Overload-Based Cascades in Multiplex Networks with Decoupled Functionalities

Orkun İrsoy, Osman Yağan, *Senior Member, IEEE*

***Abstract*—Many networked systems consist of units that perform multiple tasks in parallel, where the workload of one task can reduce performance in another. In cloud infrastructures, for example, a server may handle GPU-intensive inference and memory-intensive data services simultaneously, so that high demand in one function degrades the capacity available to the other. Such systems can be modeled as multiplex networks, with each layer representing a task type supported by the same set of nodes. These networks are prone to overload-based cascading failures: when a node fails, its task-specific load is redistributed, potentially triggering further failures. We study cascading failures under a novel failure condition, termed *layer-influenced decoupled overload*, where survival in each layer depends on loads across all layers, while failure in one layer does not necessarily disable the others. For the global redistribution rule, we derive recursive equations that characterize cascade evolution and final system sizes as functions of the attack size, the joint load/free-space distribution, and cross-layer influence parameters. We also run simulations under a local redistribution rule and find that the main qualitative conclusions remain consistent when redistribution is constrained by network topology. Our analysis and simulations reveal complex behaviors, including asymmetric layer collapses and non-monotonic robustness curves driven by cross-layer interactions. Finally, we propose and evaluate several free-space allocation strategies aimed at maximizing overall system robustness.**

***Index Terms*—Robustness, Cascading Failures, Overload-based cascades, Flow Networks, Multiplex Networks**

I. Introduction

IN October 2025, a major outage in Amazon Web Services (AWS) disrupted internet platforms worldwide after a routing fault caused congestion and traffic redistribution, overloading multiple servers and triggering cascading service disruptions [1]. Earlier that year, in April 2025, an over-voltage in the Iberian power grid initiated a cascade of generation losses and load shedding, resulting in a large-scale blackout across Spain and Portugal [2]. In both events, the failure of one component transferred excess load to dependent components, propagating through the network and leading to a system-wide collapse. Such *cascading failures* [3] frequently occur in *networked systems* such as supply chains [4], [5] and cloud data centers [6]. The severity of these disruptions highlight the importance of designing more robust infrastructure. The Iberian blackout, for instance, affected tens of millions of people and severely interrupted transportation, telecommunications, and industrial activities across the region [2].

Existing studies on cascading failures consider various mechanisms of failure propagation. The *percolation-based models* [3], [7]–[11] focus on structural connectedness and are utilized for networks where system functionality depends on mutual reachability. These models are particularly suitable for communication and cyber-physical systems, where the loss of connectivity directly impairs operation. A separate line of work focuses on *overload-based* or *flow-redistribution* models [12]–[19], which capture systems governed by physical or workload flows, such as power transmission, traffic, or distributed computing. In flow networks, failure of a node typically causes redistribution of its load among surviving nodes, which may result in overload and subsequent failures.

Most existing studies on the robustness of flow networks have focused on single-layer structures [12]–[19] or on interconnected multilayer systems in which flow can be redistributed across layers [20]–[26]. These approaches, however, do not capture settings where multiple functionalities coexist on the same node and compete for its finite resources. This limitation highlights the need to analyze *multiplex flow networks*, in which each node simultaneously supports several distinct flows, and its operational state is determined jointly by load–capacity conditions across multiple layers. Many real-world systems exhibit precisely this structure. Units often perform multiple interdependent functions, where heavy load in one task degrades performance in others. In cloud infrastructures, for example, a server may handle GPU-intensive inference and memory-intensive data services in parallel, such that heavy demand in one task degrades performance in another [6]. Similar patterns arise in management systems where each unit handles multiple tasks, and in supply-chain networks where facilities oversee the production and distribution of multiple products.

Despite the relevance of multiplex flow networks, research in this area remains limited. Existing studies either focus on specific coupling strategies between dual layers [27], [28], where node dependencies are modeled through topological connections or initial load assignments, or extend classical single-layer models, such as the sandpile model, to multiplex settings where cascades are triggered by random load increments [29]. A recent study [30] proposed a multiplex overload model in which the load in one layer influences the effective capacity in another, highlighting the impact of cross-layer coupling on failure propagation. However, this exploration remains limited to empirical datasets and does not generalize to arbitrary load–capacity distributions or analytical formulations of the cascade dynamics.

O. İrsoy and O. Yağan are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213. oirsoy@andrew.cmu.edu, oyagan@andrew.cmu.edu.

This work was supported in part by the Air Force Office of Scientific Research (AFOSR) Grant # FA9550-22-1-0233.

O. İrsoy gratefully acknowledges the support of Knight Fellowship through the IDeaS Center at Carnegie Mellon University for the 2024-2025 academic year.

In our prior work [31], we introduced a multiplex flow network model in which the overload condition of a node depends jointly on its loads across all layers. In that formulation, however, the functionality of a node was fully coupled across layers; that is, if a node failed in one layer, it simultaneously failed in all others. This assumption overlooks scenarios where functionalities can be decoupled while still sharing common resource constraints. A recent example is the 5G network-slicing infrastructure, where different slices (e.g., enhanced mobile broadband, ultra-reliable low-latency communication, and massive IoT) compete for shared physical resources such as spectrum, processing capacity, and memory [32]. Overload in one slice may degrade the available capacity for others, but a slice can remain operational for certain services depending on resource allocation and service-level requirements. A similar principle applies to supply chain networks, where facilities handle production and distribution of multiple product types. Although the effective capacity for handling one product is influenced by the workload associated with others due to shared resources, a facility may temporarily halt operations for one type while sustaining functionality for others. These examples highlight the importance of capturing systems in which layers are coupled through shared capacities but can remain operationally independent.

To this end, we propose a new failure condition, termed the *layer-influenced decoupled overload*, to model systems where different functionalities share limited resources but can continue operating independently after partial failures. Under this condition, the survival of each node in a given layer depends jointly on the load it supports across all layers, reflecting the effect of shared resource constraints. However, when a node fails in one layer, its load in that layer is set to zero and redistributed among the remaining active nodes within the same layer, while the node may continue to operate in other layers as long as their survival conditions are not violated. This mechanism captures a broad class of systems where functional interdependencies arise from shared node level resources rather than from direct operational coupling, extending the fully coupled overload case studied previously.

We analyze the overload-based cascading dynamics of this model through a mean-field framework, derive recursive equations for the evolution of surviving nodes under the new failure condition, and examine how partial decoupling affects robustness, critical transition behavior, and system performance. Our analysis reveals that partial decoupling between layers leads to qualitatively new cascading behaviors such as asymmetric layer collapses and non-monotonic changes in final system size with increasing attack size driven by cross-layer influence.

The main contributions of this paper are summarized as follows:

- We introduce a new overload condition, termed the *layer-influenced decoupled overload*, that captures systems where different functionalities share common resources yet can remain operationally independent after partial failures.
- We develop an analytical framework to characterize cascade dynamics under this condition and derive recursive equations describing the evolution of node survival across layers.
- We demonstrate, through analysis and simulations, that partial decoupling gives rise to complex phenomena such as asymmetric layer collapse and non-monotonic robustness patterns.
- We explore free-space allocation (FSA) strategies under both local and global redistribution rules and examine how they affect robustness across layers, revealing trade-offs between overall system performance and the robustness of individual layers.
- Motivated by the effectiveness of *layer-weighted equal FSA* under global redistribution, we propose *local-risk-weighted FSA* (LR-FSA), which allocates free space using first-degree neighbor load and node degree information (and cross-layer influence). Across all tested configurations and network topologies (Erdős–Rényi and scale-free), LR-FSA yields the highest robustness in our experiments.
- We discuss applications to multi-functional infrastructures such as cloud computing systems, supply chain networks, and 5G network-slicing architectures, where shared resources and partial independence coexist.

The remainder of this paper is organized as follows. Section II presents the multiplex flow network model and defines the proposed overload condition. Section III derives the analytical framework and recursive relations governing the cascade dynamics. Section IV provides numerical results and robustness evaluations, and Section V concludes the paper with key insights and future research directions.

II. System Model

A. Multiplex Flow Network Model

Consider a network where a set of nodes $\mathcal{N} = \{1, 2, \ldots, N\}$ is responsible for transporting (or supporting) $M \geq 2$ distinct types of flows (or tasks/functionalities), each supported by a different graph structure. This structure is represented as a multiplex network with M layers, where each layer corresponds to a specific flow type. We focus on the two-layer case, denoted as $\mathcal{M} = \{A, B\}$, demonstrated in Figure 1.

Each node v_x carries a flow (or load) represented by the vector $\mathbf{L}_x = [L_{x,A}, L_{x,B}]$. We define the overload condition by partitioning the load space into four regions corresponding to the possible outcomes for a node: surviving in both layers, surviving only in layer A, surviving only in layer B, or failing in both layers. Figure 2 represents such a partition for a single node, distinguishing the values of $[L_{x,A}, L_{x,B}]$ that would result in failure in one or both layers. The capacity vector $\mathbf{C}_x = [C_{x,A}, C_{x,B}]$ specifies the maximum load the node can sustain in each layer when the load from the other layer is zero. Failures occur according to the overload condition, and the loads of failed nodes are subsequently redistributed to surviving nodes, possibly causing additional failures. Since the model assumes that each functionality or flow is distinct, there is no transfer of load between layers; instead, the load redistribution is confined within each layer. However,

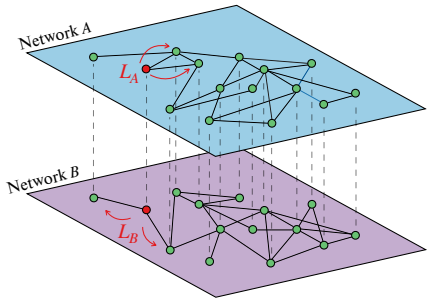


Fig. 1: Multiplex Flow Network Model proposed in [31]. Each layer is defined on the same set of vertices but with (potentially) different edge sets, and each layer is responsible for carrying/supplying a different flow type. The highlighted node v_x carries load $L_{x,A}$ in layer A and $L_{x,B}$ in layer B. If v_x fails in layer A, its type-A load is redistributed to functional nodes in layer A, but the node can still function in layer B if it does not satisfy the failure condition in that layer.

the interdependence between layers is governed by specific overload conditions.

Previously, in [31], we introduced the *layer-influenced overload* condition (Figure 2(a)), where a node's failure was influenced by the load in both layers. In this model, a node survived as long as:

$$L_{x,A} + \beta_B L_{x,B} \leq C_{x,A} \text{ and } L_{x,B} + \beta_A L_{x,A} \leq C_{x,B}. \tag{1}$$

Here, β_A and β_B are the *cross-layer influence factors*, representing the unit impact of the load in one layer on another, for layers A and B, respectively. Thus, the load in layer A contributes partially to the effective load in layer B, and vice versa. If either condition is violated, the node fails, and the loads on each layer are redistributed within their respective layers. This failure condition couples the functionalities of both layers at the node level. In other words, the failure in one layer implies the failure in all other layers.

