Joint Progressive Network and Datacenter Recovery After Large-Scale Disasters

Sifat Ferdousi[1], **Massimo Tornatore**[1,2], **Ferhat Dikbiyik**[3], and **Biswanath Mukherjee**[1]
[1]University of California, Davis, USA [2]Politecnico di Milano, Italy [3]Sakarya University, Turkey
Email: {sferdousi,mtornatore,bmukherjee}@ucdavis.edu, fdikbiyik@sakarya.edu.tr

Abstract—**Large-scale disasters affecting both network and datacenter infrastructures, upon which cloud systems are built, can cause severe disruptions in cloud-based services. During the post-disaster recovery phase, repairs are usually carried out in progressive manner due to limited availability of repair resources. The order in which network elements and DCs are repaired can impact users' reachability to important contents and services significantly. In this work, we investigate joint progressive network and DC recovery in which network and DC recovery are carried out in a coordinated manner such that users can have access to maximum possible amount of contents/services at each repair stage. We first solve the optimization problem of joint progressive recovery to find the optimal sequence of network element and DC repairs with the objective to maximize *cumulative weighted content reachability* in the network. We then propose a scalable heuristic for joint progressive recovery which schedules the sequential repair of network nodes/links and DCs such that content reachability is maximized. Our model assumes a simplified repair resource constraint that, at each stage, one network node with its adjacent links and one DC can be fully repaired. However, in realistic scenarios, limited resource availability may not guarantee full recovery of one node and one DC at each repair stage. Hence, we also propose ”resource-aware” joint progressive recovery approach which considers both full and partial recovery of elements at each stage based on available resources. We present two resource allocation strategies, namely selective allocation and adaptive allocation which decide how available repair resources can be efficiently utilized at each stage to provide maximum content reachability to users. We show that, compared to disjoint progressive recovery approach, in which network recovery and DC recovery plans are independent, our joint progressive recovery approach provide significantly higher per-stage gain in content reachability in the network.**

Index Terms—**Cloud networks; Progressive disaster recovery; Content reachability; Resource allocation;**

I. Introduction

Cloud-based services are ubiquitous in today's world. As more and more customers are migrating to cloud, content providers are offering an increasing variety of services ranging from search engines, social networks, video sharing platforms to tools for online collaboration and more. To support these low-latency and high-bandwidth services, they build (or lease from other providers and operators) *datacenters* (DCs) and *networks*, both to interconnect their DCs and to achieve customer proximity [1]. Network and DC infrastructures are generally designed in a geographically distributed manner with failure protection, but even with some baseline resiliency measures in place, they can be vulnerable to disasters. Depending on scale and intensity of disasters, network node/link failures and DC outages can cause temporary to long-term service degradation/disruptions. Many enterprise DCs are colocated in same infrastructures sharing same underlying network connections. Disasters affecting one location can disrupt services for many customers of multiple cloud providers.

The impact can be more acute for large-scale disasters, such as earthquakes, hurricanes, weapons of mass destruction (WMD) attacks (such as nuclear, chemical, EMP, biological), cascading power outages, etc. For instance, 2011 East Japan earthquake and tsunami caused long power outages that affected thousands of telecom offices and damaged tens of DCs; in 2011, a major cloud providers cloud crash due to electrical storm caused damage in many customers data [2]; Hurricane Sandy in 2012 damaged some DCs in New York Area due to floods caused by the hurricane [3]; recent Hurricane Harvey caused outages for more than 148,000 customers [4].

Considering the volume and dependency of customers in cloud networks, it is crucial to guarantee survivability of the cloud services. So far, most studies on disaster survivability in cloud networks focus on pre-disaster preparedness, but preparedness cannot guarantee full recovery at reasonable costs under multiple random/correlated failures due to disasters. Hence, there is a growing interest towards post-disaster recovery strategies. After a disaster, infrastructure repairs are usually carried out in multiple stages, since only a subset of the failed components may be repaired at a time due to limited availability and accessibility of repair resources, e.g., equipment, repair crew, etc. [5], [6]. Therefore, infrastructures can only be *progressively* recovered over time until components are fully repaired (a formal definition and modeling of the progressive network recovery problem was first introduced in [7]). Depending on the scale of failures, duration of recovery stages can be hours, days, or even weeks. In fact, after a disaster, availability of repair resources (e.g., switching units, fibers, transponders, servers, computing racks, storage disks, etc.) and accessibility to sites (factors such as, road conditions, transportation vehicles) can be severely constrained making the recovery process very challenging for repair crews.

Large-scale diasters can affect both DCs and the underlying network infrastructures causing massive service disruptions for end users. How to provide contents/services to users efficiently during post-disaster progressive recovery, considering the limited or ”partial” network and DC capacities, is a major problem to explore. For DCs to be operational after a disaster, not only should the DCs be repaired but also the underlying network connectivity should be restored. On the other hand, operational network elements can provide users access to services after a disaster only when the serving DCs are repaired. Hence, network recovery and DC recovery are interdependent and the order in which network elements and DCs are repaired has an impact on how and when contents and services are available to users. Independent/decoupled plans for network recovery and DC recovery can be inefficient and counter-productive. Rather, a coordinated network and DC recovery plan can lead to more efficient repair resource allocation and restoration of services, especially, for cloud providers that jointly own/manage DCs and networks.

In this study, we investigate *joint* progressive recovery of network and DCs after a large-scale disaster considering a physical network infrastructure that connects both DCs and end users. Generally, network recovery problems aim towards network connectivity in the failed portion of the network. In our work, to ensure delivery of cloud services, we consider content connectivity [8] (reachability of content from any point of a network). Contents are usually replicated in different DCs for redundancy; if a content can be reached by users from at least one hosting DC in the network, it is considered reachable. We employ multi-source anycast routing in which a content is delivered from any of its source DCs to a requesting user.

Our contribution to the joint progressive recovery problem is twofold: first, we formulate the problem of joint progressive network and DC recovery as an optimization model to find the optimal sequence of network nodes/links and DCs to be repaired with the objective to maximize *cumulative weighted content reachability* to users at each stage of the repair process based on importance of the contents. Since optimization approach lacks scalability, we propose a Joint Progressive network and DC Recovery (JPR) algorithm which schedules the sequential repair of network nodes/links and DCs such that users have maximum reachability to important contents at each repair stage. Our model employs a simplified repair resource constraint by assuming that at each stage, one network node with its adjacent links and one DC can be repaired.

We then propose a more realistic resource-aware joint progressive recovery approach which takes into account of both *full recovery* and *partial recovery* of the components at each stage based on the available repair resources. We present two resource allocation strategies, called selective resource allocation and adaptive resource allocation to efficiently utilize the available resources for network and DC repair at each stage. We compare the proposed joint progressive recovery with disjoint progressive recovery in which network recovery and DC recovery are independent of each other. Our results show that, for the typical US-wide networks studied in this work, joint progressive recovery provides about 4 times higher reachability compared to disjoint recovery. We also evaluate our resource-aware joint progressive recovery approach by comparing the two resource allocation strategies and demonstrate their effectiveness.

The rest of the paper is organized as follows: Section II presents brief survey of related works on network survivability and recovery. Section III presents the joint progressive network and DC recovery problem, the optimization model, and our proposed heuristic to efficiently solve the problem. In Section IV, we present resource-aware joint progressive recovery approach with two resource allocation strategies. Section V provides evaluation our approaches through illustrative results. Finally, Section VI concludes the study.

II. Related Work

It is imperative to ensure survivability in cloud networks to prevent data loss and service disruptions. Traditionally, disaster survivability in cloud networks focus on pre-disaster preparedness. Multiple works (e. g., [8]–[11]) propose design of disaster-survivable networks and pre-provision of backup resources. Other works focus on design of disaster-survivable DC networks such as [12]–[17]. Disaster resiliency requirements and techniques through network virtualization an VN mapping were studied in [18]–[21].

However, proactive measures may not guarantee comprehensive survivability in cloud networks against large-scale diasters. In a disaster-affected network, recovery of services through reactive measures can be done in stages. A few works looked into progressive network recovery problem, which is first introduced by [7] for point-to-point requests, and further investigated in [22]–[25]. Ref. [22] and [23] propose multi-stage network recovery methods to meet traffic demand using limited resources with [22] focusing on transport networks. Ref. [24], [25] study efficient recovery methods for optical core networks focusing on the role of different optical network architectures. These recovery studies lack in consideration of recovery time constraint. Ref. [6] model the traveling repairman problem for network virtualization to find optimum travel path in scheduling repairs considering travel time and repair time. However, these works do not consider DC failures which is crucial in cloud networks.

Since then, other works have advanced towards cloud-oriented recovery problems: [5], [26] study different progressive recovery schemes for virtual network services in DC networks and [27] studies the progressive DC recovery problem assuming a given network recovery plan. But, none of these works considers a joint user network and DC network. While network recovery and DC recovery are separate dimensions, it is important to look into the problem of cloud network recovery by jointly considering DCs and the underlying network infrastructures (connecting DCs and serving customers) for fast and efficient restoration of cloud services as we do in this paper.

III. Joint Progressive Network and Datacenter Recovery

In post-disaster recovery, the repair sequence of network elements and DCs is crucial in maximizing users' reachability to important contents. At any given stage of the recovery process, only a subset of network elements and DCs are operational to provide services to users. Traditionally, network repair and DC repair management teams may have different respective repair strategies, e.g., network repair team may repair the node the with highest nodal degree first to provide more connectivity in the network and DC repair team may repair the largest/most important DC first to save more data. External constraints, such as, availability and accessibility of repair resources can also affect the repair strategy (e.g., repairing the closest failed component can be more feasible than repairing others). In our study, we propose a joint progressive recovery approach in which network recovery and DC recovery are planned in a coordinated manner to achieve a common objective of maximizing content reachability to users at each stage.

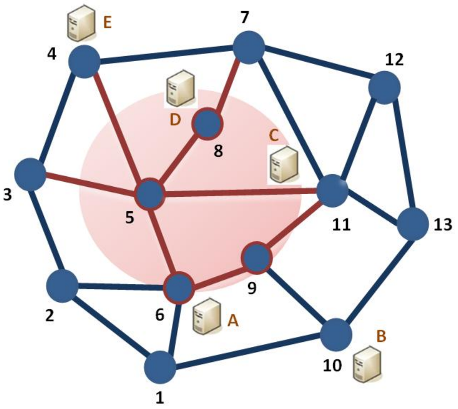

Fig. 1: Sample network for joint progressive recovery.

TABLE I: Node and DC repair selections based on *gain* in content reachability in the sample network with *latency constraint* = 2 *hop counts*.

Node	DC	Additional content reachability within 2 hop counts
9	D	DC B to nodes 9
5	D	DC E to nodes 5, 11
6	D	DC A to nodes 1, 2, 3, 6, 10
8	D	DC D to nodes 4, 8, 7, 11, 12 and DC E to node 8
8	C	DC E to nodes 8 and DC C to node 4, 7, 8, 10, 11, 12, 13
9	C	DC B to nodes 9 and DC C to node 4, 7, 9, 10, 11, 12, 13
5	C	DC E to nodes 5, 11 and DC C to nodes 3, 4, 5, 7, 10, 11, 12, 13
6	C	DC A to nodes 1, 2, 3, 6, 10 and DC C to nodes 4, 7, 10, 11, 12, 13

After a disaster, if a DC is operational but the underlying network node/links are failed, it is important to repair the node/links providing the connectivity so that the DC can resume its services immediately. On the other hand, if a DC is failed but the underlying network node/links are operational, the DC should be repaired because it already has the network connectivity to resume its services. Core network nodes supporting (providing connectivity to) DCs are crucial because such nodes provide connectivity not only to other nodes in the network but also to a DC. However, a core node which does not support a DC but has high nodal degree in the network can also be critical in terms of providing reachability to users.

We demonstrate the problem using a simple example of a 13-node network with 5 DCs with uniform content distribution and user requests. As shown in Fig. 1, DCs C and D, and network nodes 5, 6, 8, and 9 with the associated links are failed due to a disaster. Since contents in DCs C and D are not available, contents in DC A are not reachable, and available contents are not accessible to users at nodes 5, 6, 8, and 9, cumulative content reachability in the network is reduced. We assume that only one network node with its viable adjacent links (links with active end nodes) and one DC can be repaired at each stage of the recovery process. We also assume that a content should be reachable within two hop counts to satisfy users' latency constraint.

