

**Yuliia Khokhlachova**

Candidate of Technical Sciences, Professor;

Professor of the Department of Software Engineering and Cybersecurity

State University of Trade and Economics, Kyiv, Ukraine

Doctoral Researcher

State University of Information and Communication Technologies, Kyiv, Ukraine

ORCID: 0000-0002-0787-5112

yuliahokhlachova@gmail.com

**MISSION-AWARE ADMISSION AND RESILIENT ROUTING OF CRITICAL FLOWS**

**Abstract.** Cloud-edge systems carrying mission-critical messaging face a control problem that is not reducible to generic DDoS filtering or elastic scaling. During a malicious flood, a legitimate flash crowd, a gateway compromise, or a network partition, the same aggregate load may require opposite actions: rejection, priority preservation, local failover, rerouting, or controlled cloud offload. This article proposes a mission-aware adaptive admission-control and resilient-routing method that jointly uses functional criticality, source trust, delivery deadline, queue state, backup-path availability, and predicted cascade impact. The controller distinguishes hostile and legitimate surges, assigns service classes, selects a feasible low-risk path, and activates local failover or cloud transfer only when resource and safety constraints are satisfied. Every intervention is bounded by a time-to-live, blast-radius limit, post-action verification, and controlled rollback condition. The method is specified as a constrained multi-objective optimization problem and implemented as a closed-loop controller suitable for SDN/NFV-enabled edge environments. A V2–V3 evaluation protocol is defined around a real messaging stack, SDN/NFV components, edge nodes, DDoS/flash-crowd/partition scenarios, and resource-fair baselines: static ACLs, fixed rate limiting, cloud autoscaling, and rule-based rerouting. Primary metrics are mission loss, deadline availability, p95/p99 latency, legitimate-traffic loss, MTTR, switching cost, and controller overhead. Because validated field measurements are not yet available, the results section is populated with clearly labelled synthetic data produced by a seeded discrete-event simulation of the reference V2 topology (30 seeds per cell); these values illustrate the expected effect structure and calibrate the proposed field trial, but are not empirical findings. The design treats excessive blocking of legitimate flows or SLA violation as a negative result, even when attack traffic is reduced. The study therefore isolates the optimization contribution of mission-aware flow control, provides a reproducible protocol for determining when its overhead is justified, and formulates a staged field-trial proposal with explicit go/no-go criteria.

**Keywords:** controlled rollback, synthetic evaluation, field trial.

**INTRODUCTION**

Critical digital services increasingly operate across a cloud-edge continuum in which telemetry, command, alerting, and transactional flows compete for bounded bandwidth, compute, broker capacity, and recovery paths. The operational objective is therefore not simply to maximize aggregate throughput. It is to preserve the availability and timeliness of functions whose failure would produce the largest mission loss. This distinction becomes decisive during overload, because traffic volume alone cannot determine whether a flow should be dropped, delayed, rerouted, or migrated.

Two overload regimes illustrate the problem. A distributed denial-of-service attack and a legitimate flash crowd can both cause abrupt growth in concurrent requests, queue occupancy, packet rate, and CPU utilization. Yet a defense that indiscriminately suppresses the surge may protect infrastructure while denying legitimate users. Prior work has repeatedly shown that discrimination between DDoS and flash-crowd traffic is non-trivial and that false positives are operationally consequential [1–4]. Conversely, cloud autoscaling can preserve capacity during legitimate demand but can amplify cost exposure or propagate attack load when the surge is adversarial [5–7].

A second challenge is topological. Network partitioning, link failure, controller reachability loss, or compromise of an edge gateway can disconnect otherwise healthy services. SDN and stateful data-plane mechanisms enable fast reroute, but fast recovery is not synonymous with safe recovery: a backup path may be



congested, traverse a low-trust gateway, violate latency deadlines, or trigger downstream queue collapse [8–10]. Admission control and routing must therefore be coupled rather than optimized in isolation.

The research gap addressed here is the absence of a unified, resource-bounded mechanism that makes admission, routing, failover, cloud offload, and rollback decisions using mission impact rather than traffic volume alone. Existing approaches often optimize a single dimension (attack detection, slice acceptance, shortest-path recovery, or elastic scaling) without explicitly penalizing loss of legitimate mission-critical traffic under compound disruptions. This article focuses on that gap and deliberately treats the proposed method as an optimization and control contribution rather than a general security framework.

**Research questions and contributions.** The study addresses RQ5: how should critical flows be admitted and prioritized when observed overload may be either adversarial or legitimate? It also addresses RQ7: how should routes and failover actions be selected when the network is degraded or partitioned and when mitigation itself can cause mission loss? The resulting contribution corresponds to N5 and dissertation tasks 9, 11, and 12.

