Risk-Aware Dynamic Routing in Space-Terrestrial Integrated Networks

Jiayi Rong*, Yang Xu*, *Senior Member, IEEE*, Jia Liu†, *Senior Member, IEEE*,
Hiroki Takakura†, *Member, IEEE*, Tarik Taleb‡, *Senior Member, IEEE*, and Norio Shiratori§, *Life Fellow, IEEE*
*School of Computer Science and Technology, Xidian University, Xi'an, China
†Center for Strategic Cyber Resilience Research and Development, National Institute of Informatics, Tokyo, Japan
‡Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Bochum, Germany
§Tohoku University, Sendai, Japan
Email: rongjiayi@stu.xidian.edu.cn, yxu@xidian.edu.cn, jliu@nii.ac.jp,
takakura@nii.ac.jp, tarik.taleb@rub.de, norio.shiratori.e8@tohoku.ac.jp

***Abstract*—This paper proposes a dynamic routing framework for space-terrestrial integrated networks (STINs), which incorporates traffic restoration, malicious node localization, and risk-aware path selection. Specifically, the traffic restoration module employs a Satformer model with an adaptive spatiotemporal attention mechanism to capture complex nonlinear dependencies in large-scale dynamic traffic data, enabling accurate tensor-level recovery of corrupted observations on time-varying graphs. Malicious node localization is performed by comparing the ratio between observed and restored traffic row sums against a calibrated lower-quantile threshold, thereby estimating each node's risk probability and identifying compromised nodes. The derived node risk probabilities are further propagated to link-level risks, which are used to reweight edge costs in the routing layer. By executing a k-shortest path search, the algorithm achieves a balanced tradeoff between delay and packet loss. Experimental results demonstrate that the proposed routing scheme significantly reduces end-to-end packet loss while maintaining a delay close to that of the shortest path.**

***Index Terms*—Space-terrestrial integrated networks, risk-aware routing, traffic restoration, malicious node localization.**

I. Introduction

Low Earth orbit (LEO) satellite constellations are emerging as a critical complement to terrestrial networks in future 6G systems due to their global coverage and cost effectiveness [1]. However, the highly dynamic motion of satellites, time-varying inter-satellite links (ISLs) and ground-satellite links (GSLs), as well as the massive number of nodes, impose stringent requirements on monitoring, prediction, and routing in space-terrestrial integrated networks (STINs) [2]. Directly measuring traffic for all source-destination pairs is costly and often infeasible at scale, while partial observations are typically sparse, noisy, and sometimes corrupted by failures or malicious behaviors (e.g., packet-dropping attacks) [3], thereby derailing routing decisions and degrading QoS.

In this context, recent research has advanced two related directions. The first focuses on traffic estimation and reconstruction from partial measurements by learning spatiotemporal correlations over non-Euclidean topologies, where graph neural networks and attention mechanisms are leveraged to handle dynamic and sparse traffic matrices [4]–[8]. The second concentrates on dynamic routing in LEO-based networks, aiming to predict network states and adapt link weights or paths accordingly, often employing multipath strategies to hedge uncertainty [9]. Although these approaches have achieved notable progress, important challenges remain: (i) Existing traffic estimation techniques rarely establish a closed loop that feeds robust, attack-aware traffic signals back into routing decisions; and (ii) many routing algorithms rely solely on current or historical states, without explicitly correcting corrupted observations or quantifying node-level risk under adversarial conditions. These limitations can lead to congestion, retransmissions, and degraded reliability in STINs.

To address these challenges, we propose SatCure, a predicted-risk-based dynamic routing framework that unifies traffic restoration, malicious node localization, and risk-aware routing for time-varying STINs. SatCure employs a spatiotemporal encoder-decoder instantiated by Satformer blocks to reconstruct clean traffic tensors from corrupted observations, while modeling both short and long-range dependencies on dynamic graphs [4]–[6]. It then calibrates a per-node, per-time reconstruction ratio against the restored baseline to derive robust risk indicators via lower-quantile calibration [10]. Finally, SatCure performs k-shortest path routing with risk-aware edge reweighting to avoid or de-emphasize high-risk regions without sacrificing latency. Specifically, edge lengths are inflated by aggregating endpoint risks before running the k-shortest-path search, which softly steers traffic away from risky nodes and provides a tunable trade-off between delay and packet loss while maintaining path diversity.

