

Telemetry-Aware Risk-Adaptive Routing for Link-Flooding Resilience in Software-Defined Networks

Utham Kumar Anugula Sethupathy^{1*}, Vijayanand Ananthanarayanan²

¹Independent Researcher, Senior Member IEEE, Alumni, Nanyang Technological University, Atlanta, USA

²Independent Researcher Alumni, Farleigh Dickinson University, Atlanta, USA

Email: *mailuthamkumar@gmail.com, vijayanand31@gmail.com

How to cite this paper: Sethupathy, U.K.A. and Ananthanarayanan, V. (2026) Telemetry-Aware Risk-Adaptive Routing for Link-Flooding Resilience in Software-Defined Networks. *International Journal of Communications, Network and System Sciences*, 19, 97-112.

<https://doi.org/10.4236/ijcns.2026.197007>

Received: June 10, 2026

Accepted: July 28, 2026

Published: July 31, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Adaptive-routing evaluations can overestimate resilience when controllers are assumed to know attack locations. This study presents telemetry-aware risk-adaptive routing (TARA), in which the controller receives only delayed, noisy utilization and delay measurements processed by an exponentially weighted detector with explicit false positives, false negatives, and calibration error. Twenty-four paired synthetic 36-node backbones were evaluated across three attack loads, three detector-quality settings, three baselines, full TARA, and four structural ablations, producing 1728 trials. Under 100% target-capacity attack load and moderate detection, full TARA delivered 97.25% of demand (SD 1.20) versus 93.18% (SD 2.52) for reactive rerouting. The paired improvement was 4.07 percentage points (95% CI 3.39 to 4.74). Its 95th-percentile path latency was 27.15 ms, a paired difference of -6.47 ms (95% CI -7.94 to -5.00) relative to reactive routing. Delivery changed from 97.89% with strong detection to 96.71% with weak detection. Ablations identify path diversity as the dominant source of benefit; risk, overlap control, and utilization prediction produce comparatively modest, condition-dependent differences. These are controlled simulation results, not measurements from a production network.

Keywords

Software-Defined Networking, Link-Flooding Attack, Telemetry, Resilient Routing, Sensitivity Analysis

1. Introduction

Software-defined networking separates forwarding behavior from logically centralized control and enables systematic path changes [1]. Its programmability also

enlarges the consequences of inaccurate or delayed network state, so resilience claims must identify exactly what the controller can observe [2].

Low-rate link-flooding attacks can congest selected transit links without sending traffic directly to the victim [3]. Defenses that perturb routes can increase attacker cost, but their benefit depends on whether congestion can be inferred quickly enough [4]. Evaluations that disclose the attack set or instantaneous attack load to a routing controller risk overstating deployable resilience.

This study asks how much service protection risk-adaptive path diversity provides when attack state is hidden and the controller receives only imperfect telemetry. Attack ground truth exists only in the traffic generator and evaluator; routing decisions use noisy delayed measurements, detector output, and controller state.

The contributions are an observable telemetry and detection layer; fair information timing for all policies; detector-quality sensitivity; four structural ablations; and paired confidence intervals over fixed seeds. **Figure 1** summarizes the information boundary and the controller workflow.

Telemetry-derived risk-adaptive routing

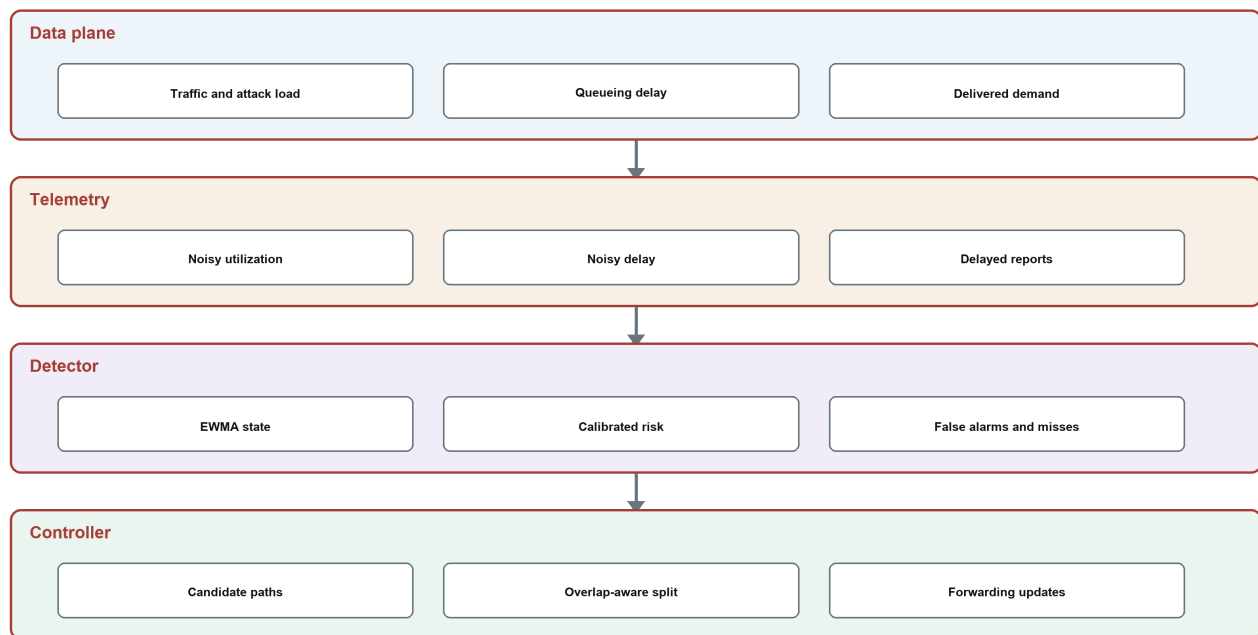


Figure 1. Information flow from generated traffic through noisy delayed telemetry to detector-derived risk and routing decisions.

2. Related Work

Software-driven wide-area traffic engineering demonstrated centralized allocation over multiple paths [5], while OSPF weight optimization established a classical link-cost baseline [6]. These systems motivate load-aware routing but do not by themselves model adversarial uncertainty.

Disjoint-path construction [7] and loopless k -shortest path enumeration [8] provide the topology-only diversity used in this study. TARA uses the same can-

didate set as its ablations, preventing candidate-generation differences from confounding the comparison.

Exponentially weighted state estimation is a standard way to smooth noisy time series [9]. Intrusion-detection guidance distinguishes sensor evidence, detection decisions, and ground truth and emphasizes false alarms and missed detections [10]. Those distinctions determine the detector model.