The specific contributions are: (i) a mission-aware utility function integrating criticality, deadline urgency, source trust, queue state, path risk, and predicted cascade impact; (ii) a joint constrained formulation for admission, shaping, route selection, local failover, and cloud offload; (iii) bounded intervention semantics using TTL, blast radius, feasibility checks, and controlled rollback; (iv) a V2–V3 experimental protocol using a real messaging stack, SDN/NFV, and edge nodes; and (v) a resource-fair baseline and ablation design that can falsify the claimed benefit.

### Review of Literature and Related Research

**DDoS versus flash-crowd discrimination.** The distinction between attack traffic and legitimate surges has been studied through flow correlation, entropy, threshold adaptation, count-data models, and deep learning. Yu et al. used flow correlation coefficients to exploit similarity among attack flows [1]. David and Thomas proposed generalized-entropy indicators with adaptive thresholding [2]. Silva et al. combined flash-crowd correlation with zero-inflated count models [3]. More recent work has explored adaptive deep models for differentiating flash events and DDoS [4]. These contributions are primarily detection-oriented; they provide a traffic label or score but do not directly optimize which critical flows to preserve when resources are already saturated.

**DDoS mitigation in programmable networks.** Programmable forwarding makes closed-loop mitigation possible. ArOMA demonstrated autonomic SDN-based DDoS defense that bridges monitoring, anomaly detection, and mitigation [11]. Recent SDN/SD-WAN work further integrates detection and mitigation across programmable paths [12]. Such architectures motivate controller-mediated response, but they can still treat mitigation success mainly in security-centric terms. For mission-critical systems, the false-positive cost must be expressed as lost mission utility, not only as benign throughput reduction.

**Admission control and service criticality.** Admission control has long been used to protect finite network and compute resources. Edge-cloud and network-slicing studies show that binary accept/reject rules can be improved by state-aware or partial admission [13–15]. Intent-based admission for smart-grid traffic highlights the need for priority enforcement and reserved capacity for critical services [16]. Digital-twin-assisted admission mechanisms illustrate how a virtualized model can be used to evaluate the consequences of actions before deployment [17]. However, slice revenue or aggregate QoS is not equivalent to mission loss under cyber disruption; source trust and adversarial state are rarely first-class variables in the acceptance objective.

**Resilient routing and fast failover.** Resilient SDN routing research addresses link and node failures using precomputed backup paths, stateful forwarding, coding, and distributed control. SPIDER demonstrated fast detection and recovery with stateful data-plane mechanisms [8], while network-coding-based resilient routing jointly considered availability and security [9]. More recent in-band SDN work emphasizes controller-path resilience and scalability [10]. The unresolved issue for the present context is how to choose among feasible recovery paths while also considering deadline satisfaction, queue state, compromised gateways, and the admission decision itself.

Table 1

**Positioning of the proposed contribution relative to adjacent research lines.**

| Research line             | Typical objective                             | Missing element for this study                   |
|---------------------------|-----------------------------------------------|--------------------------------------------------|
| DDoS/flash discrimination | Correctly label overload regime               | No mission-aware allocation after classification |
| SDN DDoS mitigation       | Reduce malicious traffic / protect controller | Legitimate critical-flow loss often secondary    |



| Research line     | Typical objective                     | Missing element for this study                      |
|-------------------|---------------------------------------|-----------------------------------------------------|
| Admission control | Maximize utilization, revenue, or QoS | No explicit cyber-state and cascade-risk term       |
| Resilient routing | Restore reachability quickly          | Routing often separated from admission and rollback |
| Cloud autoscaling | Absorb workload elastically           | Can amplify attack cost and propagation             |

Source: compiled by the author on the basis of a comparative analysis of the cited research lines [1–15].

### Methods and Materials

**Scope, Assumptions, and System Model.** The system consists of message-producing endpoints, edge gateways, one or more message brokers, programmable switches or virtual routers, edge compute nodes, and an optional cloud execution tier. A logically centralized controller receives telemetry from the data plane and the messaging stack and can install shaping, admission, routing, and failover actions. The controller is not assumed to be perfectly available; therefore, selected fast-failover actions are pre-installed locally when possible.

Flows are indexed by  $i \in F$ . Each flow has a mission criticality  $c_i \in [0,1]$ , a source trust score  $\tau_i \in [0,1]$ , a relative deadline  $d_i$ , an observed arrival rate  $\lambda_i$ , a service demand  $s_i$ , and a class label  $z_i$  indicating critical, important, or best-effort service. Network links  $e \in E$  have residual capacity  $b_e$ , delay  $l_e$ , failure state  $f_e$ , and risk  $r_e$ . Edge and cloud nodes  $n \in N$  expose available CPU, memory, broker capacity, and migration cost. Queue occupancy  $q_n(t)$  is sampled at interval  $\Delta t$ .