In this paper, we consider a novel overload condition, the *layer-influenced overload*, where the load from one layer still contributes to the overload in the other, but failure in one functionality does not necessarily cause failure in the other, hence the functionalities are decoupled. More specifically, while the *cross-layer influence factors* (β_A, β_B) remain, failures now occur independently in each layer. As a result, the condition in (1) is decoupled into two separate conditions that independently determine survival in each layer:

$$L_{x,A} + \beta_B L_{x,B} \leq C_{x,A}, \tag{2}$$

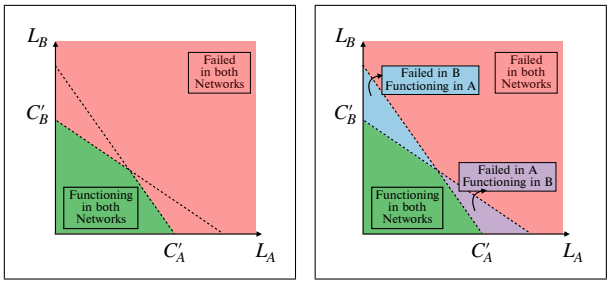
$$L_{x,B} + \beta_A L_{x,A} \leq C_{x,B} \tag{3}$$

Here, the first inequality determines survival in layer A, while the second determines survival in layer B. For example, if a node v_x satisfies the first but not the second, it fails in layer B (so $L_{x,B}$ is set to zero and redistributed among the surviving nodes in layer B), but it continues to function in layer A. This decoupling localizes failures within individual layers and increases robustness, as load from the failed layer no longer contributes to overload in the other. We refer to this condition as the "layer-influenced decoupled overload" (Figure 2(b)).

As an example, consider a supply chain network in which production and distribution facilities act as nodes responsible for multiple types of products, such as product-A and product-B. The production and distribution processes for each product type follow distinct topologies, which can be represented as separate layers in a multiplex network. The ability of each facility to meet production or distribution requirements can be modeled using a load-capacity relationship. Due to shared resources, such as raw materials and labor, the ability to meet the demand for product-A is influenced by the current load associated with product-B, and vice versa. If the load exceeds a certain threshold, a facility may no longer be able to meet the demand for product-A, resulting in unmet demand being redirected to nearby facilities, potentially causing backlogs and lost demand. However, depending on the condition, the facility may still be able to meet the demand for product-B, or in some cases, reallocating all operations to product-B may improve efficiency by dedicating all resources to a single product. Another relevant example is a distributed computing system, where servers are responsible for executing multiple types of tasks, such as CPU-intensive and I/O-intensive operations, each represented as a distinct functionality in separate layers. Although these tasks do not directly interfere with each other, they compete for shared system resources such as memory, CPU cycles, or bandwidth. Overload in one type of task may reduce the available capacity for another, but a server can still continue to function for one task type even if it fails to support the other, depending on resource availability and task-specific thresholds.

B. Redistribution Rule and Robustness Metrics

We adopt the *global redistribution rule*, where the load of a failed node is evenly distributed among all surviving nodes. Specifically, if a node v_x fails in a specific layer, each type-i flow, $L_{x,i}$, is redistributed within its respective layer-i. This assumption is widely used in overload-based cascading failure models because it captures *long-range redistribution effects*, where the failure of a single node can influence the entire system through global rebalancing of workload [12], [16], [21]. Such global effects are characteristic of systems in which tasks or demands can be reassigned without strong spatial or topological constraints. For example, during the 2025 AWS



(a) Layer-influenced Overload (as in [31])

(b) Layer-influenced Decoupled Overload

Fig. 2: Failure/overload conditions defined as a partitioning of the (L_A, L_B) plane into regions determining a node's current state on each flow as functioning or failed.

[1] outage, the failure of a small set of components triggered system-wide rerouting of requests—placing additional stress on distant resources and leading to further disruptions. This illustrates how failures in large-scale flow systems often propagate through global workload shifts rather than only through local interactions. In addition, the global rule yields a tractable analytical formulation: all surviving nodes in a layer receive the same additional load, allowing the cascade to be described through recursions for the excess loads and surviving fractions. This preserves the essential behavior of intermittent failures while keeping the model analytically tractable under general load-capacity distributions.

While the global rule is useful for analysis, it does not incorporate network structure, since redistributed load is shared regardless of connectivity. To represent settings where redistribution is constrained by topology, we also consider a *local redistribution rule*, where the load of a failed node in a layer is redistributed only among its surviving neighbors in that layer. Our analytical results focus on the global redistribution rule, but we complement them with simulations under local redistribution in Section IV-C and use these experiments to evaluate robustness and free-space allocation strategies when load redistribution is localized.

To analyze the system's robustness, we examine cascading failures initiated by *random attacks*. An initial attack removes a fraction $p \in (0,1)$ of nodes, i.e., failing them in both layers, which triggers load redistribution and potential secondary failures. In the steady state, let $\mathcal{N}_A(p)$ and $\mathcal{N}_B(p)$ denote the sets of nodes that survive in layers A and B, respectively. We further define $\mathcal{N}_{AB}(p) := \mathcal{N}_A(p) \cap \mathcal{N}_B(p)$ as the set of nodes that survive in both layers. (When needed, $\mathcal{N}_{A\setminus B}(p) := \mathcal{N}_A(p)\setminus\mathcal{N}_B(p)$ and $\mathcal{N}_{B\setminus A}(p) := \mathcal{N}_B(p)\setminus\mathcal{N}_A(p)$.) We measure system robustness using the steady-state fraction of surviving nodes in both layers ($n_{AB,\infty}(p)$), in layer A ($n_{A,\infty}(p)$), and in layer B ($n_{B,\infty}(p)$):

$$n_{i,\infty}(p) := \lim_{N\to\infty} \frac{E\left[|\mathcal{N}_i(p)|\right]}{N} \quad \text{for } i \in \{A, B, AB\} \tag{4}$$

Throughout the paper, we analyze the final system size $n_{i,\infty}(p)$ for $i \in \{A, B, AB\}$ under any attack size $0 < p < 1$. Another key objective is to determine the *critical attack size*, denoted p^*, beyond which the system fails completely. However, since failures in different layers are decoupled, the critical attack size may differ between layers. Therefore, we define the critical attack size separately for each layer as:

$$p_A^* := \sup\{p : n_{A,\infty}(p) > 0\}, \tag{5}$$

$$p_B^* := \sup\{p : n_{B,\infty}(p) > 0\}, \tag{6}$$

$$p_{AB}^* := \sup\{p : n_{AB,\infty}(p) > 0\}. \tag{7}$$

Here, p_A^* and p_B^* denote the attack sizes beyond which all nodes in layer A and layer B fail, respectively. In comparison, p_{AB}^* represents the attack size above which no node remains functional in *both* layers at the same time. Therefore, it follows that $p_{AB}^* \le p_A^*$ and $p_{AB}^* \le p_B^*$.

C. Initialization and Cascade Dynamics

Figure 3 summarizes the overall cascade progression. We model the cascade in discrete iterations $t = 0, 1, 2, \ldots$, where

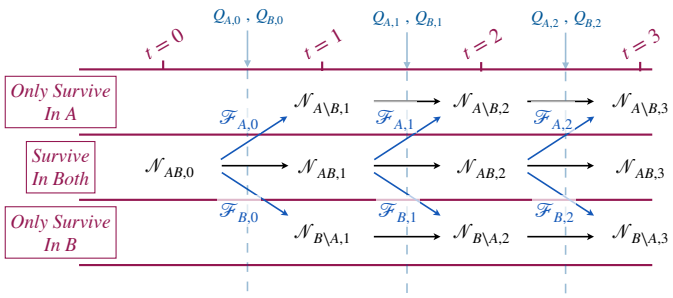


Fig. 3: Conceptual diagram of the cascade progression. Here $t = 0, 1, 2, \cdots$ indicates the rounds of flow redistribution. $\mathcal{N}_{AB,t}$ denotes the set of nodes surviving in both layers at iteration t. For $i \in \{A, B\}$, $\mathcal{N}_{i,t}$ denotes the set of nodes surviving in layer i at iteration t, and $\mathcal{N}_{i\setminus j,t} := \mathcal{N}_{i,t}\setminus\mathcal{N}_{j,t}$ denotes the set of nodes surviving only in layer i (with $j \neq i$). Moreover, $\mathcal{F}_{i,t}$ is the set of nodes that were surviving in both layers at iteration t but will survive only in layer i at iteration $t{+}1$. $Q_{A,t}, Q_{B,t}$ denote the excess load per surviving node resulting from the failures up to iteration t.

each iteration represents a round of flow redistribution. At $t = 0$, an initial random attack removes a fraction p of nodes uniformly at random, and their loads are redistributed within the corresponding layer. Subsequently, at each iteration, nodes that violate the overload conditions are removed simultaneously, their failed-layer loads are redistributed within that layer, and the process repeats until no further overload-induced failures occur.

We track the cascade through the sets $\mathcal{N}_{A,t}$ and $\mathcal{N}_{B,t}$ of nodes operating in layers A and B at iteration t, and their intersection $\mathcal{N}_{AB,t} := \mathcal{N}_{A,t} \cap \mathcal{N}_{B,t}$. The sets of nodes operating only in one layer are $\mathcal{N}_{A\setminus B,t} := \mathcal{N}_{A,t} \setminus \mathcal{N}_{B,t}$ and $\mathcal{N}_{B\setminus A,t} := \mathcal{N}_{B,t} \setminus \mathcal{N}_{A,t}$. Nodes that lose exactly one functionality transition from $\mathcal{N}_{AB,t}$ to the appropriate single-layer set, while nodes that lose both functionalities exit the system. For bookkeeping, let $\mathcal{F}_{i,t}$ denote the nodes that transition at iteration t from both-layer survival to only layer $i \in \{A, B\}$ (i.e., $\mathcal{N}_{AB,t} \to \mathcal{N}_{i\setminus j,t+1}$ with $j \neq i$). Moreover, $\mathcal{N}_{A,t}$, $\mathcal{N}_{B,t}$, and $\mathcal{N}_{AB,t}$ are monotone non-increasing in t and the process terminates when no additional overload-induced removals occur and the sets $\mathcal{N}_{AB,t}$, $\mathcal{N}_{A,t}$, and $\mathcal{N}_{B,t}$ stabilize. In the recursive equations, we track the fractional sizes of these sets:

$$n_{A,t} = \frac{|\mathcal{N}_{A,t}|}{N}, \qquad n_{B,t} = \frac{|\mathcal{N}_{B,t}|}{N}, \qquad n_{AB,t} = \frac{|\mathcal{N}_{AB,t}|}{N}. \tag{8}$$

Following the set definitions we can also write $n_{A,t} = n_{A\setminus B,t} + n_{AB,t}$ and $n_{B,t} = n_{B\setminus A,t} + n_{AB,t}$.

Each node v_x operates in layers A and B and carries an initial load vector $[L_{x,A}^{(0)}, L_{x,B}^{(0)}]$. Under the *global redistribution rule*, all surviving nodes in the same layer receive the same additional burden at iteration t. Denoting the *excess load per surviving node* by $Q_{t,A}$ and $Q_{t,B}$ in layers A and B, respectively, the instantaneous loads of a surviving node at iteration t are

$$\begin{aligned} L_{x,A}^{(t)} &= L_{x,A}^{(0)} + Q_{t,A}, \\ L_{x,B}^{(t)} &= L_{x,B}^{(0)} + Q_{t,B}, \end{aligned} \tag{9}$$

and $L_{x,i}^{(t)} = 0$ once the node has failed in layer i. Given the instantaneous loads, the overload conditions for layers A and B are

$$L_{x,A}^{(t)} + \beta_B L_{x,B}^{(t)} > C_{x,A}, \tag{10}$$

$$L_{x,B}^{(t)} + \beta_A L_{x,A}^{(t)} > C_{x,B}. \tag{11}$$

Thus, nodes satisfying (10) fail in layer A, and nodes satisfying (11) fail in layer B.