Suppose DC D hosts more important contents and hence its repair can be of high priority for DC repair crew in the first repair stage. But repairing DC D will not provide any additional content reachability in the network since its supporting node 8 is failed. On the other hand, network repair crew may chose to repair node 5 with links 5 3 and 5 4 in the first repair stage since it has the highest nodal degree in the original network. However, repair of node 5 will allow additional reachability from only DC E to users at nodes 5 and 11 because: DC C is still failed at this stage, DCs A and D (if repaired) are still unreachable, and DC B does not satisfy the two hop count constraint. Hence, the overall content reachability in the network is not greatly improved by the disjoint recovery.

To evaluate the impact of joint recovery, in Table I, we consider all possible node/links and DC repair selections in the first stage in terms of additional/recovered content reachability (or *gain* in overall content reachability) in the network. We can see that repair of node 6 (with no link repairs since adjacent nodes 5 and 9 are failed) and DC C leads to highest gain in content reachability by allowing additional content access from 2 DCs to 11 user nodes in total. However, content distribution (how contents replicas are distributed in DCs), content importance, user demands, repair resource availability, etc. also play important roles in the selection decision. Hence, it is not trivial to decide which node and/or which DC should be repaired at a stage so that maximum number of users can access maximum amount of contents. Considering all dimensions, it is intuitive that network recovery and DC recovery are interdependent and complementary in providing cloud services to users. A disjoint recovery scheme cannot reflect this interdependency because network repair may not be aware of the state of the DCs and the impact on users' reachability to important contents/services.

Our joint progressive recovery scheme determines the sequence of network elements and DCs to be repaired by maximizing the *cumulative weighted content reachability*, R at each repair stage. This metric is a measure of reachability of all contents from a set of active DCs hosting the contents to a set of active user nodes requesting the contents *within their latency constraint*. It can be formulated as $R = \sum_{s\in V} \sum_{c\in C} \} R_{(c,s)} \leq \alpha_c|$, where $R_{(c,s)} = 1$ if content c is reachable to node s within latency constraint from any DC hosting c through multi-source anycast routing. It is then multiplied with importance of content c, α_c to weigh in the priority of obtaining content c. The higher the importance of a content, the higher the priority to reach that content. By using this factor, we favor the reachability of important contents first.

A disaster may cause failure of multiple network elements and DCs and it may not be possible to repair all components at the same time due to limited availability of repair crew and repair resources. We assume, as a simplistic approach, that one network repair crew and one DC repair crew are available for post-disaster recovery and that, at each repair stage, one network node with its viable adjacent links and one DC can be *fully* repaired with the available resources. Hence, the number of repair stages is governed by the maximum of the number of failed nodes and failed DCs. For link repair, we consider that links are viable for repair only if the end-nodes are active, because repairing links among failed end-nodes is not resource efficient as it will not increase network connectivity (hence reachability). We also assume that once a DC is *fully* repaired, all contents hosted in the DC are available. We also assume that contents can be served directly from a DC to a requesting node without using any external network resources (links) if the serving DC is supported by the requesting node. We consider uniform content distribution and that all user nodes request access to all contents (a generalization of content demands). The recovery process terminates when all network elements and DCs in the network are repaired.

A. Problem Formulation

The joint progressive recovery problem can be summarized as follows: given a physical network topology, a set of DCs, a set of contents (with corresponding distribution, demands, and importance factors), and a set of failed network nodes, links, and DCs, the joint progressive network and DC recovery scheme selects one network node with its viable adjacent links and one DC to be repaired at each subsequent repair stage such that content reachability to users is maximized based on importance of the contents. We formulate the optimization problem as an integer linear program (ILP) with the objective to maximize cumulative weighted content reachability, R as follows:

- ***Given***:
  - $G(V,E)$: Physical network topology with set of all nodes, V and set of all links, E.
  - D: Set of all DCs.
  - $\bar{G}(\bar{V},\bar{E})$: Post-disaster physical network topology with set of failed nodes, $\bar{V}$ / V and set of failed links, $\bar{E}$ / E.
  - $\bar{D}$: Set of failed DCs, $\bar{D}$ / D.
  - K: Number of repair stages ($K = max\}\backslash\bar{V}\backslash, \backslash\bar{D}\backslash|$).
  - C: Set of contents.
  - α_c: Importance factor of content c / C.
  - X_d^c: Binary indicator that is 1 if DC d / D hosts content c / C and 0 otherwise.
  - $l_{(i,j)}$: Latency on link (i,j) / E.
  - l_{max}: Latency constraint for serving a content from a DC to an user node.
- ***Binary Variables***:
  - $R^k_{(c,s)}$: 1 if content c / C is reachable to node s / V at stage k and 0 otherwise.
  - $f^{(s,d),k}_{(i,j),c}$: 1 if link (i,j) / E is used to serve content c / C to node s / V from DC d / D at stage k and 0 otherwise.
  - $Q^{s,k}_{d,c}$: 1 if c / C is served to node s / V from DC d / D at stage k and 0 otherwise.
  - A^k_d: 1 if DC d / D is available at stage k and 0 otherwise.
  - L^k_i: 1 if node i / V is available at stage k and 0 otherwise.
  - $L^k_{(i,j)}$: 1 if link (i,j) / E is available at stage k and 0 otherwise.

- ***Objective***: The objective is to maximize cumulative weighted content reachability.

$$max \sum_{k=1}^{K} \sum_{c \in C} \sum_{s \in V} \alpha_c R^k_{(c,s)} \tag{1}$$

subject to

- ***Constraints***:
  - **Initialization constraints**:

    ≤ Initially at stage $k = 0$, node i is unavailable if affected by disaster, and available otherwise.

$$L^0_i = 1, \ \cup i \ / \ V \quad \bar{V} \tag{2a}$$

$$L^0_i = 0, \ \cup i \ / \ \bar{V} \tag{2b}$$

    ≤ Initially at stage $k = 0$, DC d is unavailable if affected by disaster, and available otherwise.

$$A^0_d = 1, \ \cup d \ / \ D \quad \bar{D} \tag{3a}$$

$$A^0_d = 0, \ \cup d \ / \ \bar{D} \tag{3b}$$

  - **Node availability constraint**: Node s can request content c from DC d at stage k if node s is available at stage k.

$$Q^{s,k}_{d,c} \geq L^k_s, \tag{4}$$

    $\cup c \ / \ C, \cup s \ / \ V, \cup d \ / \ D, k = 1, 2, \ldots, K$
  - **DC availability constraints**:

    ≤ DC d is available at stage k if the supporting node is also available at stage k.

$$A^k_d \geq L^k_d, \ \cup d \ / \ D, k = 1, 2, \ldots, K \tag{5}$$

    ≤ DC d can serve content c to node s at stage k if DC d is available at stage k and if DC d hosts content c.

$$Q^{s,k}_{d,c} \geq A^k_d * X^c_d, \tag{6}$$

    $\cup c \ / \ C, \cup s \ / \ V, \cup d \ / \ D, k = 1, 2, \ldots, K$
  - **Link availability constraints**:

    ≤ Link (i, j) is available at stage k if the end nodes i and j are available at stage k.

$$L^k_{(i,j)} = L^k_i \ \{ \ L^k_j, {}^1 \tag{7}$$

    $\cup(i, j) \ / \ E, \cup i \ / \ V, \cup j \ / \ V, k = 1, 2, \ldots, K$

    ≤ There can be a flow carrying content c to node s from DC d using link (i, j) at stage k if link (i, j) is available at stage k.

$$f^{(s,d),k}_{(i,j),c} \geq L^k_{(i,j)}, \tag{8}$$

    $\cup c \ / \ C, \cup s \ / \ V, \cup d \ / \ D, \cup(i, j) \ / \ E, k = 1, 2, \ldots, K$
  - **Flow-conservation constraints**:

    ≤ There is a flow carrying content c to node s from DC d using link (i, j) at stage k if content c is served to node s from DC d.

$$\sum_{j \in V} f^{(s,d),k}_{(i,j),c} \quad \sum_{j \in V} f^{(s,d),k}_{(j,i),c} = Q^{s,k}_{d,c} \ \text{ if } i = d \tag{9a}$$

$$\sum_{j \in V} f^{(s,d),k}_{(i,j),c} \quad \sum_{j \in V} f^{(s,d),k}_{(j,i),c} = \quad Q^{s,k}_{d,c} \ \text{ if } i = s \tag{9b}$$

$$\sum_{j \in V} f^{(s,d),k}_{(i,j),c} \quad \sum_{j \in V} f^{(s,d),k}_{(j,i),c} = 0 \ \text{ if } i \not\forall s, d \tag{9c}$$

    $\cup c \ / \ C, \cup i \ / \ V, \cup s \ / \ V, \cup d \ / \ D, \cup(i, j) \ / \ E, k = 1, 2, \ldots, K$

    ≤ Content c can be served to node s directly from DC d at stage k if the serving DC d is supported by the requesting node s.

$$f^{(s,d),k}_{(s,d),c} = Q^{s,k}_{d,c} \ \text{ if } s = d \tag{10}$$

    $\cup c \ / \ C, \cup s \ / \ V, \cup d \ / \ D, k = 1, 2, \ldots, K$
  - **Latency constraint**: DC d can serve content c to node s at stage k if DC d is reachable from node s within latency constraint.

$$\sum_{(i,j) \in E} f^{(s,d),k}_{(i,j),c} * l_{(i,j)} \geq l_{max}, \tag{11}$$

    $\cup c \ / \ C, \cup s \ / \ V, \cup d \ / \ D, k = 1, 2, \ldots, K$
  - **Recovery constraints**:

    ≤ At each stage k, at most one node i can be recovered.

$$\sum_{i \in V} L^{k+1}_i \quad \sum_{i \in V} L^k_i \geq 1, \ k = 0, 1, \ldots, K \tag{12}$$

    ≤ At each stage k, at most one DC d can be recovered.

$$\sum_{d \in D} A^{k+1}_d \quad \sum_{d \in D} A^k_d \geq 1, \ k = 0, 1, \ldots, K \tag{13}$$

  - **Continuity constraints**:

    ≤ After node i is recovered at stage k, it is available in the consecutive stages.

$$L^{k+1}_i \leftarrow L^k_i, \ \cup i \ / \ V, k = 0, 1, \ldots, K \tag{14}$$

    ≤ After DC d is recovered at stage k, it is available in the consecutive stages.

$$A^{k+1}_d \leftarrow A^k_d, \ \cup d \ / \ D, k = 0, 1, \ldots, K \tag{15}$$

  - **Reachability constraints**:

    ≤ Content c is served to node s from one of the available DCs d at stage k if content c is available in DC d.

$$\sum_{d \in D} Q^{s,k}_{d,c} \geq 1 \tag{16a}$$

$$\sum_{d \in D} Q^{s,k}_{d,c} \leftarrow \frac{\sum_{d \in D} A^k_d * X^c_d}{M} \tag{16b}$$

    $\cup c \ / \ C, s \ / \ V, k = 1, 2, \ldots, K$

    ≤ If content c is served to node s from any available DC d at stage k, then $R^k_{(c,s)} = 1$, otherwise it is 0.

$$R^k_{(c,s)} \geq \sum_{d \in D} Q^{s,k}_{d,c} \tag{17a}$$

$$R^k_{(c,s)} \leftarrow \frac{\sum_{d \in D} Q^{s,k}_{d,c}}{M} \tag{17b}$$

    $\cup c \ / \ C, s \ / \ V, k = 1, 2, \ldots, K$

In our joint progressive recovery ILP model, both the number of variables and the number of constraints are upper bounded by $O(\backslash C\backslash\backslash V\backslash\backslash D\backslash\backslash E\backslash\backslash K\backslash)$. With network size and number of contents (in the range of hundreds and thousands), the problem size can grow significantly leading to limited scalability.