Congestion-control principles [11], differentiated-services architecture [12], and the end-to-end argument [13] caution against treating a controller as omniscient. Simulation studies also require explicit scope because Internet behavior is difficult to reproduce in full [14]. The present model therefore isolates route selection and telemetry quality rather than claiming Internet-scale realism.

Recent work has combined link-flooding prevention with alternate-path and topology-obfuscation mechanisms [15], measured SD-WAN delay through in-band telemetry [16], and used monitored data-plane state for reliable path prediction [17]. Scalable software-defined sensor networks likewise expose controller-update and state-prediction overheads [18]. Link verification based on observed latency [19] and credibility-oriented link-flooding mitigation [20] reinforce the importance of detection uncertainty. Control-plane resilience in SD-WANs [21] and flooding defenses in information-centric networks [22] provide complementary perspectives on failure containment and adversarial traffic.

3. System Model and Information Boundary

Figure 2 shows one generated 36-node backbone. Red links are attack targets in the evaluator. The controller never receives those labels; the coloring exists only to explain the simulated ground truth.

Table 1 separates variables available to the traffic generator, telemetry process, controller, and evaluator. This separation is enforced in the executable code: routing functions accept measured risk and utilization state but not the attack dictionary.

Table 1. Information boundary.

Component	Available state	Unavailable state
Traffic generator	Demand matrix; attack schedule	Controller estimate
Telemetry process	Actual utilization and delay for measurement generation	Routing objective
Controller	Delayed noisy measurements; detector risk; topology	Attack labels; attack load; future measurements
Evaluator	Ground truth and delivered traffic	No decision authority

For link e and epoch t , Equation (1) defines measured utilization and delay as physical values plus zero-mean detector-scenario noise.

$$\tilde{u}_{e(t)} = u_{e(t)} + \varepsilon_{e(t)}^u; \tilde{d}_{e(t)} = d_{e(t)} + \varepsilon_{e(t)}^d \quad (1)$$

Noise standard deviations are 0.025, 0.060, and 0.110 utilization units and 0.35,

0.75, and 1.25 ms for strong, moderate, and weak detection.

Representative synthetic backbone and attack ground truth

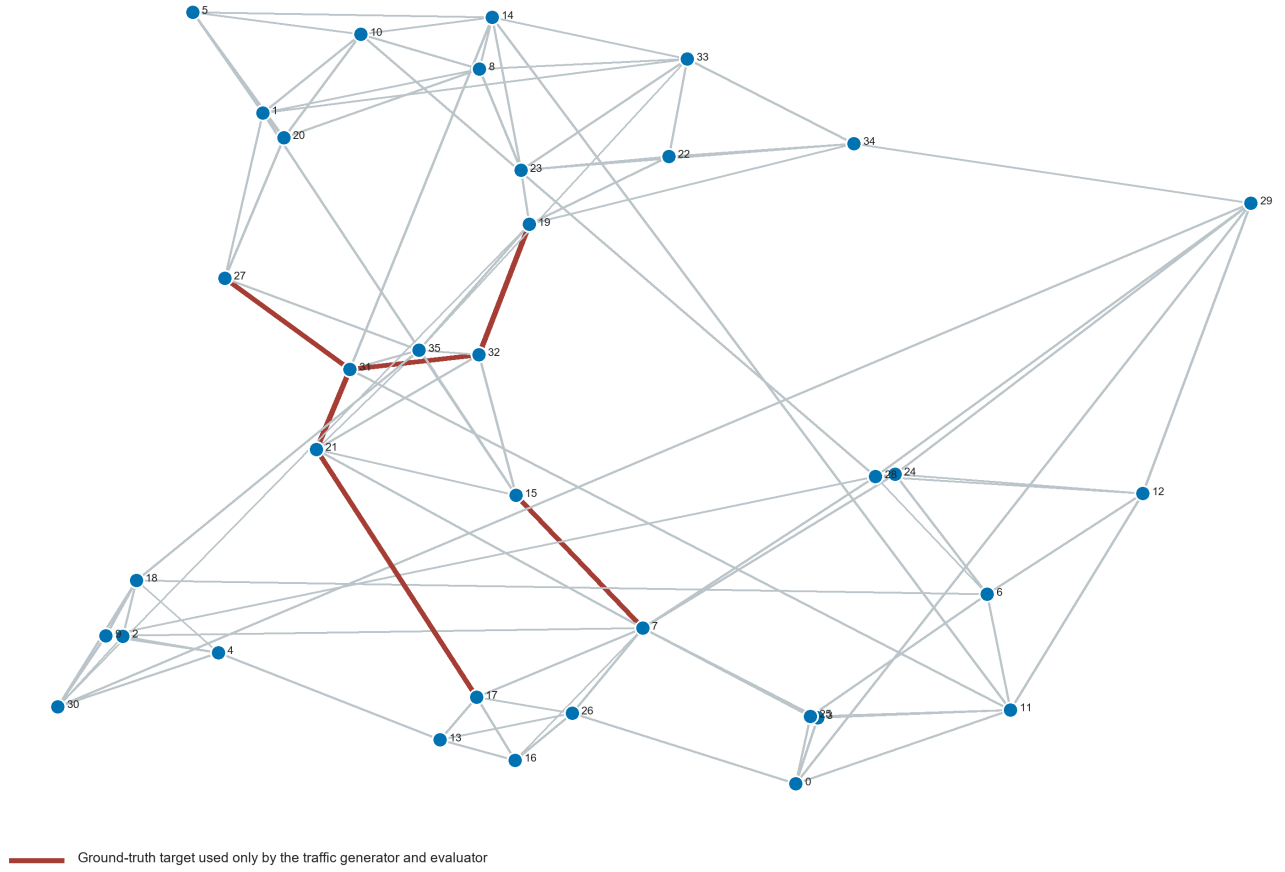


Figure 2. Representative synthetic backbone. Red links are evaluator-only attack ground truth and are not exposed to any routing policy.

