Build/ Update the probability distribution of failure $\zeta(t)$
(2)Find a feasible solution set S_t
$S_t = \emptyset$?
no
$\{n_i \in S_t\} = \emptyset$?
no
(6)Update expected costs
(5)Monitor on n_i, discovery phase
(4)Repair n_i and edges attached to it $(i, j) \in S_t$
(3)Select candidate node $n_i \in S_t$

Fig. 3: Different steps of our iterative stochastic recovery (ISR).

B. Controlling Cascading Failures in Interdependent Networks under Incomplete Knowledge

In [12, 13], we show that the inter-connectivity and dependency between different elements makes complex networks more vulnerable to failure. we study the inter-dependency between a power grid and a communication network. The power and communication networks are modeled as undirected graphs $\mathcal{G}_p = (V_p, E_p)$ and $\mathcal{G}_c = (V_c, E_c)$, respectively. Each node i in the power grid is monitored by several sensors deployed nearby. The monitoring data is then sent to the node of the communication network which hosts the control functionalities related to node i of the power grid and is responsible of sending its monitoring data to the control center consisting SCADA/EMS and RAS/SPS. In addition, control commands are sent to the dependent communication node for generator re-dispatch or load shedding.

To clarify the interdependency model between the communication network and the power grid, consider the example shown in Figure 4. The figure shows the interdependency model between a communication network with 4 nodes $\{c_1, .., c_4\}$, and a power grid with 8 nodes $\{p_1, ..., p_8\}$. The red arrows show the interdependency between the two networks. For example, c_1 controls three power nodes $\{p_1, p_2, p_3\}$ and gets power from p_3. Now consider a failure in one of the communication nodes c_1. In this case three power grid nodes $\{p_1, p_2, p_3\}$ become uncontrollable as the controller cannot send the power adjustment control commands to them. Next, consider a failure in a node in the power grid p_8. In this case, the communication node c_4 that gets power from p_8 loses power and consequently the dependent power grid node p_7 becomes uncontrollable.

We propose a failure mitigation strategy that first detects the failure and limits further propagation of the disruption by re-distributing the generator and load's power. Figure 5 illustrates our two-phase approach. As described in the figure, whenever a new failure event shows up, a preliminary monitoring activity

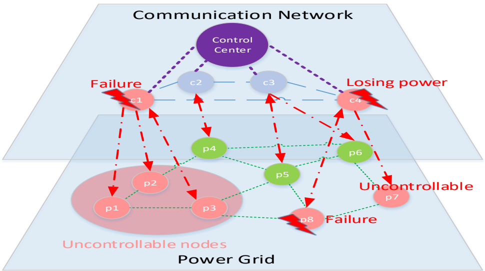


Fig. 4: Interdependency model between a power grid and a communication network.

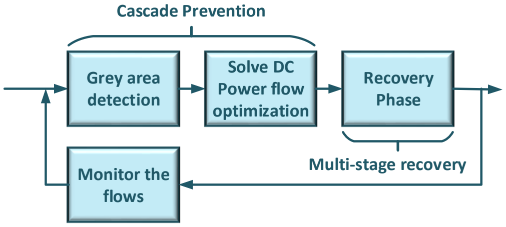


Fig. 5: Recovery Process: 1) Cascade mitigation phase, and 2) Recovery phase.

is performed to localize the failure sites. We propose a consistent failure set algorithm (*CFS*) to locate the failures. After the failure assessment it follows a first phase in which further cascades are mitigated or prevented by means of a combination of load shedding and adjustment of the generated power. The cascade mitigation problem is formulated as a linear programming optimization that minimizes the cost of new flow assignment (*Min-CFA*) and aims at finding a DC power flow setting that stops the cascading failure at minimum cost. Once the cascade is stopped, a progressive recovery activity follows. Since the recovery problem (*Max-R*) is NP-Hard, we propose a greedy (*Max-R-Greedy*) and a backward (*Max-R-Backward*) approach. We formulate the recovery plan to maximize the total amount of power delivered to the demand loads during the recovery intervention. Recovery is performed in multiple stages according to resource availability. After the system is recovered, the monitoring activity restarts, until new failures occur.