We instantiate SatCure on a time-varying Starlink constellation with 1,584 satellites and realistic ground stations. Simulation results show that SatCure recovers clean traffic under targeted packet-dropping attacks, accurately localizes malicious nodes, and achieves a favorable delay and loss tradeoff: the risk-aware k-shortest strategy attains near shortest-path latency while substantially reducing end-to-end loss compared with classical multipath baselines.

The main contributions of this paper are summarized as follows:

- We formulate attack-aware traffic restoration for STINs and implement a Satformer-based spatiotemporal encoder-decoder that captures nonlinear correlations on dynamic graphs.
- We propose a calibrated ratio thresholding detector that converts restoration signals into per-node risk scores, providing robust, interpretable evidence of malicious behavior without needing labels during operation.
- We develop a risk-aware k-shortest-path routing scheme that softly penalizes edges adjacent to risky nodes, balancing latency and reliability and enabling rapid switchover among multiple candidates.
- We build a reproducible simulation using real satellite parameters and evaluate SatCure under targeted packet-dropping attacks, demonstrating accurate restoration and localization and clear QoS gains over representative multipath baselines.

The rest of this paper is organized as follows. Section II presents the system model and design objectives. Section III elaborates on the development of the SatCure framework. The experimental setup and results are provided in Section IV, followed by the conclusion in Section V.

II. System Model and Design Objectives

A. Network Model

We consider a typical STIN that consists of a space backbone and multiple distributed ground stations as shown in Fig. 1. Let $S = \{s_1, s_2, \ldots, s_N\}$ denote the set of all satellites in the STIN, with $|S| = N$ satellites in total. Let $U = \{u_1, u_2, \ldots, u_Q\}$ denote the set of all ground stations that connect to the space backbone for ground-space communication, and Q is the number of ground stations. Let $V = S \cup U$ denote the set of all nodes in the STIN. In addition, LEO satellites interconnect to each other via inter-satellite links (ISL), and connect to ground stations via ground-satellite links (GSL). Let (i, j) denote an available link in the STIN. In practice, both ISLs and GSLs are bidirectional, and let L denote the set of all available links. Therefore, in a failure-free scenario, an STIN can be formulated as a graph $G = (V, L)$.

In the considered STIN, the inter-satellite traffic data can be represented as a spatiotemporal matrix, which reflects the data volume to be transmitted between all node-node pairs over the space backbone. Let T denote the discrete time steps under observation and $\mathbf{X} \in \mathbb{R}^{N \times N \times T}$ denote the inter-satellite traffic tensor, where $X_{i,j,t}$ represents the data transmission from satellite i to satellite j at time step t.

B. Attack Model

We assume that malicious nodes intentionally discard received data packets, thereby introducing the impact of adversarial attacks and disrupting normal network operations. These malicious nodes can initiate different types of attacks. For instance, a black hole attack involves discarding data packets with a 100% probability upon receipt, while a selfish attack deliberately drops data packets at a certain rate to conserve energy.

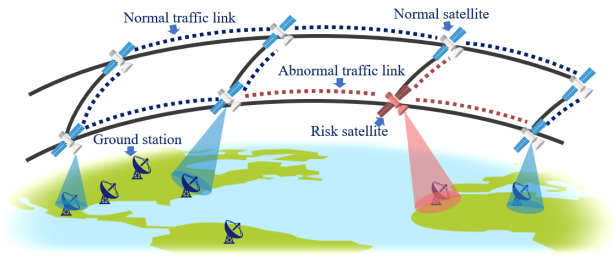


Fig. 1: System model.

We define the set of malicious nodes as $\mathbf{H}$. The scaling factor (attack function) is represented as $F_{i,j,t} \in [0, 1]$, which indicates the proportion of data packets retained after an attack. Thus, the post-attack tensor $\mathbf{X}'$ can be expressed as

$$X'_{i,j,t} = F_{i,j,t} \cdot X_{i,j,t}, \tag{1}$$

where $X'_{i,j,t}$ represents the traffic volume from satellite i to satellite j at time step t under the attack.