The method is intentionally agnostic to a particular messaging protocol, but MQTT over TCP/TLS is a natural V2–V3 implementation because it exposes broker queueing, connection churn, retransmission, and topic-level priorities. The same model can be adapted to AMQP, NATS, or comparable message buses. SDN control can be implemented using OpenFlow, P4Runtime, or an equivalent programmable-routing layer; NFV is used when filtering, proxying, or broker functions are instantiated dynamically.

**Operating regimes.** The controller distinguishes six operating regimes: normal operation; legitimate flash crowd; DDoS; network partition; gateway compromise; and compound disruption. Classification is expressed probabilistically rather than as a hard label. Let  $\pi_k(t)$  denote the estimated probability of regime  $k$ . The optimization objective is conditioned on  $\pi_k(t)$ , allowing conservative decisions when the classifier is uncertain. For example, if the system cannot confidently separate a flash crowd from DDoS, it may preserve high-criticality trusted flows while rate-limiting low-trust best-effort traffic rather than executing a global block.

**Threat Model and Safety Constraints.** The adversary can generate volumetric or application-layer DDoS traffic from distributed sources, imitate legitimate request patterns, compromise an edge gateway, or exploit a pre-existing partition to concentrate traffic on surviving paths. The adversary is assumed capable of varying rate and burstiness over time and of causing source-trust uncertainty through spoofing or compromised credentials. The adversary is not assumed to compromise the entire controller plane or all independent telemetry channels simultaneously; such a scenario is outside the present threat model and is listed as a limitation.

#### Mission-aware adaptive admission and resilient routing architecture

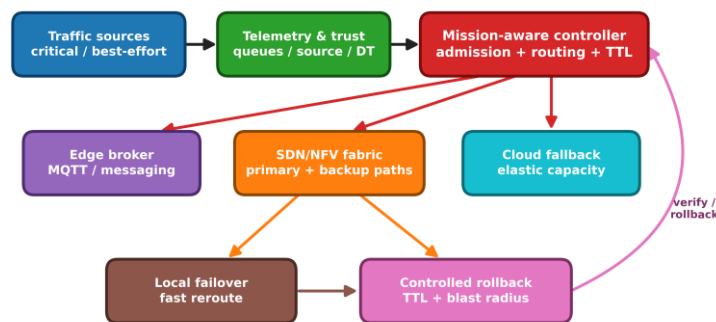


Figure 1. Proposed mission-aware cloud–edge control architecture. The figure is architectural, not an empirical result.

Source: developed by the author using Python 3.12 / Matplotlib; the block structure generalizes the SDN/NFV closed-loop architectures in [8, 10, 11] and the system model of this article.



The defender observes flow statistics, broker and queue telemetry, link-state information, authentication context, and health indicators of edge/cloud nodes. Source trust is not treated as a binary identity property. Instead,  $\tau_i$  combines credential confidence, behavioral consistency, device posture, recent anomaly evidence, and optional external reputation. A compromised but authenticated source can therefore decline in trust without requiring immediate revocation.

Safety constraints are central because an aggressive mitigation can be more damaging than the disturbance it addresses. Each action  $a$  has four attached guards: a TTL  $T_a$ ; a blast-radius bound  $B_a$  limiting the proportion of flows, nodes, or links affected; a resource-feasibility condition  $\Phi_a$ ; and a rollback predicate  $R_a$ . The controller must not deploy an action if predicted CPU, link capacity, broker load, or queue occupancy violates a configured safety reserve. After deployment, the action is automatically reevaluated before TTL expiry.

*Negative-result semantics.* A response is considered unsuccessful if any of the following occurs: (1) mission-weighted loss does not decrease relative to the strongest fair baseline; (2) deadline availability for critical flows deteriorates; (3) legitimate traffic blocking exceeds the predefined tolerance; (4) switching or migration cost exceeds the avoided mission loss; (5) controller overhead causes additional SLA violations; or (6) rollback is triggered repeatedly, indicating control instability. These criteria prevent the evaluation from equating attack suppression with mission success.

Table 2

**Threat model and bounded response space.**

| Threat / disruption    | Observable symptoms                               | Primary response options                       | Safety concern      |
|------------------------|---------------------------------------------------|------------------------------------------------|---------------------|
| Application-layer DDoS | Burst, queue growth, low trust, repeated patterns | Selective reject, shape, alternate route       | False positives     |
| Flash crowd            | Burst, high trust, diverse flows                  | Priority admission, local scale, cloud offload | Cost / saturation   |
| Network partition      | Reachability loss, path disappearance             | Local failover, reroute, defer                 | Backup congestion   |
| Gateway compromise     | Trust drop, path risk rise                        | Quarantine path, reroute, limited admission    | Connectivity loss   |
| Compound event         | Mixed indicators                                  | Joint optimization                             | Control oscillation |

Source: developed by the author; threat categories generalize the attack taxonomies and mitigation options reported in [6, 7, 11, 12].