We compare the recovery performance of the proposed heuristics (*Max-R-Greedy* and *Max-R-Backward*). Figure 6 shows the total delivered power flow over different stages of progressive recovery intervention, when using the two algorithms. As shown, the greedy approach does not consider the correlation between different steps of the recovery approach and tries to maximize the added flow at each iteration step. On the other hand, the backward algorithm solves the problem using all repair resources in the beginning and removes the repair edges with less profit from the schedule of previous stage until all repair schedules are determined. Therefore, *Max-R-Backward* performs better than the *Max-R-Greedy* approach

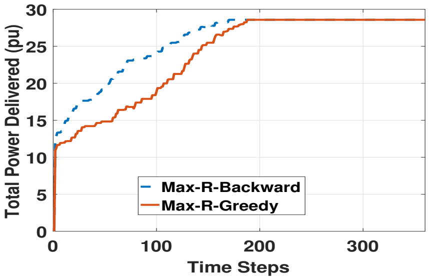


Fig. 6: Total delivered power (pu) flow over time for *Max-R-Backward* and *Max-R-Greedy* in the Italian power grid.

with larger total area behind the curve in Figure 6.

C. Optimizing Cost-Identifiability Trade-off for Probing-based Network Monitoring

In [14, 15], we study the optimal selection of monitoring paths to balance identifiablity and cost. Given an undirected graph $G(V, L)$, where V represents the network nodes and L is the set of communicating links connecting the nodes, and a set of nodes $M \subseteq V$ employed as monitors, the set P of routing paths between all pairs of monitors specifies the set of candidate probing paths that we can select from. Each link j in L is associated with an additive metric x_j (e.g., link delay). Given a set P of all possible probing paths (e.g., routing paths between all the monitors), let A be the routing matrix of size $|P| \times |L|$, such that if path $r \in P$ contains link j, then $A[r, j] = 1$ and $A[r, j] = 0$ otherwise. We can write a linear system of equations relating the link's additive metric (e.g. delay) to path metrics as $Ax =$ y. The objective of network tomography is to infer x from A and y.

The linear system of equations ($Ax =$ y) may not be invertible as the routing matrix A may not have a full column rank. To quantify the extent to which this system can be solved, we introduce two measures: **identifiability** and **rank**. The rank of P is calculated by the rank of the routing matrix A, denoted by $rank(A)$, which is the cardinality of the largest set of probing paths, such that each path in the set contains "new information" about the links (every other path is a linear combination of paths in the set and thus does not provide new information).

An illustrative example: Figure 7 shows an example of a network with 5 links and four candidate monitors $M = \{m_1, ..., m_4\}$. Using all possible paths between candidate monitors we have the following routing matrix.

$$A = \begin{array}{c} \\ \\ \end{array} \begin{array}{ccc|cc l} l1 & l2 & l3 & l4 & l5 & \\ 1 & 1 & 0 & 0 & 0 & : r_{m_1,m_2} \\ 1 & 0 & 1 & 0 & 0 & : r_{m_1,m_3} \\ 1 & 0 & 0 & 1 & 1 & : r_{m_1,m_4} \\ 0 & 1 & 1 & 0 & 0 & : r_{m_2,m_3} \\ 0 & 1 & 0 & 1 & 1 & : r_{m_2,m_4} \\ 0 & 0 & 1 & 1 & 1 & : r_{m_3,m_4} \\ & A_{*,L_1} & & \multicolumn{2}{c}{A_{*,L \setminus L_1}} & \end{array}$$

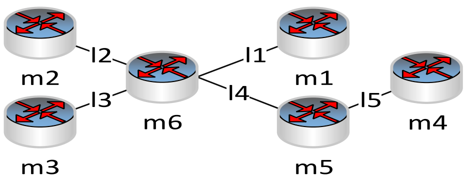


Fig. 7: A simple network example with 5 links and 4 monitors $\{m_1, \dots, m_4\}$. Candidate paths: r_{m_1,m_2}, r_{m_1,m_3}, r_{m_1,m_4}, r_{m_2,m_3}, r_{m_2,m_4}, r_{m_3,m_4}.