Previously in our work [31], to avoid trivial failures during initialization, we adopted a flexible formulation. Similarly here, rather than specifying capacities directly, for each node v_x we define two nonnegative quantities per layer: its baseline (initial) load and its free space, denoted $L_{x,A}^{(0)}, L_{x,B}^{(0)}$ and $S_{x,A}, S_{x,B}$ for layers A and B, respectively. The free space $S_{x,i}$ represents the additional load that v_x can tolerate in layer i beyond its baseline load plus the cross-layer contribution. Capacities are then

$$\begin{aligned} C_{x,A} &= L_{x,A}^{(0)} + \beta_B L_{x,B}^{(0)} + S_{x,A}, \\ C_{x,B} &= L_{x,B}^{(0)} + \beta_A L_{x,A}^{(0)} + S_{x,B}. \end{aligned} \tag{12}$$

We assume that $(L_{x,A}^{(0)}, S_{x,A}, L_{x,B}^{(0)}, S_{x,B})$ are i.i.d. across nodes with joint CDF $P_{L_A S_A L_B S_B}(y_{L_A}, y_{S_A}, y_{L_B}, y_{S_B}) = P[L_A \le y_{L_A},\, S_A \le y_{S_A},\, L_B \le y_{L_B},\, S_B \le y_{S_B}]$ and associated joint PDF $p_{L_A S_A L_B S_B}$. We further assume positive support, i.e., $L_{x,A}^{(0)} > 0$, $S_{x,A} > 0$, $L_{x,B}^{(0)} > 0$, and $S_{x,B} > 0$ for all v_x, which ensures all nodes are functioning prior to the attack while allowing arbitrary nonnegative joint distributions to capture load–capacity relationships.

With these definitions, we revisit the overload conditions (10)–(11). If a node v_x is functioning in both layers, i.e., $v_x \in \mathcal{N}_{AB,t}$, both conditions must be checked to determine its state in the next iteration. Substituting (9) and (12) into (10)–(11) cancels the baseline terms and yields the free-space comparisons:

$$S_{x,A} > Q_{t,A} + \beta_B Q_{t,B}, \tag{13}$$

$$S_{x,B} > Q_{t,B} + \beta_A Q_{t,A}. \tag{14}$$

If both are satisfied, the node continues to function in both layers; otherwise, the load in the failed layer(s) is set to zero and redistributed in the respective layer(s). This motivates the notation of *effective excess loads*

$$\begin{aligned} Q'_{t,A} &= Q_{t,A} + \beta_B Q_{t,B}, \\ Q'_{t,B} &= Q_{t,B} + \beta_A Q_{t,A}, \end{aligned} \tag{15}$$

under which the survival checks for a node that is alive in both layers reduce to

$$S_{x,A} > Q'_{t,A} \quad \text{(survive in layer } A \text{ at } t), \tag{16}$$

$$S_{x,B} > Q'_{t,B} \quad \text{(survive in layer } B \text{ at } t). \tag{17}$$

If a node v_x is functioning in only one layer (i.e., $v_x \in \mathcal{N}_{A\setminus B,t}$ or $v_x \in \mathcal{N}_{B\setminus A,t}$), a single condition determines its survival in the respective layer. Consider $v_x \in \mathcal{N}_{A\setminus B,t}$, so $L_{x,B}^{(t)} = 0$. Then substituting (9) and (12) into (10) yields

$$S_{x,A} > Q_{t,A} - \beta_B L_{x,B}^{(0)}, \tag{18}$$

with the symmetric expression for survival in B once A has failed.

In the following section, we present our results for calculating the mean fraction of nodes at each iteration until the system reaches a stable state.

III. Analytical Results

In this section, we present the analytical characterization of the cascading process under the proposed overload condition. We derive recursive equations that describe how the excess loads and the fractions of surviving nodes in each layer change following an initial attack.

Theorem 3.1: Consider the multiplex flow network model for layer-influenced decoupled overload condition as described in Section II. Assume that the initial load-free space values are drawn independently for each node from the joint probability distribution function $p_{L_A S_A L_B S_B}$. Let the cross-layer influence factors be denoted by β_A and β_B. With $n_{AB,0} = n_{A,0} = n_{B,0} = 1 - p$, $Q_{A,-1} = Q_{B,-1} = 0$, $Q_{A,0} = \frac{p\mathbb{E}[L_A]}{(1-p)}$, and $Q_{B,0} = \frac{p\mathbb{E}[L_B]}{(1-p)}$, the fraction of surviving nodes at each iteration, initiated by a random attack removing a p-fraction of the nodes, can be calculated as follows for each iteration $t = 1, 2, \ldots$

$$n_{AB,t} = n_{AB,0}.\mathbb{P}\left[S_A > Q'_{A,t-1}, S_B > Q'_{B,t-1}\right] \tag{19}$$

$$n_{A\setminus B,t} = n_{AB,0} \sum_{j=0}^{t-1} \mathbb{P}\left[\; S_A > Q_{A,t-1} - \beta_B L_B,\; S_A > Q'_{A,j},\; Q'_{B,j-1} < S_B < Q'_{B,j} \;\right] \tag{20}$$

$$n_{B\setminus A,t} = n_{AB,0} \sum_{j=0}^{t-1} \mathbb{P}\left[\; S_B > Q_{B,t-1} - \beta_A L_A,\; S_B > Q'_{B,j},\; Q'_{A,j-1} < S_A < Q'_{A,j} \;\right] \tag{21}$$

$$n_{A,t} = n_{AB,t} + n_{A\setminus B,t} \tag{22}$$

$$n_{B,t} = n_{AB,t} + n_{B\setminus A,t} \tag{23}$$

$$Q_{A,t} = \frac{p\mathbb{E}[L_A] + n_{AB,0}\left(\sum_{j=0}^{t-1} \mathbb{E}\left[L_A \mathbb{1}\left[\begin{smallmatrix} Q'_{A,j-1} < S_A < Q'_{A,j} \\ Q'_{B,j-1} < S_B \end{smallmatrix}\right]\right] + \sum_{j=0}^{t-2} \mathbb{E}\left[L_A \mathbb{1}\left[\begin{smallmatrix} Q'_{A,j} < S_A < Q_{A,t-1} - \beta_B L_B \\ Q'_{B,j-1} < S_B < Q'_{B,j} \end{smallmatrix}\right]\right]\right)}{n_{A,t}} \tag{24}$$

$$Q_{B,t} = \frac{p\mathbb{E}[L_B] + n_{AB,0}\left(\sum_{j=0}^{t-1} \mathbb{E}\left[L_B \mathbb{1}\left[\begin{smallmatrix} Q'_{B,j-1} < S_B < Q'_{B,j} \\ Q'_{A,j-1} < S_A \end{smallmatrix}\right]\right] + \sum_{j=0}^{t-2} \mathbb{E}\left[L_B \mathbb{1}\left[\begin{smallmatrix} Q'_{B,j} < S_B < Q_{B,t-1} - \beta_A L_A \\ Q'_{A,j-1} < S_A < Q'_{A,j} \end{smallmatrix}\right]\right]\right)}{n_{B,t}} \tag{25}$$

$n_{AB,t}$	The fractional size of nodes functioning in **both** layers at iteration t
$n_{A,t}, n_{B,t}$	The fractional size of nodes functioning in layer-A and layer-B at iteration t, respectively
$n_{A\setminus B,t}$, $n_{B\setminus A,t}$	The fractional sizes of nodes functioning **only** in A and **only** in B at iteration t, respectively
$Q_{A,t}$, $Q_{B,t}$	The excess load of type-A and type-B per surviving node at (the end of) iteration, respectively t
$Q'_{A,t}$, $Q'_{B,t}$	The **effective** excess load of type-A and type-B, respectively as defined in (15)
β_i	The *cross-layer influence factor* representing the unit impact of the load in layer i ($i \in \{A, B\}$) to the other

TABLE I: Notation used for analysis

Table I summarizes the notation used in the recursive calculations, i.e., (19)–(25) [1]. At any iteration, if $n_{A,t} = 0$ (or $n_{B,t} = 0$), we set $Q_{A,t} = \infty$ (or $Q_{B,t} = \infty$), indicating a complete collapse of layer A (or layer B). We provide an overview of the proof below, while the detailed proof is presented in Appendix A.

In the proposed model, a key distinction is that a node can lose functionality in one layer while remaining operational in the other. As a result, the two layers do not necessarily fail simultaneously once the cascade begins: failures in one layer affect the available free space in the other, but the layers can progress through the cascade in different ways and at different times. This requires the analysis to track transitions among four functional states: dual operation ($\mathcal{N}_{AB}$), single-layer operation ($\mathcal{N}_{A\setminus B}$ or $\mathcal{N}_{B\setminus A}$), and complete failure. To derive the recursive equations, we first characterize the fraction of nodes in each of these states, as illustrated in Figure 3, as well as the resulting excess loads per surviving nodes after the initial attack. We first describe the intuition behind each component that enters the analysis and clarify how the different terms arise from the cascade process.

We begin with the effect of the initial attack. An initial attack of size p at iteration $t = 0$ results in a system size of $n_{AB,0} = 1-p$. To compute the mean excess load per surviving node in each layer, we consider the total load redistributed from failed nodes and divide it by the fraction of surviving nodes. Since the initial attack is random, the mean excess loads per surviving node, $Q_{0,A}$ and $Q_{0,B}$, are given by $\frac{\mathbb{E}[L_A]p}{1-p}$ and $\frac{\mathbb{E}[L_B]p}{1-p}$, respectively. Starting from iteration $t = 1$, we recursively compute the mean fraction of surviving nodes in each state ($n_{A,t}$,$n_{B,t}$, $n_{AB,t}$) and the corresponding excess loads per surviving node ($Q_{t,A}$, $Q_{t,B}$) for $t = 2, 3, \ldots$, thereby obtaining expressions for them in terms of their values at previous iterations.

We next describe the update of $n_{AB,t}$, the fraction of nodes that continue to function in both layers given in (19). At iteration t, a node in $\mathcal{N}_{AB,t}$ proceeds to $\mathcal{N}_{AB,t+1}$ only if it satisfies the survival conditions in both layers, namely, its free spaces exceed the corresponding effective excess loads. In other words, among the nodes in $\mathcal{N}_{AB,t}$, only those satisfying (16) and (17) will proceed to $\mathcal{N}_{AB,t+1}$. Since these nodes are already functioning in both layers at iteration t, they should already satisfy $S_A > Q'_{A,t-1}$ and $S_B > Q'_{B,t-1}$ thus the survival probability in the next iteration should be conditioned on this event. Therefore, the relationship between $n_{AB,t+1}$ and $n_{AB,t}$ can be written as:

$$n_{AB,t+1} = n_{AB,t} \cdot \mathbb{P}\left(\begin{array}{c} S_A > Q'_{A,t} \\ S_B > Q'_{B,t} \end{array} \middle| \begin{array}{c} S_A > Q'_{A,t-1} \\ S_B > Q'_{B,t-1} \end{array} \right). \quad (26)$$

Repeatedly, applying (26) iterating from 1 through t yields (19).

The characterization of the nodes surviving in a single layer, $n_{A\setminus B,t}$ and $n_{B\setminus A,t}$ (equations (20),(21)), requires additional effort because the failure of a node in one layer increases its available free space in the other, as described in (18). To capture this effect, the recursion for $n_{A\setminus B,t}$ accounts for the specific iteration at which a node transitions from operating in both layers to operating only in layer A. In (20) and (21), this transition time is indexed by j. For example, in (20), j denotes the iteration in which the node fails in layer B while still surviving in layer A. Hence, the probability inside the summation in (20) reflects two events: (i) the node fails in layer B exactly at iteration j, and (ii) it still survives in layer A at iteration t. Since this transition can occur at any cascade step prior to t, the summation ranges over all possible values of j.