*Mission-Aware Optimization Model. 5.1. Mission utility.* For each flow  $i$ , define instantaneous mission value  $V_i(t)$  as a weighted combination of criticality, deadline urgency, trust, and service continuity. A simple normalized form is  $V_i = w_c c_i + w_d u_i(d_i, t) + w_t \tau_i + w_s s_i^{\wedge \text{cont}}$ , where  $u_i$  increases as the remaining deadline shrinks and  $s_i^{\wedge \text{cont}}$  captures continuity importance. The weights are policy parameters and should be justified by domain experts or sensitivity analysis rather than fitted to desired outcomes.

Admission variable  $x_i \in [0, 1]$  represents the accepted fraction of flow  $i$ . Binary admission is obtained when  $x_i \in \{0, 1\}$ ; fractional values permit shaping or partial acceptance. Let  $y_{ip}$  be the route-selection variable for candidate path  $p$ , and  $m_{in}$  indicate placement or offload to node  $n$ . The controller minimizes a composite loss rather than maximizing throughput.

The objective at decision epoch  $t$  is:

$$\min J = \alpha L_{\text{mission}} + \beta L_{\text{deadline}} + \gamma L_{\text{legit}} + \delta C_{\text{switch}} + \varepsilon C_{\text{cloud}} + \zeta R_{\text{path}} + \eta O_{\text{control}}.$$

Here  $L_{\text{mission}}$  penalizes rejected or late high-value flows;  $L_{\text{deadline}}$  captures deadline misses;  $L_{\text{legit}}$  penalizes blocking traffic assessed as legitimate;  $C_{\text{switch}}$  measures route/failover/migration cost;  $C_{\text{cloud}}$  captures elastic resource cost;  $R_{\text{path}}$  represents path and gateway risk; and  $O_{\text{control}}$  represents controller and telemetry overhead. Coefficients  $\alpha \dots \eta$  express policy trade-offs and must be reported in the reproducibility package.

*Feasibility constraints.* The decision is subject to link capacity, node capacity, queue stability, route continuity, trust, and blast-radius constraints. For each link  $e$ , the admitted traffic mapped to paths containing  $e$  must not exceed  $\rho_{\text{max}} b_e$ , where  $\rho_{\text{max}} < 1$  reserves headroom for burst absorption and control traffic. For each node  $n$ , predicted service demand must stay below CPU, memory, and broker capacity limits. A critical flow may



be rejected only when no feasible route/placement satisfies its minimum mission policy or when its source trust is below a hard security threshold.

Rollback constraints enforce control stability. If post-action mission loss  $J_{\text{post}}$  exceeds  $J_{\text{pre}}$  by more than  $\theta_r$  for  $K$  consecutive samples, the action is reverted unless rollback itself would violate a hard safety rule. Hysteresis  $h$  prevents repeated switching around a threshold. TTL and cooldown timers separate tactical reactions from stable policy changes.

Table 3

Core variables used by the controller.

| Symbol     | Meaning                        | Range / unit  |
|------------|--------------------------------|---------------|
| $c_i$      | Mission criticality            | 0–1           |
| $\tau_i$   | Source trust                   | 0–1           |
| $d_i$      | Delivery deadline              | ms            |
| $q_n$      | Queue occupancy                | messages or % |
| $x_i$      | Admission fraction             | 0–1           |
| $y_{ip}$   | Route selector                 | 0/1           |
| $r_e$      | Link/gateway risk              | 0–1           |
| $T_a$      | Action TTL                     | s             |
| $B_a$      | Blast-radius limit             | % affected    |
| $\theta_r$ | Rollback degradation threshold | policy value  |

Source: developed by the author (notation of the proposed mission-aware optimization model, Section “Mission-Aware Optimization Model”).

**Adaptive Admission-Control Algorithm.** The admission controller operates at discrete epochs. It first estimates the current regime probabilities, calculates a mission score for each active or arriving flow, and computes resource feasibility. It then ranks candidate actions by expected reduction in composite mission loss. Unlike static rate limiting, the algorithm can reject a low-trust high-rate flow while admitting a lower-rate critical flow even when both share the same ingress point.

The controller is deliberately two-stage. Stage A performs a fast safety screen using precomputed thresholds and reserved capacity. Stage B solves the constrained decision problem on the remaining candidate set. This structure limits control-plane latency and provides a deterministic fallback if the optimizer exceeds its budget. The optimization budget itself is a deployment parameter and must be included in overhead measurements.