4. Telemetry Derived TARA

4.1. Delayed EWMA Detector

The detector converts delayed utilization and delay measurements into bounded normalized anomaly scores. These quantities are denoted as anomaly scores rather than statistical z-scores because they are referenced to fixed operating thresholds rather than estimated sample means and standard deviations. For delayed measured utilization ($\tilde{u}_e$) and delayed measured latency ($\tilde{d}_e$), the two anomaly components are

$$\begin{aligned} a_e^u(t) &= \text{clip}\left(\left(\tilde{u}_e(t) - 0.62\right) / 0.75, 0, 1\right) \\ a_e^d(t) &= \text{clip}\left(\left(\tilde{d}_e(t) \ell_e - 1.15\right) / 2.2, 0, 1\right) \end{aligned} \quad (2)$$

where (ℓ_e) is the nominal propagation latency of link (e). Detector evidence is

$$x_e(t) = \max\left(a_e^u(t), a_e^d(t)\right)$$

The utilization baseline of 0.62 represents the nominal utilization level above which congestion evidence begins to accumulate. For delay, evidence begins when measured latency exceeds 1.15 times nominal propagation latency. Both signals are clipped to $[0, 1]$.

A detector-quality-specific calibration function is applied to the anomaly evidence:

$$C_q(x) = c_q x + (1 - c_q) 0.10$$

where (c_q) is 0.95, 0.80, or 0.62 for the strong, moderate, and weak detector settings, respectively. Link risk is then updated using

$$r_e(t) = 0.60r_e(t-1) + 0.40C_q(x_e(t - \delta_q)) \quad (3)$$

where (δ_q) is the detector delay.

To simulate imperfect detection, an attack-associated evidence value is attenuated to 12% of its original magnitude when a configured false-negative event occurs. For an unattacked link experiencing a configured false-positive event, alert-like evidence between approximately 0.58 and 0.93 is injected before calibration. These stochastic error events use the probabilities reported in **Table 2**.

For reporting detector precision and recall, a link is classified as detector-positive when $(r_e(t) \geq 0.50)$. Precision and recall are computed over link-epoch decisions during attack-active evaluation epochs by comparing this detector output with evaluator-only attack ground truth. Ground-truth attack labels are used only by the simulation generator to impose controlled false-negative/false-positive events and by the evaluator to score detector performance; they are never supplied to the routing controller.

Table 2. Detector quality scenarios.

Quality	Utilization noise SD	Delay noise SD	False positive	False negative	Delay
Strong	0.025	0.35 ms	1%	5%	1 epoch
Moderate	0.060	0.75 ms	4%	15%	2 epochs
Weak	0.110	1.25 ms	10%	30%	3 epochs

4.2. Utilization Prediction and Path Cost

For a candidate path (p) , let

$$L(p) = \sum_{e \in p} \ell_e$$

denote nominal propagation latency,

$$U(p) = \max_{e \in p} \left[\hat{u}_e(t+1) + \frac{v_e}{C_e} \right]$$

denote the maximum predicted utilization after accounting for provisional traffic (v_e) already assigned during the current routing pass, and

$$R(p) = \frac{1}{|p|} \sum_{e \in p} r_e(t)$$

denote mean detector-derived path risk.

The TARA path score is

$$J(p) = L(p) \left[1 + 1.45U(p)^3 \right] + 18R(p) \quad (4)$$

Candidate paths are initially ordered by $(J(p))$. The lowest-cost path becomes the primary path. When path splitting is enabled, the secondary path is selected from the remaining candidates by first minimizing the number of links shared with the primary path and then minimizing $(J(p))$ among equal-overlap candidates. The no-overlap ablation omits this shared-link preference. Traffic shares across the selected paths are proportional to $(1/J(p))$ and normalized to sum to one.

4.3. Policy Definitions and Fair Access

Table 3 identifies what each policy does with the common information stream. Static and topology-disjoint routing intentionally ignore telemetry after initialization; this is a policy restriction, not unequal information delivery. Reactive routing uses the same risk signal as TARA and changes a route only when its current path contains a link whose risk is at least 0.55.

Table 3. Policies and ablations.

Policy	Risk	Utilization forecast	Diversity	Overlap control
Static shortest path	No	No	No	No
Reactive rerouting	Threshold only	No	No	No
Topology disjoint	No	No	Two paths	Topology only
Full TARA	Continuous	Yes	Two paths	Yes
No risk term	No	Yes	Two paths	Yes
No overlap term	Continuous	Yes	Two paths	No
Single path	Continuous	Yes	No	No
No utilization prediction	Continuous	Current EWMA	Two paths	Yes

5. Experimental Design

5.1. Scenario Generation and Reproducibility

For each seed, 36 nodes are placed independently and uniformly in a 1000×1000 two-dimensional coordinate region. Each node is connected to its four nearest neighbors, after which 18 additional undirected node pairs are selected uniformly without replacement to increase path diversity. Duplicate undirected links are merged. Propagation latency for a link between nodes u and v is generated as $1 + \frac{d_{\{uv\}}}{190} + \xi$ ms, where d_{uv} is Euclidean distance and $\xi \sim U(0,1)$. Capacities of nearest-neighbor links are sampled from $U(125, 235)$, while the additional ran-

dom links use $U(115, 220)$.

Each topology contains 90 legitimate demands. Source and destination nodes are sampled uniformly from distinct node pairs. The offered rate for each demand is sampled from a lognormal distribution with log-location 1.05 and log-scale 0.42 and is clipped to the interval 1 - 8 simulation rate units.

Initial shortest paths are computed using propagation latency as link cost. For every demand, up to four candidate paths are generated through repeated shortest-path searches in which links used by an already discovered path have their temporary cost multiplied by 2.0. This produces alternative paths while retaining a bounded candidate set.

Attack targets are selected from links that occur most frequently in the initial shortest paths. Specifically, links are ranked by the number of baseline paths that traverse them, and the highest-ranked $\max\left(5, \frac{|E|}{15}\right)$ links are selected as attack targets. The target set remains fixed within a seed-scenario trial.

The simulation contains 10 routing epochs, with attacks active during epochs 4 - 10. For every attacked link (e) and attack-active epoch (t), attack load is generated as

$$A_e(t) = \alpha C_e F_e(t),$$

where (C_e) is link capacity, $\alpha \in \{0.60, 1.00, 1.40\}$ is the configured attack-intensity factor, and $F_e(t) \sim U(0.82, 1.18)$ introduces epoch-level fluctuation. Random streams are deterministically derived from the simulation seed, attack intensity, and detector setting so that policies evaluated within a scenario experience paired underlying conditions.

Table 4 gives the complete design. Each seed fixes node coordinates, capacities, demands, attack targets, measurement noise, false-alarm draws, and attack fluctuations. Policies within that scenario are therefore paired.

Table 4. Executed simulation design.