[1] "AND" operation in Equation (7) can be linearly calculated as follows:

$$L^k_{(i,j)} \leq L^k_i; \quad L^k_{(i,j)} \leq L^k_j; \quad L^k_{(i,j)} \geq L^k_i + L^k_j - 1$$

B. Heuristic for Joint Progressive Recovery

Following the same design considerations and assumptions used in our optimization model, we propose a scalable Joint Progressive network and DC Recovery (JPR) heuristic which selects a network node, a set of links, and a DC to be repaired at each repair stage k such that cumulative weighted content reachability at stage k, R^k is maximized.

Algorithm: The JPR algorithm is described in Algorithm 1 and its two sub-algorithms, namely *JPR Node Repair Selection* and *JPR DC Repair Selection*, are described in Algorithm 2 and Algorithm 3, respectively. The following list defines the new notations used in the algorithms.

- V^k_{rep}: Set of active and repaired nodes at stage k.
- E^k_{rep}: Set of active and repaired links at stage k.
- D^k_{rep}: Set of active and repaired DCs at stage k.
- N^k_s: Set of active neighboring nodes of node s, $N^k_s \; / \; V^k_{rep}$.
- $\mathit{\Gamma}^k_{s,N^k_s}$: Set of links between node s and N^k_s at stage k.
- V^k_{test}: Set of test nodes for reachability computation at stage k.
- E^k_{test}: Set of test links for reachability computation at stage k.
- D^k_{test}: Set of test DCs for reachability computation at stage k.
- D_c: Set of DCs hosting content c.
- P_{sc}: Set of k-shortest paths for content c delivered to node s.
- l_p: Latency of path $p \; / \; P$.
- R^k_s: Cumulative weighted content reachability obtained *if* node s is active at stage k.
- R^k_d: Cumulative weighted content reachability obtained *if* DC d is active at stage k.
- S^k_{rank}: List of nodes s with R^k_s values at stage k.
- D^k_{rank}: List of DCs d with R^k_d values at stage k.
- R^k: Cumulative weighted content reachability at stage k.

Algorithm 1 Joint Progressive network and DC Recovery (JPR)

```
Input: $G(V, E)$, $\bar{G}(\bar{V}, \bar{E})$, $D$, $\bar{D}$, $C$, $D_c$, $\alpha_c$, $l_{max}$.
Output: $V^k_{rep}$, $E^k_{rep}$, $D^k_{rep}$, $R^k$.
Initialization: $k = 1$. $\bar{V}^k \in \; \bar{V}$; $\bar{E}^k \in \; \bar{E}$; $\bar{D}^k \in \; \bar{D}$. $V^k_{rep} \in \; V \;\; \bar{V}$;
$E^k_{rep} \in \; E \;\; \bar{E}$; $D^k_{rep} \in \; D \;\; \bar{D}$.
  • While $\}\bar{V}^k$ and $\bar{E}^k$ and $\bar{D}^k| \; \not\forall \; empty$
      – Network Recovery Phase:
          ≤ Node Repair
              ∗ Invoke Algorithm 2 to select node $s_t \; / \; \bar{V}^k$ for repair.
          ≤ Link Repair
              ∗ Select set of links $\mathit{\Gamma}^k_{s_t,N^k_{s_t}} \; / \; \bar{E}^k$ for repair.
      – DC Recovery Phase:
          ≤ DC Repair
              ∗ Invoke Algorithm 3 to select DC $d_t \; / \; \bar{D}^k$ for repair.
  • Repair node $s_t$, set of links $\mathit{\Gamma}^k_{s_t,N^k_{s_t}}$, DC $d_t$.
  • $V^k_{rep} \in \; V^k_{rep} \wedge \}s_t|$ ; $E^k_{rep} \in \; E^k_{rep} \wedge \mathit{\Gamma}^k_{s_t,N^k_{s_t}}$ ; $D^k_{rep} \in \; D^k_{rep} \wedge$
    $\}d_t|$ .
  • $R^k = \sum_{s \in V^k_{rep}} \sum_{c \in C} R^k_{(c,s)} \leq \alpha_c$.
  • $\bar{V}^{k+1} \in \; \bar{V}^k \;\; \}s_t|$ ; $\bar{E}^{k+1} \in \; \bar{E}^k \;\; \mathit{\Gamma}^k_{s_t,N^k_{s_t}}$ ; $\bar{D}^{k+1} \in \; \bar{D}^k \;\; \}d_t|$ .
  • $V^{k+1}_{rep} \in \; V^k_{rep}$; $E^{k+1}_{rep} \in \; E^k_{rep}$; $D^{k+1}_{rep} \in \; D^k_{rep}$.
  • $k = k + 1$.
```

Our proposed JPR approach consists of a *network recovery phase* and a *DC recovery phase*. We consider the following cases in network recovery phase: **a)** failed core nodes supporting no DCs, **b)** failed core nodes supporting active DCs, **c)** failed core nodes supporting failed DCs, and the following cases in DC recovery phase: **d)** failed DCs with active supporting core nodes, **e)** failed DCs with failed supporting core nodes. Considering each of these cases, content reachability metric R is used to determine if repair of a failed component is feasible at stage k by assuming that the failed component is active and connected to the operational part of the network at stage k. The idea is to evaluate whether the failed component, *if repaired*, contribute to the overall maximum content reachability that can be achieved in the network at stage k.

Algorithm 2 JPR Node Repair Selection

```
1) For each test node $s_t \; / \; \bar{V}^k$,
    a) $V^k_{test} \in \; V^k_{rep} \wedge \}s_t|$ ; $E^k_{test} \in \; E^k_{rep} \wedge \mathit{\Gamma}^k_{s_t,N^k_{s_t}}$ .
    b) If node $s_t$ does not support any DC or supports an active
       DC $d \; / \; D^k_{rep}$, $D^k_{test} \in \; D^k_{rep}$.
    c) Else,
        i) Get the failed DC $d_t \; / \; \bar{D}$ supported by $s_t$.
       ii) $D^k_{test} \in \; D^k_{rep} \wedge \}d_t|$ .
    d) Compute $R^k_{s_t} = \sum_{s \in V^k_{test}} \sum_{c \in C} R^k_{(c,s)} \leq \alpha_c$ as:
        • $R^k_{s_t} = 0$.
        • For each node $s \; / \; V^k_{test}$,
            – For each content $c \; / \; C$, get set of DCs $D_c \; / \; D^k_{test}$
              hosting content $c$.
                i) If any DC $d \; / \; D_c$ is supported by node $s$,
                   $R^k_{(c,s)} = 1$, $R^k_{s_t} = R^k_{s_t} + R^k_{(c,s)} \leq \alpha_c$.
               ii) Else,
                    A) Compute set of $k$-shortest paths, $P_{sc}$ between
                       $s$ and any DC $d \; / \; D_c$ using multi-source
                       anycast routing with $E^k_{test}$.
                    B) If for any path $p \; / \; P$, latency $l_p < l_{max}$,
                       $R^k_{(c,s)} = 1$, $R^k_{s_t} = R^k_{s_t} + R^k_{(c,s)} \leq \alpha_c$.
    e) $S^k_{rank} \in \; \}R^k_{s_t}|$ .
2) Sort $S^k_{rank}$ in descending order of $R^k_{s_t}$ value.
3) Select node $s_t$ with max $R^k_{s_t}$ for repair.
    a) If multiple nodes give max $R^k_{s_t}$, select node $s_t$ which
       supports an active DC.
        i) If all nodes support active DCs, select node $s_t$ with
           max nodal degree for repair.
       ii) If no node supports an active DC, select node $s_t$ which
           supports a failed DC $d \; / \; \bar{D}^k$ for repair.
            A) If all or no nodes support failed DCs, select node
               $s_t$ with max nodal degree for repair.
```

As shown in Algorithm 1, at each repair stage k, a failed node s_t with its viable adjacent links $\mathit{\Gamma}^k_{s_t,N^k_{s_t}}$ is selected for repair in *network recovery phase* and a failed DC d_t is selected for repair in *DC recovery phase*. After the failed components are repaired, they are added to V^k_{rep}, E^k_{rep}, and D^k_{rep} and removed from $\bar{V}^k$, $\bar{E}^k$, and $\bar{D}^k$. Final cumulative weighted content reachability at stage k, R^k is computed as $R^k = \sum_{s \in V^k_{rep}} \sum_{c \in C} R^k_{(c,s)} \leq \alpha_c$ using anycast routing with V^k_{rep}, E^k_{rep}, and D^k_{rep}. Since each repair stage allows repair of one node with its adjacent links and one DC, the algorithm moves to the next stage and the process continues until all failed network nodes, links, and DCs are repaired in the network. The two recovery phases are discussed in details below.

1) Network Recovery Phase: The network recovery phase of JPR algorithm includes **node repair** and **link repair** steps. In **node repair**, a failed node s_t is selected for repair through *Node Repair Selection* algorithm. As shown in Algorithm 2 *step (1)*, at stage k, for each candidate/test node $s_t \; / \; \bar{V}^k$, a set of test nodes V^k_{test} is formed for reachability computation by assuming s_t to be active with the set

Algorithm 3 JPR DC Repair Selection

```
1) For each test DC $d_t \ / \ \bar{D}^k$,
    a) $D^k_{test} \in \ D^k_{rep} \wedge \}d_t| \,.$
    b) If DC $d_t$ has an active supporting node $s \ / \ V^k_{rep}$,
        i) $V^k_{test} \in \ V^k_{rep}$; $E^k_{test} \in \ E^k_{rep}$.
    c) Else,
        i) Get the failed supporting node $s_t \ / \ \bar{V}^k$.
        ii) $V^k_{test} \in \ V^k_{rep} \wedge \}s_t| \,$; $E^k_{test} \in \ E^k_{rep} \wedge \mathit{\Gamma}^k_{s_t, N^k_{s_t}}$ .
    d) Compute $R^k_{d_t} = \sum_{s \in V^k_{test}} \sum_{c \in C} R^k_{(c,s)} \leq \alpha_c$ as:
        • $R^k_{d_t} = 0$.
        • For each node $s \ / \ V^k_{test}$,
            – For each content $c \ / \ C$, obtain set of DCs $D_c \ / \ D^k_{test}$ hosting content $c$.
                i) If any DC $d \ / \ D_c$ is supported by node $s$, $R^k_{(c,s)} = 1$, $R^k_{d_t} = R^k_{d_t} + R^k_{(c,s)} \leq \alpha_c$.
                ii) Else,
                    A) Compute set of $k$-shortest paths, $P_{sc}$ between $s$ and any DC $d \ / \ D_c$ using multi-source anycast routing with $E^k_{test}$.
                    B) If for any path $p \ / \ P$, latency $l_p < l_{max}$, $R^k_{(c,s)} = 1$, $R^k_{d_t} = R^k_{d_t} + R^k_{(c,s)} \leq \alpha_c$.
    e) $D^k_{rank} \in \ \}R^k_{d_t}| \,.$
2) Sort $D^k_{rank}$ in descending order of $R^k_{d_t}$ value.
3) Select DC $d_t$ with max $R^k_{d_t}$ value for repair.
    a) If multiple DCs give max $R^k_{d_t}$, select DC $d_t$ which has an active supporting node. hosts important contents.
        i) If all or no DCs have active supporting nodes, select DC $d_t$ which hosts most important contents.
```

of existing active nodes V^k_{rep}. As a result, set of links between node s_t and its set of active neighboring nodes $N^k_{s_t}$, $\mathit{\Gamma}^k_{s_t, N^k_{s_t}}$ are also assumed active with the set of existing active links E^k_{rep} to form E^k_{test}. If node s_t does not support any DC or supports an active/operational DC (*cases a, b*), then the set of test DCs for reachability computation, D^k_{test} is the set of existing active DCs D^k_{rep}. If node s_t supports a failed DC (*case c*), then the failed DC d_t is also assumed to be active with D^k_{rep} to form D^k_{test}. Since such node is a DC supporting node, to evaluate its actual repair potential, the failed DC is also assumed operational in the reachability computation.