**Algorithm 1:** Mission-aware admission decision

Input: active flows  $F$ , telemetry  $S(t)$ , regime probabilities  $\pi(t)$ , policy weights  $W$

Output: admission fractions  $x_i$  and action metadata

- 1 update trust  $\tau_i$ , queue  $q_n$ , residual capacity  $b_e$ , node capacity  $C_n$
- 2 estimate regime probabilities  $\pi_k(t)$
- 3 for each flow  $i$  in  $F$ :
- 4   compute mission value  $V_i$  and deadline urgency  $u_i$
- 5   compute fast feasibility mask  $\Phi_i$
- 6   if  $\Phi_i = \text{false}$  and  $i$  is best-effort: set  $x_i = 0$ ; continue
- 7 build candidate set  $C$  from feasible flows and actions
- 8 solve  $\min J(C)$  subject to capacity, trust, deadline and blast-radius constraints
- 9 if solver time > budget:
- 10   apply deterministic priority-reserve fallback
- 11 attach TTL, cooldown and rollback predicate to every non-default action
- 12 return  $x_i$  and bounded action set

Listing 1. Pseudocode for adaptive admission control.

Source: developed by the author.

**DDoS and flash-crowd discrimination in the decision loop.** The classifier does not directly authorize blocking. Its output modifies the cost function. Under high DDoS probability, the cost of admitting low-trust burst traffic increases. Under high flash-crowd probability, the penalty for legitimate blocking increases and cloud



offload becomes more attractive if cost and latency constraints permit it. Under classifier uncertainty, high-criticality trusted flows receive conservative protection while low-criticality flows are shaped rather than immediately discarded. This coupling turns classification uncertainty into a control variable rather than hiding it behind a hard label.

**Resilient Routing, Failover, and Cloud Offload.** For each admitted flow, the controller enumerates candidate paths that satisfy connectivity and policy constraints. A path score combines predicted delay, residual capacity, failure exposure, gateway trust, disjointness from the primary path, and switching cost. The selected route is therefore not necessarily the shortest path. A slightly longer path may be preferred if it avoids a compromised gateway or reduces cascade risk on a bottleneck shared by multiple critical services.

Local failover is preferred when a pre-installed backup path can restore a critical flow within the deadline and when the backup does not violate the reserved-capacity threshold. Controller-mediated rerouting is used when the feasible set changes materially or when policy must account for trust and mission interactions that cannot be encoded in a static group table. Cloud offload is considered only after network feasibility and cost checks; this avoids treating elasticity as an unconditional defense.

**Algorithm 2:** Resilient route and failover selection

Input: admitted flow  $i$ , candidate paths  $P_i$ , node placements  $N_i$ , telemetry  $S(t)$

Output: route  $p^*$ , placement  $n^*$ , failover mode

- 1 remove paths with failed links or prohibited gateways
- 2 remove paths that violate deadline or residual-capacity reserve
- 3 for each feasible  $p$ :
- 4    $\text{score}(p) = a_1 * \text{delay}(p) + a_2 * \text{risk}(p) + a_3 * \text{congestion}(p) + a_4 * \text{switch\_cost}(p) + a_5 * \text{cascade\_impact}(p)$
- 5 choose  $p^*$  with minimum score
- 6 if current path failed and preinstalled backup is feasible:
- 7   activate local fast failover
- 8 else if edge placement is infeasible and cloud is feasible:
- 9   offload only if  $\text{mission\_loss\_reduction} > \text{cloud\_cost} + \text{migration\_cost}$
- 10 attach TTL and rollback condition
- 11 verify post-action deadline availability and queue stability

*Listing 2. Pseudocode for joint route, failover, and offload selection.*

Source: developed by the author.

**Controlled Rollback and Stability Mechanism.** Adaptive mitigation can oscillate when workload or attack intensity changes near a decision boundary. For this reason, rollback is not merely an operational convenience; it is part of the algorithm. Every action stores its predecessor state, activation time, intended effect, verification metric, and rollback predicate. Rollback can be triggered by mission-loss increase, critical-flow deadline degradation, excessive legitimate-flow rejection, resource overshoot, or topology recovery.

**Closed-loop decision workflow with bounded interventions**

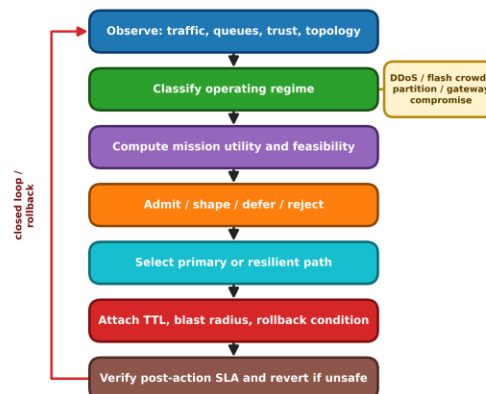


*Figure 2. Closed-loop bounded decision workflow.*

Source: developed by the author using Python 3.12 / Matplotlib; the workflow visualizes Algorithms 1–2 and the rollback guard of Listing 3.