Failure in layer B at iteration j implies that the node satisfied $S_A > Q'_{A,j}$ but not $S_B > Q'_{B,j}$, specifically falling into the interval $Q'_{B,j-1} < S_B < Q'_{B,j}$. After this failure, to survive in layer A until iteration t, the node must satisfy $S_A > Q_{A,t-1} - \beta_B L_B$, consistent with the condition in (18). Summing over all j accounts for every possible iteration when a node may have first failed in layer B prior to iteration t. Since these nodes originate from those functioning in both layers at the start, we multiply the sum by $n_{AB,0}$. The derivation for $n_{B\setminus A,t}$ follows symmetrically by interchanging layers A and B, and the same reasoning applies. After finding the fractional size of the nodes surviving only in a single layer ($n_{A\setminus B,t}$ and $n_{B\setminus A,t}$), (22) and (23) are used to compute the surviving fraction of nodes in individual layers A and B.

Finally, we describe the update of the excess loads $Q_{A,t}$ and $Q_{B,t}$ given in (24)-(25). We focus on the equation for $Q_{A,t}$ to explain the key components in the formulation as the expressions for $Q_{B,t}$ is symmetrical. Equation (24) defines the excess load of type-A per surviving node at iteration t. This quantity captures the total redistributed type-A load divided evenly among the surviving nodes in layer-A. The numerator represents the total excess load to be redistributed, normalized by the total system size N. This total consists of two parts: the excess load released due to the initial attack ($p\mathbb{E}[L_A]$), and the load released by subsequent cascading failures that occur up to iteration t (the remaining sum in the numerator). While the numerator is normalized by the total system size N, $Q_{A,t}$ is defined as the excess load of type-A per node *surviving* in layer-A. To ensure this, the numerator is divided by the relative size of the *surviving* nodes in layer-A, given by $n_{A,t}$. This normalization ensures that the computed value reflects the actual redistribution burden on each *surviving* node in layer-A.

[1]Probabilistic statements are defined with respect to the probability measure $\mathbb{P}$, and the associated expectation operator is denoted by $\mathbb{E}$. The indicator function of an event E is written as $\mathbb{1}[E]$.

The two summation terms in the numerator of (24) quantify the excess load generated by cascading failures in layer-A over time. These failures stem from two distinct sources: (i) nodes that were initially functioning in both layers and subsequently failed in A, and (ii) nodes that were initially functioning only in A and later failed. The first summation corresponds to failures of nodes that survive in both layers at iteration $j-1$, i.e., $S_A > Q'_{A,j-1}$ and $S_B > Q'_{B,j-1}$, but then fail in layer-A at iteration j, i.e., $S_A < Q'_{A,j}$. Summing over $j = 0$ to $t-1$ gives the total expected load released by these failures. The second summation captures failures of nodes that were functioning only in layer-A. These nodes survive in layer-A at iteration j, i.e., $S_A > Q'_{A,j}$, but fail in layer B exactly at that iteration, i.e., $Q'_{B,j} > S_B > Q'_{B,j-1}$. To fail in layer-$A$ by iteration t, the excess load in A should exceed their free space in A, i.e., $S_A < Q_{A,t-1} - \beta_B L_B$ as presented in (18). Again, summing over $j = 0$ to $t-1$ yields the cumulative contribution of this second group to the total excess load.

Overall, equations (19)–(25) enable the calculation of the mean fraction of nodes that survive in layer A, in layer B, or in both layers, given the initial load and free-space distribution ($p_{L_A S_A L_B S_B}$), the cross-layer influence factors (β_A, β_B), and the initial attack size (p). We begin the calculation at $t = 1$ and proceed iteratively until the system reaches a *steady state*, that is, when $n_{A,t+1} = n_{A,t}$, $n_{B,t+1} = n_{B,t}$, and $n_{AB,t+1} = n_{AB,t}$, indicating that no further failures occur. The values obtained at this point represent the final system sizes in each state.

IV. Numerical Results

In this section, we present numerical results that validate the analytical solutions introduced in the previous sections and explore free-space allocation strategies for improving system robustness. In Section IV-A, we verify the recursive equations that describe the cascade evolution and evaluate their accuracy across a range of initial conditions. We also study how the cross-layer influence factors (β_A, β_B) affect the outcome, and we discuss the non-monotone behavior of the final system size as the initial attack size increases. In Section IV-B, we assess several free-space allocation strategies from the literature and compare their ability to improve robustness. Finally, in Section IV-C, we analyze robustness under the *local redistribution* rule, where the load of failed nodes is redistributed only to their immediate neighbors, and we consider distinct network topologies in each layer. We then propose a free-space allocation strategy that uses local information (first-degree neighbor load and node degree) and evaluate its performance against the other strategies.

To perform numerical simulations, we use different initial load-free space distributions from the Uniform, Pareto, and Weibull families[2]. These distributions offer a representative range of load and free-space characteristics commonly observed in real-world systems. The uniform distribution serves as a baseline, modeling scenarios where values are evenly distributed over a fixed interval. The Pareto distribution captures heavy-tailed behavior, where a small fraction of nodes may have disproportionately large loads or capacities. The Weibull distribution provides flexibility to model various statistical profiles, including the Exponential and Rayleigh distributions, and is widely used to describe workloads in engineered systems such as distributed computing platforms [33]. Together, these distribution families support a comprehensive and realistic evaluation of system robustness under diverse operating conditions.

[2]The uniform distribution is $U(U_{\min}, U_{\max})$, where $U_{\min}$ and $U_{\max}$ are the bounds. The Weibull distribution is $Wei(W_{\min}, \lambda, k)$, with $W_{\min}$ as the minimum, λ as the scale, and k as the shape parameter. The Pareto distribution is $Par(P_{\min}, b)$, where $P_{\min}$ is the minimum and b is the shape parameter.

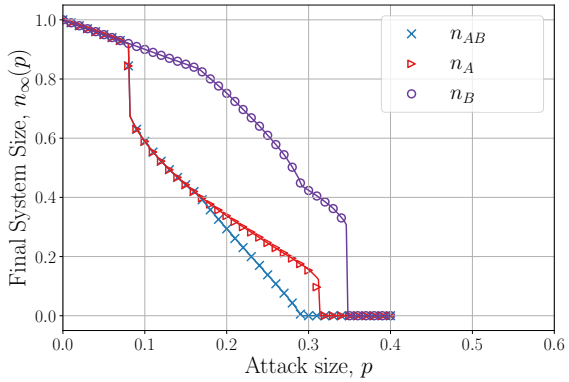


Fig. 4: Final system size for nodes surviving in A (red triangle), B (purple circle), and both layers (blue cross), for initial load-free space distributions: $L_A \sim Wei(10, 100, 0.4)$, $S_A \sim 3L_A$, $L_B \sim Wei(10, 10.78, 6)$, $S_B \sim U(20, 100)$ with $\beta_A = \beta_B = 0.1$. Solid lines represent analytical predictions obtained from Theorem 3.1, and markers indicate averages over 50 independent simulation runs with $N = 10^6$.

In Sections IV-A and IV-B, all simulations are conducted under the *global redistribution* rule with $N = 10^6$. In the figures, markers (triangle, cross, circle) report averages over 50 simulation runs, while the solid lines show the corresponding analytical results calculated by equations (19)–(25). In Section IV-C, we study the *local redistribution* rule using networks of size $N = 10^5$ and 100 replications. Since analytical results are not available in the *local redistribution* setting, both the markers and solid lines represent averages over the 100 replications.

A. Numerical Validation of Recursive Equations

In this subsection, we use simulations to validate the recursive equations (19)–(25) describing the cascade evolution and to quantify their accuracy across representative initial conditions. Figure 4 shows the final system size measured by the fractions of nodes surviving in both layers (n_{AB}), in layer A (n_A), and in layer B (n_B) as a function of the initial attack fraction p. In this setting, the initial loads follow $L_A \sim \text{Wei}(10, 100, 0.4)$ and $L_B \sim \text{Wei}(10, 10.78, 6)$, while the free-space (excess capacity) variables are $S_A \sim 3L_A$ (proportional to each node's A-layer load) and $S_B \sim U(20, 100)$, with cross-layer influence factors $\beta_A = \beta_B = 0.1$. For the analytical curves, we compute the probabilities required by

Theorem 3.1 directly from these distributions, whereas for the simulations we draw node-level loads and free spaces from the same distributions and report Monte Carlo averages over 50 independent runs with $N = 10^6$. We also experimented with a broad set of initial configurations by combining Uniform, Weibull, and Pareto families for (L_A, S_A, L_B, S_B), and these settings lead to different failure patterns as p increases including sudden collapses, gradual degradation, or one type followed by the other. Across all cases, the analytical calculations from Theorem 3.1 align almost perfectly with the simulation averages. In Fig. 4, we focus on one representative setting for a more detailed discussion.

In Figure 4, we observe that as the attack size increases, nodes initially functioning in both layers undergo an abrupt transition around $p \approx 0.09$, failing in layer A while continuing to operate in layer B. This is reflected in the survival curve for layer B, which closely follows the linear decay of $1 - p$, indicating that these nodes are unaffected in layer B despite losing functionality in layer A. As p increases beyond approximately 0.18, we begin to observe a reverse transition, where some nodes previously functioning in both layers fail in B and continue operating only in A. Consequently, in the range $p \in [0.18, 0.3]$, the system stabilizes with distinct subsets of nodes functioning exclusively in A, exclusively in B, or in both layers. While the decline in n_{AB} appears nearly linear, the transitions in the layer A and layer B survival curves are abrupt, suggesting critical threshold behavior.

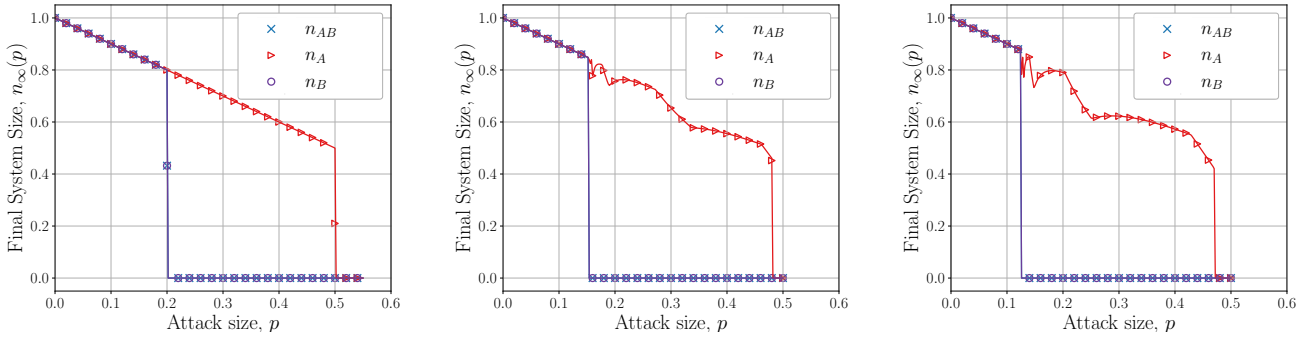
Figure 5b illustrates another novel phenomenon: oscillations in the final system size of layer A as the attack size increases. Although counter-intuitive, this non-monotone behavior is consistently reproduced in both the analytical calculations and simulation results, also reported by the papers discussing a similar node level overload condition [30]. This behavior appears to emerge after the complete collapse of layer-B, which releases some additional capacity in layer-A that was previously consumed due to cross-layer influence. A possible explanation is that at larger attack sizes, the collapse of layer-B occurs earlier in the load redistribution process, allowing the freed capacity in layer-A to be leveraged sooner and partially mitigate subsequent failures. This may result in a slightly larger final system size in layer-A, depending on the timing of collapse and the strength of cross-layer influence. We further explore this hypothesis by varying the cross-layer influence parameter β_B, as shown in Figure 5a and Figure 5c.