Content reachability at stage k with assumed active node s_t, $R^k_{s_t}$ is computed in *step* $(1.d)$ with V^k_{test}, E^k_{test}, and D^k_{test} using anycast routing. For each node $s \ / \ V^k_{test}$ and each content $c \ / \ C$, we obtain $D_c \ / \ D^k_{test}$, set of DCs hosting content c. Request for content c from node s can be served from any of the multiple source DCs in D_c. If a DC $d \ / \ D_c$ is supported by the requesting node s, c can be served to the self supporting node s without any route computation. Otherwise, we compute set of k-shortest paths, P_{sc} from any DC $d \ / \ D_c$ to node s using multi-source anycast routing with E^k_{test}. If any path $p \ / \ P_{sc}$ satisfies the users' latency constraint ($l_p < l_{max}$), content c is reachable to node s and hence $R^k_{(c,s)} = 1$. It is then multiplied with importance of content c, α_c to weigh in the priority of obtaining the content (thus, we ensure that more important contents are reachable to the users first). $R^k_{s_t}$ gives the cumulative weighted reachability of all contents to all active nodes in the network including the assumed active node s.

The algorithm then ranks the nodes $s_t \ / \ \bar{V}^k$ based on the respective reachability values, $R^k_{s_t}$ in *step (2)*, and selects the node which leads to maximum reachability for repair in *step (3)*. During node selection in *step (3)*, there can be multiple nodes that satisfy the selection criteria. In such cases, priorities are given to nodes which: i) support an active DC since it can immediately provide connectivity to an active DC - *step (3.a)*, ii) have the highest nodal degree in the original physical network as it will lead to increased network connectivity and hence facilitate content reachability in later stages - *step (3.a.i)* and *(3.a.ii.A)*, iii) support a failed DC because repair of such node can provide connectivity to a failed DC when it is repaired - *step (3.a.ii)*.

After selection of node s_t, in **link repair**, set of failed links between node s_t and $N^k_{s_t}$, $\mathit{\Gamma}^k_{s_t, N^k_{s_t}} \ / \ \bar{E}^k$ are selected for repair.

2) DC Recovery Phase: The DC recovery phase of JPR algorithm consists of **DC repair** step in which a failed DC d_t is selected for repair through *DC Repair Selection* algorithm. As shown in Algorithm 3 *step (1)*, at stage k, for each candidate/test DC $d_t \bar{D}^k$, a set of test DCs D^k_{test} is formed for reachability computation by assuming DC d_t to be active with D^k_{rep}. If DC d_t has an active/operational supporting core node (*case d*), then V^k_{test} for reachability computation is V^k_{rep} and E^k_{test} is E^k_{rep}. If DC d_t has a failed supporting core node (*case e*), then the failed node s_t is also assumed to be active (for DC d_t to be assumed to be active it also has to be connected to the network) with V^k_{rep} to form V^k_{test}. As a result, set of links $\mathit{\Gamma}^k_{s_t, N^k_{s_t}}$ are also assumed active with E^k_{rep} to form E^k_{test}. Content reachability at stage k with assumed active DC d_t, $R^k_{d_t}$ is computed in *step (1.d)* as described in **node repair**. The DC which leads to maximum reachability at stage k is selected for repair. During DC selection in *step (3)*, priorities are given to DCs which: i) has an active supporting core node so that the DC has immediate connectivity - *step (3.a)*, ii) hosts more important contents - *step (3.a.i)*.

Complexity of heuristic: In JPR node and DC repair selection algorithms, we computed k-shortest paths for each content $c \ / \ C$ and each node $s \ / \ V$ for reachability computation, which is a dictating factor in the complexity of our algorithm. Overall, the time complexity of JPR is $O(\backslash K \backslash[(\backslash V \backslash + \backslash D \backslash)(\backslash V \backslash\backslash C \backslash\} k \backslash V \backslash(\backslash E \backslash + \backslash V \backslash log \backslash V \backslash)| \,)])$.

C. Disjoint Progressive Recovery

To evaluate our joint progressive network and DC recovery approach, we consider disjoint progressive network and DC recovery in which network repair phase and DC repair phase are independent. The disjoint recovery does not consider content reachability to users as the common objective since the network and the DC repair teams follow their individual repair strategies. We developed two disjoint approaches, called greedy disjoint progressive recovery and random disjoint progressive recovery. In greedy approach, at each stage, network repair phase focuses on improving network connectivity rather than content reachability since it is not aware of the content distribution in DCs and hence selects nodes/links for repair based on nodal degree in the physical network. On the other hand, DC repair phase focuses on content availability (repairing a DC makes the contents available) rather than content reachability since it is not aware of the underlying core network connectivity. For each DC $d \ / \ \bar{D}^k$, we evaluate the total importance of contents hosted in DC d by computing content weight metric, W_d as $W_d = \sum_{c \in C} d_c \leq \alpha_c$ where d_c indicates if content c is hosted in DC d. DC with the highest W_d value is selected for repair and thus, at each stage, more important contents are made available but they are not necessarily reachable to users since DC recovery is disjoint from network recovery. In random approach, nodes/links and DCs are selected for repair individually at random in network repair phase and DC repair phase, respectively.

IV. Resource-Aware Joint Progressive Recovery

In our joint progressive recovery approach we had the simplistic assumption that, at any stage, enough resources are available to *fully* repair one node with its viable adjacent links and one DC. However, in a more realistic scenario, full recovery of one node, a set of links, and one DC may not be feasible in one stage as such repairs require considerable amount of resources, manpower, and time. External issues such as road and transportation access, travel time of repair crew etc. also become more critical after a disaster. In this section, we consider a more realistic approach for our JPR algorithm which takes into account limited availability of repair resources that may not guarantee full recovery of network elements and DCs. We develop ”resource-aware joint progressive network and DC recovery approaches” in which, at any stage, network elements and DCs are repaired partially or fully based on available resources with the objective to maximize content reachability.

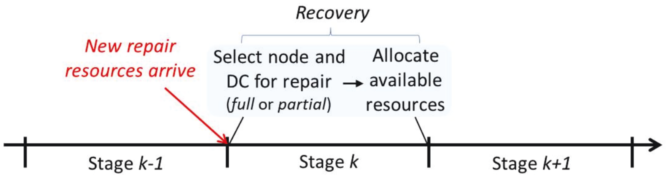
Recovery
New repair resources arrive
Select node and DC for repair (*full* or *partial*)
Allocate available resources
Stage *k-1*
Stage *k*
Stage *k+1*

Fig. 2: Progressive recovery with resource allocation.

In a post-disaster scenario, repair resources usually become available in batches and hence the repair process utilizes maximum available resources at a given stage and waits for next batch of resources to be available as shown in Fig. 2. In between stages, elements can be fully recovered or partially recovered meaning that some/partial services can be served to users even if the elements are not fully operational. In our study, partial network recovery means that network nodes and links can be operational with partial capacity (e.g., degraded services with delay and packet loss), and partial DC recovery means that DCs can serve partial services (e.g., limited number of content requests) to users. Note that, for links, there is no notion of partial repair; once enough link repair resources (e.g., fibers, amplifiers, etc) are available, a link is fully repaired in terms of providing connectivity between working nodes. The capacity in which links can serve (bandwidth) depends on the capacity of the end nodes and hence node repair resources (switching units, transponders, regenerators) control the potential serving capacity in the network. Similarly, DC repair resources (servers, computing racks, storage disks, etc.) control how many content requests can be served.

We consider that, with partial node and DC recovery, content reachability to users is also ”partial” or reduced (meaning that, not all user requests for contents can be served) in proportion to the partial capacities of the nodes and DCs. We compute cumulative content reachability at stage k considering the partial capacities as $R^k = \sum_{s\in V_{rep}^k} \sum_{c\in C} \} R_{(c,s)}^k \leq X_{s\in V_{rep}^k}^k \leq Y_{d\in D_{rep}^k}^k \leq \alpha_c |$, where X_s^k and Y_d^k denote partial capacity of node s and DC d respectively. In a disaster regions, providing even partial services can be crucial for users, especially at the early stages. As resources become available in subsequent stages, repairs of network elements and DCs continue until they are operational with full capacity. Hence, it is possible that a node or DC requires multiple stages to be fully recovered.

A. Heuristics for Resource Allocation

We propose two repair resource allocation strategies, namely, *selective resource allocation (SR)* and *adaptive resource allocation (AR)* which are employed in our JPR algorithm. We assume that, at each stage k, A_V^k units of node repair resources, A_E^k units of link repair resources, and A_D^k units of DC repair resources are available, and that each node, link, and DC require F_s, F_e, and F_d units of repair resources, respectively, to be fully repaired. Since network elements and DCs are selected sequentially for repair in JPR algorithm, repair resources are also allocated sequentially to the components at each stage and we consider one network repair crew and one DC repair crew (working in a coordinated manner) for the repair process. Here, we also consider repair crew travel time and equipment repair time [6] during the selection of nodes and DCs for repair. Depending on equipment type, repair time can be considered to be insignificant compared to travel time and vice versa. So, we consider a total recovery time T_s for repairing node s (and any adjacent links) and T_d for repairing DC d. We assume that each repair crew has respective work hour constraint as the maximum allowed duration of a repair work. Hence, we enforce that T_s and T_d comply with the repair duration constraint T_N for network repair crew and T_D for DC repair crew, respectively.

Algorithm: The JPR-SR algorithm is described in Algorithm 4. The JPR-SR algorithm with its two sub-algorithms, namely *JPR-AR Node Repair Selection* and *JPR-AR DC Repair Selection*, are described in Algorithm 5, Algorithm 6, and Algorithm 7, respectively. The following list defines the new notations used in the algorithms.

- V_{pend}: A binary indicator if any node repair is pending.
- D_{pend}: A binary indicator if any DC repair is pending.
- A_V^k: Available node repair resources at stage k.
- A_E^k: Available link repair resources at stage k.
- A_D^k: Available DC repair resources at stage k.
- $r_{s\in\bar{V}}^k$: Pending resource requirement for node s at stage k.
- $r_{d\in\bar{D}}^k$: Pending resource requirement for DC d at stage k.
- $F_{s\in\bar{V}}$: Repair resource requirement for full recovery of node s.
- $F_{e\in\bar{E}}$: Repair resource requirement for full recovery of link e.
- $F_{d\in\bar{D}}$: Repair resource requirement for full recovery of DC d.
- $X_{s\in\bar{V}}^k$: Serving capacity of node s at stage k.
- $Y_{d\in\bar{D}}^k$: Serving capacity of DC d at stage k.
- T_s / $\bar{V}$: Recovery time of node s and viable adjacent links.
- T_d / $\bar{D}$: Recovery time of DC d.
- T_N: Maximum repair duration for network repair crew.
- T_D: Maximum repair duration for DC repair crew.

1) JPR-SR Algorithm: At each stage k, JPR-SR heuristic selects a failed network node and a failed DC which can lead to maximum content reachability *if fully recovered* at stage k as in Algorithm 1. The *SR* strategy then allocates available resources to the selected element and if resources are insufficient for full recovery, the strategy marks the repair as ”pending”. Resource allocation is continued in subsequent stages until the element is fully repaired. Essentially, the idea is to select the most critical node and DC (in terms of content reachability) and allocate maximum available resources across the stages until full recovery and then select a new critical node and DC.

As shown in Algorithm 4, the network recovery phase consists of **node repair resource allocation** and **link repair resource allocation** steps. Under **node repair resource allocation**, in *step (a)*, if V_{pend} is 0, i.e., no prior node repair is pending, a failed node s_t is selected for repair using Algorithm 2 given that T_{s_t} satisfies T_N. After selection, JPR-SR allocates the maximum available resources A_V^k to node s_t and if the resources are sufficient, i.e., $(A_V^k \leftarrow F_{s_t})$, then node s_t is fully repaired in *step (a.i)*. Any remaining resources $(A_V^k \quad F_{s_t})$ are reserved for use at stage $k+1$ and node s_t is added to V_{rep}^k and removed from $\bar{V}^k$. Otherwise, node s_t is partially repaired in *step (a.ii)* and V_{pend} is set to 1. Resource requirement of node s_t in stage $k+1$, is updated as $r_{s_t}^{k+1} = F_{s_t} \quad A_V^k$ and node s_t is added to V_{rep}^k as a partially repaired node. Note that, node s_t is not removed from $\bar{V}^k$ since it is still partially failed and requires further repair.