The rank of this matrix is 4 while the null space shows only 3 identifiable links $l1, l2, l3$. If we only probe paths in $R = \{r_{m_1,m_2}, r_{m_1,m_3}, r_{m_2,m_3}\}$, the corresponding routing matrix A_R can identify all 3 identifiable links.

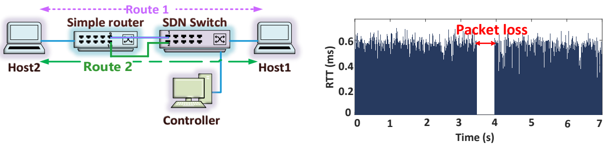
$$A_R = \begin{array}{c} \\ \left(\begin{array}{ccc|cc} l1 & l2 & l3 & l4 & l5 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{array} \right) \\ \quad A_{R,L_1} \qquad\qquad A_{R,L\setminus L_1} \end{array} \begin{array}{l} \\ : r_{m_1,m_2} \\ : r_{m_1,m_3} \\ : r_{m_2,m_3} \end{array}$$

Meanwhile, it is also clear that probing these paths suffices to identify l_1, l_2 and l_3. We considered four closely related optimization problems: (1) *Max-IL-Cost* that maximizes the number of identifiable links under a probing budget, (2) *Max-Rank-Cost* that maximizes the rank of selected paths under a probing budget, (3) *Min-Cost-IL* that minimizes the probing cost while preserving identifiability, and (4) *Min-Cost-Rank* that minimizes the probing cost while preserving rank. We show that while (1) and (3) are hard to solve, (2) and (4) posses desirable properties that allow efficient computation while providing good approximation to (1) and (3). We proposed an optimal greedy-based approach for (4) and proposed a $(1 - 1/e)$-approximation algorithm for (2). Experimental analysis reveals that, compared to several greedy approaches, our rank-based optimization performs better in terms of identifiability and probing cost.

D. A Minimally Disruptive Rule Update in Software Defined Networking

In this section, we study the problem of re-routing existing flows in a software defined network (SDN) to enable the admission of new flows while minimizing the disruption of existing flows under link capacity and Quality of Service (QoS) constraints [4, 16].

Software Defined Networks (SDN), which decouple the control plane and data plane, provide a powerful tool for network management, traffic engineering, and network policy enforcement. Decoupling the control functions and data plane brings significant advantages, including routing flexibility, being vendor agnostic, and centralized control and programmability [17, 18]. While SDN provides a rich framework for packet forwarding, forwarding rules may change frequently due to new traffic demands, topology changes, network congestion or failures. One of the important challenges in software defined networking is the ability to quickly react to changing network conditions, which requires fast and safe update of the flow table entries, without causing major disruptions to existing flows. Rule updates can disrupt the existing traffic by causing packet losses, delays, and security holes in the system [16, 19, 20, 21, 22].



(a) Testbed setup.

(b) Measured disruption.

Fig. 8: Disruption caused by re-routing the existing flow to accommodate new flows.

Figure 8 shows a preliminary experiment that characterizes the disruption due to flow rule updates in our SDN testbed which is shown in Figure 8a. We first install rules on a 24-port Brocade (ICX 6610) SDN switch to connect two hosts via a (non-SDN) router using *Route 1*. Then, we update the flow table in the Brocade switch to use *Route 2*. During the update, we send ICMP packets from *Host1* to *Host2* to measure the round trip time (RTT). Figure 8b shows RTT measurements, before, during and after the rerouting. The results show that rerouting the *Host1* → *Host2* flow from *Route 1* to *Route 2* disrupts the flow for about 500 ms, i.e. ICMP packets are lost during the 500 ms period.

