AEGIS: Throughput-Guaranteed Resilient Routing via a Conditional Value-at-Risk Approach

Zhaofeng Zhang, *Member, IEEE*, Xuanli Lin, *Graduate Student Member, IEEE*, Guoliang Xue, *Fellow, IEEE*, and Kevin S. Chan, *Senior Member, IEEE*

***Abstract*—The past decade has witnessed significant progress in next-generation wireless networks. Resilient routing is essential for maintaining reliability in mission-critical network services, particularly in dynamic and adversarial environments. Traditional traffic engineering (TE) approaches rely on pre-computed paths. Still, they face performance limitations when the number of pre-computed paths is small and scalability challenges when the number is large. This study seeks to answer the fundamental question: *"How can we achieve throughput-guaranteed resilient routing under network failures without pre-computing routing paths?"* We propose AEGIS, a novel throughput-guaranteed resilient routing scheme leveraging a conditional value-at-risk (CVaR) approach, which proactively guarantees the required throughput under normal conditions and enables recovery during network failures. Specifically, we propose an optimization problem that minimizes the CVaR of total throughput loss across all the failure situations while respecting user budget and network constraints. The above optimization problem is non-differentiable and non-linear; we then reformulate it as an equivalent linear program (LP) and develop an optimal solution. However, the above solution will induce cyclic flows due to resource reservation behaviors. To achieve a more resource-efficient routing, we propose a bisection approach to obtain a CVaR upper bound so that the corresponding routing is acyclic. Extensive numerical evaluations demonstrate the trade-offs among various approaches and highlight the advantages of AEGIS.**

***Index Terms*—Resilient routing, throughput guarantee, conditional value-at-risk, resource-efficient.**

I. Introduction

THE past decade has seen rapid advancements in next-generation wireless technology. Under this emerging trend, next-generation communication networks are expected to comprise ultra-densely deployed devices with integrated sensing, communication, and computation capabilities [16], [17]. However, meeting the requirements of mission-critical network applications remains a complex challenge due to environmental uncertainty, network congestion, and limited resources [37], [38]. For example, a military sensor in a harsh environment may need to transmit collected images for further processing. Such data transfer operates under constrained resources (e.g., limited network bandwidth) in a dynamic and adversarial environment. Furthermore, the network may become congested due to high traffic demands from multiple competing nodes, further degrading transmission reliability and increasing latency. Specifically, network components may fail unexpectedly during transmission due to environmental instability, while congestion can cause severe packet loss and unpredictable delays, making resilient data delivery even more challenging. These challenges are not unique to military networks but also arise in large-scale, real-world network infrastructures such as networks of commercial fiber providers, which must deal with dynamic traffic patterns, congestion, and unexpected failures while ensuring high availability and resilience. Therefore, throughput-guaranteed resilient routing schemes are critical to ensure the reliability of next-generation network services.

Resilient routing and recovery schemes in networks have been studied extensively [23], [27], [28]. However, designing throughput-guaranteed resilient routing schemes under *environmental uncertainty* remains a challenge. A key challenge for network operators is balancing network utilization and availability while accounting for potential node and link failures [14], [15]. These objectives are intrinsically conflicting: ensuring high availability necessitates maintaining network utilization at a sufficiently low level to accommodate traffic shifts during failures. This has motivated a large body of work on network utilization-availability trade-off facing possible failures [10], [18], [19]. Among these approaches, traffic engineering (TE) stands out due to its ability to optimize network performance by dynamically managing traffic flows, efficiently representing diverse traffic as network flows compactly, and its practicality for implementation, particularly with the support of systems like Software-Defined Networking (SDN) [20].

Notably, one recent TE work, TEAVAR [10], aims to tackle the above challenge by drawing inspiration from financial risk theory. In particular, TEAVAR leverages a probabilistic model of network failures and minimizes the throughput loss by applying a conditional value-at-risk (CVaR) approach. This

Received 17 March 2025; revised 8 September 2025 and 25 January 2026; accepted 30 March 2026; approved by IEEE TRANSACTIONS ON NETWORKING Editor J. Wu. Date of publication 16 April 2026; date of current version 6 May 2026. This work was supported in part by the DOD Center of Excellence in Future Generation Wireless Technology (CoE-FutureG); and in part by the Army Research Laboratory, under Agreement W911NF-23-2-0225. *(Zhaofeng Zhang and Xuanli Lin contributed equally to this work.) (Corresponding author: Guoliang Xue.)*

Zhaofeng Zhang is with Delaware State University, Dover, DE 19901 USA (e-mail: zzhang@desu.edu).

Xuanli Lin and Guoliang Xue are with Arizona State University, Tempe, AZ 85287 USA, and also with CoE-FutureG, Tempe, AZ 85281 USA (e-mail: xlin54@asu.edu; xue@asu.edu).

Kevin S. Chan is with DEVCOM U.S. Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: kevin.s.chan.civ@army.mil).

This article has supplementary downloadable material available at https://doi.org/10.1109/TON.2026.3684425, provided by the authors.

Digital Object Identifier 10.1109/TON.2026.3684425

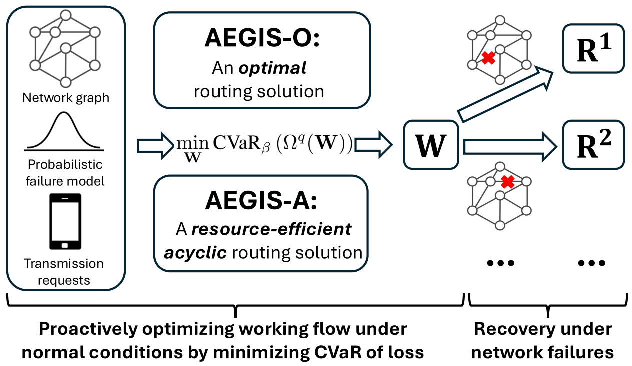


Fig. 1. AEGIS: throughput-guaranteed resilient routing based on CVaR optimization.

TABLE I
TABLE OF NOTATIONS AND DEFINITIONS

Symbol	Type	Definition / First appearance
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	graph	Undirected network graph with set of nodes $\mathcal{V}$, and set of links $\mathcal{E}$ (§2.A).
$n = \lvert V\rvert$, $m = \lvert E\rvert$	int	Number of nodes and links (§2.A).
$e = (u, v) \in E$	link	Link between nodes u and v (§2.A).
c_e	real	Capacity of link e. $c_e > 0$ (§2.A).
κ_e	real	Per-unit cost on link e. $\kappa_e > 0$ (§2.A).
$\mathcal{K} = \{1, \ldots, K\}$	set	Set of users (§2.A).
$\mathcal{Z}$	set	Set of SRLG failure events (§2.B).
$z \in \mathcal{Z}$	event	SRLG failure event (§2.B).
p_z	real	Probability of failure event z (§2.B).
$q = (q_1, \ldots, q_{\lvert Z\rvert})$	vector	Failure situation; $q_z \in \{0, 1\}$ indicates whether event z occurs (§2.B).
$\mathcal{Q}$	set	Set of all failure situations (§2.B).
$p_{\hat{q}}$	real	Probability of situation $\hat{q}$ (§2.B; Eq. (1)).
$\mathcal{E}_q$	set	Affected links under failed situation q (§2.B).
(s_k, t_k, d_k, b_k)	tuple	User k's request: source, destination, demand, budget (§2.C).
$\gamma \in (0, 1]$	real	Throughput guarantee level (§2.D).
$\mathbf{W} = (W_1, \ldots, W_K)$	flows	Working flow under normal conditions (§2.D; Def. 1).
$W_k : V \times V \to \mathbb{R}_+$	flow	Working flow for commodity k (§2.D; Def. 1).
f_k	real	Throughput of commodity k under normal condition (§2.D; Def. 1).
$R^q = (R^q_1, \ldots, R^q_K)$	flows	Recovery flow under failure situation q (§2.E;Def. 2).
$R^q_k : V \times V \to \mathbb{R}_+$	flow	Recovery flow for commodity k (§2.E; Def. 2).
g^q_k	real	Recovery throughput for commodity k (§2.E; P2).
$\Omega_q(W)$	real	Total throughput loss in failure situation q (§3.A; Eq. (4)).
β	real	CVaR confidence level. $\beta \in [0, 1)$ (§3.A).
$\mathrm{VaR}_\beta(\Omega_q(W))$	real	Value-at-Risk at level β (§3.A; Eq. (5)).
$\mathrm{CVaR}_\beta(\Omega_q(W))$	real	CVaR at level β (§3.A; Eq. (6)).
α	real	Auxiliary variable in LP reformulations (§3.A).
Φ^q	real	Per-situation slack for CVaR linearization (§3.B).
λ	real	CVaR tolerance level used by AEGIS-A. $\lambda \geq 0$ (§4.A; Eq. (10)).
ϵ	real	Convergence threshold for bisection in Alg. 1. $\epsilon > 0$ (§4.C; Alg. 1).
$\lambda^{\mathrm{UB}}_{\gamma,\beta}$, $\lambda^{\mathrm{LB}}_{\gamma,\beta}$	real	Upper/lower bounds in bisection (§4.C; Alg. 1).
$\lambda^{(t)}_{\gamma,\beta}$	real	Midpoint at iteration t in bisection (§4.C; Alg. 1).

approach can enable network operators to strike the utilization-availability balance that best suits their goals and operational reality. However, like most existing TE approaches, TEAVAR's approach relies on pre-computed path information. On the one hand, the number of paths will exponentially increase as the scale of the network topology becomes more extensive, which will significantly increase the computational complexity when computing the routing strategy. On the other hand, when the routing paths are restricted to being among only a few pre-computed paths, the network throughput achieved could be low. Furthermore, TEAVAR does not consider the user budget constraints, which could be a critical bottleneck for designing the resilient routing scheme. Motivated by the above, compared to the existing TE approaches, our work takes a step further and seeks to answer the following fundamental question *"With possible network failures, how can we design a throughput-guaranteed resilient routing scheme without pre-computing the routing paths while the user budget and network constraints are satisfied?"*

In this paper, we adopt a CVaR methodology to answer the above question. We develop AEGIS, a throughput-guaranteed resilient routing scheme based on a CVaR approach, proactively guaranteeing the required throughput under normal conditions and providing recovery routing during network failures. Specifically, as illustrated in Fig. 1, we compute a working flow by minimizing the CVaR of throughput loss across all failure situations while adhering to the user budget and network constraints; meanwhile, the network throughput is guaranteed following users' requirements under normal conditions. Then, we can enable a recovery process based on the optimized working flow above under a specific network failure situation. The above CVaR minimization problem is non-differentiable and non-linear. We reformulate the above optimization problem as an equivalent linear program (LP) and propose a corresponding optimal solution. However, a key observation is that the optimal solution above can result in cyclic working flows due to resource reservation behaviors for recovery. To achieve a more resource-efficient routing, we propose a bisection approach to obtain an upper bound on the CVaR of total throughput loss, which is close to its optimal value, ensuring that a corresponding routing strategy remains acyclic. The main contributions of this paper are the following:

- We propose a novel resilient routing scheme by formulating an optimization problem to minimize the CVaR of the total throughput loss across all the failure situations while satisfying the user budget and network constraints.
- We transform the original non-differentiable and non-linear optimization problem into an equivalent LP and propose an optimal solution.
- We observe that the above optimal solution will induce cyclic flows due to resource reservation behaviors for recovery. We propose a bisection approach that eliminates cyclic flows while achieving performance close to the optimal and resource-efficient routing by obtaining an upper bound on the CVaR of total throughput loss.
- We conduct extensive numerical evaluations over various real-world and synthetic network topologies and provide insights into the trade-offs among various approaches, highlighting the advantages of AEGIS.

The rest of this paper is organized as follows. We present the system model in Section II. We introduce the problem formulation of AEGIS and its optimal solution in Section III. We present how we achieve acyclic and resource-efficient routing in Section IV. We present evaluation results in Section V, introduce related work in Section VI, and conclude the paper in Section VII.

II. SYSTEM MODEL

In this section, we formally introduce the network model, probabilistic failure model, user requests, working flow, and recovery flow. Table I summarizes the key notations used throughout our system model and problem formulation.