C. Design Objectives

Given the time-varying STIN graph $G = (V, L)$ and the inter-satellite traffic tensor $\mathbf{X} \in \mathbb{R}^{N \times N \times T}$, we consider that only the attacked observations $\mathbf{X}'$ are available during operation due to the presence of malicious nodes $\mathbf{H}$. The objective of this study is threefold: (i) recover a clean estimate $\widehat{\mathbf{X}}$ of the original traffic $\mathbf{X}$ from the corrupted $\mathbf{X}'$; (ii) identify the set of malicious nodes $\mathbf{H}_t$ at each time step t; and (iii) for any demanded source-destination pair (s, d), compute end-to-end paths with low latency and low loss that avoid or mitigate the impact of the detected malicious nodes.

1) Traffic Restoration with Spatiotemporal Modeling: Let $\mathbf{A}_t \in \{0, 1\}^{N \times N}$ denote a time-varying adjacency that reflects instantaneous connectivity or activity. Given a lookback window W, we learn a spatiotemporal predictor f_θ that maps the historical corrupted sequences and their adjacencies to a clean traffic estimate as follows:

$$\widehat{\mathbf{X}}_t = f_\theta\big(\mathbf{X}'_{t-W:t-1},\ \mathbf{A}_{t-W:t-1}\big), \quad t = W+1, \cdots, T. \tag{2}$$

Here, f_θ is instantiated by the Satformer architecture (a spatiotemporal graph neural block combined with local attention and residual refinement). The objective of training is to minimize the reconstruction loss over unattacked segments, that is,

$$\min_\theta \sum_{t \in \mathcal{T}_{\text{train}}} \left\| \widehat{\mathbf{X}}_t - \mathbf{X}_t \right\|_F^2. \tag{3}$$

2) Malicious Node Localization via Calibrated Ratio Thresholding: Let $s'_{i,t}$ and $\hat{s}_{i,t}$ denote the outgoing row sums of corrupted and predicted traffic, respectively, which can be expressed as

$$s'_{i,t} = \sum_{j=1}^{N} X'_{ijt}, \quad \hat{s}_{i,t} = \sum_{j=1}^{N} \widehat{X}_{ijt}. \tag{4}$$

We define the per-node reconstruction ratio as

$$r_{i,t} = \frac{s'_{i,t}}{\hat{s}_{i,t} + \varepsilon}, \tag{5}$$

which quantifies the relative depletion of observed traffic against the model restored baseline, where ε is a very small constant introduced to prevent the denominator from becoming 0 (set to 10^{-8} in our simulations). By comparing $r_{i,t}$ to calibrated thresholds (detailed in Section III-B), we identify the set of malicious nodes $\mathbf{H}_t$ at each time step and derive per-node risk score $p_{i,t} \in [0, 1]$ for routing.

3) Routing with Risk-aware Delay Penalization: For any source-destination pair $(s, d) \in V \times V$ and time t, let $\mathcal{P}_{s,d}(t)$ denote the set of feasible s-d paths in the instantaneous graph G_t. The malicious node localization module yields node-level risk score $p_{i,t} \in [0, 1]$ that quantifies the likelihood or severity of malicious behavior at satellite i and time t. For each path $P \in \mathcal{P}_{s,d}(t)$, we quantify propagation delay and end-to-end packet loss as follows.

- Propagation delay:

$$D(P, t) = \sum_{e \in P} \frac{\ell_e(t)}{v}, \qquad v \approx c, \tag{6}$$

  where $\ell_e(t)$ is the geometric length of link e at time t and v is the propagation speed along inter-satellite links.
- End-to-end packet loss:

$$L(P, t) = 1 - \prod_{i \in V(P)} \big(1 - d_{i,t}\big), \tag{7}$$

  where $d_{i,t} \in [0, 1]$ denotes the ground-truth packet loss probability introduced by the simulated attack at satellite i and time t.