Algorithm 4 Joint Progressive network and DC Recovery with Selective Resource Allocation (JPR-SR)

```
Input: $G(V,E)$, $\bar{G}(\bar{V},\bar{E})$, $D$, $\bar{D}$, $C$, $D_c$, $\alpha_c$, $l_{max}$, $A_V^k$, $A_E^k$, $A_D^k$, $r_{s\in\bar{V}}^k$, $r_{d\in\bar{D}}^k$, $F_{s\in\bar{V}}$, $F_{e\in\bar{E}}$, $F_{d\in\bar{D}}$, $T_{s\in\bar{V}}$, $T_{d\in\bar{D}}$, $T_N$, $T_D$.
Output: $V_{rep}^k$, $E_{rep}^k$, $D_{rep}^k$, $R^k$
Initialization: $k=1$. $\bar{V}^k \leftarrow \bar{V}$; $\bar{E}^k \leftarrow \bar{E}$; $\bar{D}^k \leftarrow \bar{D}$. $V_{rep}^k \leftarrow V \setminus \bar{V}$; $E_{rep}^k \leftarrow E \setminus \bar{E}$; $D_{rep}^k \leftarrow D \setminus \bar{D}$. $V_{pend}=0$, $D_{pend}=0$.
  1) While $\{\bar{V}^k$ and $\bar{E}^k$ and $\bar{D}^k\} \neq empty$
       • Network Recovery Phase:
         – Node Repair Resource Allocation
             a) If $\boldsymbol{V_{pend}=0}$, invoke Algorithm 2 to select node $s_t \in \bar{V}^k$ which satisfies $T_N$.
                 i) If $A_V^k \geq F_{s_t}$, fully repair node $s_t$.
                     A) $V_{rep}^k \leftarrow V_{rep}^k \cup \{s_t\}$; $\bar{V}^{k+1} \leftarrow \bar{V}^k \setminus \{s_t\}$.
                     B) $A_V^{k+1} = A_V^{k+1} + (A_V^k - F_{s_t})$.
                ii) Else, partially repair node $s_t$ with $A_V^k$ units.
                     A) $\boldsymbol{V_{pend}=1}$. $V_{rep}^k \leftarrow V_{rep}^k \cup \{s_t\}$.
                     B) $r_{s_t}^{k+1} = F_{s_t} - A_V^k$.
             b) Else, get node $s_t \in V_{rep}^k$ with $r_{s_t}^k$ units required.
                 i) If $A_V^k \geq r_{s_t}^k$, fully repair node $s_t$.
                     A) $\boldsymbol{V_{pend}=0}$. $\bar{V}^{k+1} \leftarrow \bar{V}^k \setminus \{s_t\}$.
                     B) $A_V^{k+1} = A_V^{k+1} + (A_V^k - r_{s_t}^k)$.
                ii) Else, partially repair node $s_t$ with $A_V^k$ units.
                     A) $r_{s_t}^{k+1} = r_{s_t}^k - A_V^k$.
             c) $X_{s_t}^k = [(F_{s_t} - r_{s_t}^{k+1})/F_{s_t}]$.
         – Link Repair Resource Allocation
             a) For each link $e \in \{\mathit{\Gamma}_{s_t,N_{s_t}^k}^k \cap \bar{E}^k\}$
                 i) If $A_E^k \geq F_e$, repair link $e$.
                     A) $E_{rep}^k \leftarrow E_{rep}^k \cup \{e\}$; $\bar{E}^{k+1} \leftarrow \bar{E}^k \setminus \{e\}$.
                     B) $A_E^k = A_E^k - F_e$.
             b) $A_E^{k+1} = A_E^{k+1} + A_E^k$.
       • DC Recovery Phase:
         – DC Repair Resource Allocation
             a) If $\boldsymbol{D_{pend}=0}$, invoke Algorithm 3 to select DC $d_t \in \bar{D}^k$ which satisfies $T_D$.
                 i) If $A_D^k \geq F_{d_t}$, fully repair DC $d_t$.
                     A) $D_{rep}^k \leftarrow D_{rep}^k \cup \{d_t\}$; $\bar{D}^{k+1} \leftarrow \bar{D}^k \setminus \{d_t\}$.
                     B) $A_D^{k+1} = A_D^{k+1} + (A_D^k - F_{d_t})$.
                ii) Else, partially repair node $d_t$ with $A_D^k$ units.
                     A) $\boldsymbol{D_{pend}=1}$. $D_{rep}^k \leftarrow D_{rep}^k \cup \{d_t\}$.
                     B) $r_{d_t}^{k+1} = F_{d_t} - A_D^k$.
             b) Else, get DC $d_t \in D_{rep}^k$ with $r_{d_t}^k$ units required.
                 i) If $A_D^k \geq r_{d_t}^k$, fully repair DC $d_t$.
                     A) $\boldsymbol{D_{pend}=0}$. $\bar{D}^{k+1} \leftarrow \bar{D}^k \setminus \{d_t\}$.
                     B) $A_D^{k+1} = A_D^{k+1} + (A_D^k - r_{d_t}^k)$.
                ii) Else, partially repair DC $d_t$ with $A_D^k$ units.
                     A) $r_{d_t}^{k+1} = r_{d_t}^k - A_D^k$.
             c) $Y_{d_t}^k = [(F_{d_t} - r_{d_t}^{k+1})/F_{d_t}]$.
  2) $R^k = \sum_{s\in V_{rep}^k} \sum_{c\in C} \{R_{(c,s)}^k * X_{s\in V_{rep}^k}^k * Y_{d\in D_{rep}^k}^k * \alpha_c\}$.
  3) $V_{rep}^{k+1} \leftarrow V_{rep}^k$; $E_{rep}^{k+1} \leftarrow E_{rep}^k$; $D_{rep}^{k+1} \leftarrow D_{rep}^k$.
  4) $k = k+1$.
```

Algorithm 5 Joint Progressive Network and DC Recovery with Adaptive Resource Allocation (JPR-AR)

```
Input: $G(V,E)$, $\bar{G}(\bar{V},\bar{E})$, $D$, $\bar{D}$, $C$, $D_c$, $\alpha_c$, $l_{max}$, $A_V^k$, $A_E^k$, $A_D^k$, $r_{s\in\bar{V}}^k$, $r_{d\in\bar{D}}^k$, $F_{s\in\bar{V}}$, $F_{e\in\bar{E}}$, $F_{d\in\bar{D}}$, $T_{s\in\bar{V}}$, $T_{d\in\bar{D}}$, $T_N$, $T_D$.
Output: $V_{rep}^k$, $E_{rep}^k$, $D_{rep}^k$, $R^k$
Initialization: $k=1$. $\bar{V}^k \leftarrow \bar{V}$; $\bar{E}^k \leftarrow \bar{E}$; $\bar{D}^k \leftarrow \bar{D}$. $V_{rep}^k \leftarrow V \setminus \bar{V}$; $E_{rep}^k \leftarrow E \setminus \bar{E}$; $D_{rep}^k \leftarrow D \setminus \bar{D}$.
  1) While $\{\bar{V}^k$ and $\bar{E}^k$ and $\bar{D}^k\} \neq empty$
       • Network Recovery Phase:
         – Node Repair Resource Allocation
             a) Invoke Algorithm 6 to select node $s_t$ with $r_{s_t}^k$ units required.
             b) If $A_V^k \geq r_{s_t}^k$, fully repair node $s_t$.
                 i) $V_{rep}^k \leftarrow V_{rep}^k \cup \{s_t\}$; $\bar{V}^{k+1} \leftarrow \bar{V}^k \setminus \{s_t\}$.
                ii) $A_V^{k+1} = A_V^{k+1} + (A_V^k - r_{s_t}^k)$.
             c) Else, partially repair node $s_t$ with $A_V^k$ units.
                 i) $V_{rep}^k \leftarrow V_{rep}^k \cup \{s_t\}$.
                ii) $r_{s_t}^{k+1} = r_{s_t}^k - A_V^k$.
             d) $X_{s_t}^k = [(F_{s_t} - r_{s_t}^{k+1})/F_{s_t}]$.
         – Link Repair Resource Allocation
             a) For each link $e \in \{\mathit{\Gamma}_{s_t,N_{s_t}^k}^k \cap \bar{E}^k\}$
                 i) If $A_E^k \geq F_e$, repair link $e$.
                     A) $E_{rep}^k \leftarrow E_{rep}^k \cup \{e\}$; $\bar{E}^{k+1} \leftarrow \bar{E}^k \setminus \{e\}$.
                     B) $A_E^k = A_E^k - F_e$.
             b) $A_E^{k+1} = A_E^{k+1} + A_E^k$.
       • DC Recovery Phase:
         – DC Repair Resource Allocation
             a) Invoke Algorithm 7 to select DC $d_t \in \bar{D}^k$ with $r_{d_t}^k$ units required.
             b) If $A_D^k \geq r_{d_t}^k$, fully repair DC $d_t$.
                 i) $D_{rep}^k \leftarrow D_{rep}^k \cup \{d_t\}$; $\bar{D}^{k+1} \leftarrow \bar{D}^k \setminus \{d_t\}$.
                ii) $A_D^{k+1} = A_D^{k+1} + (A_D^k - r_{d_t}^k)$.
             c) Else, partially repair node $d_t$ with $A_D^k$ units.
                 i) $D_{rep}^k \leftarrow D_{rep}^k \cup \{d_t\}$.
                ii) $r_{d_t}^{k+1} \leftarrow r_{d_t}^k - A_D^k$.
             d) $Y_{d_t}^k \leftarrow [(F_{d_t} - r_{d_t}^{k+1})/F_{d_t}]$.
  2) $R^k = \sum_{s\in V_{rep}^k} \sum_{c\in C} \{R_{(c,s)}^k * X_{s\in V_{rep}^k}^k * Y_{d\in D_{rep}^k}^k * \alpha_c\}$.
  3) $V_{rep}^{k+1} \leftarrow V_{rep}^k$; $E_{rep}^{k+1} \leftarrow E_{rep}^k$; $D_{rep}^{k+1} \leftarrow D_{rep}^k$.
  4) $k = k+1$.
```

On the other hand, in *step (b)*, if V_{pend} is 1, i.e., repair of node s_t is pending, available resources A_V^k are allocated to the partially active node s_t. In *step (b.i)*, if the resources are sufficient for full recovery, V_{pend} is set to 0 and A_V^k, V_{rep}^k, and $\bar{V}^k$ are updated accordingly. Otherwise, node s_t is still partially repaired in *step (b.ii)* and $r_{s_t}^{k+1}$ is updated accordingly. In *step (c)*, serving capacity of node s_t at stage k, $X_{s_t}^k$ is obtained as fraction of the allocated resources out of total required resources, $((F_{s_t} \quad r_{s_t}^{k+1})/F_{s_t})$; $X_{s_t}^k$ is 1 for fully repaired nodes and any value $0 < n < 1$ for partially repaired nodes.

Under **link repair resource allocation**, there are no partial repairs and hence depending on resources availability, links are either repaired or not repaired. In *step (a)*, for each failed link e between node s_t and its set of neighboring nodes $N_{s_t}^k$, if available resources are sufficient, i.e., $(A_E^k \leftarrow F_e)$, then link e is repaired and e is added to E_{rep}^k and removed from $\bar{E}^k$. Any remaining resources $(A_E^k \quad F_e)$ are reserved for further repairs.