A. Network Model

We model a network by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of links. We assume that the network has $|\mathcal{V}| = n$ nodes and $|\mathcal{E}| = m$ links. Each link $e = (u, v) \in \mathcal{E}$ is associated with a capacity, denoted by $c_e > 0$, and a per unit cost, denoted by $\kappa_e > 0$. We note that $\kappa_e > 0$ reflects a generic measure of resource usage or penalty associated with routing traffic over link e. In practice, this can represent diverse physical metrics such as energy consumption, latency-based penalties, or economic costs. This abstraction ensures that the framework remains adaptable across various network contexts, allowing practitioners to tailor the cost model to their specific operational needs. We also have a set $\mathcal{K} = \{1, 2, \ldots, K\}$ of K users.

B. Probabilistic Failure Model

Following the previous work [10], we consider a general failure model consisting of a set of failure events $\mathcal{Z}$. A failure event $z \in \mathcal{Z}$ represents a single shared risk link group (SRLG) becoming unavailable. Here, SRLG is a concept in network survivability that describes the situation where different links may suffer from a common failure (e.g., node/switch failure) if they share a common risk [25], [30]. Each failure event z occurs with probability p_z. We assume that the probabilities of two events are independent and uncorrelated. By $q = (q_1, \ldots, q_{|\mathcal{Z}|})$, we denote a network failure situation, where each element q_z is a binary random variable, indicating whether failure event z occurred ($q_z = 1$) or not. More formally, let $\mathcal{Q}$ be the set of all possible failure situations, and let $p_{\hat{q}}$ denote the probability of failure situation $\hat{q} = (\hat{q}_1, \ldots \hat{q}_{|\mathcal{Z}|}) \in \mathcal{Q}$. The probability of $\hat{q}$ can be obtained using the following equation:

$$p_{\hat{q}} = \Pi_z \left[\hat{q}_z p_z + (1 - \hat{q}_z)(1 - p_z)\right], \quad (1)$$

where $\hat{q}_z \in \{0, 1\}$. For each $q \in \mathcal{Q}$, we use $\mathcal{E}^q$ to denote the set of failed links under the network failure situation q.

C. Transmission Request and Routing Objective

The network accepts data transmission requests from network users. The request from user $k \in \mathcal{K}$ is specified by a four-tuple (s_k, t_k, d_k, b_k), where $s_k, t_k \in \mathcal{V}$ are the source and destination nodes, respectively, $d_k > 0$ is the bandwidth demand, and $b_k > 0$ is the user's budget. Note that these network data transmission requests originate from their respective source nodes. Routing strategies are determined using SDN techniques, leveraging a global view of the network state information. Our routing objective is to achieve the following key functions and properties:

- **Ensuring Reliable Network Operations:** The routing strategy guarantees the required network throughput under normal conditions, ensuring stable and efficient data transmission.
- **Proactive and Seamless Failure Recovery:** The routing strategy mitigates the impact of network failures by minimizing throughput loss through forward-looking recovery mechanisms. These recovery mechanisms must ensure that each commodity does not consume additional network resources under each failure situation.
- **Adhering to User Budget and Network Constraints:** The routing strategy must satisfy the network's physical constraint and each user's budget constraint.

These properties collectively ensure the feasibility and effectiveness of the proposed routing scheme. In the following, we formally define our routing strategy based on the well-known multi-commodity flow model [12].

D. Working Flow

We treat the network transmission from each user as a commodity, i.e., the network transmission from user $k \in \mathcal{K}$ is denoted as commodity-k. Assuming that the network is capable of multi-path routing, we need to determine the routing paths and the bandwidth allocation for each link to accommodate a commodity. We use the following working flow to characterize the network transmission under normal conditions.

Definition 1 (γ-Feasible Working Flow): Given a throughput guarantee level $\gamma \in (0, 1]$, we define a γ-feasible working flow by $\mathbf{W} = (W_1, W_2, \ldots, W_K)$, where for $k \in \mathcal{K}$, $W_k : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}^+$ is a flow for commodity-$k(s_k, t_k, d_k, b_k)$, and commodity-k's throughput $f_k = \sum_{(u,t_k)\in\mathcal{E}} W_k(u, t_k) - \sum_{(t_k,v)\in\mathcal{E}} W_k(t_k, v)$ is at least γd_k, $\forall k \in \mathcal{K}$.

Given $\gamma \in (0, 1]$, the following LP problem checks if a γ-feasible $\mathbf{W}$ exists. If this LP problem is feasible, its solution is a γ-feasible $\mathbf{W}$.

$$\mathbf{P1}(\gamma)\text{:} \ \texttt{Find} \ \ W_k, \forall k \in \mathcal{K} \quad (2)$$

$$\text{s. t.} \ \sum_{(u,v)\in\mathcal{E}} W_k(u, v) - \sum_{(u,v)\in\mathcal{E}} W_k(v, u) = 0, \quad \forall k \in \mathcal{K}, v \in \mathcal{V} \setminus \{s_k, t_k\}; \quad \text{(a)}$$

$$\sum_{(u,t_k)\in\mathcal{E}} W_k(u, t_k) - \sum_{(t_k,v)\in\mathcal{E}} W_k(t_k, v) \geq \gamma d_k, \forall k \in \mathcal{K}; \quad \text{(b)}$$

$$\sum_{k\in\mathcal{K}} W_k(u, v) + W_k(v, u) \leq c_e, \ \ \forall e = (u, v) \in \mathcal{E}; \quad \text{(c)}$$

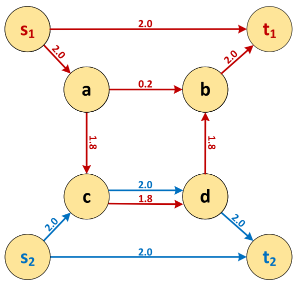
$$\sum_{e=(u,v)\in\mathcal{E}} \kappa_e \left(W_k(u, v) + W_k(v, u)\right) \leq b_k, \ \ \forall k \in \mathcal{K}; \quad \text{(d)}$$

$$W_k(u, v) \geq 0, W_k(v, u) \geq 0, \forall k \in \mathcal{K}, (u, v) \in \mathcal{E}. \quad \text{(e)}$$

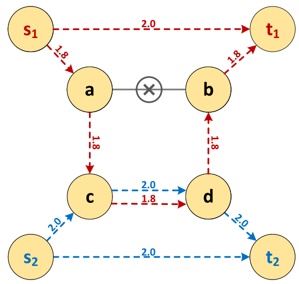
Explanation: $\mathbf{P1}(\gamma)$ is a feasibility problem. It may be infeasible for large γ. Constraint (a) ensures flow conservation at each node, excluding the source and the destination. Constraint (b) ensures the throughput of commodity-k is at least γd_k. Constraint (c) ensures link capacity constraints. Constraint (d) ensures user budget constraints. Constraint (e) defines the range of each variable.

E. Recovery Flow

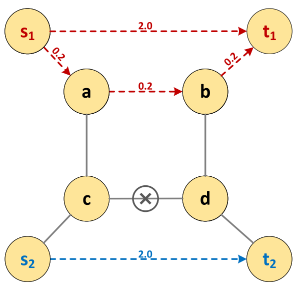
To ensure resilience, in addition to deciding the bandwidth allocation of the working flow under normal conditions, the network operator must also determine the paths and allocation against each possible network failure situation. Meanwhile, we aim for a seamless recovery mechanism where each commodity does not consume additional network resources under each failure situation while the total recovery throughput is maximized, which follows the same philosophy of the recovery mechanism used in TEAVAR [10]. Specifically, given a network failure situation, each commodity's recovery flow



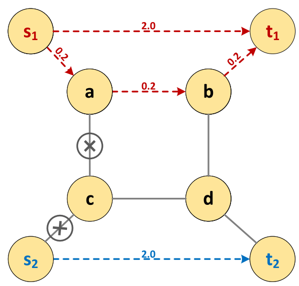
(a) Working flow $\mathbf{W}$



(b) Recovery flow corresponding to $\mathbf{W}$ when link (a, b) fails



(c) Recovery flow corresponding to $\mathbf{W}$ when link (c, d) fails



(d) Recovery flow corresponding to $\mathbf{W}$ when links (a, c) and (s_2, c) fail

Fig. 2. An example illustrating how a working flow induces recovery flows given different network failure situations.

must be 0 on the failed links, and each commodity's recovery flow cannot exceed its working flow on the operational links. We use the following example to illustrate the recovery mechanism.

Example 1: Consider a network topology with two transmission requests, one from s_1 to t_1, with a demand of 4, and one from s_2 to t_2, with a demand of 4. Each link has a capacity of 3.8. The per-unit cost on all the links is zero in this example. One working flow $\mathbf{W}$ is presented in Fig. 2. The flow value for commodity-1 is illustrated in solid red arrows. The flow value for commodity-2 is illustrated in solid blue arrows.

Fig. 2b shows a recovery flow corresponding to $\mathbf{W}$ when link (a, b) fails. The recovery flow is 0 on link (a, b). The flows on link (s_1, a) and (b, t_1) decrease from 2.0 to 1.8 due to the flow conservation. Fig. 2c presents a recovery flow corresponding to $\mathbf{W}$ when link (c, d) fails. For commodity-1, the recovery flow is 0 on link (c, d). The flows on link (a, c) and (d, b) decrease from 1.8 to 0, and the flows on link (s_1, a) and (b, t_1) decrease from 2.0 to 0.2 due to the flow conservation. For commodity-2, the recovery flow is 0 on link (c, d). The flows on link (s_2, c) and (d, t_2) decrease from 2.0 to 0 due to the flow conservation. Fig. 2d presents a recovery flow corresponding to $\mathbf{W}$ when both links (a, c) and (s_2, c) fail. For commodity-1, the recovery flow is 0 on link (a, c). The flows on link (c, d) and (d, b) decrease from 1.8 to 0, and the flows on link (s_1, a) and (b, t_1) decrease from 2.0 to 0.2 due to the flow conservation. For commodity-2, the recovery flow is 0 on link (s_2, c). The flows on link (c, d) and (d, t_2) decrease from 2.0 to 0 due to the flow conservation.

Given a working flow and a network failure situation, the recovery flow is formally defined in the following.

Definition 2: [Recovery Flow] Given a working flow $\mathbf{W}$ and a failure situation q, a corresponding recovery flow is defined by $\mathbf{R}^q = (R_1^q, R_2^q, \dots, R_K^q)$ where for $k \in \mathcal{K}$ $R_k^q : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}^+$ is the recovery flow for commodity-$k(s_k, t_k, d_k, b_k)$. $\mathbf{R}^q$ can be calculated by solving the optimization problem $\mathbf{P2(W}, q)$.

Let $\mathbf{W}$ be a solution of $\mathbf{P1}(\gamma)$. Given a failure situation $q \in Q$, we define commodity-k's recovery throughout as $g_k^q = \sum_{(u,t_k)\in\mathcal{E}} R_k^q(u, t_k) - \sum_{(t_k,v)\in\mathcal{E}} R_k^q(t_k, v)$. $\mathbf{R}^q$ can be computed by solving the following LP problem.