As expected, higher values of β_B in Figure 5 lead to lower critical attack sizes, indicating reduced robustness due to increased coupling between layers. When $\beta_B = 0$, no oscillatory behavior is observed in the final system size, confirming that these dynamics are driven by inter-layer interactions. As the cross-layer influence increases, we observe mild oscillations in the final system size of layer A, which become more pronounced with larger β_B values. This is consistent with the intuition that a greater degree of coupling allows more capacity to be released from the failed layer, temporarily boosting the robustness of the remaining one. The effect is most prominent in Figure 5c, where the amplitude of the oscillations increases with higher cross-layer influence.

B. Different Free Space Allocation Strategies

In many motivating applications, such as cloud infrastructures and supply chains, the total available free space or capacity is inherently limited due to limited resources. To avoid large scale disruptions in such systems, how this limited capacity is allocated across different components can play a critical role. Motivated by this, we consider settings where the total free space across the two layers is fixed, i.e., $\mathbb{E}[S_A] + \mathbb{E}[S_B] = S_{\text{total}}$, where S_{total} is constant. Under the *layer-influenced decoupled overload* condition studied here, characterizing the optimal allocation of this fixed free space is analytically challenging, particularly because of the existence of multiple states that requires distinct failure evaluations. Given these limitations, we focus on a numerical comparison of three practical **free-space allocation (FSA)** strategies often employed in the literature [16], [21], [31]:

i) Layer-weighted equal FSA: The total free space is divided between layer A and layer B in proportion to their expected loads adjusted by the *cross-layer influence factors*, i.e., in proportion to $\mathbb{E}[L_A] + \beta_B \mathbb{E}[L_B]$ and $\mathbb{E}[L_B] + \beta_A \mathbb{E}[L_A]$, respectively. Then, within each layer,



(a) $\beta_A = 0.25, \beta_B = 0$. (b) $\beta_A = \beta_B = 0.25$. (c) $\beta_A = 0.25, \beta_B = 0.5$.

Fig. 5: Final system size for nodes surviving in A (red triangle), B (purple circle), and both layers (blue cross), for three different cross-layer influence factors for initial distributions: $L_A \sim U(10, 50)$, $S_A \sim U(30, 60)$, $L_B \sim \text{Par}(10, 2)$, $S_B \sim 0.5 L_B$. Solid lines represent analytical predictions obtained from Theorem 3.1, and markers indicate averages over 50 independent simulation runs with $N = 10^6$.

the allocated free space is distributed evenly across all nodes.

ii) **Equal FSA:** The total free space is split equally between the two layers and distributed evenly across all nodes in each layer.

iii) **Equal tolerance factor:** Each node is assigned free space in proportion to its load, i.e., $S_{x,i} = \alpha L_{x,i}$, where α is a constant for all $x \in \mathcal{N}$.

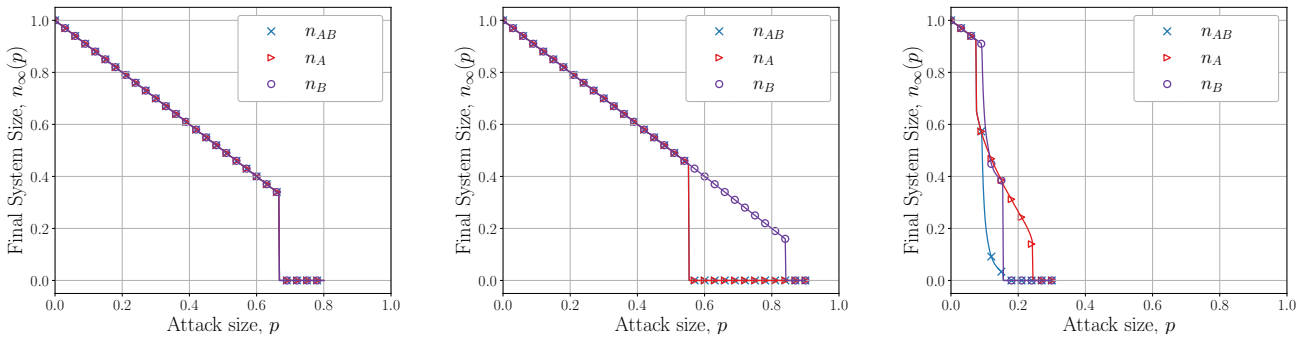
We evaluated these strategies under various initial load configurations, including Weibull–Uniform, Uniform–Pareto, and Weibull–Pareto distributions. As the observed trends were consistent across all configurations, we present results for the Weibull–Pareto case in Figure 6, which is representative of the overall behavior.

Figure 6 compares the final fractions of nodes that remain functional in layer A (red triangles), in layer B (purple circles), and in both layers (blue crosses) under three free-space allocation strategies. For both *Layer-weighted equal FSA* (Figure 6a) and *Equal FSA* (Figure 6b), the surviving fractions follow the line $1-p$ over a wide range of attack sizes and then drop suddenly to zero at a critical threshold. This behavior is consistent across different initial load distributions in our experiments hence it is not coincidental. When each node in a layer receives the same free space, the allocation effectively fixes a common excess load threshold per node. As we increase the attack size, we are effectively increasing the excess load per node in each layer for the initial round of flow redistribution. As a result, until this excess load exceeds the threshold, no secondary failures occur and once it is exceeded, all nodes fail simultaneously. The difference between the two strategies is that the *Layer-weighted equal FSA* accounts for differences between layers (through expected loads and cross-layer influence). Therefore, the collapse threshold of both layers losing functionality are aligned at the same attack size, as in Figure 6a. In contrast, *Equal FSA* ignores these differences and can lead to one layer failing earlier than the other, as in Figure 6b. As a result, if robustness of the joint functionality is the priority, the *Layer-weighted equal FSA* provides the largest critical attack size for nodes functioning in both layers, p^*_{AB}, across our experiments. If, instead, one functionality is more important, priority can be shifted toward that layer by allocating a larger share of the total free space to it (while still distributing free space equally among its nodes), which can increase the single-layer critical attack size p^*_A or p^*_B at the expense of the other layer and of p^*_{AB}. For example, relative to Figure 6a ($S_A = 584$, $S_B = 136$), the *Equal FSA* in Figure 6b ($S_A = S_B = 360$) increases p^*_B, while reducing p^*_A and p^*_{AB}. Finally, the *Equal tolerance factor* strategy in Figure 6c leads to earlier degradation and smaller critical attack sizes, and it consistently performs worse than the other two strategies across the load-distribution combinations we tested.

C. Experiments with Local Redistribution Rule

So far, our analysis and numerical validation have focused on the *global redistribution* rule, where the load of a failed node is shared evenly among all surviving nodes in the same layer. This rule captures long-range cascading effects and enables analytical tractability. In many applications, however, overload effects are more localized and are better represented by an explicit network topology that encodes which nodes can directly share load. For example, in the supply-chain setting discussed earlier, excess demand at a facility is more likely to be absorbed by nearby facilities that can serve in close proximity, which can naturally be modeled through a network of local interactions. Motivated by this, in this subsection, we evaluate the effectiveness of different free-space allocation (FSA) strategies under *local redistribution* rule.

Under the local redistribution rule, when a node fails in a given layer, its load in that layer is redistributed evenly among its surviving neighbors. If all nodes in a connected component fail (i.e., redistribution to *neighboring* nodes becomes impossible), the total load of that component is redistributed evenly among all surviving nodes. This implementation detail preserves total load across realizations and avoids artificially



(a) Layer-weighted equal FSA (b) Equal FSA (c) Equal tolerance factor

Fig. 6: Final system size for nodes surviving in A (red triangle), B (purple circle), and both layers (blue cross), for three different free-space allocation strategies for initial distributions: $L_A \sim Wei(10, 84.25, 0.4)$, $L_B \sim Par(5, 2)$, $\beta_A = \beta_B = 0.2$. For *equal free space*, each layer receives 360 free space, distributed equally among nodes. For *equal tolerance factor*, $\alpha = 2.4$ satisfies numerical constraints. For *layer-weighted, equal free space*, free space is allocated per layer proportional to expected loads, adjusted by *cross-layer influence factors* (β_A, β_B), then evenly distributed among nodes which results in $S_A = 584, S_B = 136$.

overestimating robustness due to stranded load in isolated failed components.

The previous results in Section IV-B suggest that *layer-weighted equal FSA* performs well under global redistribution because it reduces secondary failures: once the initial attack is applied, the system typically remains functional until a critical threshold is reached, after which it collapses completely. Under local redistribution, however, such secondary failures will be shaped by the network topology, since failed load is passed to immediate neighbors. This motivates an allocation rule that follows the same principle of limiting secondary failures, but now uses local information about potential exposure from neighboring nodes. Accordingly, we introduce below a fourth strategy that we refer to as *local-risk-weighted FSA* (LR-FSA).

iv) Local-risk-weighted FSA (LR-FSA): For each node i, we estimate the overload *risk*, i.e., the amount of load it would receive from its neighbors upon their failure. We approximate this by the total load it would inherit if all of its neighbors were to fail. Specifically, we define

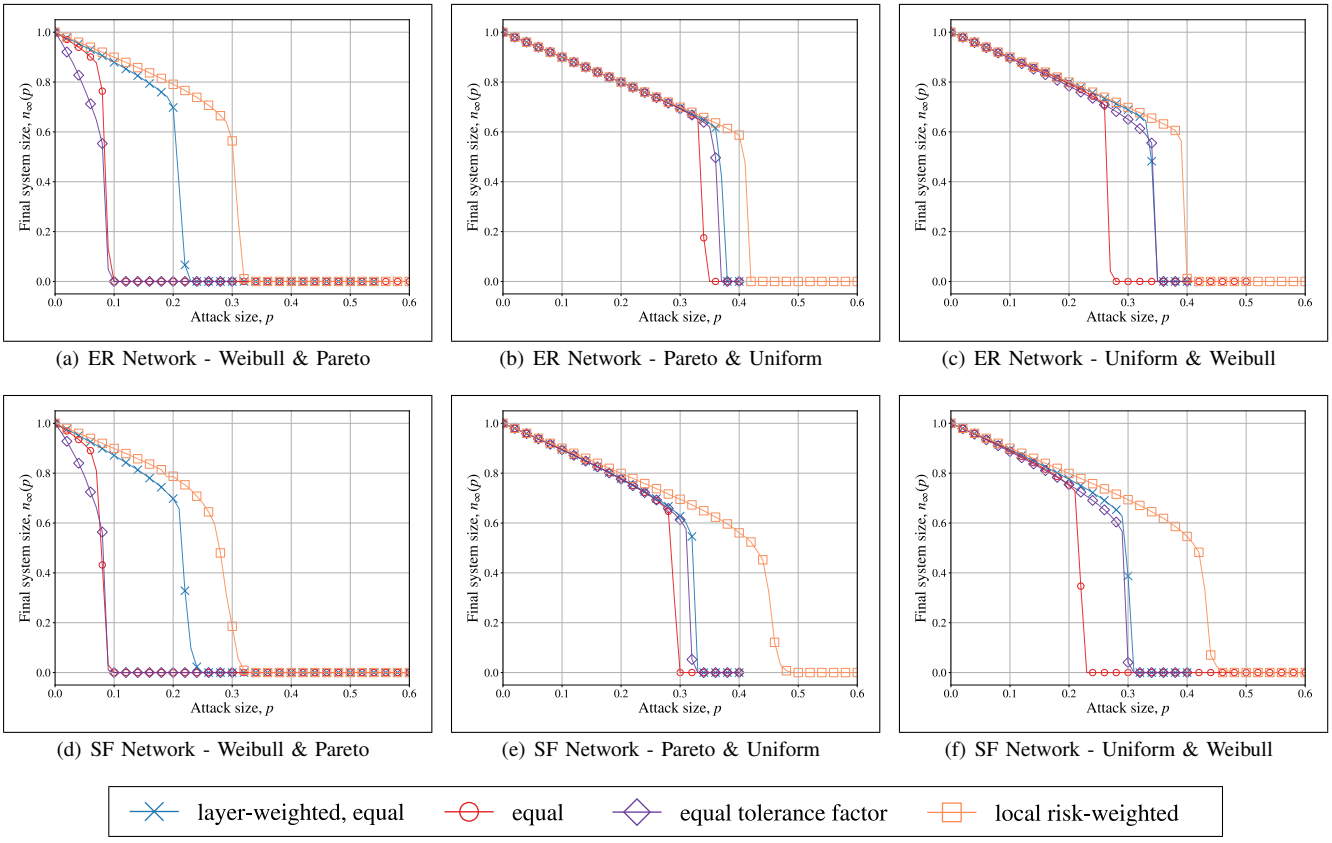
$$r_{i,A} = \sum_{v \in \mathcal{N}_A(i)} \frac{L_{v,A}}{\deg_A(v)}, \qquad r_{i,B} = \sum_{v \in \mathcal{N}_B(i)} \frac{L_{v,B}}{\deg_B(v)},$$