In general, it is not always possible to find an update schedule that (i) preserves policy consistency, (ii) avoids congestion during the update, and (iii) satisfies all the demands. In [4, 16], we use two existing approaches to perform concurrent updates of multiple switch tables: (1) the two-phase update approach [19], which we call the ***synchronous*** update approach, and (2) the sequential update approach [23], which we call the ***asynchronous*** update approach. In the synchronous update approach, all updates need to wait for the slowest switch to complete the update, while in the asynchronous update approach, each flow gets an update independent of other flows, which privileges update time at the expense of temporary congestion and lack of consistency. We then show the trade-off between (ii), (iii) and disruption cost.

We introduce a minimally disruptive rule update problem (*Min-touch*) and show that it is NP-Hard. We propose two randomized rounding algorithms *RR-Cong* and *RR-Demand* with bounded approximation factors on congestion and demand loss. We show that under RR-Cong, the probability of violating the link capacity constraint for any link (i, j) by a factor of $1 + 5log(|E|)$ is no greater than $1/|E|^2$, i.e.,

$$Pr\left(\exists(i,j) \in E : \sum_{h\in H} f_{ij}^h \geq (1 + 5log(|E|)) \cdot c_{ij}\right) \leq 1/|E|^2 \qquad (1)$$

Under RR-Demand, the probability of violating the link capacity constraint for any link (i,j) is no greater than $1/|E|^2$, i.e.,

$$Pr\left(\exists(i,j)\in E:\ \sum_{h\in H} f_{ij}^{h} \geq c_{ij}\right) \leq 1/|E|^2 \tag{2}$$

By reducing the amount of routed flow on a chosen path by a factor of $(6\cdot log(|E|))$, RR-Demand satisfies link capacity constraints with high probability, and if the minimum link capacity is greater than or equal to the maximum demand, i.e. $min_{(i,j)\in E}(c_{ij}) \geq max_{h\in H}(d_h)$, we can satisfy a total demand of at least $\sum_{h\in H} d_h/(6\cdot log(|E|))$.

Experimental results on real network topologies demonstrated the effectiveness of the proposed approaches in terms of disruption cost, congestion and demand loss. The results indicate that our approaches have a disruption cost close to the optimal while having a low congestion factor and a low demand loss.

III. Conclusion

In this paper, we presented the contributions of the thesis [4] and provided comprehensive solutions to recover a network after massive disruption. We proposed novel schemes to monitor and recover a network under uncertain knowledge of failure while targeting four main goals: (1) minimizing the number of necessary repaired elements, (2) minimizing the amount of demand loss, (3) minimizing the recovery time and (4) minimizing the cost of monitoring probes. These critical goals were in conflict with each other and we studied the trade-off among them. The recovery approach and failure detection mechanism with incomplete information is one of the first steps towards understanding disruption management techniques under uncertainty and opens up the area of designing reliable systems under incomplete on noisy information. We then studied the disruption caused by updating flow rules in software defined networks. We then proposed two randomized rounding algorithms with bounded approximation on congestion and demand loss.

Acknowledgement

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not with standing any copyright notation here on. Special thanks to Prof. Novella Barolini and Dr. Hana Khamfroush for the co-supervision of this work and all coauthors of all papers that have been written during this work.

References

[1] A. Kwasinski, W. W. Weaver, P. L. Chapman, and P. T. Krein. Telecommunications power plant damage assessment for hurricane katrina–site survey and follow-up results. *IEEE Systems Journal*, 2009.

[2] D. Bienstock. *Electrical Transmission System Cascades and Vulnerability: An Operations Research Viewpoint*. SIAM, 2015.