$\mathbf{P2(W}, q)$**:**

$$\max_{\mathbf{R}^q} \sum_{k\in\mathcal{K}} \left(\sum_{(u,t_k)\in\mathcal{E}} R_k^q(u, t_k) - \sum_{(t_k,v)\in\mathcal{E}} R_k^q(t_k, v) \right) \tag{3}$$

$$\text{s. t.} \sum_{(u,v)\in\mathcal{E}} R_k^q(u, v) - \sum_{(u,v)\in\mathcal{E}} R_k^q(v, u) = 0, \quad \forall k \in \mathcal{K}, v \in \mathcal{V} \setminus \{s_k, t_k\}; \tag{f}$$

$$R_k^q(u, v) = 0, R_k^q(v, u) = 0, \ \forall k \in \mathcal{K}, (u, v) \in \mathcal{E}^q; \tag{g}$$

$$R_k^q(u, v) \le W_k(u, v), R_k^q(v, u) \le W_k(v, u), \quad \forall k \in \mathcal{K}, (u, v) \in \mathcal{E}; \tag{h}$$

$$R_k^q(u, v) \ge 0, \ R_k^q(v, u) \ge 0, \ \forall k \in \mathcal{K}, (u, v) \in \mathcal{E}; \tag{i}$$

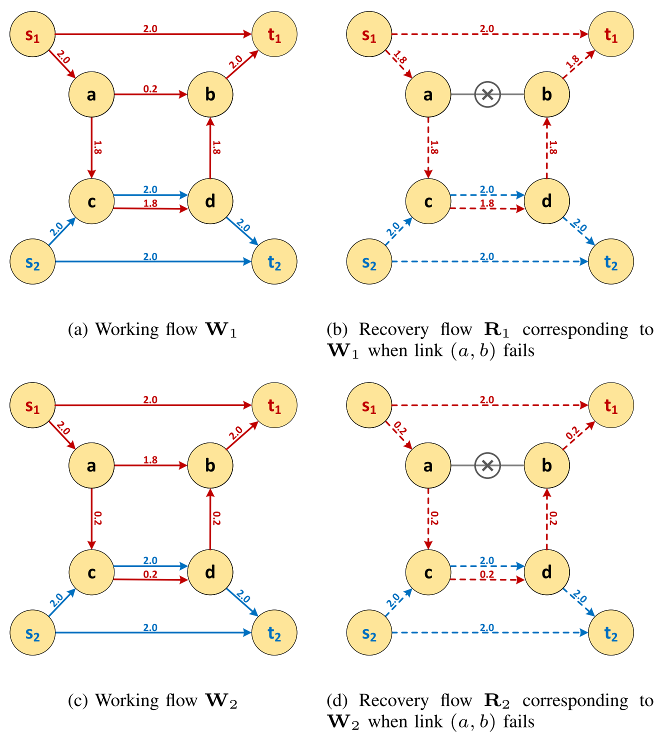
$$\sum_{(u,t_k)\in\mathcal{E}} R_k^q(u, t_k) - \sum_{(t_k,v)\in\mathcal{E}} R_k^q(t_k, v) \le \sum_{(u,t_k)\in\mathcal{E}} W_k(u, t_k) - \sum_{(t_k,v)\in\mathcal{E}} W_k(t_k, v), \ \forall k \in \mathcal{K}. \tag{j}$$

Explanation: The objective of $\mathbf{P2(W}, q)$ is to maximize the total recovery throughput so that each commodity can receive enough bandwidth during the recovery operation when one specific network failure situation happens. Constraint (f) ensures flow conservation at each node during the recovery operation, excluding the source and the destination. Constraint (g) ensures no bandwidth is allocated to the affected links. Constraint (h) ensures that each commodity's recovery flow cannot exceed its working flow. Constraint (i) defines the range of each variable. Constraint (j) ensures that each commodity's recovery throughput does not exceed its working throughput. The recovery flow R_k^q can be solved independently by $\mathbf{P2(W}, q)$ for each network failure situation $q \in Q$. Note that link capacity constraints and user budget constraints are automatically satisfied since constraint (h) ensures each commodity's recovery flow cannot exceed its working flow.

III. Throughput-Guaranteed Resilient Routing: Minimizing CVaR of Throughput Loss

Given a working flow, we can obtain a recovery flow to mitigate the impact of one specific network failure situation $q \in Q$ by solving $\mathbf{P2(W}, q)$. Nevertheless, given a specific failure situation, different working flows will induce different recovery flows, hence different total throughput losses, illustrated by the following example.

Example 2: Consider a network topology with two transmission requests, one from s_1 to t_1, with a demand of 4, and one from s_2 to t_2, with a demand of 4. Each link has a capacity of 3.8. The link (a, b) can fail. The per-unit cost on all the links is zero.



(a) Working flow $\mathbf{W}_1$

(b) Recovery flow $\mathbf{R}_1$ corresponding to $\mathbf{W}_1$ when link (a, b) fails

(c) Working flow $\mathbf{W}_2$

(d) Recovery flow $\mathbf{R}_2$ corresponding to $\mathbf{W}_2$ when link (a, b) fails

Fig. 3. An example illustrating how different working flows lead to varying recovery flows and throughput losses under a specific network failure situation. Working flow throughput is not necessarily a good indicator of the performance of the induced recovery operation. Both $\mathbf{W}_1$ and $\mathbf{W}_2$ have a total network throughput of 8.0. However, after link (a, b) fails, recovery flow $\mathbf{R}_1$ has a total network throughput loss of 0.2, while recovery flow $\mathbf{R}_2$ has a total network throughput loss of 1.8.

One feasible working flow $\mathbf{W}_1$ is presented in Fig. 3a. The flow value for commodity-1 is illustrated in solid red arrows. The flow value for commodity-2 is illustrated in solid blue arrows. The total throughput of these two network transmission requests is 8. Fig. 3b shows the recovery flow corresponding to $\mathbf{W}_1$ shown in Fig. 3a when (a, b) fails. The total throughput of these two network transmission requests is 7.8 during the recovery operation; hence, the total network throughput loss is 0.2. Fig. 3c presents another feasible working flow $\mathbf{W}_2$. The total throughput of these two network transmission requests is 8. Fig. 3d shows the recovery flow corresponding to $\mathbf{W}_2$ shown in Fig. 3c when (a, b) fails. The total throughput of these two network transmission requests is 6.2 during the recovery operation; hence, the total network throughput loss is 1.8.

The above example shows that two different working flows, even with the same value of total throughput, can induce different performance degradations during recovery operations. Thus, the following natural questions arise:

- *How can we proactively obtain a working flow to mitigate the performance degradation during recovery operations across all possible network failure situations?*
- *What optimization metric properly guides the development of this routing strategy?*

In this section, we propose a throughput-guaranteed resilient routing scheme to address these questions, in which we formulate the routing problem by minimizing the CVaR of total throughput loss. Then, we present an optimal solution to the above routing problem based on an equivalent LP formula.

A. Minimizing CVaR of Total Throughput Loss

1) Preliminaries: As shown in Example 2, different working flows and their induced recovery flows can result in varying total throughput losses. This insight motivates us to use the total throughput loss to quantify the quality of the proactive routing strategy we aim to design. Formally, let $\mathbf{W}$ be a working flow, Let $\mathbf{R}^q$ be a corresponding recovery flow computed by $\mathbf{P2}(\mathbf{W}, q)$, for $q \in \mathcal{Q}$. We define $\Omega^q(\mathbf{W})$ as the total throughput loss during the recovery operation under the network failure situation q

$$\Omega^q(\mathbf{W}) = \sum_{k \in \mathcal{K}} \left(f_k - g_k^q\right), \ \forall q \in \mathcal{Q}. \tag{4}$$

Given a $\mathbf{W}$, $\Omega^q(\mathbf{W})$ is a random variable depending on the realization of q.

Due to the stochastic nature of $\Omega^q(\mathbf{W})$, a natural approach to obtain a working flow is to minimize the expectation of $\Omega^q(\mathbf{W})$ over all the failure situations:

$$\min_{\mathbf{W}} \mathbb{E}\left(\Omega^q(\mathbf{W})\right) = \sum_{q \in \mathcal{Q}} p_q \cdot \Omega^q(\mathbf{W}). \tag{5}$$

However, expectation optimization often underestimates the impact of rare but catastrophic events [33], [34]. The minimax criterion seeks to minimize the worst-case loss across all failure situations:

$$\texttt{Minimax}\left(\Omega^q(\mathbf{W})\right) = \min_{\mathbf{W}} \max_{q \in \mathcal{Q}} \Omega^q(\mathbf{W}), \tag{6}$$

which is robust but overly conservative and blind to probabilistic structure. To achieve risk-averse performance while incorporating the stochastic nature of $\Omega^q(\mathbf{W})$, we minimize the CVaR of $\Omega^q(\mathbf{W})$, a common risk measure in financial risk management. Formally, given a confidence level $\beta \in [0, 1)$, define

$$\mathrm{VaR}_\beta(\Omega^q(\mathbf{W})) = \min\left\{\omega \,|\, \Pr(\Omega^q(\mathbf{W}) \le \omega) \ge \beta\right\}, \tag{7}$$

$$\mathrm{CVaR}_\beta(\Omega^q(\mathbf{W})) = \mathbb{E}\left[\Omega^q(\mathbf{W}) | \Omega^q(\mathbf{W}) \ge \mathrm{VaR}_\beta(\Omega^q(\mathbf{W}))\right]. \tag{8}$$

Equation (7) defines the value-at-risk (VaR) with the confidence level β, representing the minimum value such that the network throughput loss will not exceed this value with confidence level β. Equation (8) defines its CVaR as the expected value of network throughput loss across all the failure situations in Q where the loss exceeds the VaR.

In formula (8), computing CVaR requires first computing VaR, which is not easy to compute. Fortunately, Rockafellar and Uryasev have shown in [24] that the CVaR of $\Omega^q(\mathbf{W})$ can be computed efficiently using the following formula:

$$\mathrm{CVaR}_\beta(\Omega^q(\mathbf{W})) = \min_{\alpha \in \mathbb{R}} \left\{\alpha + \frac{1}{1-\beta} \sum_{q \in \mathcal{Q}} p_q \left(\Omega^q(\mathbf{W}) - \alpha\right)^+\right\}.$$

2) Problem Formulation: We present the problem formulation of our throughput-guaranteed resilient routing scheme, AEGIS, which *optimizes a working flow in a forward-looking manner by minimizing the CVaR of total throughput loss*:

$$\texttt{AEGIS-O}(\gamma,\beta)\textbf{:} \min_{\mathbf{W}} \mathrm{CVaR}_\beta\left(\Omega^q(\mathbf{W})\right)$$
$$\text{s. t. } (3)-(7);$$
$$(9)-(13),\ \forall q \in \mathcal{Q}. \tag{9}$$

Explanation: The objective of **AEGIS-O**(γ,β) is to minimize the CVaR of total throughput loss during the recovery operation, subject to all the constraints proposed in $\mathbf{P1}(\gamma)$ and $\mathbf{P2}(\mathbf{W},q)$, including flow conservation constraints, link capacity constraints, user budget constraints, and constraints ensuring the flow value of commodity-k is at least γd_k under normal conditions.

Remark 1: The proposed CVaR approach properly balances expectation and robust optimization objectives, controlled by the confidence level β: If $\beta=0$, **AEGIS-O**(γ,β) is equivalent to minimizing the expectation of total throughput loss over all the failure situations; If $\beta \to 1$, $\mathrm{CVaR}_\beta\left(\Omega^q(\mathbf{W})\right)$ converges to the maximum loss over all possible failure situations, and **AEGIS-O**(γ,β) equals to the minimax approach in robust optimization. While the expectation optimization lacks robustness, the minimax approach addresses only the extreme cases; in practice, the minimax approach could reduce worst-case loss but drastically increase loss in all other cases.

B. Optimal Solution Based on Equivalent LP

It is challenging to solve **AEGIS-O**(γ,β) directly since it is non-differentiable and non-linear due to the CVaR term in the objective. To address this, we transform **AEGIS-O**(γ,β) into an equivalent LP problem **AEGIS-O-LP**(γ,β), enabling efficient optimization, as presented in the following theorem.

Theorem 1: The problem **AEGIS-O**(γ,β) can be transformed into an equivalent LP problem:

$$\texttt{AEGIS-O-LP}(\gamma,\beta): \min_{\mathbf{W},\alpha,\Phi^q} \alpha + \frac{1}{1-\beta}\sum_{q\in\mathcal{Q}} p_q \Phi^q$$
$$\text{s. t. } \Phi^q \geq 0,\ \forall q \in \mathcal{Q};$$
$$\Phi^q \geq \Omega^q(\mathbf{W}) - \alpha,\ \forall q \in \mathcal{Q};$$
$$(3)-(7); \tag{10}$$
$$(9)-(13),\ \forall q \in \mathcal{Q}, \tag{11}$$

which can be solved in $O((Km|\mathcal{Q}| + Km + |\mathcal{Q}|)^3\mathcal{L})$. Here, m is the number of links, K is the number of commodities, $|\mathcal{Q}|$ is the number of failure situations, and $\mathcal{L}$ is the input size, which denotes the total bit-length required to encode all numerical parameters in the LP formulation, including link capacities, link costs, user demands, failure situation-specific availabilities, and relevant threshold parameters (e.g., γ,β).

We present the proof of the above theorem in the appendix for a smoother paper flow.

Remark 2: The above theorem proposes an optimal solution based on an equivalent LP formula of the original routing problem. This solution does not depend on pre-computing routing paths. Although TEAVAR also reformulates its TE optimization as an equivalent LP problem, there are as many variables as the number of paths. In general, since the number of routing paths grows exponentially with the network size, solving TEAVAR's equivalent LP would be much more computationally demanding than solving **AEGIS-O-LP**(γ,β) for large-scale real-world network topologies. We will elaborate on it in Section V.