Factor	Values
Seeds	24, numbered 1001 - 1024
Topology	36 nodes; geometric links; 90 demands
Epochs	10 total; attack active in epochs 4 - 10
Attack load	60%, 100%, or 140% of target-link capacity
Detector quality	Strong, moderate, weak
Policies	3 baselines, full TARA, 4 structural ablations
Trials	1728 seed-scenario-policy records
Primary outcomes	Delivered demand and 95th-percentile path latency
Attack-active epochs	4 - 10 of 10
Attack targets	Highest baseline-path-centrality links; $\max(5, E /15)$ targets per topology
Attack fluctuation	Independent multiplier $F_e(t) \sim U(0.82, 1.18)$ per target link and epoch
Candidate paths	Up to 4 per demand

The 95% paired interval for a comparison uses Equation (5), where each difference is computed between policies sharing a seed and scenario.

The two-sided 95% paired confidence interval is computed using the Student- (t) distribution:

$$\bar{\Delta} \pm t_{0.975, n-1} \frac{s_{\Delta}}{\sqrt{n}} \quad (5)$$

where $\bar{\Delta}$ is the mean paired difference, s_{Δ} is the sample standard deviation of the paired differences, and $n = 24$. For the reported comparisons, $df = 23$ and $t_{\{0.975, 23\}} = 2.069$.

The interval is descriptive for the executed seed population; no claim is made that synthetic topology seeds are a random sample of deployed networks.

5.2. Attack Effects and Evaluation Metrics

Attack traffic and legitimate routed traffic contribute jointly to link load. For each link e , utilization is

$$u_e(t) = \frac{L_e^{\text{legit}}(t) + A_e(t)}{C_e}$$

For a legitimate path (p) , the deliverable fraction is limited by its most capacity-constrained link. The delivered fraction associated with each path share is therefore computed from

$$D_p(t) = \min_{e \in p} \min \left(1, \frac{1}{u_e(t)} \right)$$

Demand delivery is the weighted sum of the fractions delivered through its selected paths.

The performance-latency model uses total utilization to approximate congestion-induced delay. For link e , the queueing factor is

$$q_e(t) = \begin{cases} \frac{u_e(t)^2}{\max(0.08, 1 - u_e(t))}, & u_e(t) < 1, \\ \frac{u_e(t)^2}{0.08}, & u_e(t) \geq 1. \end{cases}$$

The primary results aggregate only the seven attack-active epochs (epochs 4 - 10). Trial-level delivered demand is the mean delivery percentage across those epochs. At each epoch, the 95th percentile of modeled path latency is computed across the 90 legitimate demands; the reported trial-level P95 latency is the 95th percentile of those seven epoch-level P95 values.

Dropped or partially undelivered demand is represented through the delivery-ratio metric and is not assigned an artificial infinite or penalty latency. The latency statistic therefore represents modeled path delay under the offered load rather than packet-completion latency for successfully delivered traffic only.

6. Results

6.1. Principal Scenario

Figure 3 and **Table 5** report the moderate-detector, 100%-attack scenario. Full TARA delivered 97.25% (SD 1.20) and reactive rerouting delivered 93.18% (SD 2.52). The paired difference was 4.07 percentage points, with a two-sided 95% paired t -interval from 3.39 to 4.74 percentage points.

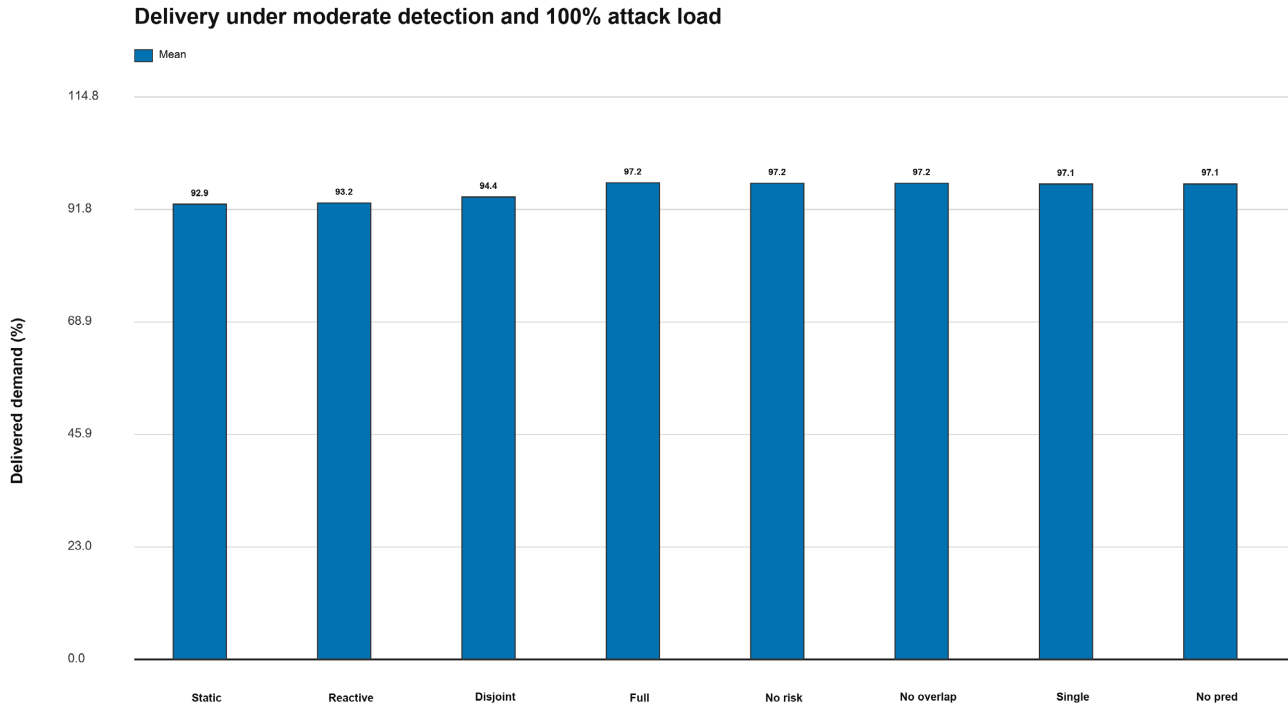


Figure 3. Mean delivered demand for the principal scenario across 24 paired seeds.

Table 5. Principal scenario results.