Instead of hard-removing nodes before routing, we adopt a soft risk-aware path metric that inflates link lengths proportionally to the risk of their incident nodes. Let

$$p_{e,t} = \begin{cases} \max\{p_{u,t}, p_{v,t}\}, \\ \frac{p_{u,t}+p_{v,t}}{2}, \end{cases} \qquad \text{for } e = (u, v) \tag{8}$$

and define the effective (risk penalized) length as

$$\ell_e^{\text{eff}}(t) = \ell_e(t)\,\big(1 + \lambda\, p_{e,t}\big), \qquad \lambda \geq 0. \tag{9}$$

We then select paths by minimizing the total effective length (equivalently, effective delay), that is

$$\min_{P \in \mathcal{P}_{s,d}(t)} J_\lambda(P, t) = \sum_{e \in P} \ell_e^{\text{eff}}(t). \tag{10}$$

III. Design of SatCure Routing Framework

Fig. 2 illustrates the overall workflow of our proposed SatCure framework. SatCure first takes as input the dynamic STIN topology and the attacked traffic window $\mathbf{X}'$ over the past W time slots. Then, a reconstruction network equipped with spatiotemporal attention and graph convolution restores a clean traffic tensor $\widehat{\mathbf{X}}$. Next, the malicious node localization module compares the observed and reconstructed row sums,

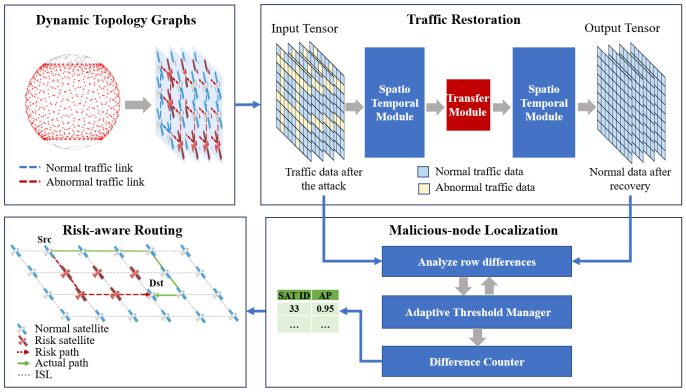
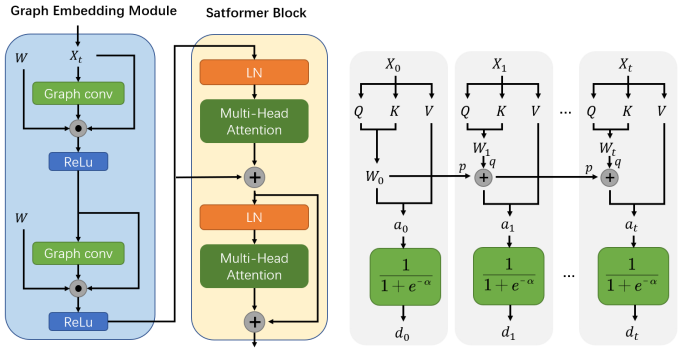


Fig. 2: Workflow of the SatCure framework.



(a) Spatiotemporal module. (b) Transfer module.

Fig. 3: Structure of Satformer modules.

applies a node-wise calibrated quantile, and generates time-varying risk scores. Finally, the routing module computes k-shortest paths on a risk-weighted graph, where each link weight is inflated in proportion to the aggregated risk of its incident nodes, thereby smoothly steering traffic away from high-risk regions while preserving path diversity. The details of each module are presented in the following subsections.

A. Traffic Restoration

Satformer was originally developed as a tensor completion model for accurate and robust global traffic estimation in satellite networks. We apply Satformer in the SatCure framework with appropriate modifications. As shown in Fig. 2, the modified Satformer adopts an encoder-decoder with both components featuring multiple spatiotemporal modules connected by residual links. Each module couples a Graph Embedding (GCN) with a Satformer Block that performs adaptive spatiotemporal attention to extract informative patterns from large traffic tensors. A Transfer Module bridges the encoder and decoder to preserve long-range temporal context, enabling the decoder to reconstruct missing or corrupted entries. Module details (spatiotemporal module and transfer module) are presented in Fig. 3.