Remark 3: **AEGIS-O-LP**(γ,β) can be solved in polynomial-time, given a fixed failure situation set $\mathcal{Q}$, as stated in Theorem 1. Here, we do not assume that $|\mathcal{Q}|$ is polynomial in the number of SRLGs. In fact, under general SRLG-based failure models, the size of $\mathcal{Q}$ can grow exponentially. Our analysis treats $\mathcal{Q}$ as a given input, and the complexity bounds apply under the assumption that this set is explicitly enumerated. For large-scale networks, techniques such as sampling or situation pruning may be employed to reduce $|\mathcal{Q}|$ to a tractable size.

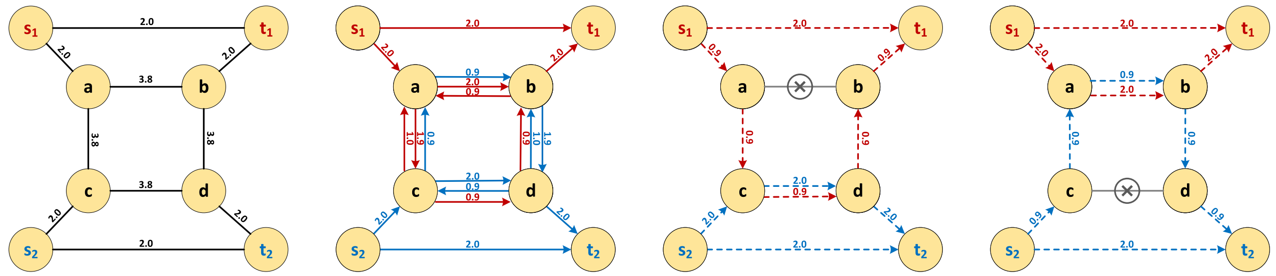
Remark 4: In Theorem 1, we analyze the complexity of solving the problem **AEGIS-O-LP**(γ,β) using the classical bit-length input size $\mathcal{L}$ [32], which is defined as the total bit-length of numerical inputs (e.g., link capacities, costs, and failure situation-specific link availabilities). While our formulation explicitly defines the failure situation set $\mathcal{Q}$, we emphasize that the LP input data exhibits strong structural similarity across these failure situations. In particular, many situations share overlapping failed SRLGs, resulting in similar constraint matrices. Thus, we assume that such input data similarity enables the LP input size $\mathcal{L}$ to remain polynomial in the number of nodes, links, and SRLGs. This justifies the use of LP complexity bounds that are polynomial in $\mathcal{L}$, supporting the claimed computational tractability of our approach.

IV. Acyclic Resource-Efficient Routing With Throughput Guarantee and CVaR Constraint

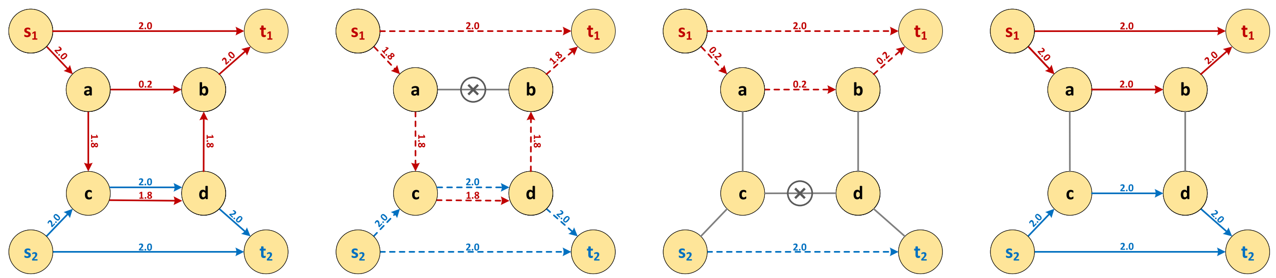
How does the solution of **AEGIS-O**(γ,β) perform? Indeed, solving **AEGIS-O**(γ,β) settles a trade-off between the expectation and robust optimization. Furthermore, unlike existing TE approaches, solving **AEGIS-O**(γ,β) does not rely on any pre-computed path information. Nevertheless, we observe that a working flow, obtained by solving **AEGIS-O**(γ,β), could be *cyclic*. Moreover, *the minimum CVaR value subject to the constraint that the flows are acyclic can be strictly larger than the minimum CVaR value without the acyclic constraint, which further indicates that a cyclic working flow is unavoidable when the objective goal is to minimize the CVaR of total throughput loss*. The following example illustrates the above key observations and the difficulties of eliminating the cycles in a working flow.

Example 3: Consider a network topology, shown in Fig. 4a, where the number on each link indicates the link's bandwidth. Assuming there are two connection requests, one from s_1 to t_1, with a demand of 4, and one from s_2 to t_2, with a demand of 4. Two links possibly fail: link (a,b) may fail with the probability of 0.90; link (c,d) may fail with the probability of 0.05. The per-unit cost on all the links is zero in this example.

A working flow that directly minimizes CVaR of throughput loss is presented in Fig. 4b, with $\beta = 0.90$. The flow for commodity-1 is illustrated in solid red arrows. The flow for



(a) Network topology. (b) Working flow that minimizes CVaR. (c) Recovery flow when link (a, b) fails from (b). (d) Recovery flow when link (c, d) fails from (b).



(e) Working *acyclic* flow that minimizes CVaR. (f) Recovery flow when link (a, b) fails from (e). (g) Recovery flow when link (c, d) fails from (e). (h) Simply removing cycles from (b)

Fig. 4. An example showing that a cyclic working flow is unavoidable when the goal is to minimize the CVaR of total throughput loss. (a) Example network, where the number on each link indicates the link's bandwidth. There are two connection requests, one from s_1 to t_1, with a demand of 4, one from s_2 to t_2, with a demand of 4. Link (a, b) has a failure probability of 0.90. Link (c, d) has a failure probability of 0.05. (b) A working flow that minimizes CVaR, with $\beta = 0.90$. **The CVaR has a value of 2.00**. Flow value for commodity-1 is illustrated in red. Flow value for commodity-2 is illustrated in blue. (c) Recovery flow corresponding to (b) when (a, b) fails. (d) Recovery flow corresponding to (b) when (c, d) fails. (e) A working flow that minimizes the CVaR, *subject to the constraint that flows are acyclic*. **The CVaR has a value of 2.09 which is larger than the CVaR achieved without the acyclic constraint**. (f) Recovery flow corresponding to (e) when link (a, b) fails. (g) Recovery flow corresponding to (e) when link (c, d) fails. (h) A working flow obtained by simply removing cycles in (b). The corresponding CVaR value is 2.90, which is larger than the CVAR value in (e).

commodity-2 is illustrated in solid blue arrows. The value of CVaR of throughput loss during the recovery operation is 2.00. Fig. 4c shows a recovery flow corresponding to Fig. 4b when (a, b) fails. Fig. 4d shows a recovery flow corresponding to Fig. 4b when (c, d) fails.

Fig. 4e presents a working flow that minimizes the CVaR of throughput loss, subject to the constraint that flows are acyclic. Using this approach, the CVaR of throughput loss has a value of 2.09, which is larger than the CVaR achieved without the acyclic constraint. This indicates that a cyclic working flow is unavoidable when directly minimizing the CVaR of the total throughput loss. Fig. 4f shows a recovery flow corresponding to Fig. 4e when link (a, b) fails. Fig. 4g shows a recovery flow corresponding to Fig. 4e when link (c, d) fails. Fig. 4h shows a working flow obtained by simply removing cycles in Fig. 4b. The corresponding CVaR value is 2.90, larger than the CVaR value in Fig. 4e. This indicates that naively removing cycles will significantly decrease the performance.

Remark 5: The essential reason why **AEGIS-O**(γ, β) produces a cyclic solution of working flow is as follows: The objective of **AEGIS-O**(γ, β) and constraint (h) ensure that a working flow always reserves backup bandwidth as much as possible for recovery operations against network failure situations. Thereby, the value of a recovery flow can be pushed up. This backup resource reservation behavior will cause cycles on some links, increasing bandwidth usage and user budget costs.

In the following, we present an approach to eliminating cycles in working flows. It guarantees total throughput under normal conditions and follows the CVaR constraint of throughput loss across all possible failure situations. The resulting acyclic routing is more resource-efficient.

A. Computing Working Flow With Throughput Guarantee and CVaR Constraint

Ensuring an acrylic working flow would decrease network resource usage but is highly non-trivial. As illustrated in Example 3, a working flow obtained by solving **AEGIS-O**(γ, β) always reserves extra bandwidth to push up the value of an induced recovery flow. Then, a natural question arises: *can we sacrifice some performance during recovery operations to achieve an acyclic and more resource-efficient working flow?* To tackle this, we present an approach to attain acyclic routing with throughput guarantee and CVaR constraint.

Define $\lambda \geq 0$ as a tolerance level of the $\mathrm{CVaR}_\beta(\Omega^q(\mathbf{W}))$. Given a throughput guarantee level $\gamma \in (0, 1]$, a confidence level $\beta \in [0, 1)$, and a $\lambda \geq 0$, we aim to compute *an acyclic working flow such that commodity-k's throughput is at least γd_k under normal conditions for each $k \in \mathcal{K}$, while the $\mathrm{CVaR}_\beta(\Omega^q(\mathbf{W}))$ is at most the tolerance level λ, thereby ensuring that the performance sacrifice remains within acceptable limits during recovery operations under failures.*

A natural approach is to solve the following problem.

$$\begin{aligned} &\texttt{AEGIS-A}(\gamma,\beta,\lambda): \\ &\min_{\mathbf{W}} \sum_{k\in\mathcal{K}} \sum_{(u,v)\in\mathcal{E}} (W_k(u,v) + W_k(v,u)) \\ &\text{s. t. } \mathrm{CVaR}_\beta(\Omega^q(\mathbf{W})) \le \lambda; \\ &\qquad (3)-(7); \\ &\qquad (9)-(13),\ \forall q \in \mathcal{Q}. \end{aligned} \tag{12}$$

Explanation: Cycles in a working flow increase the network resource usage. The objective of $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ is to minimize the total network resource usage under normal conditions; intuitively, the number of cycles in a working flow may be decreased. Meanwhile, constraint (12) ensures that the CVaR of network throughput loss is within the prescribed tolerance level λ, thereby ensuring that the performance degradation during recovery operations is controlled. In addition, all the constraints proposed in **P1**(γ) and **P2**$(\mathbf{W}, q)$ are satisfied, including flow conservation constraints, link capacity constraints, user budget constraints, and constraints ensuring the flow value of commodity-k is at least γd_k under normal conditions.

In the following, we show that $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ can be transformed into an equivalent LP problem, which can be solved efficiently. Then, we present a bisection approach to obtain a proper tolerance level λ of CVaR.

B. Equivalent LP of $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$

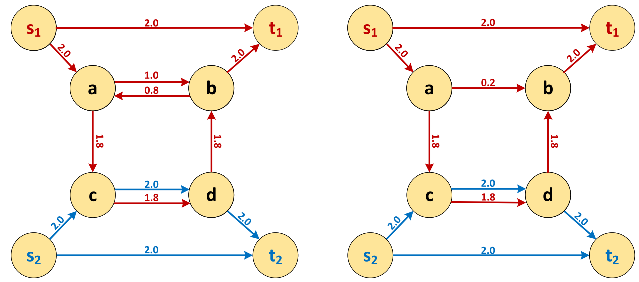
Directly solving $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ is challenging since it is non-differentiable and non-linear due to the CVaR term in constraint (12). To address this, we transform $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ into an equivalent LP problem $\texttt{AEGIS-A-LP}(\gamma,\beta,\lambda)$, as presented in the following theorem.

Theorem 2: The problem $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ can be transformed into an equivalent LP problem:

$$\begin{aligned} &\texttt{AEGIS-A-LP}(\gamma,\beta,\lambda): \\ &\min_{\mathbf{W},\alpha,\Phi^q} \sum_{k\in\mathcal{K}} \sum_{(u,v)\in\mathcal{E}} (W_k(u,v) + W_k(v,u)) \\ &\text{s. t. } \Phi^q \ge 0,\ \forall q \in \mathcal{Q}; \\ &\qquad \Phi^q \ge \Omega^q(\mathbf{W}) - \alpha,\ \forall q \in \mathcal{Q}; \qquad (13) \\ &\qquad \alpha + \frac{1}{1-\beta}\sum_{q\in\mathcal{Q}} p_q \Phi^q \le \lambda; \qquad (14) \\ &\qquad (3)-(7); \\ &\qquad (9)-(13),\ \forall q \in \mathcal{Q}, \qquad (15) \end{aligned}$$

which can be solved in $O((Km|\mathcal{Q}| + Km + |\mathcal{Q}|)^3\mathcal{L})$. Here, m is the number of links, K is the number of commodities, $|\mathcal{Q}|$ is the number of failure situations, and $\mathcal{L}$ is the input size, which denotes the total bit-length required to encode all numerical parameters in the LP formulation, including link capacities, link costs, user demands, failure situation-specific availabilities, and relevant threshold parameters (e.g., γ, β).