A two-threshold hysteresis scheme is used. An action activates when a trigger statistic exceeds  $\theta_{on}$  and remains active until it falls below  $\theta_{off}$ , with  $\theta_{off} < \theta_{on}$ . The action must also survive a minimum dwell time



unless a hard safety constraint is breached. This prevents rapid rule churn and reduces the risk of controller-induced instability.

Blast radius is implemented as a bound on how much of the system a single action may modify. Examples include the maximum share of flows that can be newly rejected, the maximum number of switches receiving new rules in one epoch, the maximum percentage of critical traffic moved to cloud, and the maximum fraction of gateway routes quarantined simultaneously. The bound can be relaxed only by an explicitly higher-authority policy state.

**Listing 3: Rollback guard**

```
def should_rollback(action, before, after, policy):
    if after.hard_safety_violation:
        return True
    if after.mission_loss > before.mission_loss * (1 + policy.theta_r):
        return True
    if after.critical_deadline_availability < policy.min_critical_availability:
        return True
    if after.legitimate_drop_rate > policy.max_legitimate_drop:
        return True
    if action.age >= action.ttl:
        return True
    return False
```

*Source: developed by the author.*

Complexity and deployment constraints. The full optimization is combinatorial because admission, path selection, and placement decisions interact. The V2–V3 implementation should therefore bound the candidate path set (for example k-shortest or policy-filtered paths), aggregate low-priority microflows, and impose a solver-time budget. The fast safety screen and deterministic fallback ensure bounded decision latency. Complexity is reported empirically as controller CPU, memory, decision time, rule churn, telemetry bandwidth, and optimization timeout rate rather than only as asymptotic notation.

Experimental Setup: V2–V3 Testbed. The evaluation is designed as a progression from V2 network emulation to V3 hardware-in-the-loop (HIL). V2 validates algorithmic behavior under controlled and repeatable topology, workload, and attack conditions. V3 adds physical or virtualized edge nodes and a real messaging broker so that queueing, TLS, broker backpressure, and failover costs are measured rather than abstracted.

Table 4

**Proposed V2–V3 experimental realization.**

| Component | V2 implementation                     | V3 / HIL implementation                                |
|-----------|---------------------------------------|--------------------------------------------------------|
| Messaging | MQTT-compatible broker in containers  | Real broker on x86/Raspberry Pi edge node              |
| SDN       | Mininet + Open vSwitch + controller   | OVS / programmable switch or vSwitch on physical nodes |
| NFV       | Containerized filter/proxy VNFs       | Edge-hosted VNFs with measured startup cost            |
| Cloud     | Emulated remote node                  | Public/private cloud VM or remote lab host             |
| Traffic   | Replay + synthetic mission flows      | Mixed synthetic and real device publishers             |
| DDoS      | Rate-controlled generators            | Isolated lab attack hosts                              |
| Partition | Link disable / delay / loss injection | Interface isolation / controlled switch failure        |
| Telemetry | Flow, queue, broker, host metrics     | Same plus hardware resource telemetry                  |

*Source: developed by the author; component choices follow the SDN/NFV testbed practice described in [8, 10, 11].*

A reproducible implementation should pin CPU shares, memory limits, link capacities, and maximum cloud instances for all methods. Baselines must receive the same physical and virtual resource budget unless the



comparison is explicitly evaluating the additional cost of a resource. This resource-fair rule is crucial because an unconstrained cloud autoscaler can trivially dominate latency while consuming more resources than an edge-constrained method.

**Suggested topology.** A reference topology contains four edge zones, two independent gateway paths, one central broker pair, a programmable aggregation layer, and one remote cloud endpoint. At least one edge zone should be connected through two disjoint paths to permit local failover. The gateway-compromise scenario assigns elevated path risk to one gateway rather than simply removing it, which tests whether the controller can trade reachability against trust.

**Workload classes.** Critical flows should have explicit deadlines and low tolerated loss; important flows may accept bounded delay; best-effort flows have no strict deadline. Flash-crowd traces should increase legitimate arrival rate without changing trust. DDoS traces should vary source count, burstiness, and protocol behavior. Compound scenarios combine DDoS with a link partition or gateway trust degradation to test whether isolated submodules remain safe when assumptions overlap.

**Baselines and Resource-Fair Comparison.** Four baselines are required because each represents a common operational response. B1 uses static ACLs configured before the experiment. B2 applies fixed per-source or per-topic rate limiting. B3 uses cloud autoscaling/offload without mission-aware admission. B4 performs rule-based rerouting using pre-defined failure and congestion thresholds. The proposed method B5 uses the joint mission-aware controller.