1) Spatiotemporal Module: Each Spatiotemporal module first processes the topology information via a graph embedding

module, which contains two graph convolutional (Graph Conv) layers. Its output is then passed to a Satformer block.

a) Graph Embedding (GCN): Given the time-varying adjacency A and self-connections matrix I, we define

$$\tilde{A} = A + I, \qquad \tilde{D} = \sum_j \tilde{A}_{ij}. \tag{11}$$

A two-layer normalized GCN propagates as

$$H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\,H^{(l)}W^{(l)}\Big), \tag{12}$$

and yields the embedding of a traffic slice $\mathcal{X}$ under A:

$$\begin{aligned}\mathcal{Z} &= f(\mathcal{X}, A)\\ &= \sigma\Big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\,\mathrm{ReLU}\big(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\,\mathcal{X}W^{(0)}\big)\,W^{(1)}\Big),\end{aligned} \tag{13}$$

where $W^{(0)}, W^{(1)}$ are learnable weights and $\sigma(\cdot)$ is an activation.

b) Satformer Block: We partition $\mathcal{Z}$ into local regions $\mathcal{Z}_{\mathrm{div}}$ and, for each region, compute the query (Q), key (K), and value (V) tensors via linear projections:

$$Q = \mathcal{Z}_{\mathrm{div}}W^q, \qquad K = \mathcal{Z}_{\mathrm{div}}W^k, \qquad V = \mathcal{Z}_{\mathrm{div}}W^v. \tag{14}$$

Here, W^q, W^k, and W^v denote the projection weights for Q, K, and V, respectively.

The local attention in each region i is implemented using a mask matrix Ψ, which is given by

$$\alpha_i = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{C}} \odot \Psi\right), \tag{15}$$

Then, the final output α_t can be calculated as

$$\begin{aligned}\alpha_{s_i} &= \mathrm{softmax}\big(W^s\,\mathrm{ReLU}(1 - W^r\alpha_i) \odot V\big),\\ \alpha_t &= \mathrm{concat}(\alpha_{s_i}).\end{aligned} \tag{16}$$

Here, W^r is the weighted matrix of L1 regularization, W^s is the scaling matrix, both W^r and W^s are trainable parameters. C is a scaling constant, and $\odot$ denotes elementwise product.

2) Transfer Module: Let $E = [e_1, \ldots, e_T]$ denote the encoder outputs. Before decoding, we reweight the full sequence with self-attention as follow:

$$Q = EW^Q, \qquad K = EW^K, \qquad V = EW^V, \tag{17}$$

$$\alpha_{t,t-1} = \mathrm{softmax}\left(p\sum_{i=1}^{t-1}\frac{Q_iK_t^\top}{\sqrt{C_i}} \;+\; qQ_t\frac{K_t^\top}{\sqrt{C_t}}\right)V_t, \tag{18}$$

Specifically, p adjusts the aggregated attention responses from previous time steps $\{1, \ldots, t-1\}$, while q controls the contribution of the current time step t.

$$d_t \;=\; \frac{1}{1 + e^{-\alpha_{t,t-1}}}. \tag{19}$$

Collecting $\{d_t\}_{t=1}^T$ gives $D = [d_1, \ldots, d_T]$, which supplies long-range temporal context to the decoder.

B. Malicious Node Localization

Given the restored traffic $\widehat{\mathbf{X}}$ from the traffic restoration module, the malicious node localization module operationalizes Eqs. (4)-(5) by contrasting per-node row sums to obtain reconstruction ratios $r_{i,t}$.

1) Threshold Calibration: Before testing, we calibrate node-specific thresholds on unattacked training data. Using the trained model f_θ, we run inference on the clean training set $\mathbf{X}_{\text{train}}$ to collect normal-regime ratios:

$$\mathcal{R}_i^{\text{normal}} = \{r_{i,t} : t \in \mathcal{T}_{\text{train}}\}. \tag{20}$$

For each node i, we compute the lower α-quantile as its threshold:

$$\tau_i = \mathrm{Quantile}_\alpha(\mathcal{R}_i^{\text{normal}}). \tag{21}$$

We use $\alpha = 0.05$ (5th percentile) because malicious nodes exhibit *abnormally low* ratios due to packet dropping.