We present the proof of the above theorem in the appendix for a smoother paper flow.



(a) $\lambda = 2.05$ (b) $\lambda = 2.09$

Fig. 5. An example showing that acyclic working flows can not be guaranteed without selecting a proper tolerance level λ.

C. Selecting CVaR Tolerance Level: A Bisection Approach

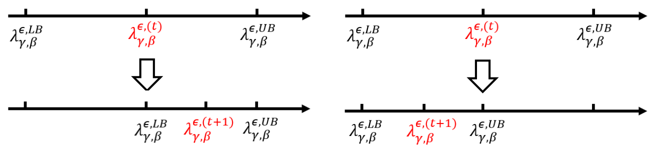
Now, we have transformed $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ into an equivalent problem $\texttt{AEGIS-A-LP}(\gamma,\beta,\lambda)$. While the objective function in $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ tries to eliminate cycles in a working flow by minimizing network resource usage under normal conditions, it cannot guarantee a feasible acyclic working flow without carefully selecting tolerance level λ, which can be illustrated in the following example.

Example 4: Consider the same network topology and settings in Example 3. As illustrated in Fig. 5a, if we choose $\gamma = 1$, $\beta = 0.90$, and $\lambda = 2.05$, the solution of $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ leads to a working flow with cycles. As shown in Fig. 5b, if we choose $\gamma = 1$, $\beta = 0.90$, and $\lambda = 2.09$, the solution of $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ leads to an acyclic working flow.

Remark 6: As illustrated in Example 4, $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ may be infeasible or produce a cyclic working flow when λ is small; $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$ is more likely to produce a feasible acyclic working flow when λ is large, but the CVaR may be larger. A proper tolerance level λ must be carefully selected to achieve an acyclic feasible working flow with a small CVaR. Intuitively, given γ and β, there exists $\lambda^{\text{opt}}_{\gamma,\beta}$ denoting the smallest λ such that there is an acyclic working flow $\mathbf{W}$ satisfying the constraints in $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$, not necessarily minimizing the objective function of $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$. However, it is challenging to directly compute $\lambda^{\text{opt}}_{\gamma,\beta}$.

In the following, we present a bisection approach to obtain an λ that is close to $\lambda^{\text{opt}}_{\gamma,\beta}$ such that there is an acyclic working flow $\mathbf{W}$ which satisfies the constraint in $\texttt{AEGIS-A}(\gamma,\beta,\lambda)$. Our proposed approach is summarized in Alg. 1. Specifically, given the convergence threshold $\epsilon > 0$, let $\lambda^{\epsilon,\text{UB}}_{\gamma,\beta}$ denote an upper bound of $\lambda^{\text{opt}}_{\gamma,\beta}$, and $\lambda^{\epsilon,\text{LB}}_{\gamma,\beta}$ denote an lower bound of $\lambda^{\text{opt}}_{\gamma,\beta}$. At the start, we set $\lambda^{\epsilon,\text{LB}}_{\gamma,\beta} = 0$ and $\lambda^{\epsilon,\text{UB}}_{\gamma,\beta} = \gamma\sum_k d_k$ since clearly $0 \le \lambda^{\text{opt}}_{\gamma,\beta} \le \gamma\sum_k d_k$. At iteration t, we set $\lambda^{\epsilon,(t)}_{\gamma,\beta} = \frac{\lambda^{\epsilon,\text{LB}}_{\gamma,\beta} + \lambda^{\epsilon,\text{UB}}_{\gamma,\beta}}{2}$ and solve $\texttt{AEGIS-A}(\gamma,\beta,\lambda^{\epsilon,(t)}_{\gamma,\beta})$ based on its equivalent LP formula $\texttt{AEGIS-A-LP}(\gamma,\beta,\lambda^{\epsilon,(t)}_{\gamma,\beta})$. As illustrated in Fig. 6, $\lambda^{\epsilon,\text{UB}}_{\gamma,\beta}$ and $\lambda^{\epsilon,\text{LB}}_{\gamma,\beta}$ are updated using the following rules:

- If $\texttt{AEGIS-A-LP}(\gamma,\beta,\lambda^{\epsilon,(t)}_{\gamma,\beta})$ is infeasible or its solution is cyclic, we update the value of $\lambda^{\epsilon,\text{LB}}_{\gamma,\beta}$ to $\lambda^{\epsilon,(t)}_{\gamma,\beta}$ and let $\lambda^{\epsilon,\text{UB}}_{\gamma,\beta}$ remain the same;
- If $\texttt{AEGIS-A-LP}(\gamma,\beta,\lambda^{\epsilon,(t)}_{\gamma,\beta})$ has an acyclic solution, we update the value of $\lambda^{\epsilon,\text{UB}}_{\gamma,\beta}$ to $\lambda^{\epsilon,(t)}_{\gamma,\beta}$ and let $\lambda^{\epsilon,\text{LB}}_{\gamma,\beta}$ remain the same.



(a) **AEGIS-A-LP**$(\gamma, \beta, \lambda^{\epsilon,(t)}_{\gamma,\beta})$ is infeasible or its solution is cyclic. (b) **AEGIS-A-LP**$(\gamma, \beta, \lambda^{\epsilon,(t)}_{\gamma,\beta})$ has an acyclic solution

Fig. 6. Illustration of proposed bisection approach for obtaining a proper λ.

Algorithm 1 `Bisection`$(\gamma, \beta, \epsilon)$

Input: γ: throughput guarantee level; β: confidence level; ϵ: convergence threshold

Output: $\lambda^{\epsilon}_{\gamma,\beta}$: CVaR tolerance level

Set $\lambda^{\epsilon,\mathrm{UB}}_{\gamma,\beta} \leftarrow \gamma \sum_k d_k$, $\lambda^{\epsilon,\mathrm{LB}}_{\gamma,\beta} \leftarrow 0$, and $t \leftarrow 0$.
while $\lambda^{\epsilon,UB}_{\gamma,\beta} - \lambda^{\epsilon,LB}_{\gamma,\beta} > \epsilon$ **do**
  $t \leftarrow t + 1$
  $\lambda^{\epsilon,(t)}_{\gamma,\beta} \leftarrow \frac{\lambda^{\epsilon,\mathrm{LB}}_{\gamma,\beta} + \lambda^{\epsilon,\mathrm{UB}}_{\gamma,\beta}}{2}$;
  Solve **AEGIS-A-LP**$(\gamma, \beta, \lambda^{\epsilon,(t)}_{\gamma,\beta})$;
  if ***AEGIS-A-LP***$(\gamma, \beta, \lambda^{\epsilon,(t)}_{\gamma,\beta})$ *is infeasible or its solution is cyclic* **then**
    $\lambda^{\epsilon,\mathrm{LB}}_{\gamma,\beta} \leftarrow \lambda^{\epsilon,(t)}_{\gamma,\beta}$
  if ***AEGIS-A-LP***$(\gamma, \beta, \lambda^{\epsilon,(t)}_{\gamma,\beta})$ *has an acyclic solution* **then**
    $\lambda^{\epsilon,\mathrm{UB}}_{\gamma,\beta} \leftarrow \lambda^{\epsilon,(t)}_{\gamma,\beta}$
Output $\lambda^{\epsilon}_{\gamma,\beta}$

The above process will repeat until the stopping criteria are met: $\lambda^{\epsilon,\mathrm{UB}}_{\gamma,\beta} - \lambda^{\epsilon,\mathrm{LB}}_{\gamma,\beta} \le \epsilon$. Once we have obtained $\lambda^{\epsilon}_{\gamma,\beta}$, we can compute an acyclic and resource-efficient working flow by solving AEGIS-A-LP$(\gamma, \beta, \lambda^{\epsilon}_{\gamma,\beta})$. We emphasize that Alg. 1 provides a sub-optimal solution to $\lambda^{\mathrm{opt}}_{\gamma,\beta}$ via a bisection-based feasibility check. Since directly obtaining $\lambda^{\mathrm{opt}}_{\gamma,\beta}$ is challenging, Alg. 1 searches for a λ such that a feasible working flow exists. The bisection procedure terminates in at most $\log_2\left(\gamma \sum_i d_i\right) + \log_2 \frac{1}{\epsilon}$ iterations. While global optimality is not guaranteed, our experiments show that the resulting solutions are empirically effective for practical deployment. We will further clarify the choice of ϵ in Section V.

V. Performance Evaluation

This section presents AEGIS's performance evaluation results compared to multiple baselines over different experimental settings and network topologies. The original TEAVAR approach does not consider throughput guarantee under normal conditions and user budget. Thus, we add the throughput-guaranteed constraint (constraint (b)) and the user budget constraint (constraint (d)) to the TEAVAR-based baselines, ensuring fair comparisons over all the experiments. All the network topologies used in our evaluation are undirected graphs. Although TEAVAR assumes a setting with directed graphs, we note that TEAVAR can be generalized to the undirected graph setting.

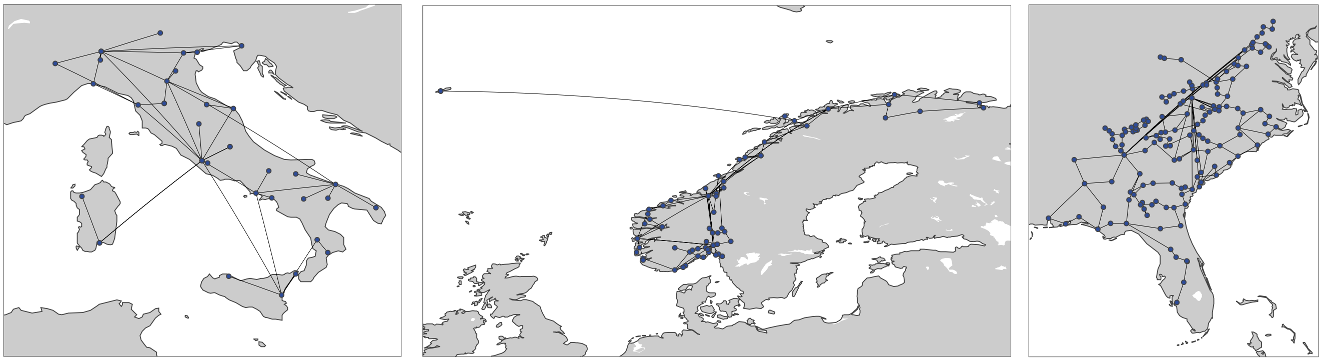
A. Evaluation Setup

Our experiments contain two parts. In the first part, we use the GARR-X network topology (shown in Fig. 7a), an ultra-broadband optical network with 54 nodes and 68 edges dedicated to the Italian research and education community [2], to evaluate AEGIS against the following TE-based baselines:

- Tea3: TEAVAR with 3-shortest path information,
- TeaA: TEAVAR with all path information,
- MinEA: Minimizing the expected throughput loss during recovery operations with all path information.

We are interested in the behavior of AEGIS in different β and γ values. Thus, we conduct two sets of experiments. In the first set, we fix $\beta = 0.9$ and sweep γ value from 0.7 to 1.0 in 0.1 increments, whereas in the second set we fix $\gamma = 0.9$ and sweep β values in 0.0, 0.6, 0.9, and 0.95. In the appendix, we present the detailed experimental settings of the commodities and SRLGs under the GARR-X network topology for a smoother paper flow.