The **DC repair resource allocation** step in DC recovery phase works in a similar manner as **node repair resource allocation**. A failed DC d_t is selected for repair using Algorithm 3 and available

Algorithm 6 JPR-AR Node Repair Selection

```
1) For each test node $s_t\ /\ \bar{V}^k$ with $r^k_{s_t}$ required units,
    a) If $s_t\ /\ V^k_{rep}$ and $\Gamma^k_{s_t,N^k_{s_t}}\ /\ E^k_{rep}$, $V^k_{test} \in\ V^k_{rep}$; $E^k_{test} \in E^k_{rep}$.
    b) Else, $V^k_{test} \in\ V^k_{rep} \wedge \}s_t|$ ; $E^k_{test} \in\ E^k_{rep} \wedge \Gamma^k_{s_t,N^k_{s_t}}$.
    c) If $A^k_V \leftarrow r^k_{s_t}$, assumed node capacity $X^k_{s_t} = 1$.
    d) Else, $X^k_{s_t} = [(F_{s_t} \quad (r^k_{s_t} \quad A^k_V))/F_{s_t}]$.
    e) If $s_t$ supports an active DC $d\ /\ D^k_{rep}$ (or no DC), $D^k_{test} \in\ D^k_{rep}$. Get DC capacity, $Y^k_d$ (if any).
    f) Else, get failed DC $d_t\ /\ \bar{D}$ supported by $s_t$ with $r^k_{d_t}$ units required. $D^k_{test} \in\ D^k_{rep} \wedge \}d_t|$ .
        i) If $A^k_D \leftarrow r^k_{d_t}$, assumed DC capacity $Y^k_{d_t} = 1$.
        ii) Else, $Y^k_{d_t} = [(F_{d_t} \quad (r^k_{d_t} \quad A^k_D))/F_{d_t}]$.
    g) $R^k_{s_t} = \sum_{s\in V^k_{test}} \sum_{c\in C} \}R^k_{(c,s)} \leq \alpha_c \leq X^k_{s\in V^k_{test}} \leq Y^k_{d\in D^k_{test}}|$ .
    h) $S^k_{rank} \in\ \}R^k_{s_t}|$ .
2) Sort $S^k_{rank}$ in descending order of $R^k_{s_t}$ value.
3) Select node $s_t$ with max $R^k_{s_t}$ value.
    a) If recovery time $T_{s_t} \geq T_N$, select node $s_t$ for repair.
    b) Else, select next best node $s_t$ within $T_N$ constraint.
```

Algorithm 7 JPR-AR DC Repair Selection

```
1) For each test DC $d_t\ /\ \bar{D}^k$ with $r^k_{d_t}$ units required,
    a) If $d_t\ /\ D^k_{rep}$, $D^k_{test} \in\ D^k_{rep}$.
    b) Else, $D^k_{test} \in\ D^k_{rep} \wedge \}d_t|$ .
    c) If $A^k_D \leftarrow r^k_{d_t}$, assumed DC capacity $Y^k_{d_t} = 1$.
    d) Else, $Y^k_{d_t} = [(F_{d_t} \quad (r^k_{d_t} \quad A^k_D))/F_{d_t}]$.
    e) If DC $d_t$ has an active supporting node $s\ /\ V^k_{rep}$, get node capacity, $X^k_s$. $V^k_{test} \in\ V^k_{rep}$; $E^k_{test} \in\ E^k_{rep}$.
    f) Else, get failed supporting node $s_t\ /\ \bar{V}^k$ with $r^k_{s_t}$ units required. $V^k_{test} \in\ V^k_{rep} \wedge \}s_t|$ ; $E^k_{test} \in\ E^k_{rep} \wedge \Gamma^k_{s_t,N^k_{s_t}}$.
        i) If $A^k_V \leftarrow r^k_{s_t}$, assumed node capacity $X^k_{s_t} = 1$.
        ii) Else, $X^k_{s_t} = [(F_{s_t} \quad (r^k_{s_t} \quad A^k_V))/F_{s_t}]$.
    g) $R^k_{d_t} = \sum_{s\in V^k_{test}} \sum_{c\in C} \}R^k_{(c,s)} \leq \alpha_c \leq X^k_{s\in V^k_{test}} \leq Y^k_{d\in D^k_{test}}|$ .
    h) $D^k_{rank} \in\ \}R^k_{d_t}|$ .
2) Sort $D^k_{rank}$ in descending order of $R^k_{d_t}$ value.
3) Select DC $d_t$ with max $R^k_{d_t}$ value.
    a) If recovery time $T_{d_t} \geq T_D$, select DC $d_t$ for repair.
    b) Else, select next best DC $d_t$ within $T_D$ constraint.
```

DC resources A^k_D are allocated following the *SR* strategy. After full/partial repair of DC d_t, A^k_D, D^k_{rep}, and $\bar{D}^k$ are updated accordingly. Serving capacity of DC d_t, $Y^k_{d_t}$ is obtained as fraction of the allocated resources out of total required resources, $((F_{d_t} \quad r^{k+1}_{d_t})/F_{d_t})$; $Y^k_{d_t}$ is 1 for fully repaired DCs and any value $0 < n < 1$ for partially repaired DCs.

In *step (2)*, final cumulative weighted content reachability at stage k, R^k is computed using anycast routing with $V^k_{rep}, D^k_{rep}, E^k_{rep}$, and the partial capacities X^k_s and Y^k_d. The algorithm then moves to the next stage and allocates resources for ongoing repairs if V_{pend} and/or D_{pend} are 1. Otherwise, the algorithm selects new elements for repair and the process continues until all failed components are repaired.

2) JPR-AR Algorithm: JPR-AR heuristic incorporates repair resource availability *in the selection* of network nodes and DCs to be repaired. In contrast to *SR* strategy, where JPR-SR selects nodes and DCs for repair with the assumption of full recovery (and hence full capacity) irrespective of resources availability, JPR-AR selects nodes and DCs based on actual recovery or serving capacity potential of the elements with the available resources. Hence, the selection process is *resources-aware* during reachability metric computation to reflect the actual content reachability that can be achieved at a stage with the given resources.

As shown in Algorithm 5, node selection is determined in the network recovery phase using *JPR-AR Node Repair Selection* algorithm and DC selection is determined in the DC recovery phase using *JPR-AR DC Repair Selection* algorithm. In Algorithm 6, in *steps (1.a, 1.b)*, for each test node s_t in $\bar{V}^k$, test set of nodes, links, and DCs, V^k_{test}, E^k_{test}, and D^k_{test} are formed for reachability computation as in Algorithm 2 with the exception that if node s_t is a partially repaired node (i.e., $s_t\ /\ V^k_{rep}$) and if its adjacent links are repaired (i.e., $\Gamma^k_{s_t,N^k_{s_t}}\ /\ E^k_{rep}$), then V^k_{test} is V^k_{rep} and E^k_{test} is E^k_{rep}. In *steps (1.c, 1.d)*, possible serving capacity of node s_t, $X^k_{s_t}$ is determined by *assuming* allocation of available resources, A^k_V. If resources are sufficient for full recovery, $X^k_{s_t}$ is assumed to be 1, otherwise, $X^k_{s_t}$ is assumed based on A^k_V and resource requirement F_{s_t}. In *steps (1.e, 1.f))*, if node s_t supports a fully or partially repaired DC d_t with serving capacity $Y^k_{d_t}$, then D^k_{test} is D^k_{rep} and if s_t supports a failed DC d_t, D^k_{test} is formed by joining d_t with D^k_{rep} and $Y^k_{d_t}$ is assumed based on F_{d_t} and A^k_D. Content reachability with test node s_t, $R^k_{s_t}$ is computed in *step (1.g)* with V^k_{test}, E^k_{test}, and D^k_{test} considering $X^k_{s_t}$ and $Y^k_{d_t}$. The algorithm then selects the node s_t with highest $R^k_{s_t}$ value such that T_{s_t} satisfies T_N. Similarly a failed or a partially repaired DC is selected for repair in Algorithm 7.

After the node and DC selection, the resource allocation procedure is similar to JPR-SR except that there is no notion of *pending* repairs. In Algorithm 5, at stage k, the selected node s_t and DC d_t are either fully repaired or partially repaired depending on available resources at that stage. In stage $k+1$, JPR-AR runs new selection process through Algorithm 6 and 7 with the newly available repair resources. As a result, an element currently under repair or a new element can be selected for repair, whichever leads to maximum content reachability at that stage. Hence, a partially repaired element may be fully recovered at a much later stage and in the meantime, other elements can be selected for repair. The process continues until all failed nodes, links, and DCs are fully recovered.

The *AR* strategy allows the selection process to ”adapt” to the available repair resources at each stage. However, such strategy requires flexible resources management, meaning that the repair crew and resources have to be able to move to different repair sites in between stages as the strategy may select different locations at different stages even if recovery at one location is not complete. On the other hand, *SR* strategy requires repair process at one location to be fully complete before moving to other locations. Depending on the flexibility of the repair crews, the two strategies can be chosen accordingly.

V. Illustrative Numerical Examples

A. Simulation Setup

To evaluate the performance of different progressive recovery algorithms discussed in this study, we use *gain* in cumulative weighted content reachability (R) in the network at each stage of the repair process as the performance metric. It represents the amount of additional reachability that is recovered in the network after services are disrupted due to disaster. Hence, it excludes any existing reachability in the network prior to the disaster. We simulated over a number of network topologies with different settings as discussed in the following sub-sections. For our simulation, we consider that contents are uniformly distributed among the DCs and that all user nodes request all contents. Each content is assigned α_c on a scale of 1-10 using a uniform distribution. We consider l_{max} as the number of hops between the serving DCs and the requesting nodes.

B. Benchmark with ILP

We start our numerical analysis by evaluating our proposed JPR heuristic and ILP model on 14-node NSFnet topology with a possible EMP attack as shown in Fig. 3. There are 6 DCs located on 6 supporting nodes in the network and, for our simulation, we consider that nodes, links, and DCs in the red disaster zone are failed with probability 1. Number of contents = 30 (low numbers due to limited scalability of ILP), number of replicas per content = 2 to 3, $l_{max} = 4$.

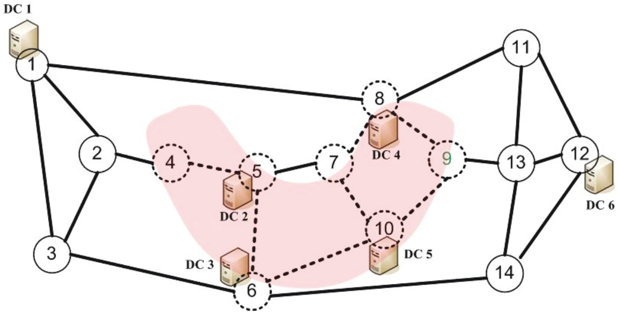

Fig. 3: 14-node NSFnet topology with possible EMP attack span.

We compare the per-stage *gain* in content reachability R of JPR algorithm and ILP model as shown in Fig. 4. Upon full recovery in the network, all approaches attain the maximum content reachability and hence the reachability values are normalized to the highest value and other values are adjusted accordingly. Therefore, the lowest value 0 represents start of disruption in content reachability due to disaster and the highest value 1 represents maximum content reachability due to full network recovery. The subsequent stages show the recovered cumulative weighted content reachability in the progressively recovering network after the disaster.

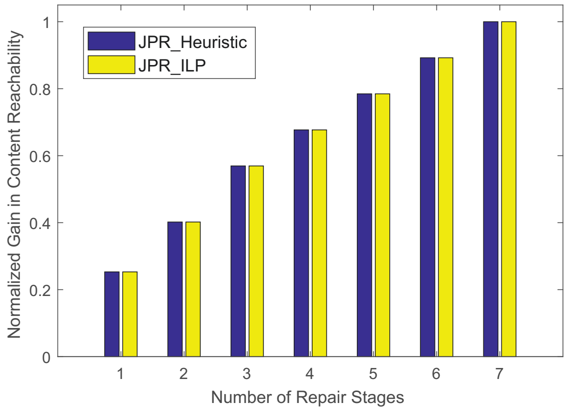

Fig. 4: Comparison of recovered cumulative weighted content reachability **R** at each stage from ILP model and heuristic.

As per our initial joint progressive recovery model assumption, number of repair stages, K = max}*number of failed nodes, number of failed DCs*| which is 7 in this case. We show that our JPR algorithm follows closely the upper bound obtained from ILP. The repair sequences of the failed nodes and the failed DCs obtained from the ILP model are: }8, 6, 5, 7, 10, 9, 4| and }4, 3, 2, 5| , respectively. The repair sequences of the failed nodes and the failed DCs obtained from our heuristic are: }8, 6, 5, 7, 10, 4, 9| and }4, 3, 2, 5| , respectively. The first two selected nodes, nodes 8 and 6, have two active end nodes each, hence after repair, the DCs on these nodes (DCs 4 and 3) can serve most number of users in the network leading to maximum content reachability at the initial stages.