2) Test Time Detection: At the test period, we define a continuous, time-varying risk score $p_{i,t} \in [0, 1]$ for each node i. This score quantifies the severity of the deviation from normal behavior by measuring how far the observed traffic ratio $r_{i,t}$ falls below its calibrated lower-quantile threshold τ_i.

Specifically, the risk score is calculated as

$$p_{i,t} = \max\left(0, 1 - \frac{r_{i,t}}{\tau_i}\right). \tag{22}$$

This formulation has the following intuitive properties:

- If the observed ratio $r_{i,t}$ is normal or better (i.e., $r_{i,t} \geq \tau_i$), the fraction $\frac{r_{i,t}}{\tau_i}$ is 1 or greater, making $p_{i,t} = 0$. This correctly assigns zero risk to normally behaving nodes.
- If the observed ratio $r_{i,t}$ drops to the threshold level (i.e., $r_{i,t} = \tau_i$), the risk score $p_{i,t}$ is still 0, representing the boundary of normal behavior.
- If the node is malicious and drops packets, causing $r_{i,t}$ to fall below τ_i, the risk score $p_{i,t}$ increases. In the most extreme case of a black hole attack where $r_{i,t} = 0$, the risk score becomes $p_{i,t} = 1$, indicating maximum risk.

The resulting set of continuous, time-varying scores $\{p_{i,t}\}$ provides a more nuanced input for the risk-aware routing algorithm (Algorithm 1), allowing it to differentiate between slightly suspicious and highly malicious nodes.

C. Routing Algorithm

In satellite networks, SPF routing that uses only geometric distance can inadvertently traverse high-risk nodes. To address this, we adopt a risk-aware k-shortest-path routing strategy (k-RA) that softly penalizes paths passing through high-risk regions, thereby prioritizing reliability while maintaining low latency and providing multiple candidate routes.

As summarized in Algorithm 1, k-RA consists of two functions. The weighting function $w(\cdot)$ fuses the physical link lengths with predicted node risks to produce a risk weighted edge matrix: it (i) optionally hard-prunes nodes whose risk exceeds a threshold, (ii) computes edge risk by aggregating endpoint risks (maximum or mean), and (iii) inflates each edge

Algorithm 1: Risk-aware k-shortest-path routing

Input: Instantaneous graph $G_t = (V, E_t)$ with edge lengths $L_e(t)$;
Node risk scores $\{p_{i,t} \in [0,1] \mid i \in V\}$ for time t;
Risk coefficient $\lambda \geq 0$; optional threshold τ;
Source-destination pair (s, d); number of paths k.
Output: Path set $\mathcal{P}$ with up to k risk-aware paths.

1 if τ *is specified* **then**
  Remove nodes i with $p_{i,t} \geq \tau$ from G_t
end
2 foreach *edge* $e = (u, v) \in E_t$ **do**
  $p_{e,t} \leftarrow \max(p_{u,t}, p_{v,t})$ or $\frac{p_{u,t}+p_{v,t}}{2}$
  $W_e(t) \leftarrow L_e(t) \cdot (1 + \lambda \cdot p_{e,t})$
end
3 Initialize $\mathcal{P} \leftarrow \emptyset$
4 Compute k shortest paths from s to d using weight $W_e(t)$
5 foreach *path* π *found* **do**
  Compute physical length: $L(\pi) = \sum_{e \in \pi} L_e(t)$
  Add $(\pi, L(\pi))$ to $\mathcal{P}$
end
6 return $\mathcal{P}$

length with a risk coefficient λ. The routing function $r(\cdot)$ then runs a k-shortest simple paths search on the weighted graph, selecting paths by the minimum sum of effective weights to form the path set P and derive the routing table R.

When $\lambda = 0$, it reduces to standard shortest-path routing, while as λ increases, it more strongly avoids risky nodes, achieving a tunable tradeoff between end-to-end delay and packet loss.

IV. Simulation Results

We build a Python-based simulation for a time-varying Starlink constellation, with the main parameters summarized in Table I. We generate corresponding traffic datasets using real satellite parameters and ground station coordinates. Satformer instantiates the predictor f_θ for traffic restoration. For the model's architecture, we configure the hidden size to 128 and employ 16 local attention heads. The dataset comprises $T = 200$ time steps, which we partition into 100 steps for training and 100 for testing.