In the second part of our evaluation, we experimented with several real-world and synthetic network topologies to show that AEGIS can efficiently provide routing solutions with competitive performance. For real-world network topologies, the following testing network topologies are used. The first network topology is Norway's National Research and Education Network (Uninett) [3], with 74 nodes and 101 edges. The second network topology is US Carrier [1], a network of fiber providers on the United States East Coast with 158 nodes and 189 edges, as shown in Fig. 7c. In the appendix, we present the detailed experimental settings for the commodities and SRLGs under the two real-world network topologies mentioned above, to facilitate a smoother paper flow. For synthetic network topologies, the following testing network topologies are used. The first network topology is a grid graph in which links form a uniform grid, and nodes are placed everywhere where links intersect. We use a 6×6 grid with randomly generated edge capacity and edge cost. The capacity for each edge is drawn from a normal distribution $\mathcal{N}(1.0, 0.5)$, and the cost is drawn from $\mathcal{N}(1.0, 0.1)$. We consider three commodities: one goes from the upper-left node to the lower-right node, one goes from the lower-left node to the upper-right node, and one goes horizontally across. Each commodity has a budget of 60. SRLGs are randomly selected among all links and assigned a random failure probability in the range $[0.0, 1.0)$. Moreover, we examine the performance of AEGIS under different network sizes by using a Waxman model [29] with a random seed, where each pair of nodes at distance d_w is joined by an edge with probability $\beta_w \cdot \exp\left(-d_w/\alpha_w\right)$. Here, we let $\alpha_w = 0.1, \beta_w = 0.2$. We generate graphs with 100, 120, 180 and 240 nodes. The bandwidth of each edge is drawn from a normal distribution $\mathcal{N}(1.0, 0.5)$, and the cost is drawn from $\mathcal{N}(1.0, 0.1)$. Each test case consists of three commodities, with the source and destination randomly chosen uniformly among all nodes, a demand of 2, and a budget equal to the number of nodes. SRLGs are set to links frequently used by paths going from a source to a destination. Since we aim to examine the impact of the number of failure situations on the average total runtime of AEGIS, for both 6×6 grid and the

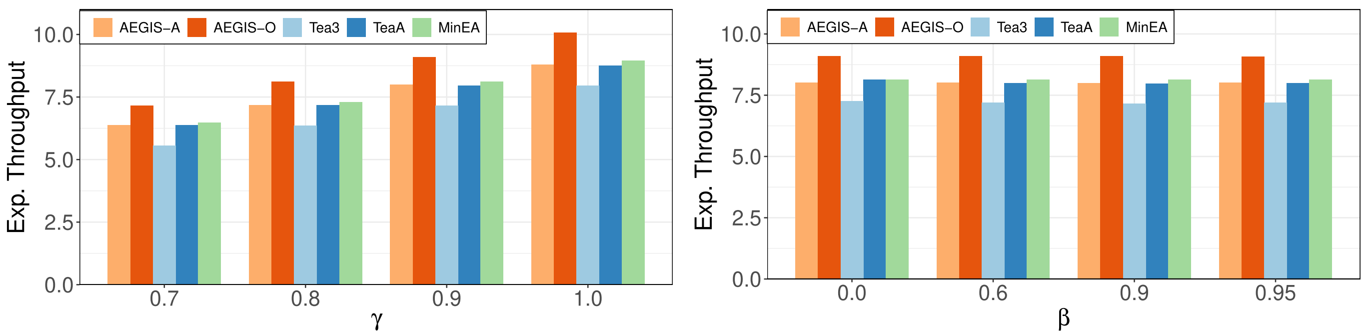


(a) GARR-X network topology with 54 nodes and 68 edges.

(b) Uninett network topology with 74 nodes and 101 edges.

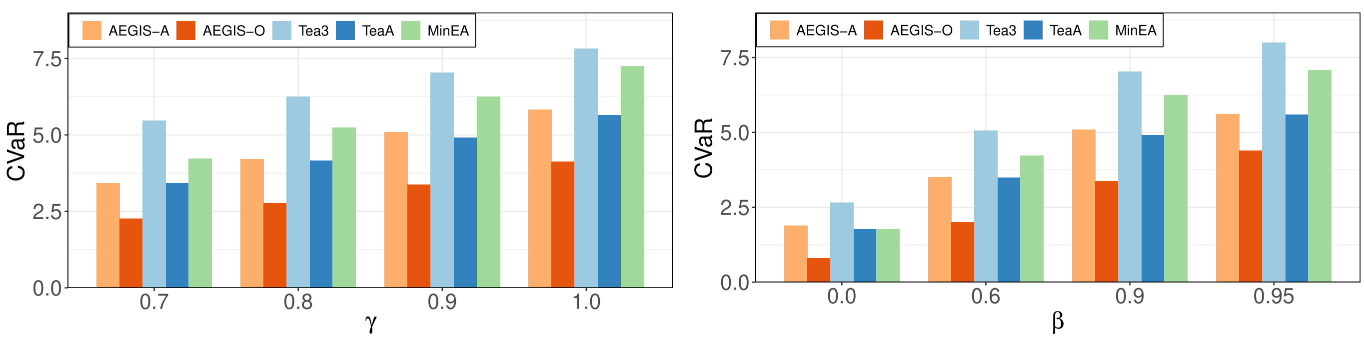
(c) US Carrier network topology with 158 nodes and 189 edges.

Fig. 7. Real-world network topologies used in performance evaluation.



(a) Expected recovery throughput with $\beta = 0.9$ and varying γ

(b) Expected recovery throughput with $\gamma = 0.9$ and varying β



(c) CVaR of throughput loss with $\beta = 0.9$ and varying γ

(d) CVaR of throughput loss with $\gamma = 0.9$ and varying β

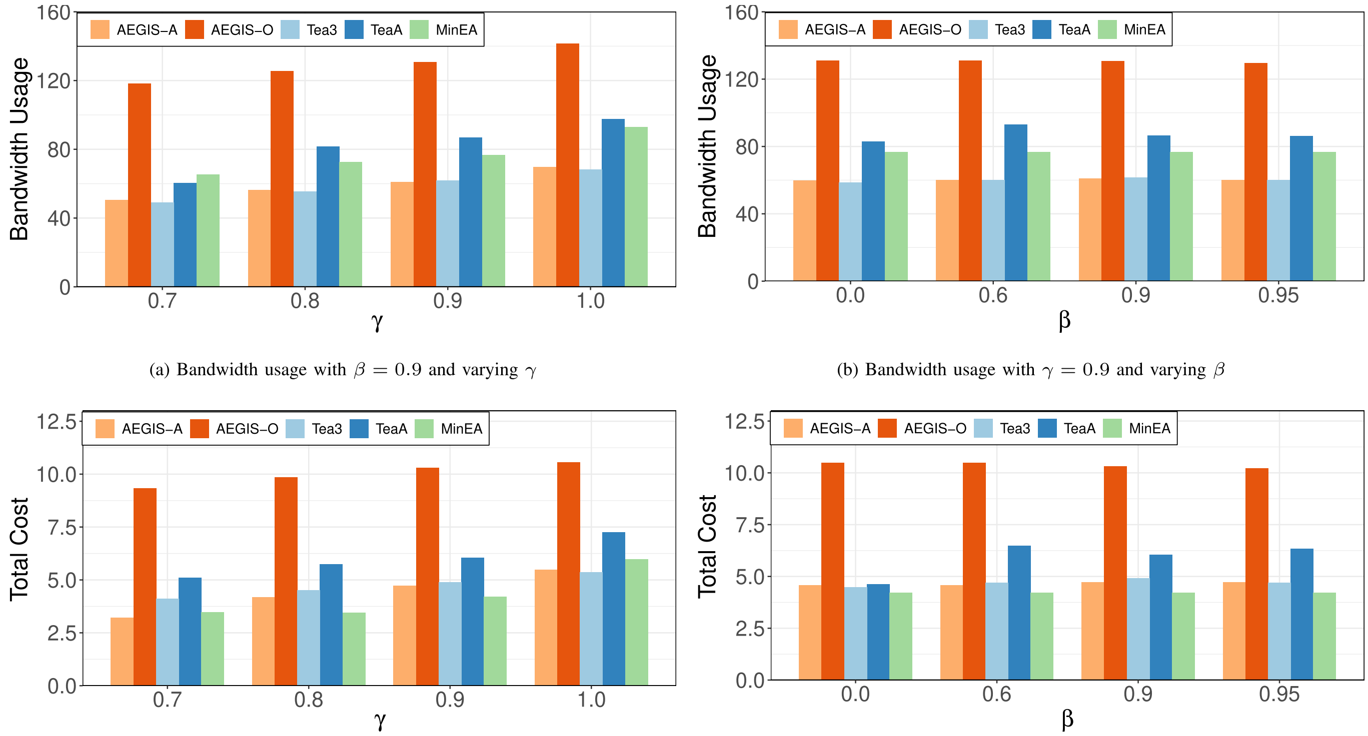
Fig. 8. Performance evaluation for GARR-X network topology.

Waxman models, we report the performance and runtime of `AEGIS` with different numbers (i.e., 3, 4, 5, and 6) of SRLGs.

All experiments are conducted on a workstation equipped with an Intel i9-12900 CPU and 128 GB of memory. All algorithms are implemented and executed in Python 3.12. We use Gurobi 11 to solve all the proposed LP problems. We set $\epsilon = 10^{-6}$ in `AEGIS-A` across all the experiments.

B. Evaluation Results and Discussion

1) Comparison Against Baselines: **`AEGIS` vs. `TEAVAR`.** Fig. 8a reveals that given the confidence level $\beta = 0.9$, `AEGIS-O`'s expected recovery network throughput is 14% to 17% higher than `TeaA` and 25% to 27% higher than `Tea3` over varying γ. `AEGIS-A` has 12% to 14% higher expected recovery network throughput than `Tea3`'s and has a comparable expected recovery network throughput with `TeaA` over varying γ. For fixed throughput guarantee level $\gamma = 0.9$, Fig. 8b shows a similar trend, with `AEGIS-O`'s expected recovery network throughput 15% higher than `TeaA` and 28% higher than `Tea3` over all the values of β. `AEGIS-A` has 11% higher expected recovery network throughput than `Tea3`'s and has a comparable expected recovery network throughput with `TeaA` over varying β. For CVaR, Fig. 8c shows that given $\beta = 0.9$, `AEGIS-O`'s CVaR of throughput loss is 22% to 43% lower than `TeaA`, and 46% to 64% lower than `Tea3` over varying γ. `AEGIS-A` has a comparable CVaR of throughput loss with `TeaA` and has a 29% to 35% lower CVaR of



(a) Bandwidth usage with $\beta = 0.9$ and varying γ

(b) Bandwidth usage with $\gamma = 0.9$ and varying β

(c) Total user budget cost with $\beta = 0.9$ and varying γ

(d) Total user budget cost with $\gamma = 0.9$ and varying β

Fig. 9. Network resource usage over different approaches for GARR-X network topology.

throughput loss than Tea3's over varying γ. A similar trend can be observed in Fig. 8d with $\gamma = 0.9$ and varying β. In a nutshell, when the routing paths are restricted to being among only a few pre-computed paths, TEAVAR performs worse than AEGIS; when all the routing paths information is given, AEGIS-A performs worse than TEAVAR, but AEGIS-O still outperforms TEAVAR. The reason why AEGIS-O consistently outperforms TEAVAR can be explained as follows: The routing paths induced by AEGIS-O's optimal solution could be cyclic due to its resource reservation behaviors; hence, the contributions of total throughput loss on these cyclic paths are zero under network failures.

AEGIS vs. MinEA. Fig. 8a-8d shows that AEGIS-O consistently outperforms MinEA for both the CVaR of total throughput loss and expected recovery throughput, with AEGIS-O performing 9% to 11% better with $\beta = 0.9$ and varying γ and 15% better with $\gamma = 0.9$ and varying β. AEGIS-O also achieves 39% to 50% less CVaR with $\beta = 0.9$ and varying γ compared to MinEA. A similar trend can also be observed with $\gamma = 0.9$ and varying β. In addition, AEGIS-A's expected recovery throughput is on par with MinEA, while achieving 10% to 29% lower CVaR with $\beta = 0.9$ and varying γ. Fig. 8d illustrates that the performance gap between AEGIS-O and MinEA increases as the confidence β increases. This is because AEGIS-O converges to minimize the worst loss among all possible failure situations as $\beta \to 1$. In other words, as β increases, AEGIS-O focuses more on the extreme cases, reducing extreme-case loss but increasing the loss of others. AEGIS-A performs similarly to MinEA when $\beta = 0$. This is because when $\beta = 0$, the CVaR-based approach is equivalent to minimizing the expected total throughput loss across all failure cases.

Network resource usage. As illustrated in Fig. 9, the total bandwidth usage and user cost of AEGIS-A are significantly less than AEGIS-O, ranging from 46% to 58% reduction in bandwidth, and 50% to 67% reduction in total cost across all settings. Here, the total user budget cost is computed as

$$\sum_{k\in\mathcal{K}} \sum_{e=(u,v)\in\mathcal{E}} \kappa_e \left(W_k(u,v) + W_k(v,u)\right),$$

i.e., the sum (over all users) of per-link unit costs weighted by the traffic routed on each edge in both directions. Compared to TEAVAR approaches, AEGIS-A consumes a similar amount of bandwidth and total cost compared to Tea3, and 18% to 31% less bandwidth and 21% to 38% less cost compared to TeaA. These results indicate that, compared to AEGIS-O, AEGIS-A successfully reduces network resource usage by eliminating cycles in working flows, thereby achieving more resource-efficient routing strategies. AEGIS-A also holds a competitive edge over TEAVAR approaches in terms of resource consumption, while maintaining competitive throughput and CVaR performance.