C. Results on Larger Networks

1) Results with 24-node USnet topology: To demonstrate the benefits of JPR algorithm over disjoint approaches, we simulated over 24-node USnet topology with a possible EMP attack as shown in Fig. 5. There are 10 DCs located on 10 supporting nodes in the network and nodes, links, and DCs in the red disaster zone are failed with probability 1. Number of contents = 100, number of replicas per content = 2 to 4, l_{max} = 2. As per our model, depending on the number of failed components in the network, total number of stages required for full recovery is 8.

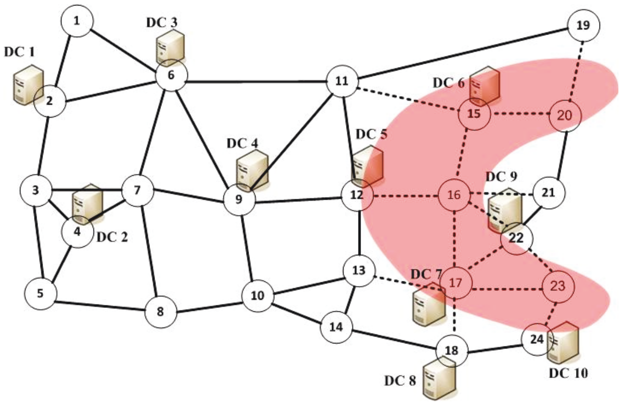

Fig. 5: 24-node USnet topology with possible EMP attack span.

TABLE II: Node and DC repair sequences for JPR, DJPR-Greedy, and DJPR-Random algorithms for 24-node USnet topology.

	Node repair sequence
JPR	12 - 17 - 16 - 15 - 22 - 23 - 24 - 20
DJPR-Greedy	16 - 17 - 12 - 22 - 15 - 20 - 23 - 24
DJPR-Random	23 - 16 - 22 - 20 - 24 - 15 - 12 - 17
	DC repair sequence
JPR	5 - 7 - 9 - 6 - 10
DJPR-Greedy	9 - 10 - 6 - 5 - 7
DJPR-Random	6 - 5 - 10 - 9 - 7

We compare our JPR algorithm with greedy and random DJPR approaches and the sequences in which nodes and DCs are repaired in the three approaches are shown in Table II. As described in Section III-B, greedy disjoint approach selects nodes and DCs for repair independently with individual objectives rather than a joint objective and random disjoint approach selects nodes and DCs randomly. Hence, in stage 1, DJPR-Greedy selects node 16 which has the highest nodal degree in the physical network and DC 9 which contains more important contents (and DJPR-Random selects node 23 and DC 6 randomly). However, these selections do not contribute to content reachability as the selected DCs do not have active supporting nodes and the selected nodes do not have many active neighboring nodes at stage 1. Our JPR approach selects node 12 and DC 5 which leads to maximum content reachability.

We also compare the per-stage *gain* in content reachability R of the three approaches as shown in Fig. 6. The results for DJPR-Random approach report an average over multiple simulation runs of the random DJPR algorithm. The reachability values are normalized to the highest value and other values are adjusted accordingly. We show that our proposed JPR algorithm provides significantly higher content reachability at the initial stages compared to the disjoint

(DJPR-Greedy and DJPR-Random) approaches. In the first repair stage, only our JPR approach provides additional content reachability in the network after the disaster, while the other approaches fail to provide any content reachability to users. In stages 2 and 3, JPR provides almost 4 times and 3 times more reachability than the disjoint approaches, respectively. In post-disaster recovery, the first few stages are more crucial due to immediate urgency and our approach is most effective in the first four stages. As nodes and DCs get repaired progressively in both joint and disjoint approaches, network gets more connected and more contents become available providing higher flexibility in routing and reachability. Hence, all the approaches converge towards same reachability even though the node and DC repair sequences are different.

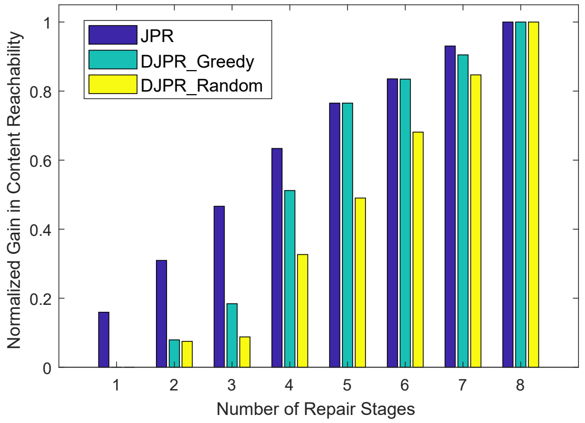

Fig. 6: Comparison of gain in cumulative weighted content reachability **R** between joint and disjoint progressive recovery approaches.

In Fig. 7, we compare the total amount of recovered user requests (denoted by $R^k_{c,s}$) in the post-disaster network at each repair stage by joint and disjoint approaches. Here, we also show the distribution of content priority in the user requests based on α_c. The contents are classified as low-priority ($1 < \alpha_c \geq 4$), medium-priority ($4 < \alpha_c \geq 7$), and high-priority ($7 < \alpha_c \geq 10$). All the values are normalized to the highest value (total amount of contents); The bars represent different approaches based on color map and the segments in each bar represent the content priority.

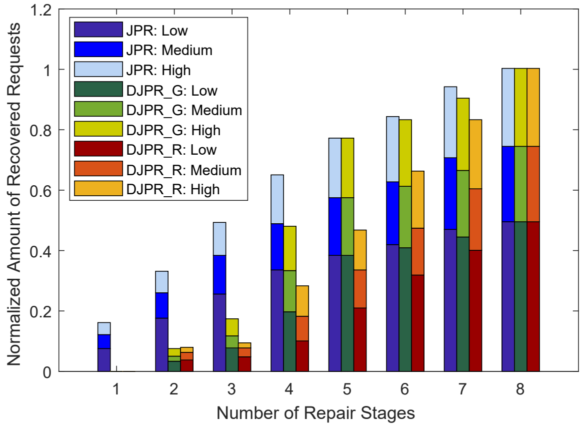

Fig. 7: Comparison of amount of recovered user requests $R^k_{c,s}$ with priority distribution between joint and disjoint approaches.

We show that in the first stage, only JPR approach allows users to access all types of contents. In the next two stages, users can access on average 3 and 4 times more high-priority contents compared to disjoint recovery approaches, respectively. Our JPR algorithm favors reachability to important contents since reachability computation takes into account of content priority. Access to important contents immediately after a disaster can be very crucial.

Next, we evaluate our *resource-aware* progressive recovery approaches by comparing our joint approach with greedy disjoint approach. We employ the two resource allocation strategies called selective resource allocation (SR) and adaptive resource allocation (AR) in both JPR and DJPR-Greedy algorithms. The simulation parameters used in resource allocation strategies (such as, per-stage resource availability, per-element resource requirement, etc.) are given in Table III.

TABLE III: Simulation parameters used in resource-aware recovery approaches for 24-node USnet topology.
$A^k_{V/E/D}$ = available network / DC repair resources in units at stage k
$F_{s/e/d}$ = Total resource requirement in units of node s / link e / DC d
$T_{s/d}$ = Total recovery time of node s / DC d

$T_N = 24$ hrs	**Node repair resources**
A^k_V, $k = 1..11$	100, 80, 20, 70, 90, 100, 50, 60, 30, 100, 10
Nodes	**12 15 16 17 20 22 23 24**
F_s	110, 80, 50, 130, 60, 100, 90, 70
T_s	16, 12, 8, 20, 15, 10, 14, 10
$T_N = 24$ hrs	**Link repair resources**
A^k_E, $k = 1..11$	50, 50, 50, 30, 40, 50, 10, 20, 40, 60, 20
Links	All failed links
F_e	30 for all links
$T_D = 60$ hrs	**DC repair resources**
A^k_D, $k = 1..11$	100, 50, 20, 70, 80, 50, 100, 20, 40, 80, 100
DCs	**5 6 7 9 10**
F_d	120, 70, 50, 100, 90
T_d	30, 15, 23, 30, 28

We can see that with the simplistic resource constraint (full repair of one node and one DC per stage), the recovery approaches required 8 stages in total for full recovery of the network as shown in Fig. 6. Whereas with realistic resource allocation strategies, the same network with same setup requires 3 additional stages for full recovery as shown in Fig. 8 because, based on available resources, some components require multiple stages to be fully repaired. The sequence of node repair and DC repair of JPR and DJPR-Greedy approach with *SR* and *AR* strategy are shown in Table IV. Here, *(p)* and *(f)* denotes partial repair and full repair, respectively.

TABLE IV: Resource-aware node and DC repair sequences for JPR-SR, JPR-AR, DJPR-Greedy-SR, and DJPR-Greedy-AR algorithms for 24-node USnet topology (based on Table III).

	Node repair sequence
JPR-SR	12p - 12f - 17p - 17f - 16f - 15f - 22f - 23f - 24p - 24f - 20f
JPR-AR	12p - 17p - 16p - 15p - 22f - 23f - 24f - 20f - 16f - 17f - 15f - 12f
DJPR-Greedy-SR	16f - 17f - 12p - 12p - 12f - 22f - 15f - 20p - 23pf - 23f - 24f
DJPR-Greedy-AR	16f - 17f - 12p - 12p - 12f - 22f - 15f - 20p - 23pf - 23f - 24f
	DC repair sequence
JPR-SR	5p - 5f - 7f - 9p - 9f - 6f - 10f
JPR-AR	5p - 7f - 9p - 6f - 9f - 10p - 10f - 5f
DJPR-Greedy-SR	9f - 10p - 10p - 10f - 6f - 5p - 5f - 7f
DJPR-Greedy-AR	9f - 10p - 10p - 10f - 6f - 5p - 5f - 7f

We compare the per-stage *gain* in cumulative weighted content reachability R of JPR-SR and JPR-AR approaches with DJPR-Greedy-SR and DJPR-Greedy-AR approaches as shown in Fig. 8. The reachability values are normalized to the highest value and other values are adjusted accordingly. We show that with both resource allocation strategies *SR* and *AR*, our joint approach performs significantly better than the disjoint approach. In the first stage, only JPR provides content reachability to users. In the initial stages 2 - 5, compared to DJPR-Greedy approach, our JPR approach provides about 2 to 2.5 times higher content reachability with *SR* strategy, and about 3 times higher content reachability with *AR* strategy.

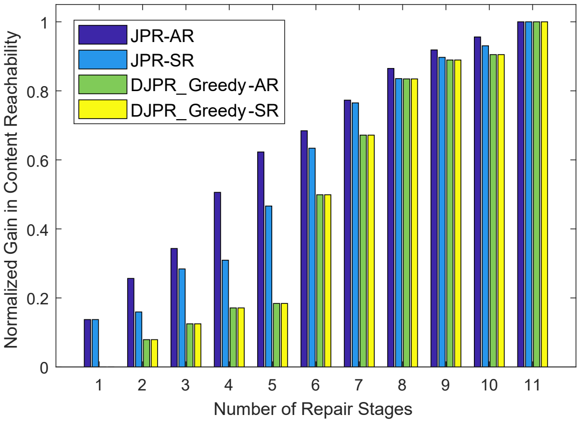

Fig. 8: Comparison of joint and greedy disjoint approaches with selective (SR) and adaptive (AR) resource allocation strategies.

The performance of the two strategies is similar at the later stages but at the initial stages, *AR* strategy performs better (significantly in stage 5) since it "adapts" the resource availability at each stage in the node and DC repair selection algorithms and hence resources are better utilized towards providing higher content reachability. We also show that the different strategies do not affect DJPR-Greedy approach since there is no intelligent *resource-aware joint selection* of repair elements in disjoint recovery approach.