TABLE I: Simulation parameters.

Parameter	Value
Number of satellites	1584 (Starlink)
Number of ground stations	910
Polar boundary latitude	80°
Bandwidth (link capacity)	1024 Mbps
Simulation time	200 s
Traffic data collection interval	1 s

To evaluate robustness under targeted impairments, we inject an attack as follows. We compute two ranking-based indicators over the constellation: (i) the top-400 satellites with the largest cumulative number of established connections, and (ii) the top-400 satellites with the longest cumulative service time. We then take the intersection of these two sets as the attack set. For every satellite in this intersection, we reduce all outgoing traffic rates by 80% during the evaluation period, thereby degrading its forwarding capability while leaving incoming traffic unchanged.

We compare our risk-aware k-shortest-paths (k-RA) with five multipath baselines:

- **k-SP** *Shortest paths:* This strategy load-balances traffic among the k-shortest source–destination (s–d) paths.
- **k-DN** *Node-disjoint satellite-to-satellite Paths:* This strategy only considers the k shortest paths that are node-disjoint.
- **k-DI** *ISL-disjoint satellite-to-satellite Paths:* This strategy is similar to the above, with the exception that disjointness is enforced exclusively on the ISLs.
- **k-LO** *Limited-overlap shortest paths:* We implement the ESX algorithm by Chondrogiannis et al. [11] to find the shortest k paths with a similarity score of no more than 50%.
- **k-HA** *Hard-avoid shortest-paths:* An ablation of our method in which risk nodes are completely removed before running the k-sp algorithm.

After 300 training epochs, we visualize the Satformer predictions on the last batch of the test set. Fig. 4 shows the traffic sent from the source satellite with ID 1584 to all 1,584 destination nodes at the 10th time step. The ground-truth curve is red and the Satformer prediction is blue. Overall, the blue curve follows the red curve's oscillatory, sparse pattern well, reproducing most spike timings and extended near-zero intervals, which indicates the model's ability to capture the global trend across destinations.

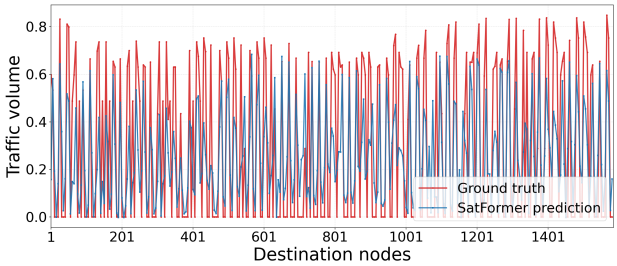
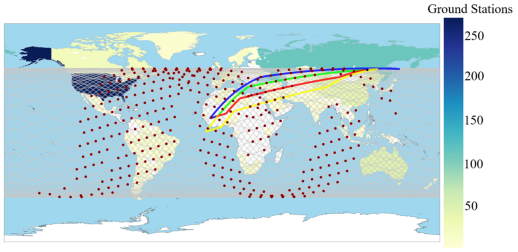
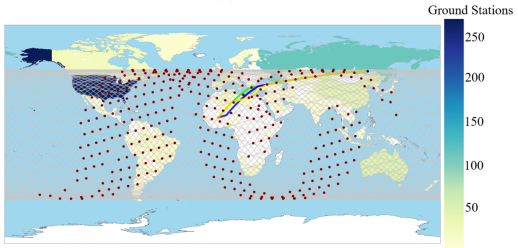


Fig. 4: Traffic comparison across destination nodes.

Fig. 5 compares k-DN and k-RA for one source-destination pair in the global constellation. The map shows ground-station density, e.g., 268 stations in the USA; brown, blue, and orange dots indicate malicious, source, and destination satellites, respectively, and four alternative paths (k = 4) are shown in green, red, blue, and yellow. k-DN enforces node-disjoint paths, which improves fault isolation but still routes some paths through high-risk (red) satellites, increasing loss. In contrast, k-RA applies soft penalties to risky nodes via edge reweighting, steering paths around malicious regions while remaining close to geometric



(a) k-DN



(b) k-RA

Fig. 5: Comparison of routing strategies under the same source-destination pair.

shortest routes. Allowing limited path overlap, k-RA achieves markedly lower packet loss with comparable or better end-to-end delay.