2) Running Time: **Real-world network topologies.** Table. II shows the running time and the performance of TeaA and AEGIS over two real-world network topologies. Compared to TeaA, AEGIS-O requires around 5500 times less running time for the Uninett network and around 87 times less running time for the US Carrier network, achieving at least the same performance in terms of the CVaR of total throughput loss. Moreover, compared to TeaA, AEGIS-A

TABLE II
RUNNING TIME AND PERFORMANCE EVALUATION FOR UNINETT AND US CARRIER NETWORK TOPOLOGIES

β	Approach	Uninett		US Carrier	
		Time(s)	CVaR	Time(s)	CVaR
0.8	AEGIS-O	0.70	2.375	2.89	1.805
	AEGIS-A	9.56	2.375	40.76	1.969
	TeaA [10]	4076.46	2.375	256.09	1.969
0.9	AEGIS-O	0.71	2.750	2.89	2.086
	AEGIS-A	9.54	2.750	40.86	2.391
	TeaA [10]	3918.63	2.750	254.96	2.391
0.95	AEGIS-O	0.69	3.000	2.89	2.297
	AEGIS-A	9.39	3.000	40.39	2.531
	TeaA [10]	3948.77	3.000	253.54	2.531

requires around 420 times less running time for the Uninett network, and around 6 times less running time for the US Carrier network, to achieve comparable performance in terms of CVaR for total throughput loss. This indicates that AEGIS does not rely on pre-determined paths and can perform well for the CVaR of total throughput loss in a much shorter time compared to TEAVAR. Note that we exclude the initial path pre-computing time from the runtime reported for TeaA.

Synthetic network topologies. To further evaluate the scalability and performance of our proposed approaches, we conduct experiments on both Waxman and 6×6 grid network topologies. For the Waxman networks, we increase the number of nodes from 100 to 240. For both Waxman and 6×6 grid network topologies, we vary the number of SRLGs from 3 to 6 to simulate increasingly complex failure models. The results are summarized in Tables III and IV. Across all test cases, AEGIS-O consistently yields the lowest CVaR, validating its optimality under the CVaR objective. AEGIS-A achieves comparable performance while maintaining resource efficiency. Notably, AEGIS-A remains tractable as the number of nodes and SRLGs increases, with runtimes staying within a practical range even at 240 nodes and 6 SRLGs. Meanwhile, the baseline TeaA becomes computationally challenging due to its exponential path enumeration, and in some Waxman cases (beyond 120 nodes), fails to return results within the allowed runtime limit. TeaA's significantly high computing time consumption can be attributed to the specific LP formation of TEAVAR, where there are as many terms as the number of paths [10]. For instance, in 6×6 grid topology, there are more than 1.26 million acyclic paths from the source to the destination, resulting in a huge number of terms for the LP. Generally, the number of paths increases exponentially with the network size. As such, the solving time for the LP of TEAVAR also increases exponentially. In summary, these results confirm that both AEGIS algorithms exhibit scalable and resilient performance across diverse network settings.

VI. Related Work

A. Resilient Routing

Most conventional resilient routing solutions predominantly concentrate on TE-based techniques. In these TE-based approaches, paths are pre-determined, where each path under normal conditions is supplemented by one or more backup paths to safeguard against various failure situations. References [23] and [27] introduced the concept of path sensitivity, which characterizes the impact of traffic bursts and network component failures on the path. Reference [19] proposed a novel approach to TE by providing varying robustness enhancements customized according to the distinct traffic characteristics of various source-destination pairs. Reference [11] ensured robust TE performance by introducing a novel metric called the TE performance-centric ratio to assess the relevance of different path programmability values. Nevertheless, these TE-based approaches highly rely on pre-computed path information. That is to say, the number of paths will exponentially increase as the scale of the network topology becomes more extensive, which will significantly increase the computational complexity when computing the routing strategy. Along a different research line, there are some existing resilient routing works based on robust optimization; Acharya et al. [4] identified that dedicated path-based protection often exceeds the necessary measures for attaining reliability objectives and introduced the first flow-based reliability mechanism. This work was subsequently expanded by Zhang et al. [35] to include differential delay constraints. However, the approaches mentioned above always provide over-conservative routing strategies. Instead of directly reasoning about network availability and failures in deterministic settings, TEAVAR [10] leverages a probabilistic failure model. It aims to minimize the CVaR of the network throughput loss when network failure happens, which strikes a proper balance between network availability and utilization.

B. Probabilistic Risk Management in Networks

Risk management in networks exhibiting stochastic behavior has been extensively examined in the literature across various contexts, such as transportation and wireless networks. The methodologies to address uncertainty encompass stochastic optimization [6], [7], [13]. For instance, [22] proposes a TE framework that accounts for demand uncertainty, contrasting with our focus on network failures, with risk quantified by standard deviation. Additionally, [8] investigates the impact of demand fluctuations, suggesting a TE approach that modifies network topology in the long term while re-routing traffic in the short term. Shoofly [26] develops a fast and low-redundancy failure recovery framework by minimizing the hardware cost of backup deployment across SRLG failures. FLAIR [36] constructs a probabilistic failure-aware routing framework that utilizes probabilistic models to establish fast and low-redundancy backup tunnels. Furthermore, a substantial body of literature exists on Conditional Value at Risk (CVaR) following the pioneering work in [24]. Considering uncertain link availability, Boginski et al. [9] apply CVaR to the minimum-cost flow problem. As mentioned earlier, the recent work, TEAVAR [10], proposes a TE scheme that utilizes the CVaR approach to balance utilization and availability effectively. Inspired by this, AEGIS takes a step further by minimizing the CVaR of the network throughput loss, subject to various network requirements and considering additional

TABLE III
RUNNING TIME AND PERFORMANCE EVALUATION FOR 6 × 6 GRID NETWORK OVER DIFFERENT NUMBERS OF SRLGS

Size	β	Approach	3 SRLGs		4 SRLGs		5 SRLGs		6 SRLGs	
			Time(s)	CVaR	Time(s)	CVaR	Time(s)	CVaR	Time(s)	CVaR
6×6	0.80	AEGIS-O	0.16	0.532	0.41	1.592	0.92	1.867	3.11	2.214
		AEGIS-A	2.86	0.532	4.24	1.791	13.77	1.966	33.98	3.001
		TeaA [10]	3016.67	0.532	3522.02	1.592	4546.30	1.901	6678.91	2.314
	0.90	AEGIS-O	0.16	0.623	0.39	1.622	0.94	1.993	3.50	2.388
		AEGIS-A	3.12	0.623	4.01	1.828	13.03	2.047	34.76	3.260
		TeaA [10]	3060.43	0.623	3497.43	1.622	4556.63	1.993	6734.45	2.399
	0.95	AEGIS-O	0.17	0.623	0.36	1.622	0.91	1.993	3.42	2.415
		AEGIS-A	2.56	0.623	3.96	1.828	12.58	2.529	27.68	3.323
		TeaA [10]	3018.71	0.623	3496.26	1.622	4561.42	1.993	6682.90	2.415

TABLE IV
RUNNING TIME AND PERFORMANCE EVALUATION FOR THE WAXMAN MODEL OVER DIFFERENT SIZES AND NUMBERS OF SRLGS

Size	β	Approach	3 SRLGs		4 SRLGs		5 SRLGs		6 SRLGs	
			Time(s)	CVaR	Time(s)	CVaR	Time(s)	CVaR	Time(s)	CVaR
$n = 100$	0.80	AEGIS-O	0.15	0.771	0.29	0.800	0.59	0.845	1.19	0.916
		AEGIS-A	1.76	0.771	3.55	0.800	7.52	0.845	17.06	0.916
		TeaA [10]	1.32	0.771	1.46	0.800	1.70	0.845	2.22	0.916
	0.90	AEGIS-O	0.14	1.980	0.30	2.117	0.59	2.335	1.17	2.617
		AEGIS-A	1.84	1.980	3.65	2.117	7.67	2.335	15.55	2.617
		TeaA [10]	1.39	1.980	1.52	2.117	1.76	2.335	2.31	2.617
	0.95	AEGIS-O	0.15	2.215	0.29	2.490	0.58	2.617	1.19	2.617
		AEGIS-A	1.77	2.215	3.57	2.490	7.10	2.617	15.64	2.617
		TeaA [10]	1.41	2.215	1.51	2.490	1.78	2.617	2.31	2.617
$n = 120$	0.80	AEGIS-O	0.20	0.059	0.39	0.059	0.76	0.059	1.63	0.059
		AEGIS-A	1.98	0.059	4.36	0.059	8.92	0.062	25.79	0.062
		TeaA [10]	356.77	0.059	381.69	0.059	431.67	0.059	533.44	0.059
	0.90	AEGIS-O	0.20	1.829	0.39	1.829	0.83	1.829	1.84	1.829
		AEGIS-A	2.58	1.829	5.50	1.829	11.27	1.829	31.18	1.829
		TeaA [10]	469.41	1.829	500.73	1.829	565.89	1.829	699.19	1.829
	0.95	AEGIS-O	0.20	2.046	0.39	2.046	0.83	2.046	2.04	2.046
		AEGIS-A	2.48	2.046	5.34	2.046	12.15	2.046	31.66	2.046
		TeaA [10]	470.11	2.046	501.62	2.046	565.21	2.046	695.74	2.046
$n = 180$	0.80	AEGIS-O	0.48	1.715	0.95	1.749	2.99	1.749	9.63	1.749
		AEGIS-A	7.63	1.715	15.31	1.749	54.79	1.764	223.64	1.764
		TeaA [10]	-	-	-	-	-	-	-	-
	0.90	AEGIS-O	0.48	4.093	0.98	4.161	2.53	4.161	10.23	4.161
		AEGIS-A	7.52	4.093	13.55	4.161	69.67	4.161	274.03	4.161
		TeaA [10]	-	-	-	-	-	-	-	-
	0.95	AEGIS-O	0.47	4.580	0.94	4.716	2.72	4.716	10.51	4.716
		AEGIS-A	8.98	4.580	14.05	4.716	51.11	4.716	238.41	4.716
		TeaA [10]	-	-	-	-	-	-	-	-
$n = 240$	0.80	AEGIS-O	0.75	1.228	1.85	1.273	5.18	1.273	29.88	1.273
		AEGIS-A	16.73	1.228	44.03	1.273	214.26	1.273	3070.07	1.273
		TeaA [10]	-	-	-	-	-	-	-	-
	0.90	AEGIS-O	0.87	2.993	1.79	3.150	7.43	3.150	23.79	3.150
		AEGIS-A	19.94	2.993	54.67	3.150	268.36	3.150	2267.76	3.150
		TeaA [10]	-	-	-	-	-	-	-	-
	0.95	AEGIS-O	0.80	3.150	1.93	3.150	6.52	3.150	28.60	3.150
		AEGIS-A	18.83	3.150	53.96	3.150	263.50	3.150	2673.36	3.150
		TeaA [10]	-	-	-	-	-	-	-	-

user budget constraints. While we also consider resource-efficient routing, this is further addressed in our cycle-free algorithm AEGIS-A. Recent years have seen a surge in Machine Learning (ML)-augmented TE solutions that seek to improve availability by learning from historical failure traces. For instance, [31] builds graph neural networks to capture WAN connectivity and network flows, while [21] leverages fine-grained optical telemetry to anticipate link degradations via neural predictors, and [5] presents a neural model for TE explicitly capable of handling variations in topology, including those not observed in training. However, these ML-based TE approaches typically rely on path-based formulations, where the routing decision is restricted to selecting from a limited set of candidate or learned paths. Compared to the above TE approaches, AEGIS does not rely on pre-computed path information from TE and maintains the routing flexibility.