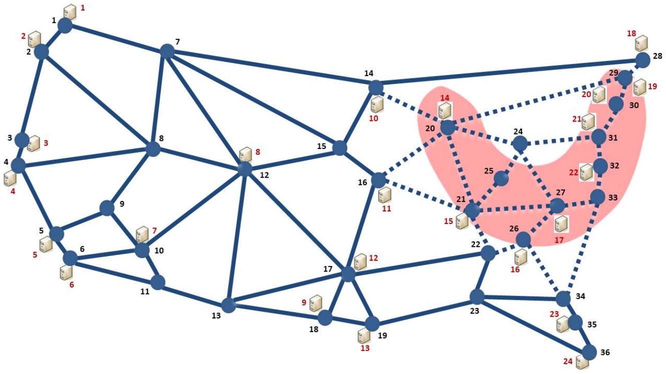

Fig. 9: 36-node USnet topology with 24 DCs and possible EMP attack span.

2) Results with 36-node topology: Post-disaster recovery approaches can be highly dependent on topological attributes (such as, network connectivity, DC placements), content and demand distribution, resource availabilities, etc. In order to get a better understanding of our proposed recovery schemes, we simulated over a more realistic layout of US-wide 36-node network topology with 24 DCs (loosely based on a major operator's DC location map [28], [29]) as shown in Fig. 9. Note that there are higher number of DCs in the US west coast and east coast regions due to higher strategic importance and population density. Similar to our previous simulation, we consider a possible EMP attack and nodes, links, and DCs in the red disaster zone are failed with probability 1. Number of contents = 100, number of replicas per content = 2 to 6, l_{max} = 2.

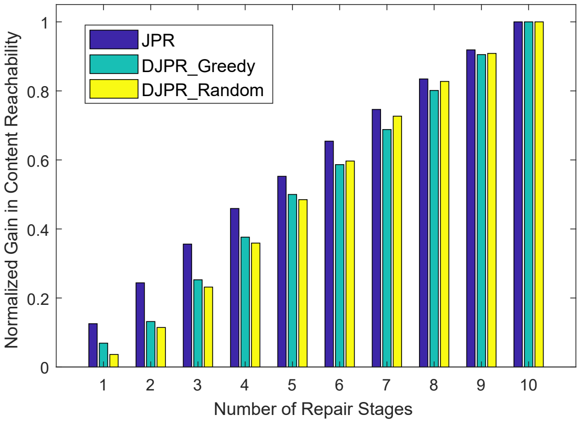

Fig. 10: Comparison of gain in cumulative weighted content reachability **R** between joint and disjoint progressive recovery approaches.

We first evaluate our JPR approach and the greedy and random DJPR approaches without any resource allocation. The total number of stages required for full recovery is 10. We compare the per-stage *gain* in cumulative weighted content reachability R of the three approaches as shown in Fig. 10. We show that JPR algorithm provides significantly higher content reachability at the initial stages compared to the disjoint (DJPR-Greedy and DJPR-Random) approaches. In the first three stages, our JPR approach provides about 1.5 to 3 times higher recovered content reachability in the network after the disaster compared to the disjoint approaches.

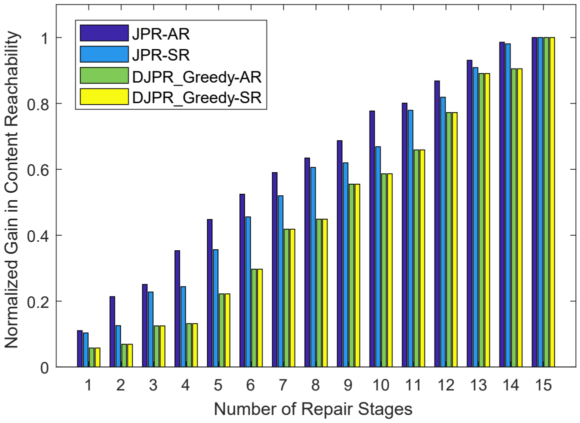

Fig. 11: Comparison of joint and greedy disjoint approaches with selective (SR) and adaptive (AR) resource allocation strategies.

To analyze our resource allocation strategies in this network, we compare the per-stage *gain* in cumulative weighted content reachability R of our joint and greedy disjoint approaches with *SR* and *AR* strategies as shown in Fig. 11. We can see that with resource allocation, the recovery approaches require 15 stages in total for full recovery. We show that with both resource allocation strategies, our joint approach performs significantly better than the greedy disjoint approach. In the initial stages 1 to 6, compared to DJPR-Greedy approach, JPR approach provides about 1.5 to 1.9 times higher content reachability with *SR* strategy, and about 1.8 to 3 times higher content reachability with *AR* strategy. The *AR* strategy is more effective (significantly in stages 2 and 4) due better resource utilization.

VI. Conclusion

In today's world, any disruption in cloud networks can mean severe impact on businesses and individual users. Enterprises providing cloud services through owned or leased DCs can lose millions in terms of customer data if their infrastructures fail. That is why, disaster recovery is a huge part DC and network management and providers are exploring new ways to secure their infrastructures against failures. However, failures due to large-scale disasters can be unavoidable at times. After a disaster in cloud network affecting both DCs and the underlying network, it is very important to plan a recovery strategy which can resume maximum amount of services possible with post-disaster limited resources. We proposed a joint progressive network and DC recovery (JPR) scheme which selects the sequence in which failed network elements and DCs should be repaired in a coordinated plan such that cumulative weighted content reachability to users is maximized at each repair stage. We solved an optimization model as a benchmark of our heuristic and showed that our algorithm performs closely. Compared to disjoint progressive recovery approach in which network repair and DC repair are independent with separate objectives, our joint approach provides significant improvement in users's reachability to important contents especially at the initial repair stages which can be crucial. Our joint progressive recovery model initially assumed a simplistic repair resource constraint which allows full recovery of one network node and one DC at each repair stage. We then employed more realistic resource-aware approach in our joint recovery scheme and proposed *selective* and *adaptive* resource allocation strategies. These strategies consider both full and partial recovery of network nodes and DCs and allocate repairs based on resource availability at each stage. Our results show that *adaptive* resource allocation strategy lead to better resource utilization in providing higher content reachability to users. In post-disaster cloud recovery, our methods can benefit fast service restoration and efficient resource allocation through a joint and coordinated network and DC repair scheme.

References

[1] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, "Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure," *Proc. ACM SIGCOMM*, Florianpolis, Brazil, Aug. 2016.

[2] http://www.datacenterknowledge.com/archives/2009/06/11/lightning-strike-triggers-amazon-ec2-outage.

[3] http://www.huffingtonpost.com/2012/10/30/hurricane-sandy-websites-floods-data-centers_n_2046034.html.

[4] https://arstechnica.com/information-technology/2017/08/tropical-storm-harvey-takes-out-911-centers-cell-towers-and-cable-networks/.

[5] M. Pourvali, K. Liang, F. Gu, H. Bai, K. Shaban, S. Khan, N. Ghani, "Progressive recovery for network virtualization after large-scale disasters," *Proc. IEEE ICNC*, Kauai, HI, Feb. 2016.

[6] C. Ma, J. Zhang, Y. Zhao, M. F. Habib, S. S. Savas and B. Mukherjee, "Traveling repairman problem for optical network recovery to restore virtual networks after a disaster [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B81-B92, Nov. 1 2015.

[7] J. Wang, C. Qiao and H. Yu, "On progressive network recovery after a major disruption," *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011.

[8] B. Mukherjee, M. F. Habib, and F. Dikbiyik, "Network adaptability from disaster disruptions and cascading failures," *IEEE Commununications Magazine*, vol. 52, no. 5, pp. 230-238, May 2014.

[9] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, "Assessing the vulnerability of fiber infrastructure to disasters," *IEEE/ACM Transactions on Networking*, vol. 19, no. 6, pp. 1610-1623, Dec. 2011.

[10] S. Banerjee, S. Shirazipourazad, P. Ghosh, and A. Sen, "Beyond connectivity - new metrics to evaluate robustness of networks," *Proc. IEEE HPSR*, Cartagena, Spain, July, 2011.

[11] F. Dikbiyik, M. Tornatore, and B. Mukherjee, "Minimizing the risk from disaster failures in optical backbone networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 32, no. 18, pp. 3175-3183, Sept. 2014.

[12] M. F. Habib,M. Tornatore, M. D. Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks, *Journal of Lightwave Technology*, vol. 30, no. 16, pp. 2563-2573, Aug. 2012.

[13] Z. I. Rauen, B. Kantarci, H. T. Mouftah, "Resiliency versus energy sustainability in optical inter-datacenter networks," *Optical Switching and Networking*, vol. 23, no. 2, pp. 144-155, Jan. 2017.

[14] S. Ferdousi, F. Dikbiyik, M. F. Habib, M. Tornatore and B. Mukherjee, "Disaster-aware datacenter placement and dynamic content management in cloud networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 7, pp. 681-694, July 2015.

[15] J. Xiao, H. Wen, B. Wu, X. Jiang, P. H. Ho and L. Zhang, "Joint Design on DCN Placement and Survivable Cloud Service Provision over All-Optical Mesh Networks," *IEEE Transactions on Communications*, vol. 62, no. 1, pp. 235-245, Jan. 2014.

[16] R. S. Couto, S. Secci, M. E. M. Campista, and L. H. M. K. Costa, "Server placement with shared backups for disaster-resilient clouds," *Computer Networks*, vol. 93, no. 3, pp. 423-434, Dec. 2015.

[17] L. Ma, X. Jiang, B. Wu, A. Pattavina, N. Shiratori, "Probabilistic region failure-aware data center network and content placement," *Computer Networks*, vol. 103, no. 1, pp. 56-66, July 2016.

[18] C. Colman-Meixner, C. Develder, M. Tornatore and B. Mukherjee, "A Survey on Resiliency Techniques in Cloud Computing Infrastructures and Applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2244-2281, Feb. 2016.

[19] R. D. Souza Couto, S. Secci, M. E. Mitre Campista, and L. H. Maciel Kosmalski Costa, "Network design requirements for disaster resilience in IaaS clouds," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 52-58, Oct. 2014.

[20] I. B. B. Harter, D. A. Schupke, M. Hoffmann and G. Carle, "Network virtualization for disaster resilience of cloud services," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 88-95, Dec. 2014.

[21] F. Gu, K.Shaban, N. Ghani, S.Khan, M. R. Naeini, M. M. Hayat, C. Assi, "Survivable Cloud Network Mapping for Disaster Recovery Support," *IEEE Transactions on Computers*, vol. 64, no. 8, pp. 2353-2366, Aug. 2015.

[22] S. Kamamura, D. Shimazaki, K. Genda, K. Sasayama, and Y. Uematsu, "Disaster recovery for transport network through multiple restoration stages," *IEICE Transactions on Communications*, vol. E98-B, no. 1, pp. 171-179, Jan. 2015.

[23] H. Yu and C. Yang, "Partial network recovery to maximize traffic demand," *IEEE Commununications Letter*, vol. 15, no. 12, pp. 1388-1390, Dec. 2011.

[24] K. A. Sabeh, M. Tornatore, F. Dikbiyik, "Progressive network recovery in optical core networks," *Proc. IEEE RNDM*, Munich, Germany, Oct. 2015.

[25] G. Ellinas, D. Papadimitriou, J. Rak, D. Staessens, J. Sterbenz, and K. Walkowiak, "Practical issues for the implementation of survivability and recovery techniques in optical networks," *Optical Switching and Networking*, vol. 14, no.2, pp. 179-193, Aug. 2014.

[26] M.Pourvali, C.Cavdar, K.Shaban, J.Crichigno, and N.Ghani, "Post-failure repair for cloud-based infrastructure services after disasters," *Computer Communications*, vol. 111, no.1, pp. 29-40, Oct. 2017.

[27] S. Ferdousi, F. Dikbiyik, M. Tornatore, B. Mukherjee, "Progressive datacenter recovery over optical core networks after a large-scale disaster," *Proc. IEEE DRCN*, Paris, France, Mar. 2016.

[28] http://www.level3.com/en/products/data-centers/north-america-data-centers/.

[29] https://www.business.att.com/solutions/Service/cloud/colocation/data-center-locations/.