where $\mathcal{N}_A$ (respectively, $\mathcal{N}_B$) denotes the set of *neighbors* of node i in layer-A (respectively, in layer-B), and $\deg_A(v) = |\mathcal{N}_A|$ (respectively, $\deg_B(v) = |\mathcal{N}_B|$) denotes the *degree* (i.e., total number of neighbors) of node i in layer-A (respectively, in layer-B). Put differently, $r_{i,A}$ and $r_{i,B}$ represent the load that node i would receive under local redistribution if all of its neighbors were to fail. To incorporate inter-layer coupling, we form the effective risks

$$r_{i,A}^{\text{eff}} = r_{i,A} + \beta_B r_{i,B}, \qquad r_{i,B}^{\text{eff}} = r_{i,B} + \beta_A r_{i,A}.$$

Given a fixed total free-space budget corresponding to an average free space S_{tot} per node, we allocate free space across node-layer pairs in proportion to these effective risks. Therefore the nodes with higher load exposure or higher risk would receive higher free spaces.

We consider three initial load configurations that we also used in Section IV-B: Weibull–Pareto with $L_A \sim Wei(10, 84.25, 0.4)$ and $L_B \sim Par(5, 2)$, Pareto–Uniform with $L_A \sim Par(100, 5)$ and $L_B \sim U(150, 200)$, and Uniform–Weibull with $L_A \sim U(80, 100)$ and $L_B \sim Wei(10, 225.68, 2)$. Across all configurations, we fix $\mathbb{E}[L_A] + \mathbb{E}[L_B] = 300$, $\mathbb{E}[S_A] + \mathbb{E}[S_B] = 720$, and $\beta_A = \beta_B = 0.2$. We evaluate two network families: Erdős–Rényi (ER) networks with mean degree 5, and scale-free (SF) networks with power-law degree distribution with exponent $\gamma \approx 2.55$ [34]. In each



(a) ER Network - Weibull & Pareto (b) ER Network - Pareto & Uniform (c) ER Network - Uniform & Weibull

(d) SF Network - Weibull & Pareto (e) SF Network - Pareto & Uniform (f) SF Network - Uniform & Weibull

Fig. 7: Final system size for the nodes surviving in both layers (n_{AB}) under different free space allocation strategies with local redistribution. The same three configurations and allocation strategies from Fig. 6 are evaluated here under local redistribution rule with the addition of *local-risk-weighted free-space allocation* strategy. Subplots (a)–(c) correspond to Erdős–Rényi (ER) networks with mean degree 5, and subplots (d)–(f) to scale-free (SF) networks with power-law degree distribution ($\gamma = 2.55$). Since analytical results are not available for the local case, all points are averages over 100 simulation runs with $N = 10^5$; lines connect the averaged data for visualization.

experiment, the two layers are generated independently so they are distinct but they share the same structural parameters within each network family.

Figure 7 reports the resulting final system size under local redistribution, focusing on the fraction of nodes that remain functional in *both* layers, n_{AB}. Across all load configurations and both network families, *local-risk-weighted FSA* yields the largest critical attack size p^*_{AB} and the highest overall final system size. In several cases, the improvement in p^*_{AB} relative to the second-best strategy reaches to about 15%. Among the remaining strategies, *layer-weighted equal FSA* typically performs the second best, while its advantage relative to the *equal tolerance factor* strategy often being small. These results suggest that when local information is available and node-level capacity assignment is feasible, LR-FSA provides a clear advantage under local redistribution. When such information is unavailable, *layer-weighted equal FSA* remains a strong baseline. The observed performance gaps are generally larger in SF networks (Fig. 7d–f), which is consistent with their higher degree heterogeneity: local exposure varies substantially across nodes, and exploiting this variation improves robustness more significantly.

V. Conclusion

In this paper, we analyzed the robustness of multiplex networks against overload-based cascading failures initiated by random attacks under the failure condition, termed *layer-influenced decoupled overload*. The condition captures systems in which the overload of a node depends jointly on the workloads across all layers/functionalities, while functionalities remain only partially dependent: a node may lose functionality in one layer but continue operating in another as long as the corresponding capacity constraint is satisfied. This setting reflects multi-functional infrastructures where different services share limited resources but can still remain operationally independent after partial failures.

For the *global redistribution* rule, we derived recursive equations that characterize the cascade evolution and the final surviving fractions in each layer and in their intersection. Across a wide range of initial load–free-space distributions, the analytical predictions align almost perfectly with Monte Carlo simulations. The results show that partial decoupling leads to behaviors not present in fully dependent multiplex models, including asymmetric layer collapses and non-monotonic robustness curves as the attack size increases, driven by cross-layer influence.

We also examined how limited excess capacity should be allocated to improve robustness. Under a fixed total free-space budget, robustness can be tuned depending on system priorities: allocating free space to balance layers improves joint functionality, whereas shifting a larger share of the free-space budget toward one layer can increase its single-layer critical attack size at the expense of the other layer and of joint performance. Among the benchmark free-space allocation (FSA) strategies we tested under global redistribution, *layer-weighted equal FSA* consistently provided the highest robustness for nodes functioning in both layers, largely by limiting secondary failures until a critical threshold is reached.

Finally, we complemented the global-redistribution analysis with simulations under a *local redistribution* rule, where the load of failed nodes is redistributed only to surviving neighbors. Motivated by the effectiveness of *layer-weighted equal FSA* in reducing secondary failures under global redistribution, we proposed *local-risk-weighted FSA* (LR-FSA), which allocates free space using first-degree neighbor load and node degree information together with the cross-layer influence factors. Across all tested load configurations and for both Erdős–Rényi and scale-free networks, LR-FSA yielded the largest critical attack size for joint functionality and the highest final system size. Taken together, these results suggest that allocation rules that aim to limit secondary failures provide a useful and broadly effective design principle for improving robustness, although an analytical proof of this empirical characterization remains an open problem for future work.

Several directions follow naturally from this work. On the modeling side, it would be valuable to incorporate correlations between loads and free spaces, and dependencies across layers, which are common in practice and may affect the cascading failure dynamics. On the analytical side, extending the framework to topology-constrained local redistribution, and deriving bounds or approximations for critical attack sizes and optimal FSA strategies would be of interest.

References

[1] M. Hicks, "Explaining the aws outage & other recent incidents," Oct 2025. [Online]. Available: https://www.thousandeyes.com/blog/internet-report-aws-outage#aws-outage

[2] ENTSO-E, "Grid incident in spain and portugal on 28 april 2025 – factual report," European Network of Transmission System Operators for Electricity (ENTSO-E), Tech. Rep., May 2025, accessed: November 2025. [Online]. Available: https://www.entsoe.eu/publications/blackout/28-april-2025-iberian-blackout/#Publications_&_Documents

[3] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, Apr. 2010.

[4] Z. Zeng, N. Wang, D. Xu, and R. Chen, "Cascading failure modeling and resilience analysis of coupled centralized supply chain networks under hybrid loads," *Systems*, vol. 13, no. 9, 2025. [Online]. Available: https://www.mdpi.com/2079-8954/13/9/729

[5] Q. Yang, C. M. Scoglio, and D. M. Gruenbacher, "Robustness of supply chain networks against underload cascading failures," *Physica A: Statistical Mechanics and its Applications*, vol. 563, p. 125466, Feb. 2021.

[6] H. Wang, H. Shen, and Z. Li, "Approaches for resilience against cascading failures in cloud datacenters," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 706–717.

[7] S. V. Buldyrev, N. W. Shere, and G. A. Cwilich, "Interdependent networks with identical degrees of mutually dependent nodes," *Phys. Rev. E*, vol. 83, p. 016112, Jan 2011.

[8] M. A. Di Muro, S. V. Buldyrev, H. E. Stanley, and L. A. Braunstein, "Cascading failures in interdependent networks with finite functional components," *Phys. Rev. E*, vol. 94, no. 4, p. 042304, Oct. 2016.

[9] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Networks formed from interdependent networks," *Nature Physics*, vol. 8, no. 1, pp. 40–48, Jan. 2012.

[10] R. Parshani, S. V. Buldyrev, and S. Havlin, "Interdependent Networks: Reducing the Coupling Strength Leads to a Change from a First to Second Order Percolation Transition," *Phys. Rev. Lett.*, vol. 105, no. 4, p. 048701, 2010.

[11] F. Radicchi, "Percolation in real interdependent networks," *Nature Physics*, vol. 11, no. 7, pp. 597–602, Jul. 2015.

[12] J. V. Andersen, D. Sornette, and K.-t. Leung, "Tricritical Behavior in Rupture Induced by Disorder," *Phys. Rev. Lett.*, vol. 78, no. 11, pp. 2140–2143, 1997.