Policy	Delivery mean $\pm$ SD (%)	P95 latency mean $\pm$ SD (ms)	Route changes
Static shortest path	92.93 $\pm$ 2.77	33.47 $\pm$ 5.65	0.0
Reactive rerouting	93.18 $\pm$ 2.52	33.61 $\pm$ 5.74	15.2
Topology disjoint	94.44 $\pm$ 2.41	27.70 $\pm$ 5.36	0.0
Full TARA	97.25 $\pm$ 1.20	27.15 $\pm$ 4.93	91.8
No risk term	97.19 $\pm$ 1.18	27.37 $\pm$ 4.97	44.2
No overlap term	97.16 $\pm$ 1.22	27.45 $\pm$ 4.44	91.4
Single path	97.07 $\pm$ 1.04	32.67 $\pm$ 5.25	94.0
No utilization prediction	97.08 $\pm$ 1.26	27.23 $\pm$ 4.95	89.3

Full TARA's mean 95th-percentile latency was 27.15 ms. Its paired difference from reactive routing was -6.47 ms (two-sided 95% paired t -interval: -7.94 to -5.00 ms); negative values favor TARA.

6.2. Detector Quality Sensitivity

Figure 4 shows degradation as detector quality weakens. Full TARA delivery was 97.89% with strong detection and 96.71% with weak detection. **Table 6** pairs those service results with the detector precision and recall produced by the simulated measurement pipeline.

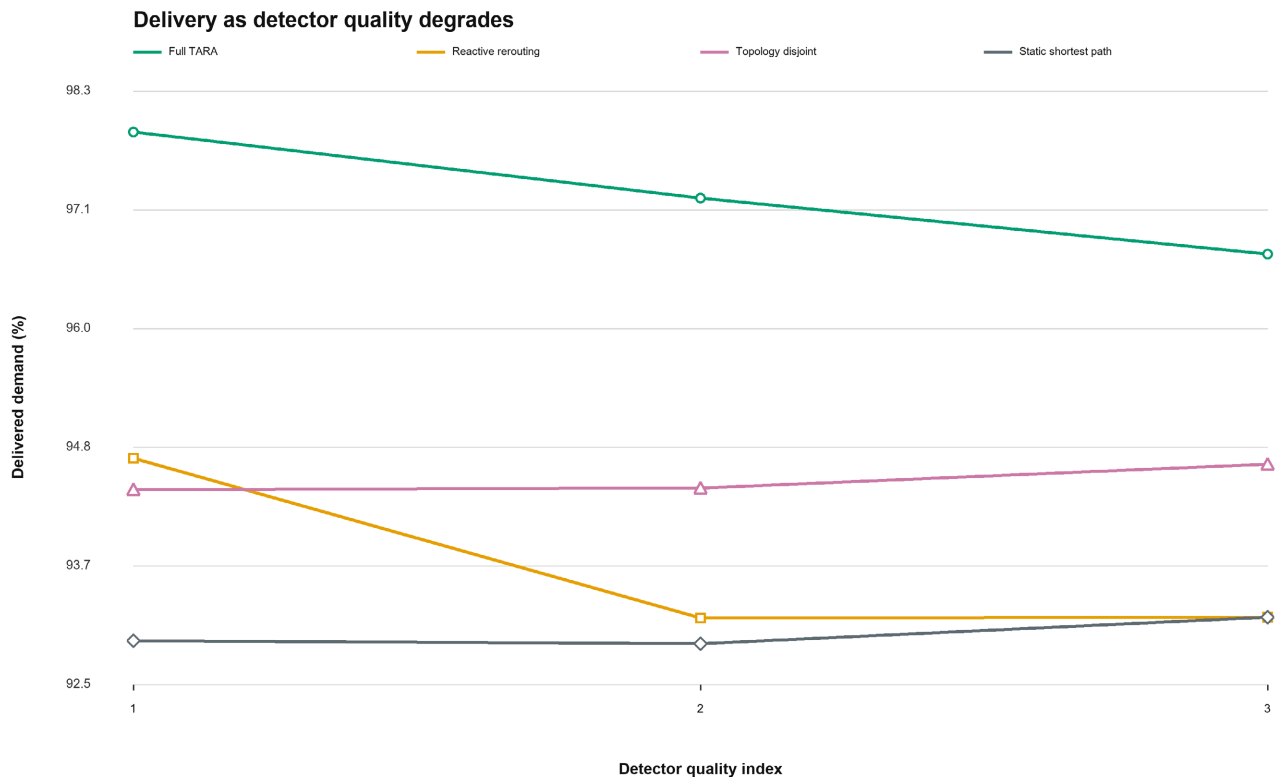


Figure 4. Delivery under strong, moderate, and weak detector settings. Axis values 1, 2, and 3 denote strong, moderate, and weak.

Table 6. Full TARA detector sensitivity at 100% attack.

Detector	Delivery mean $\pm$ SD (%)	Precision (%)	Recall (%)
Strong	97.89 $\pm$ 0.95	99.65	25.06
Moderate	97.25 $\pm$ 1.20	93.85	7.74
Weak	96.71 $\pm$ 1.31	20.83	0.78

6.3. Attack Load Sensitivity

Figure 5 reports all attack levels under moderate detection. The separation among policies grows as capacity pressure increases, but detector delay means that no telemetry-driven policy avoids the initial attacked epochs.

6.4. Ablation and Overhead

Figure 6 compares Full TARA with its four structural ablations in the principal scenario. The single-path ablation has the largest latency, showing that diversity rather than the risk term alone explains much of the resilience.