VII. CONCLUSION AND FUTURE WORK

This study addresses the challenge of achieving throughput-guaranteed resilient routing under network failures without relying on pre-computed paths. We introduce AEGIS, a novel resilient routing scheme that leverages a CVaR approach to proactively ensure throughput guarantees under normal conditions while enabling failure recovery. Specifically, we propose an optimization problem that minimizes the CVaR of total throughput loss across all the failure situations while respecting user budget and network constraints. The above optimization problem is non-differentiable and non-linear; we transform it into an equivalent LP and develop an optimal polynomial-time solution. Furthermore, we address the issue of cyclic flows by introducing a bisection approach that obtains the CVaR upper bound, leading to acyclic and more resource-efficient routing. Extensive numerical evaluations highlight trade-offs among different approaches and demonstrate the superiority of `AEGIS`.

In summary, the two proposed algorithms, `AEGIS-O` and `AEGIS-A`, offer complementary advantages tailored to different network scenarios and deployment priorities. `AEGIS-O` is ideal for highly dynamic and adversarial environments where maximizing resilience is critical. It achieves superior CVaR performance at the cost of extra resource usage. In contrast, `AEGIS-A` eliminates cycles using a bisection approach, resulting in acyclic and more resource-efficient routing, making it preferable in cost- or energy-sensitive settings. Theoretical guarantees and empirical results collectively suggest that `AEGIS-O` is preferable when worst-case performance is the primary concern, while `AEGIS-A` is more suitable when resource efficiency is prioritized. Together, these solutions provide a flexible framework for resilient routing under diverse application scenarios.

However, both approaches inherit certain limitations. First, although our LP-based formulation admits polynomial-time solutions with respect to the input size and demonstrates practical efficiency in the evaluated topologies, the input itself may still grow substantially when the failure model requires a large number of explicitly represented failure situations. This can create scalability challenges in large-scale or highly dynamic networks. Second, the current framework relies on a probabilistic risk model constructed from a predefined set of SRLG-induced failure situations. While this is a standard and useful abstraction, in practice, the probabilities of individual failures and their combinations may be difficult to estimate accurately and may vary over time. Moreover, the independence assumption used to construct the joint failure distribution may not always hold, and cascading or strongly correlated failures are not explicitly modeled in the present formulation.

There are several interesting questions and directions for future work. Firstly, an important direction for future work is to refine the current model by representing the failure situation set $\mathcal{Q}$ implicitly using binary decision variables and linear constraints. Such a formulation would enable joint optimization of both routing strategies and failure patterns within a mixed-integer programming (MIP) framework. By encoding structured failure models (e.g., SRLG overlaps or bounded network failures) directly into the constraints, one can avoid explicitly enumerating all situations in $\mathcal{Q}$, which may grow exponentially with the number of SRLGs. In a nutshell, this extension would enable the use of risk-aware objectives, such as CVaR, within a compact, flexible optimization framework, offering a promising avenue to improve the scalability and expressiveness of resilient routing under uncertainty. Secondly, it is important to improve the fidelity and robustness of the underlying risk model. In the current framework, the joint distribution of failure situations is constructed from a predefined set of SRLG-induced events under simplifying assumptions. In practice, however, failure probabilities may be estimated imperfectly or drift over time. It is therefore important to study how sensitive the proposed routing strategies are to such modeling errors Thirdly, while `AEGIS` currently addresses static traffic demands and single-shot recovery scenarios, practical networks often encounter temporally correlated failures and dynamic recovery needs. A promising extension is to incorporate Markovian models of failure evolution, where the network transitions between failure states according to a probabilistic process. This enables the design of multi-stage recovery strategies that proactively reserve resources or reroute flows in anticipation of future failures. Fourthly, introducing a new mission-critical commodity into the network may incur migration costs. The development of a resilient routing scheme to accommodate a newly introduced commodity remains unclear. Another possible extension is to consider fairness across different network users, as each commodity's throughput guarantee level would be different and needs to be carefully quantified. Lastly, we will extend the proposed routing scheme to the setting where communication is wireless.

Acknowledgment

The work of Zhaofeng Zhang was primarily done as a Post-Doctoral Researcher with Arizona State University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] *The Internet Topology Zoo*. Accessed: Apr. 27, 2026. [Online]. Available: https://topology-zoo.org/
[2] *GARR*. Accessed: Apr. 27, 2026. [Online]. Available: https://www.garr.it/en/
[3] *Uninett*. Accessed: Apr. 27, 2026. [Online]. Available: https://snl.no/Uninett
[4] S. Acharya, B. Gupta, P. Risbood, and A. Srivastava, "PESO: Low overhead protection for Ethernet over SONET transport," in *Proc. IEEE INFOCOM*, vol. 1, Mar. 2004, pp. 165–175.
[5] A. A. AlQiam et al., "Transferable neural WAN TE for changing topologies," in *Proc. ACM SIGCOMM Conf.*, Aug. 2024, pp. 86–102.
[6] R. Banner and A. Orda, "The power of tuning: A novel approach for the efficient design of survivable networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 737–749, Aug. 2007.
[7] D. Bertsimas and M. Sim, "Robust discrete optimization and network flows," *Math. Program.*, vol. 98, nos. 1–3, pp. 49–71, Sep. 2003.
[8] Y. Bi and A. Tang, "Uncertainty-aware optimization for network provisioning and routing," in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2019, pp. 1–6.
[9] V. L. Boginski, C. W. Commander, and T. Turko, "Polynomial-time identification of robust network flows under uncertain arc failures," *Optim. Lett.*, vol. 3, no. 3, pp. 461–473, Jul. 2009.
[10] J. Bogle et al., "TEAVAR: Striking the right utilization-availability balance in WAN traffic engineering," in *Proc. ACM Special Interest Group Data Commun.*, Aug. 2019, pp. 29–43.

[11] S. Dou, L. Qi, J. Wang, and Z. Guo, "EPIC: Traffic engineering-centric path programmability recovery under controller failures in SD-WANs," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 4871–4884, Dec. 2024.

[12] D. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton, NJ, USA: Princeton University Press, 2010.

[13] G. D. Glockner, G. L. Nemhauser, and C. A. Tovey, "Dynamic network flow with uncertain arc capacities: Decomposition algorithm and computational results," *Comput. Optim. Appl.*, vol. 18, no. 3, pp. 233–250, Mar. 2001.

[14] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, "Evolve or die: High-availability design principles drawn from Googles network infrastructure," in *Proc. ACM SIGCOMM Conf.*, Florianópolis, Brazil, Aug. 2016, pp. 58–72.

[15] C.-Y. Hong et al., "B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined WAN," in *Proc. Conf. ACM Special Interest Group Data Commun.*, New York, NY, USA, 2018, pp. 74–87.

[16] A. Liu et al., "A survey on fundamental limits of integrated sensing and communication," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 994–1034, 2nd Quart., 2022.

[17] F. Liu et al., "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, Jun. 2022.

[18] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter, "Traffic engineering with forward fault correction," in *Proc. ACM Conf. SIGCOMM*, New York, NY, USA, 2014, pp. 527–538.

[19] X. Liu, S. Zhao, Y. Cui, and X. Wang, "FIGRET: Fine-grained robustness-enhanced traffic engineering," in *Proc. ACM SIGCOMM Conf.*, Aug. 2024, pp. 117–135.

[20] N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[21] C. Miao et al., "PreTE: Traffic engineering with predictive failures," in *Proc. ACM SIGCOMM Conf.*, Sep. 2025, pp. 780–795.

[22] D. Mitra and Q. Wang, "Stochastic traffic engineering for demand uncertainty and risk-aware network revenue management," *IEEE/ACM Trans. Netw.*, vol. 13, no. 2, pp. 221–233, Feb. 2005.

[23] L. Poutievski et al., "Jupiter evolving: Transforming Google's datacenter network via optical circuit switches and software-defined networking," in *Proc. ACM SIGCOMM Conf.*, Aug. 2022, pp. 66–85.

[24] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *J. Risk*, vol. 2, no. 3, pp. 21–42, 2000.

[25] L. Shen, X. Yang, and B. Ramamurthy, "Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 918–931, Aug. 2005.

[26] R. Singh, N. Bjorner, S. Shoham, Y. Yin, J. Arnold, and J. Gaudette, "Cost-effective capacity provisioning in wide area networks with shoofly," in *Proc. ACM SIGCOMM Conf.*, 2021, pp. 534–546.

[27] M. Y. Teh, S. Zhao, P. Cao, and K. Bergman, "COUDER: Robust topology engineering for optical circuit switched data center networks," 2020, *arXiv:2010.00090*.

[28] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: Traffic engineering in dynamic networks," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Aug. 2006, pp. 99–110.

[29] B. M. Waxman, "Routing of multipoint connections," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[30] D. Xu, Y. Xiong, and C. Qiao, "Protection with multi-segments (PROMISE) in networks with shared risk link groups (SRLG)," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, 2002, vol. 40, no. 2, pp. 987–996.

[31] Z. Xu, F. Y. Yan, R. Singh, J. T. Chiu, A. M. Rush, and M. Yu, "Teal: Learning-accelerated optimization of WAN traffic engineering," in *Proc. ACM SIGCOMM Conf.*, 2023, pp. 378–393.

[32] Y. Ye, "An O(n^3L) potential reduction algorithm for linear programming," *Math. Program.*, vol. 50, no. 1, pp. 239–258, 1991.

[33] R. Yu, G. Xue, V. T. Kilari, and X. Zhang, "Deploying robust security in Internet of Things," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May 2018, pp. 1–9.

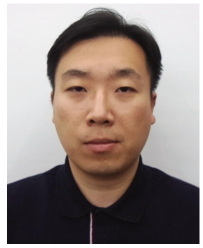
[34] R. Yu, G. Xue, Y. Wan, J. Tang, D. Yang, and Y. Ji, "Robust resource provisioning in time-varying edge networks," in *Proc. 21st Int. Symp. Theory, Algorithmic Found., Protocol Design Mobile Netw. Mobile Comput.*, Oct. 2020, pp. 21–30.

[35] W. Zhang, J. Tang, C. Wang, and S. de Soysa, "Reliable adaptive multipath provisioning with bandwidth and differential delay constraints," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

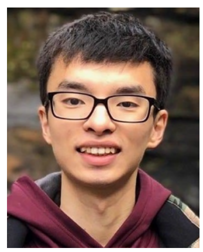
[36] Y. Zhang et al., "FLAIR: A fast and low-redundancy failure recovery framework for inter data center network," *IEEE Trans. Cloud Comput.*, vol. 12, no. 2, pp. 737–749, Apr. 2024.

[37] Z. Zhang, X. Lin, G. Xue, Y. Zhang, and K. S. Chan, "Joint optimization of task offloading and resource allocation in tactical edge networks," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2024, pp. 703–708.

[38] Z. Zhang, X. Lin, G. Xue, Y. Zhang, and K. S. Chan, "Multi-hop relay-aided task offloading for tactical edge computing," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2025, pp. 1572–1577.



Zhaofeng Zhang (Member, IEEE) received the B.Eng. degree in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015, and the M.S. and Ph.D. degrees in electrical engineering from Arizona State University, Tempe, AZ, USA, in 2016 and 2023, respectively. Currently, he is an Assistant Professor in computer science with Delaware State University, Dover, DE, USA. His research interests include edge computing, next-generation wireless networks, and machine learning.



Xuanli Lin (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in computer science from Arizona State University in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include network optimization, attack graphs, and the Internet of Things.

Guoliang Xue (Fellow, IEEE) is currently a Professor in computer science with Arizona State University. His research interests include wireless networking, security and privacy, and optimization. He received the IEEE Communications Society William R. Bennett Prize in 2019. He is an Editor-at-Large of IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. He served as the VP-Conferences for the IEEE Communications Society from January 2016 to December 2017 and the Steering Committee Chair for IEEE INFOCOM from January 2020 to December 2025. He served as an Editor for IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE TRANSACTIONS ON MOBILE COMPUTING, an Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS overseeing 12 editors in the wireless networks area, and an Area Editor for IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING overseeing 17 editors in the computer networks area.

Kevin S. Chan (Senior Member, IEEE) received the B.S. degree in electrical and computer engineering and public policy from Carnegie Mellon University, Pittsburgh, PA, USA, and the M.S. and Ph.D. degrees in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA. He is currently the Lead Research Electronics Engineer and the Network Science Team Lead with the Networks, Cyber, and Computational Sciences Division, U.S. Army Combat Capabilities Development Command, Army Research Laboratory, Adelphi, MD, USA. He is actively involved in research on network science, edge computing, and cybersecurity. He received the 2021 IEEE Communications Society Leonard G. Abraham Prize and multiple best paper awards. He also recently served as a Co-Editor for *IEEE Communications Magazine*—Military Communications and Networks Series.