Fig. 6 and Fig. 7 report packet-loss and end-to-end-delay CDFs under the same offered load. By softly penalizing risky nodes, k-RA achieves a favorable delay-loss tradeoff, whereas k-HA achieves low loss but can be brittle as the risky set grows.

In Fig. 6, the four classical baselines (k-SP, k-DN, k-DI, k-LO) cluster together: only about one third of flows are loss-free (CDFs at rate 0 are $\approx$ 0.3-0.4), with noticeable mass at moderate loss. In contrast, k-RA concentrates probability mass near zero loss: roughly 0.85-0.9 of flows incur no loss, and its CDF strictly dominates the baselines across the range, indicating consistently lower end-to-end loss. k-HA also has very low loss by removing risky nodes a priori, but becomes less robust as the risky set grows; thus it is an ablation rather than the default.

In Fig. 7, at low traffic regimes where queuing is negligible, k-SP achieves the smallest delays by strictly minimizing path length. k-RA closely tracks k-SP at the median (substantial overlap of the curves) and shows a shorter tail at high percentiles, reflecting the benefit of steering away from risky nodes that tend to cause retransmissions or local congestion. k-DN exhibits the largest delays because node-disjointness forces detours even when they are long; k-DI and k-LO fall between k-SP and k-DN due to partial overlap constraints. In summary, k-RA provides near shortest-path latency while significantly reducing packet loss, yielding a better overall QoS than the classical baselines.

V. Conclusion

SatCure combines traffic restoration, malicious-node localization, and risk-aware routing to reduce loss with near-shortest-path latency under targeted attacks.

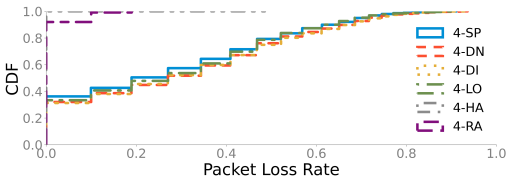


Fig. 6: Comparison of the packet loss CDF under different routing strategies.

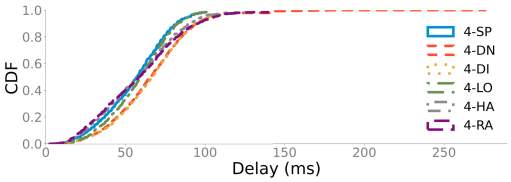


Fig. 7: Comparison of the delay CDF under different routing strategies.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62372361; in part by JSPS KAKENHI Grant Number JP25K15087; in part by the Fundamental Research Funds for the Central Universities under Grant No. QTZX26149; in part by the Project of Cyber Security Establishment with Inter-University Cooperation; and in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY24F020011.

References

[1] O. Kodheli *et al.*, "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 70–109, 2021.

[2] D. Zhou, M. Sheng, J. Li, and Z. Han, "Aerospace integrated networks innovation for empowering 6g: A survey and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 975–1019, 2023.

[3] Y. Zhang *et al.*, "Building an efficient satellite network routing scheme: Challenges and opportunities," *IEEE Commun. Standards Mag.*, vol. 9, no. 2, pp. 30–38, 2025.

[4] L. Qin, X. Liu, W. Wei, L. Chengbin, and H. Gu, "Satformer: Accurate and robust traffic data estimation for satellite networks," in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2024, pp. 47 530–47 558.

[5] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learning Representations*, 2018.

[6] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

[7] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 753–763.

[8] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[9] M. Handley, "Delay is not an option: Low latency routing in space," in *Proc. 17th ACM Workshop Hot Topics Netw.*, 2018, pp. 85–91.

[10] A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster, "Conformal risk control," *arXiv preprint arXiv:2208.02814*, 2022.

[11] T. Chondrogiannis, P. Bouros, J. Gamper, U. Leser, and D. B. Blumenthal, "Finding k-shortest paths with limited overlap," *VLDB J.*, vol. 29, no. 5, pp. 1023–1047, 2020.