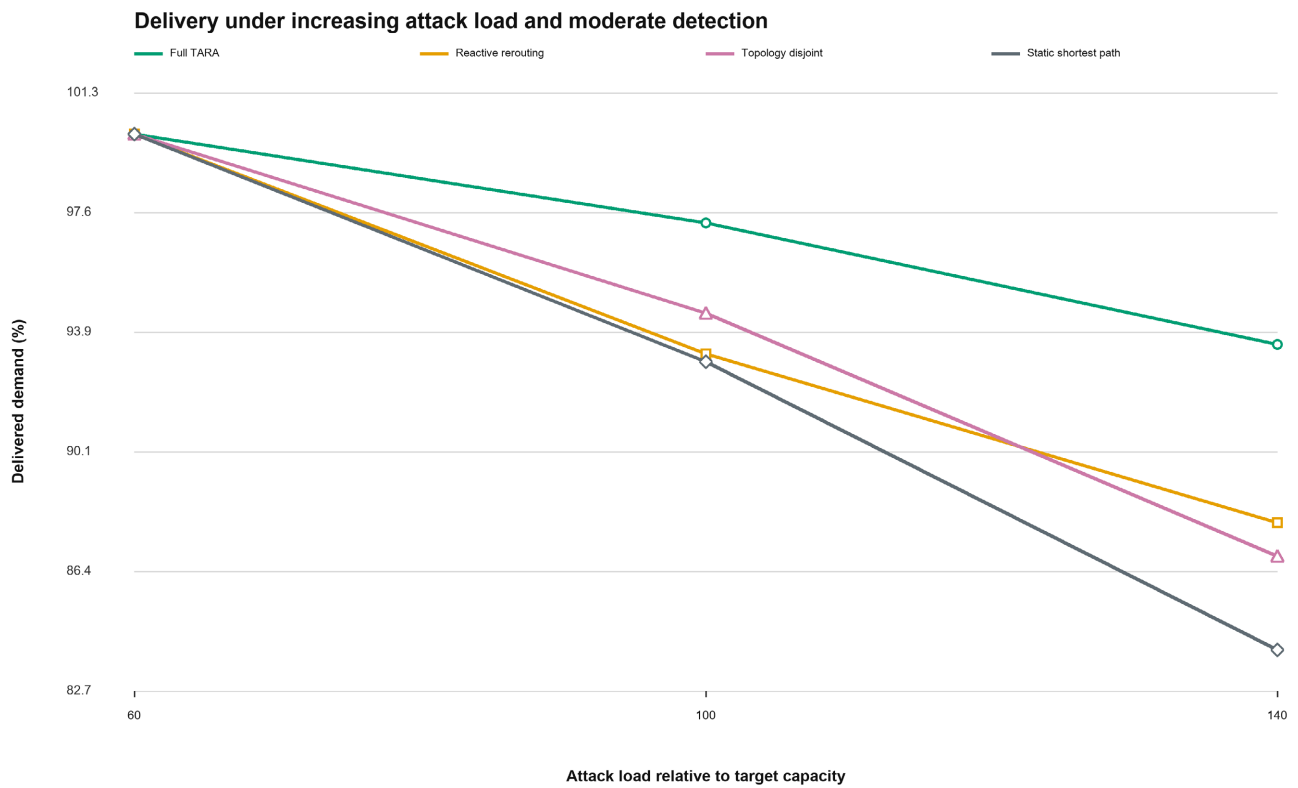


Figure 5. Delivered demand under 60%, 100%, and 140% target-capacity attack load with moderate detector quality.

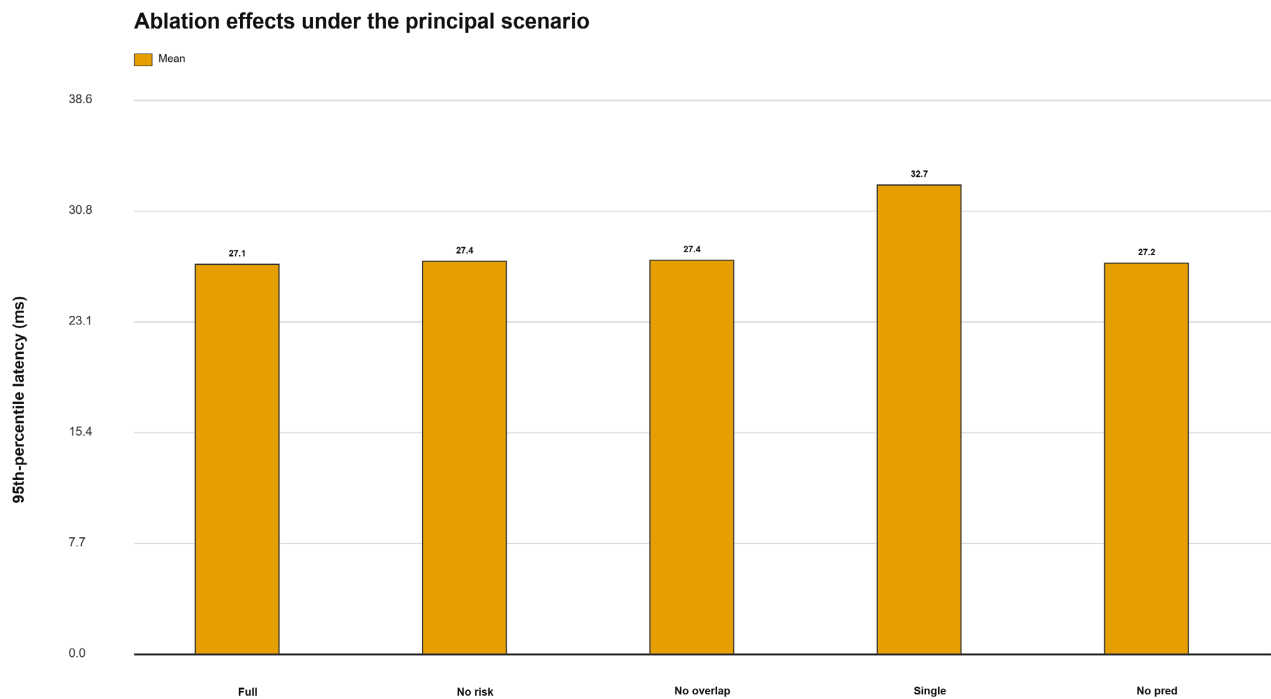


Figure 6. Mean 95th-percentile latency for full TARA and four ablations in the principal scenario.

Figure 7 confirms that strong detection improves both precision and recall; weak detection reduces recall and increases false-alert exposure. **Figure 8** shows

that this additional adaptivity requires substantially more route changes than reactive rerouting.