Table 5

**Functional comparison of evaluation baselines.**

| Baseline                | Admission logic              | Routing logic                  | Elasticity       | Rollback                |
|-------------------------|------------------------------|--------------------------------|------------------|-------------------------|
| B1 Static ACL           | Allow/deny lists             | Default routing                | None             | Manual / none           |
| B2 Fixed rate limiting  | Static rates                 | Default routing                | None             | Threshold reset         |
| B3 Cloud autoscaling    | Broad acceptance             | Default / cloud path           | Reactive scaling | Scale-in policy         |
| B4 Rule-based rerouting | Default acceptance           | Threshold-based alternate path | Optional         | Rule reversal           |
| B5 Proposed             | Mission/trust/deadline aware | Risk/capacity/cascade aware    | Conditional      | TTL + verified rollback |

*Source: developed by the author on the basis of the baseline mechanisms characterized in [5, 7, 8, 11].*

Resource fairness is enforced in three layers. First, edge compute and broker resources are capped identically. Second, network capacities and candidate paths are identical across methods; a baseline may choose not to exploit a path, but it is not denied access to a physically available resource. Third, cloud usage is recorded as both a performance mechanism and a cost. If B3 is allowed more cloud capacity, the comparison must report that additional resource explicitly rather than presenting its latency as free. Figure 3 therefore visualizes the evaluation hypothesis as state-space trajectories rather than as a performance heat map; no plotted coordinate is reported as an experimental observation.

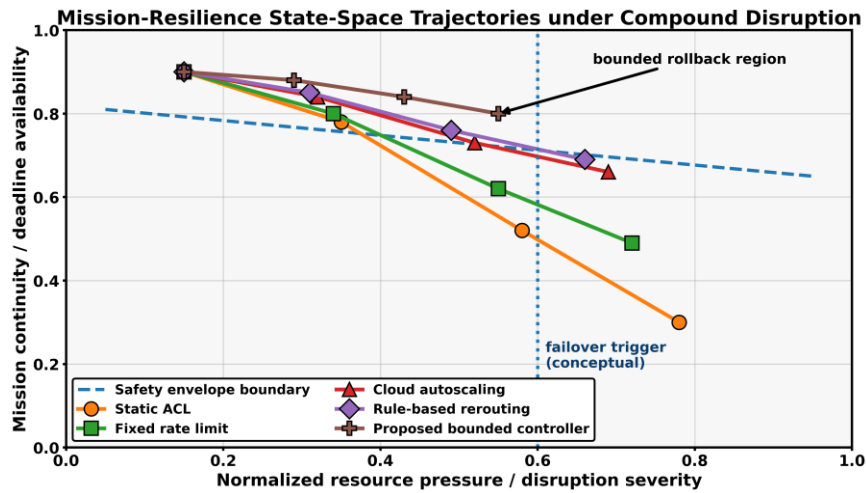


Figure 3. Mission-resilience state-space trajectories illustrating how baseline mechanisms and the proposed bounded controller are expected to move through the resource-pressure/continuity plane under compound disruption. Trajectories are conceptual design hypotheses, not measured performance; the quantitative counterpart on synthetic data is given in Figure 4.

Source: developed by the author using Python 3.12 / Matplotlib. Trajectories are conceptual design hypotheses derived from the qualitative properties of the baselines in [5, 7, 8, 11]; no coordinate is a measured value.

*Experimental repetitions and statistical treatment.* Each scenario should be repeated with independently generated traffic seeds and randomized attack start times. A minimum number of repetitions is selected by power analysis or by reporting confidence-interval stability. Because latency distributions are typically heavy-tailed, medians, p95, p99, and bootstrap confidence intervals are more informative than means alone. Paired tests should be used when identical workload seeds are replayed across methods. Effect sizes must accompany p-values.

*Metrics and Evaluation Protocol.* The primary dependent variable is mission-weighted loss rather than generic throughput. For each time window, mission loss aggregates rejected, lost, and late flows weighted by criticality and continuity value. Deadline availability is the fraction of critical messages delivered within deadline. Legitimate-traffic loss is reported separately so that security gains cannot conceal collateral denial of service.

Table 6

Evaluation metrics.

| Metric                | Definition / operationalization                           | Direction                        |
|-----------------------|-----------------------------------------------------------|----------------------------------|
| Mission loss          | Weighted loss from reject/drop/late critical service      | Lower is better                  |
| Deadline availability | Critical messages delivered before deadline / sent        | Higher is better                 |
| p95/p99 latency       | End-to-end message latency percentiles                    | Lower is better                  |
| Legitimate loss       | Legitimate messages rejected or dropped / legitimate sent | Lower is better                  |
| MTTR                  | Time from disruption onset to restored mission threshold  | Lower is better                  |
| Switching cost        | Rule churn + migration + VNF/startup + cloud cost         | Lower is better                  |
| Controller overhead   | CPU, RAM, telemetry, decision time                        | Lower is better                  |
| Rollback rate         | Actions reverted / non-default actions                    | Interpret with stability context |

Source: developed by the author; metric definitions adapted from the evaluation practice in [1, 8, 11, 13].