[13] B. Mirzasoleiman, M. Babaei, M. Jalili, and M. Safari, "Cascaded failures in weighted networks," *Phys. Rev. E*, vol. 84, no. 4, p. 046114, 2011.
[14] S. Pahwa, C. Scoglio, and A. Scala, "Abruptness of Cascade Failures in Power Grids," *Scientific Reports*, vol. 4, no. 1, p. 3694, 2014.
[15] W.-X. Wang and G. Chen, "Universal robustness characteristic of weighted networks against cascading failure," *Phys. Rev. E*, vol. 77, no. 2, p. 026101, 2008.
[16] Y. Zhang and O. Yağan, "Optimizing the robustness of electrical power systems against cascading failures," *Scientific Reports*, vol. 6, no. 1, p. 27625, 2016, osman Yagan.
[17] C. Chen, Y. Hu, X. Meng, and J. Yu, "Cascading failures in power grids: A load capacity model with node centrality," *Complex System Modeling and Simulation*, vol. 4, no. 1, p. 1–14, Mar. 2024.
[18] P. Crucitti, V. Latora, and M. Marchiori, "Model for cascading failures in complex networks," *Phys. Rev. E*, vol. 69, p. 045104, 2004.
[19] A. E. Motter and Y.-C. Lai, "Cascade-based attacks on complex networks," *Phys. Rev. E*, vol. 66, p. 065102, Dec 2002.
[20] I.-C. Lin, O. Yağan, and C. Joe-Wong, "Dynamic Coupling Strategy for Interdependent Network Systems Against Cascading Failures," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 4, pp. 2265–2282, 2023.
[21] A. Scala, P. G. De Sanctis Lucentini, G. Caldarelli, and G. D'Agostino, "Cascades in interdependent flow networks," *Physica D: Nonlinear Phenomena*, vol. 323-324, pp. 35–39, 2016.
[22] Y. Zhang, A. Arenas, and O. Yağan, "Cascading failures in interdependent systems under a flow redistribution model," *Physical Review E*, vol. 97, no. 2, p. 022307, 2018.
[23] R. Kumar, S. Kumari, and M. Bala, "Minimizing the effect of cascade failure in multilayer networks with optimal redistribution of link loads," *Journal of Complex Networks*, vol. 9, no. 6, p. cnab043, Oct. 2021.
[24] J. Pei, Y. Liu, W. Wang, and J. Gong, "Cascading failures in multiplex network under flow redistribution," *Physica A: Statistical Mechanics and its Applications*, vol. 583, p. 126340, 2021.
[25] S. Hong, X. Zhang, J. Zhu, T. Zhao, and B. Wang, "Suppressing failure cascades in interconnected networks: Considering capacity allocation pattern and load redistribution," *Modern Physics Letters B*, vol. 30, no. 05, p. 1650049, Feb. 2016.
[26] N. Wang, Z.-Y. Jin, and J. Zhao, "Cascading failures of overload behaviors on interdependent networks," *Physica A: Statistical Mechanics and its Applications*, vol. 574, p. 125989, Jul. 2021.
[27] D. Zhou and A. Elmokashfi, "Overload-based cascades on multiplex networks and effects of inter-similarity," *PLOS ONE*, vol. 12, no. 12, p. e0189624, Dec. 2017.
[28] J. Ma and J. Xin, "Robustness of dual-layer networks considering node load redistribution," *International Journal of Modern Physics C*, vol. 35, no. 04, p. 2450046, Apr. 2024.
[29] K.-M. Lee, K. I. Goh, and I. M. Kim, "Sandpiles on multiplex networks," *Journal of the Korean Physical Society*, vol. 60, no. 4, p. 641–647, Feb. 2012.
[30] J. Wang, R. He, H. Sun, and H. He, "Cascading dynamics on coupled networks with load-capacity interplay and concurrent recovery-failure," *Physica A: Statistical Mechanics and its Applications*, vol. 661, p. 130373, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437125000251
[31] O. İrsoy and O. Yağan, "Analysis and optimization of robustness in multiplex flow networks against cascading failures," 2025. [Online]. Available: https://arxiv.org/abs/2502.06159
[32] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5g telecommunication networks: A survey of principles and models," *IEEE Network*, vol. 33, no. 6, p. 172–179, Nov. 2019.
[33] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 208–221, 2014.
[34] A.-L. Barabási, *Network science*. Cambridge University Press, 2016.

ACKNOWLEDGMENTS

APPENDIX A
A PROOF OF THEOREM 3.1

We begin by recalling the fractional sizes of the state sets in Figure 3. As defined in Eq. (8), $n_{AB,t}$ is the fraction of nodes surviving in both layers at iteration t, while $n_{A,t}$ and $n_{B,t}$ are the fractions surviving in layers A and B, respectively. The fractions surviving only in A and only in B are denoted by $n_{A\setminus B,t}$ and $n_{B\setminus A,t}$, respectively, and satisfy $n_{A,t} = n_{AB,t} + n_{A\setminus B,t}$ and $n_{B,t} = n_{AB,t} + n_{B\setminus A,t}$. In Figure 3, $\mathcal{F}_{A,t}$ (resp. $\mathcal{F}_{B,t}$) denotes the set of nodes that, at iteration t, fail in the other layer but remain functional in layer A (resp. layer B)—i.e., the nodes that transition from $\mathcal{N}_{AB,t}$ to $\mathcal{N}_{A\setminus B,t+1}$ (resp. $\mathcal{N}_{B\setminus A,t+1}$). Analogous to Eq. (8), we define their normalized sizes as

$$f_{A,t} = \frac{|\mathcal{F}_{A,t}|}{N}, \qquad f_{B,t} = \frac{|\mathcal{F}_{B,t}|}{N}. \tag{27}$$

In what follows, we derive recursive expressions for $n_{AB,t}$, $f_{A,t}$, $f_{B,t}$, $n_{A\setminus B,t}$, and $n_{B\setminus A,t}$, and then use these to obtain the recursions for the excess loads $Q_{A,t}$ and $Q_{B,t}$.

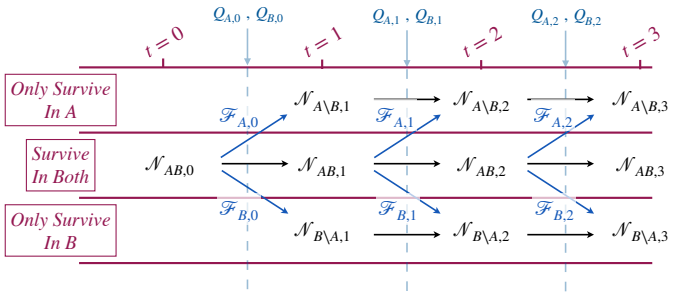


Fig. 8: Conceptual diagram of the cascade progression. Here $t = 0, 1, 2, \cdots$ indicates the rounds of flow redistribution. $\mathcal{N}_{AB,t}$ denotes the set of nodes surviving in both layers at iteration t. For $i \in \{A, B\}$, $\mathcal{N}_{i,t}$ denotes the set of nodes surviving in layer i at iteration t, and $\mathcal{N}_{i\setminus j,t} := \mathcal{N}_{i,t} \setminus \mathcal{N}_{j,t}$ denotes the set of nodes surviving only in layer i (with $j \neq i$). Moreover, $\mathcal{F}_{i,t}$ is the set of nodes that were surviving in both layers at iteration t but will survive only in layer i at iteration t+1. $Q_{A,t}, Q_{B,t}$ denote the excess load per surviving node resulting from the failures up to iteration t.

A. Calculation of $n_{AB,t}$

After the initial attack of size p, a fraction p of nodes fail in both layers, yielding $n_{AB,0} = 1 - p$. At the next iteration, the set of nodes surviving in both layers, $\mathcal{N}_{AB,1}$, is a subset of $\mathcal{N}_{AB,0}$. In other words, nodes that satisfy the survival conditions (16) and (17) proceed to the next stage surviving in both layers. Thus,

$$n_{AB,1} = n_{AB,0} \cdot \mathbb{P}[S_A > Q'_{A,0}, S_B > Q'_{B,0}]. \tag{28}$$

By similar reasoning, $\mathcal{N}_{AB,2} \subseteq \mathcal{N}_{AB,1}$. However, the nodes in $\mathcal{N}_{AB,1}$ have already survived the effective excess loads $(Q'_{A,0}, Q'_{B,0})$. Therefore, the survival probability at the next step conditions on this event:

$$n_{AB,2} = n_{AB,1} \cdot \mathbb{P}\left(\begin{array}{c|c} S_A > Q'_{A,1} & S_A > Q'_{A,0} \\ S_B > Q'_{B,1} & S_B > Q'_{B,0} \end{array} \right). \tag{29}$$

In general, for $t = 1, 2, \ldots,$

$$n_{AB,t} = n_{AB,t-1} \cdot \mathbb{P}\left(\begin{array}{c|c} S_A > Q'_{A,t-1} & S_A > Q'_{A,t-2} \\ S_B > Q'_{B,t-1} & S_B > Q'_{B,t-2} \end{array} \right), \tag{30}$$

with the convention $Q'_{A,-1} = Q'_{B,-1} = 0$.

By definition, $Q'_{A,t}$ and $Q'_{B,t}$ are the effective excess loads per surviving node resulting from failures up to iteration t, so they are nondecreasing in t. Using this monotonicity, the conditional probability in (30) can be written as a ratio of probabilities. Iterating from 1 through t gives the telescoping product

$$\begin{aligned} n_{AB,1} &= n_{AB,0}\, P\big[S_A > Q'_{A,0},\, S_B > Q'_{B,0}\big], \\ n_{AB,2} &= n_{AB,1}\, \frac{P\big[S_A > Q'_{A,1},\, S_B > Q'_{B,1}\big]}{P\Big[S_A > Q'_{A,0},\, S_B > Q'_{B,0}\Big]}, \\ &\vdots \\ n_{AB,t} &= n_{AB,t-1}\, \frac{P\big[S_A > Q'_{A,t-1},\, S_B > Q'_{B,t-1}\big]}{P\Big[S_A > Q'_{A,t-2},\, S_B > Q'_{B,t-2}\Big]}. \end{aligned}$$

Multiplying these identities yields the compact expression

$$n_{AB,t} \;=\; n_{AB,0} \cdot P\big[S_A > Q'_{A,t-1},\, S_B > Q'_{B,t-1}\big]. \quad (31)$$

B. Calculation of $n_{A\setminus B,t}$ and $n_{B\setminus A,t}$

The calculation of single-layer survivors involves two sources. For iteration t, nodes surviving only in layer A either (i) already survived only in A at $t-1$ and remain active, or (ii) survived in both layers at $t-1$ and fail in B at t. Formally, $(\mathcal{N}_{A\setminus B,t-1} \cup \mathcal{F}_{A,t}) \supseteq \mathcal{N}_{A\setminus B,t}$. Thus, we first characterize the fractional sizes of $\mathcal{F}_{A,t}$ and $\mathcal{F}_{B,t}$, and then write the recursion for the size of single-layer survivors. For brevity, we present the expressions for layer A; the formulas for layer B are obtained symmetrically.

The nodes that fail in layer B but survive in layer A at the end of iteration t (i.e., $\mathcal{F}_{A,t}$) have more free space in A than the effective load in A and less free space in B than the effective load in B. Hence, the fractional size at iteration t is

$$f_{A,t} = n_{AB,t} \cdot \mathbb{P}\left(\begin{array}{c|c} S_A > Q'_{A,t} & S_A > Q'_{A,t-1} \\ S_B < Q'_{B,t} & S_B > Q'_{B,t-1} \end{array} \right). \quad (32)$$

Substituting $n_{AB,t}$ from (31) gives

$$\begin{aligned} f_{A,t} = n_{AB,0} \cdot \mathbb{P}\left[S_A > Q'_{A,t-1}, S_B > Q'_{B,t-1}\right] \\ \cdot \mathbb{P}\left(\begin{array}{c|c} S_A > Q'_{A,t} & S_A > Q'_{A,t-1} \\ S_B < Q'_{B,t} & S_B > Q'_{B,t-1} \end{array} \right), \end{aligned}$$

which yields

$$f_{A,t} = n_{AB,0} \cdot \mathbb{P}\left[S_A > Q'_{A,t}, Q'_{B,t-1} < S_B < Q'_{B,t}\right]. \quad (33)$$

At $t = 1$, the set of nodes surviving only in layer A consists of nodes that survive the initial attack but fail in layer B due to the excess loads $(Q_{A,0}, Q_{B,0})$. Thus, $\mathcal{N}_{A\setminus B,1} = \mathcal{F}_{A,0}$, which implies $n_{A\setminus B,1} = f_{A,0}$. For $t = 2$, $\mathcal{N}_{A\setminus B,2}$ includes all nodes in $\mathcal{F}_{A,1}$ and those in $\mathcal{F}_{A,0}$ that can survive $(Q_{A,1}, Q_{B,1})$. Therefore,

$$n_{A\setminus B,2} = f_{A,1} + f_{A,0} \cdot \mathbb{P}\left[S_A > Q_{A,1} - \beta_A L_A \;\middle|\; \begin{array}{c} S_A > Q'_{A,0} \\ S_B < Q'_{B,0} \end{array} \right].$$

Nodes in $\mathcal{F}_{A,0}$ have already failed in B, so the cross-layer contribution from B no longer consumes capacity in A. Accordingly, their survival is given by $S_A > Q_{A,1} - \beta_A L_A$ (consistent with (18)). Moreover, by the definition of $\mathcal{F}_{A,0}$, these nodes survived in A and failed in B at $t = 0$, hence the conditioning on $S_A > Q'_{A,0}$ and $S_B < Q'_{B,0}$.