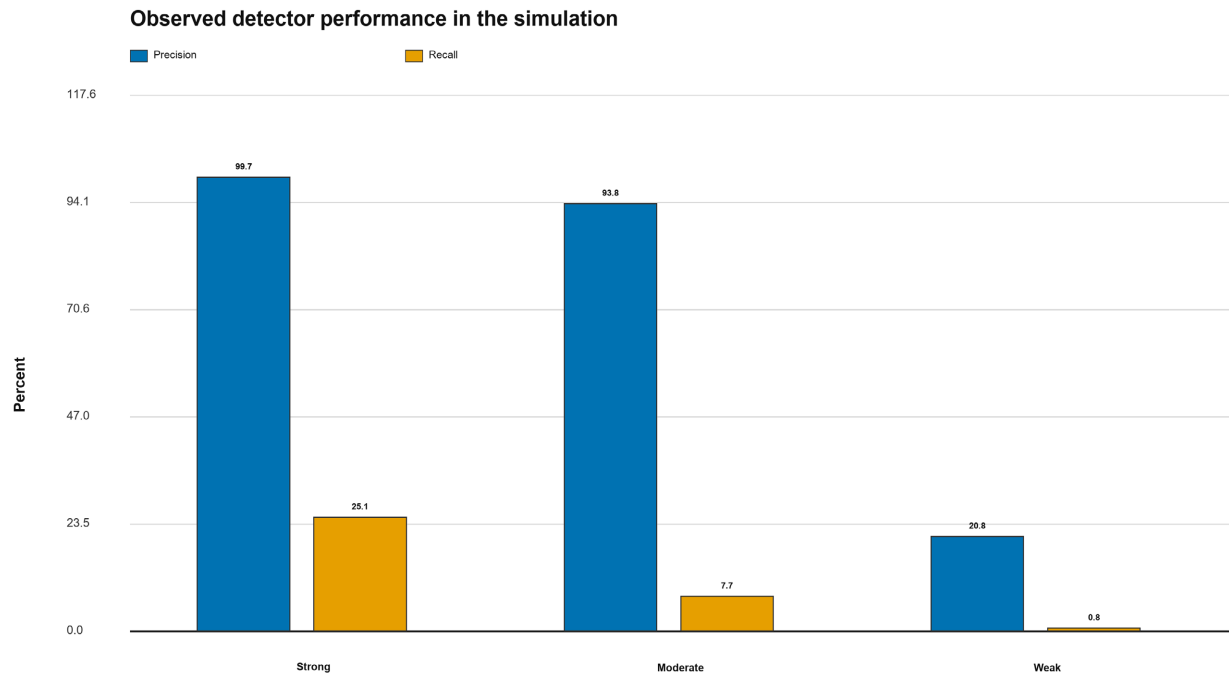


Figure 7. Mean detector precision and recall obtained from the simulated telemetry pipeline.

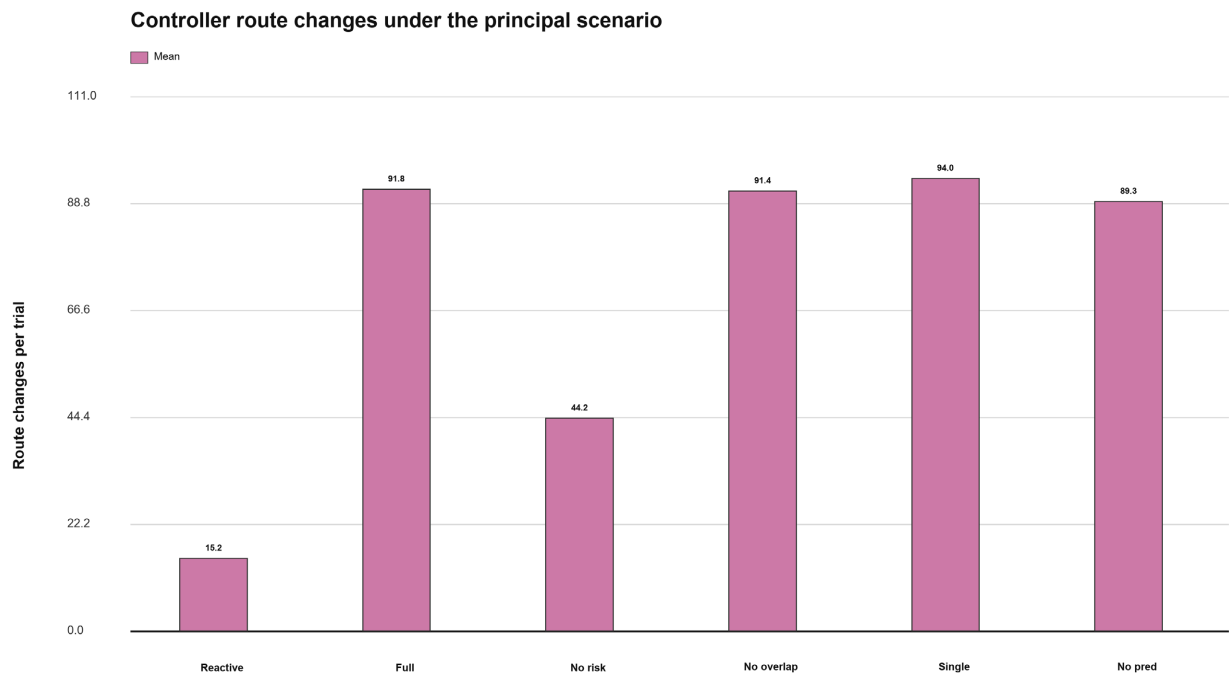


Figure 8. Mean controller route changes in the principal scenario.

7. Discussion

TARA preserves more demand than reactive and static routing in the modeled

scenarios when risk must be inferred from imperfect telemetry. This result does not imply that the controller identifies attacked links correctly in every epoch. Detection delay guarantees a period in which route decisions use stale pre-attack state.

Ablation differences are modest among full TARA, no-risk, no-overlap, and no-prediction variants in the principal scenario. Path splitting produces the largest structural change. The continuous risk term becomes more relevant as detector quality improves; under weak detection its benefit can shrink or reverse because false positives redirect flows unnecessarily.

Operationally, the delivery gain must be balanced against route churn. Full TARA changes more routes than reactive rerouting because it adjusts split paths as observed utilization evolves. Rate limiting, minimum dwell times, and rule-capacity constraints should be added before controller deployment.

8. Threats to Validity and Limitations

The topology generator is geometric and smaller than a carrier backbone. Traffic is stationary within an epoch, links are undirected, queueing is an analytic approximation, and the attacker does not adapt to observed route changes. These assumptions limit external validity.

Detector errors are parameterized rather than learned from packet traces. The strong, moderate, and weak settings are controlled sensitivity cases, not measured operating points. Ground truth is used only to generate false-negative and false-positive events and to score detector performance; it is never passed to routing code.

The paired confidence intervals describe variation across the fixed synthetic seeds. They do not account for uncertainty in the assumed traffic, detector, or queueing models. Hardware-in-the-loop evaluation with controller and switch timing remains necessary.

9. Conclusion

Full TARA delivered 97.25% of demand in the principal scenario, 4.07 percentage points more than reactive rerouting on paired seeds, while its paired 95th-percentile latency difference was -6.47 ms. Path diversity provides the dominant benefit, whereas the risk, overlap, and utilization-prediction terms show comparatively modest differences. Detector degradation changed full-TARA delivery from 97.89% to 96.71%, confirming that resilience depends on telemetry quality. These findings are simulation-based and do not constitute evidence of production deployment performance.

Data and Code Availability

The study uses no proprietary or personal data. The complete simulation source, fixed random-seed ranges, seed-level result files, aggregated tables, and publication figures are included in the accompanying reproducibility package. The code regenerates all numerical results and figures reported in this manuscript.