[3] S. Abraham et al. *Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations*. US-Canada Power System Outage Task Force, 2004.

[4] D. Z. Tootaghaj. *Modeling, Monitoring and Scheduling Techniques for Network Recovery from Massive Failures*. PhD thesis, the Pennsylvania State University, University Park, USA, 2018. https://etda.libraries.psu.edu/catalog/15485dxz149/.

[5] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 2011.

[6] The internet topology zoo. `http://www.topology-zoo.org/`, accessed in May, 2015.

[7] D. Z. Tootaghaj, H. Khamfroush, N. Bartolini, S. Ciavarella, S Hayes, and T. La Porta. Network recovery from massive failures under uncertain knowledge of damages. In *IFIP Proceedings of Networking (IFIP NETWORKING)*, 2017.

[8] N. Bartolini, S. Ciavarella, T. F. La Porta, and S. Silvestri. Network recovery after massive failures. In *Dependable Systems and Networks (DSN)*, 2016.

[9] K. Al Sabeh, M. Tornatore, and F. Dikbiyik. Progressive network recovery in optical core networks. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*. IEEE, 2015.

[10] J. Wang, C. Qiao, and H. Yu. On progressive network recovery after a major disruption. In *IEEE INFOCOM*, 2011.

[11] D. Z. Tootaghaj, N. Bartolini, H. Khamfroush, and T. La Porta. On progressive network recovery from massive failures under uncertainty. *IEEE Transactions on Network and Service Management*, 2018.

[12] D. Z. Tootaghaj, N. Bartolini, H. Khamfroush, and T. La Porta. Network recovery from massive failures under uncertain knowledge of damages. In *IEEE Proceedings of the International Symposium on Reliable Distributed Systems (SRDS)*, 2017.

[13] D. Z. Tootaghaj, N. Bartolini, H. Khamfroush, T. He, N. R. Chaudhuri, and T. La Porta. Mitigation and recovery from cascading failures in interdependent networks under uncertainty. *IEEE Transactions on Control of Network Systems*, 2018.

[14] D. Z. Tootaghaj, T. He, and T. La Porta. Parsimonious tomography: Optimizing cost-identifiability trade-off for probing-based network monitoring. In *IFIP Proceedings of Performance (IFIP Performance)*, 2017.

[15] D. Z. Tootaghaj, T. He, and T. La Porta. Parsimonious tomography: Optimizing cost-identifiability trade-off for probing-based network monitoring. *ACM SIGMETRICS Performance Evaluation Review*, 2018.

[16] S. Achleitner, N. Bartolini, T. He, T. La Porta, and D. Z. Tootaghaj. Fast network configuration in software defined networking. *IEEE Transactions on Network and Service Management*, 2018.

[17] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 2014.

[18] S. H. Yeganeh, A. Tootoonchian, and Y. Ganjali. On scalability of software-defined networking. *IEEE Communications Magazine*, 2013.

[19] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker. Abstractions for network update. In *ACM SIGCOMM*. ACM, 2012.

[20] S. Vissicchio, L. Vanbever, L. Cittadini, G. Xie, O. Bonaventure, et al. Safe updates of hybrid sdn networks. *Université catholique de Louvain, Tech. Rep*, 2013.

[21] M. Kuźniar, P. Perešíni, and D. Kostić. What you need to know about sdn flow tables. In *International Conference on Passive and Active Network Measurement*. Springer, 2015.

[22] X. Wen, B. Yang, Y. Chen, L. E. Li, K. Bu, P. Zheng, Y. Yang, and C. Hu. Ruletris: Minimizing rule update latency for tcam-based sdn switches. In *ICDCS*. IEEE, 2016.

[23] A. Ludwig, S. Dudycz, M. Rost, and S. Schmid. Transiently secure network updates. In *Sigmetrics*, 2016.