For $t = 3$, $\mathcal{N}_{A\setminus B,3}$ consists of all nodes in $\mathcal{F}_{A,2}$, plus those in $\mathcal{F}_{A,1}$ and $\mathcal{F}_{A,0}$ that can survive $Q_{A,2}$:

$$\begin{aligned} n_{A\setminus B,3} = f_{A,2} \\ + f_{A,1} \cdot \mathbb{P}\left[S_A > Q_{A,2} - \beta_A L_A \;\middle|\; \begin{array}{c} S_A > Q'_{A,1} \\ Q'_{B,0} < S_B < Q'_{B,1} \end{array} \right] \\ + f_{A,0} \cdot \mathbb{P}\left[S_A > Q_{A,2} - \beta_A L_A \;\middle|\; \begin{array}{c} S_A > Q'_{A,0} \\ S_B < Q'_{B,0} \end{array} \right]. \end{aligned}$$

These probabilities share the same survival inequality $S_A > Q_{A,2} - \beta_A L_A$, but they condition on different events reflecting the iteration at which failure in B first occurred. Using that $\mathbb{P}\big[S_A > Q_{A,2} - \beta_A L_A \mid S_A > Q'_{A,2}\big] = 1$ (since $L_A > 0$), we can write

$$\begin{aligned} n_{A\setminus B,3} = \sum_{j=0}^{2} f_{A,j} \cdot \\ \mathbb{P}\left[S_A > Q_{A,2} - \beta_A L_A \;\middle|\; \begin{array}{c} S_A > Q'_{A,j} \\ Q'_{B,j-1} < S_B < Q'_{B,j} \end{array} \right]. \end{aligned} \quad (34)$$

Here, j denotes the iteration at which the node first failed in layer B. Thus, for general t,

$$\begin{aligned} n_{A\setminus B,t} = \sum_{j=0}^{t-1} f_{A,j} \cdot \\ \mathbb{P}\left[S_A > Q_{A,t-1} - \beta_A L_A \;\middle|\; \begin{array}{c} S_A > Q'_{A,j} \\ Q'_{B,j-1} < S_B < Q'_{B,j} \end{array} \right], \end{aligned} \quad (35)$$

and substituting (33) for $f_{A,j}$ yields

$$\begin{aligned} n_{A\setminus B,t} &= \sum_{j=0}^{t-1} n_{AB,0} \cdot \mathbb{P}\left[S_A > Q'_{A,j}, Q'_{B,j-1} < S_B < Q'_{B,j}\right] \\ &\quad \cdot \mathbb{P}\left[S_A > Q_{A,t-1} - \beta_A L_A \;\middle|\; \begin{array}{c} S_A > Q'_{A,j} \\ Q'_{B,j-1} < S_B < Q'_{B,j} \end{array} \right] \\ &= n_{AB,0} \sum_{j=0}^{t-1} \mathbb{P}\left[\begin{array}{c} S_A > Q_{A,t-1} - \beta_A L_A \\ S_A > Q'_{A,j} \\ Q'_{B,j-1} < S_B < Q'_{B,j} \end{array} \right]. \end{aligned} \quad (36)$$

Finally, the total fractions surviving in layers A and B are $n_{A,t} = n_{AB,t} + n_{A\setminus B,t}$ and $n_{B,t} = n_{AB,t} + n_{B\setminus A,t}$.

C. Calculation of $Q_{A,t}, Q_{B,t}$

We derive the recursion for $Q_{A,t}$; the expression for $Q_{B,t}$ follows by symmetry.

To compute the excess load per surviving node, we use conservation of total load under global redistribution: the load released by failed nodes must be borne by the surviving nodes. This yields the decomposition

$$\begin{array}{c}\text{Total Excess} \\ \text{Load}\end{array} = \begin{array}{c}\text{Borne Load from} \\ \text{Initial attack}\end{array} + \begin{array}{c}\text{Borne Load from} \\ \text{Cascading Effect}\end{array}. \quad (37)$$

At iteration t, the per-survivor excess load in layer A multiplied by the fraction of nodes still functional in layer A equals the cumulative type-A load (normalized by N) released up to time t:

$$\begin{array}{c}\text{Total Excess} \\ \text{Load}\end{array} = n_{A,t} Q_{A,t}. \quad (38)$$

The contribution from the initial attack is immediate: a fraction p of nodes is removed at $t = 0$, and each removed node releases its baseline type-A load, so

$$\begin{matrix}\text{Borne Load from}\\ \text{Initial attack}\end{matrix} = p\mathbb{E}[L_A]. \tag{39}$$

Borne load from cascading effect: Subsequent failures add load to layer A through two disjoint channels:

(i) nodes that remain functional in both layers through iteration j and then newly fail in A at $j{+}1$;

(ii) nodes that had already failed in B at some earlier step and later fail in A.

We treat these separately because (i) uses the two-layer survival test, whereas (ii) invokes the single-layer test with additional free-space, consistent with (18).

(i) From both-layer survivors that newly fail in A.

At iteration $j{+}1$, the fraction $n_{AB,j}$ is still alive in both layers. Among them, the nodes that newly fail in A satisfy $Q'_j > S_A$ while also having $S_B > Q'_{j-1}$ and $S_A > Q'_{j-1}$ from the previous step. Then, we can obtain the total *normalized* type-A load they release as

$$\sum_{j=0}^{t-1} n_{AB,j}\mathbb{P}\left[S_A < Q'_{A,j} \,\middle|\, \begin{matrix}S_A > Q'_{j-1}\\ S_B > Q'_{j-1}\end{matrix}\right] \mathbb{E}\left[L_A \,\middle|\, \begin{matrix}Q'_j > S_A > Q'_{j-1}\\ S_B > Q'_{j-1}\end{matrix}\right]. \tag{40}$$

Here the probability identifies those that *newly* cross the layer-A threshold at the end of step j, and the conditional expectation captures their type-A loads at failure. Substituting $n_{AB,j}$ from (31) collapses the product of probabilities and gives

$$\begin{aligned}&\sum_{j=0}^{t-1} n_{AB,0}\mathbb{P}\left[\begin{matrix}Q'_j > S_A > Q'_{j-1}\\ S_B > Q'_{j-1}\end{matrix}\right] \mathbb{E}\left[L_A \,\middle|\, \begin{matrix}Q'_j > S_A > Q'_{j-1}\\ S_B > Q'_{j-1}\end{matrix}\right]\\ &= \sum_{j=0}^{t-1} n_{AB,0}\cdot\mathbb{E}\left[L_A\cdot\mathbb{1}\left(\begin{matrix}Q'_j > S_A > Q'_{j-1}\\ S_B > Q'_{j-1}\end{matrix}\right)\right],\end{aligned} \tag{41}$$

which is precisely the expectation of L_A over the nodes failing in A at iteration j. Thus, we sum over all possible $j < t$.

(ii) From nodes that failed in B earlier and later fail in A.

Let $\mathcal{F}_{A,j}$ be the nodes that, at iteration j, move from both-layer survival to only-A survival (i.e., they fail in B at j). By time $t > j$, their failure in A is governed by the single-layer condition (18). Summing their (normalized) contributions over $j < t{-}1$ yields

$$\sum_{j=0}^{t-2} f_{A,j}\cdot\mathbb{P}\left[S_A < Q_{A,t-1} - \beta_B L_B \,\middle|\, \begin{matrix}Q'_{A,j} < S_A\\ Q'_{B,j-1} < S_B < Q'_{B,j}\end{matrix}\right] \cdot\mathbb{E}\left[L_A \,\middle|\, \begin{matrix}Q'_{A,j} < S_A < Q_{A,t-1} - \beta_B L_B\\ Q'_{B,j-1} < S_B < Q'_{B,j}\end{matrix}\right]. \tag{42}$$

The probability is the chance that a node in $\mathcal{F}_{A,j}$ fails in A by time t under the single-layer test, and the conditional expectation is its type-A load at failure. Using (33) for $f_{A,j}$ gives

$$\sum_{j=0}^{t-2} n_{AB,0}\cdot\mathbb{E}\left[L_A\cdot\mathbb{1}\left(\begin{matrix}Q'_{A,j} < S_A < Q_{A,t-1} - \beta_B L_B\\ Q'_{B,j-1} < S_B < Q'_{B,j}\end{matrix}\right)\right]. \tag{43}$$

With these pieces in hand, we return to the load-balance identity (37) and combining the initial-attack term with the two cascading channels:

$$\begin{aligned}&n_{A,t}Q_{A,t} =\\ &\quad p\mathbb{E}[L_A] + \sum_{j=0}^{t-1} n_{AB,0}\cdot\mathbb{E}\left[L_A\cdot\mathbb{1}\left(\begin{matrix}Q'_j > S_A > Q'_{j-1}\\ S_B > Q'_{j-1}\end{matrix}\right)\right]\\ &\quad + \sum_{j=0}^{t-2} n_{AB,0}\cdot\mathbb{E}\left[L_A\cdot\mathbb{1}\left(\begin{matrix}Q'_{A,j} < S_A < Q_{A,t-1} - \beta_B L_B\\ Q'_{B,j-1} < S_B < Q'_{B,j}\end{matrix}\right)\right].\end{aligned}$$

Solving for $Q_{A,t}$ yields (24). By symmetry, $Q_{B,t}$ is obtained by swapping the A and B in the expressions above.

Orkun İrsoy is a Ph.D. student at Electrical and Computer Engineering department in Carnegie Mellon University. He received his B.S. and M.S. degrees in Industrial Engineering from Boğaziçi University, Istanbul (Türkiye), in 2019 and 2022, respectively. His research broadly focuses on analyzing complex systems using network science, simulation, and data science, spanning a wide range of application areas, including engineering, health, and social systems."

Osman Yağan is a Research Professor of Electrical and Computer Engineering at Carnegie Mellon University. He received his Ph.D. degree in Electrical and Computer Engineering from the University of Maryland at College Park, MD in 2011, and his B.S. degree in Electrical and Electronics Engineering from the Middle East Technical University, Ankara (Turkey) in 2007. Dr. Yağan's research focuses on modeling, analysis, and performance optimization of computing systems, and uses tools from applied probability, network science, data science, and machine learning. He is a recipient of a CIT Dean's Early Career Fellowship, an IBM Academic Award, and best paper awards in ICC 2021, IPSN 2022, and ASONAM 2023.