Funding

This research received no external funding.

Author Contributions

Utham Kumar Anugula Sethupathy: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft, writing—review and editing, and visualization. **Vijayanand Ananthanarayanan:** Validation, formal analysis, writing—review and editing.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., *et al.* (2008) OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, **38**, 69-74. <https://doi.org/10.1145/1355734.1355746>
- [2] Kreutz, D., Ramos, F.M.V., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S. and Uhlig, S. (2015) Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, **103**, 14-76. <https://doi.org/10.1109/jproc.2014.2371999>
- [3] Kang, M.S., Lee, S.B. and Gligor, V.D. (2013) The Crossfire Attack. 2013 *IEEE Symposium on Security and Privacy*, Berkeley, 19-22 May 2013, 127-141. <https://doi.org/10.1109/sp.2013.19>
- [4] Kang, M.S., Gligor, V.D. and Sekar, V. (2016) SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks. *Proceedings 2016 Network and Distributed System Security Symposium*, San Diego, 21-24 February 2016, 1-15. <https://doi.org/10.14722/ndss.2016.23147>
- [5] Hong, C., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., *et al.* (2013) Achieving High Utilization with Software-Driven WAN. *ACM SIGCOMM Computer Communication Review*, **43**, 15-26. <https://doi.org/10.1145/2534169.2486012>
- [6] Fortz, B. and Thorup, M. (2000) Internet Traffic Engineering by Optimizing OSPF Weights. *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Tel Aviv, 26-30 March 2000, 519-528. <https://doi.org/10.1109/infcom.2000.832225>
- [7] Suurballe, J.W. (1974) Disjoint Paths in a Network. *Networks*, **4**, 125-145. <https://doi.org/10.1002/net.3230040204>
- [8] Yen, J.Y. (1971) Finding the k Shortest Loopless Paths in a Network. *Management Science*, **17**, 712-716. <https://doi.org/10.1287/mnsc.17.11.712>
- [9] Brown, R.G. (1959) *Statistical Forecast for Inventory Control*. McGraw-Hill.
- [10] Scarfone, K. and Mell, P. (2007) Guide to Intrusion Detection and Prevention Systems. NIST SP 800-94. <https://doi.org/10.6028/NIST.SP.800-94>
- [11] Floyd, S. (2000) Congestion Control Principles. RFC 2914. <https://doi.org/10.17487/RFC2914>
- [12] Baker, F., Black, D., Blake, S., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998)

- An Architecture for Differentiated Services. RFC 2475.
<https://doi.org/10.17487/RFC2475>
- [13] Saltzer, J.H., Reed, D.P. and Clark, D.D. (1984) End-to-End Arguments in System Design. *ACM Transactions on Computer Systems*, **2**, 277-288.
<https://doi.org/10.1145/357401.357402>
 - [14] Paxson, V. and Floyd, S. (1997) Why We Don't Know How to Simulate the Internet. *Proceedings of the 29th conference on Winter Simulation*, Atlanta, 7-10 December 1997, 1037-1044.
 - [15] Murtuza, S. and Asawa, K. (2024) Early Prevention and Mitigation of Link Flooding Attacks in Software Defined Networks. *Journal of Network and Computer Applications*, **224**, Article 103832. <https://doi.org/10.1016/j.jnca.2024.103832>
 - [16] Troia, S., Gregorini, E., Asdikian, J.P., Li, M. and Maier, G. (2024) Real-Time Delay Measurement in SD-WAN Based on In-Band Network Telemetry. 2024 *IEEE 25th International Conference on High Performance Switching and Routing (HPSR)*, Pisa, 22-24 July 2024, 149-154. <https://doi.org/10.1109/hpsr62440.2024.10635944>
 - [17] Jisi, C., Roh, B. and Ali, J. (2024) Reliable Paths Prediction with Intelligent Data Plane Monitoring Enabled Reinforcement Learning in SD-IoT. *Journal of King Saud University- Computer and Information Sciences*, **36**, Article 102006.
<https://doi.org/10.1016/j.jksuci.2024.102006>
 - [18] Alsaeedi, M., Mohamad, M.M. and Al-Roubaiey, A. (2024) SSDWSN: A Scalable Software-Defined Wireless Sensor Networks. *IEEE Access*, **12**, 21787-21806.
<https://doi.org/10.1109/access.2024.3362353>
 - [19] Soltani, S., Shojafar, M., Mostafaei, H. and Tafazolli, R. (2023) Real-Time Link Verification in Software-Defined Networks. *IEEE Transactions on Network and Service Management*, **20**, 3596-3611. <https://doi.org/10.1109/tnsm.2023.3238691>
 - [20] Jiang, X., Shi, Q., Miao, H., Cao, W., He, H., Chen, S., *et al.* (2024) Credible Link Flooding Attack Detection and Mitigation: A Blockchain-Based Approach. *IEEE Transactions on Network and Service Management*, **21**, 3537-3554.
<https://doi.org/10.1109/tnsm.2024.3357660>
 - [21] Guo, Z., Dou, S., Liu, S., Feng, W., Jiang, W., Xu, Y., *et al.* (2022) Maintaining Control Resiliency and Flow Programmability in Software-Defined Wans during Controller Failures. *IEEE/ACM Transactions on Networking*, **30**, 969-984.
<https://doi.org/10.1109/tnet.2021.3128771>
 - [22] Al-Share, R.A., Shatnawi, A.S. and Al-Duwairi, B. (2022) Detecting and Mitigating Collusive Interest Flooding Attacks in Named Data Networking. *IEEE Access*, **10**, 65996-66017. <https://doi.org/10.1109/access.2022.3184304>

Appendix A. Reproduction Procedure

Python `run_experiments.py --study ijcn`s. It writes every seed-policy record, grouped summaries, figures, and an experiment manifest. The seed, detector, attack load, and policy columns uniquely identify each trial; **Table A1** lists the expected checks.

A complete trial is uniquely determined by the topology seed, attack-intensity setting, detector-quality setting, and routing policy. The seed determines topology geometry and base link characteristics, while deterministic derived random streams determine demands, attack fluctuations, telemetry noise, false-positive events, and false-negative events. Policies within a common seed/intensity/detector scenario therefore operate under paired stochastic conditions. The main experiment contains $24 \times 3 \times 3 \times 8 = 1728$ seed-scenario-policy records.

Table A1. Reproduction checks.

Check	Expected value
Seed range	1001 - 1024
Main records	1728
Attack information in routing function	Absent
Figure resolution	500 dpi PNG
Primary uncertainty	Paired 95% confidence interval