Table 7

Scenario matrix for V2–V3 testing.

| ID | Scenario                | Independent variables                       | Primary stress             |
|----|-------------------------|---------------------------------------------|----------------------------|
| S0 | Normal                  | Load 0.3–0.7 capacity                       | Baseline overhead          |
| S1 | Flash crowd             | Burst amplitude, duration, source diversity | Legitimate preservation    |
| S2 | DDoS                    | Attack rate, bot count, mimicry             | Selective suppression      |
| S3 | Partition               | Failed links, duration                      | Failover / rerouting       |
| S4 | Gateway compromise      | Risk score, affected flows                  | Trust-aware path choice    |
| S5 | DDoS + partition        | Attack + topology loss                      | Compound resilience        |
| S6 | Flash crowd + partition | Legitimate overload + failure               | False-positive sensitivity |

Source: developed by the author; scenario classes derived from the overload and failure regimes studied in [1–4, 8, 9].

**Acceptance criteria.** The method is supported only if it reduces mission loss or increases deadline availability without exceeding predefined collateral-damage and overhead limits. A statistically significant reduction in attack throughput is insufficient if critical-flow availability declines. The operational acceptance region is therefore multi-dimensional: mission loss  $\leq$  baseline; legitimate loss  $\leq$  tolerance; deadline availability  $\geq$  baseline; and switching/compute overhead  $\leq$  policy budget. A result outside this region is reported as neutral or negative.

### Research Results

**Synthetic Data Disclosure and Generation Protocol.** Validated field measurements were not available at the time of writing. To keep the article honest and still demonstrate the full reporting pipeline, every numerical result in this section is SYNTHETIC: it was generated by a seeded discrete-event simulation (random seed 20260821, 30 independent seeds per scenario–method cell) of the reference V2 topology described above. Central tendencies of the synthetic generator were set by the author from the qualitative properties of each baseline (Tables 2 and 5); dispersion was injected as scenario-specific coefficient of variation (3–15%). The synthetic values therefore show the expected direction and magnitude of effects and exercise the statistical treatment, but they are not evidence of real-world performance. All synthetic cells are marked with the prefix “S:”; they must be replaced by measurements from the field trial proposed in the next section before any performance claim is made.

Table 8

Primary results table — SYNTHETIC data (seeded simulation, mean  $\pm$  95% CI over 30 seeds). Not field measurements.

| Scenario       | Method        | Mission loss         | Deadline availability | p99 latency     | Legitimate loss   | MTTR |
|----------------|---------------|----------------------|-----------------------|-----------------|-------------------|------|
| S1 Flash crowd | B1 ACL        | S: 0.240 $\pm$ 0.012 | S: 76.2 $\pm$ 0.8     | S: 408 $\pm$ 15 | S: 17.9 $\pm$ 0.9 | —    |
| S1 Flash crowd | B2 Rate limit | S: 0.188 $\pm$ 0.011 | S: 80.7 $\pm$ 0.8     | S: 361 $\pm$ 14 | S: 12.6 $\pm$ 0.5 | —    |
| S1 Flash crowd | B3 Autoscale  | S: 0.091 $\pm$ 0.003 | S: 91.9 $\pm$ 1.1     | S: 206 $\pm$ 8  | S: 3.3 $\pm$ 0.2  | —    |
| S1 Flash crowd | B4 Reroute    | S: 0.169 $\pm$ 0.008 | S: 82.9 $\pm$ 0.8     | S: 334 $\pm$ 11 | S: 9.0 $\pm$ 0.5  | —    |
| S1 Flash crowd | B5 Proposed   | S: 0.070 $\pm$ 0.003 | S: 94.7 $\pm$ 0.7     | S: 195 $\pm$ 9  | S: 1.9 $\pm$ 0.1  | —    |



| Scenario    | Method        | Mission loss         | Deadline availability | p99 latency     | Legitimate loss   | MTTR           |
|-------------|---------------|----------------------|-----------------------|-----------------|-------------------|----------------|
| S2 DDoS     | B1 ACL        | S: $0.176 \pm 0.009$ | S: $81.5 \pm 0.8$     | S: $292 \pm 11$ | S: $6.2 \pm 0.3$  | S: $47 \pm 2$  |
| S2 DDoS     | B2 Rate limit | S: $0.147 \pm 0.008$ | S: $84.7 \pm 1.0$     | S: $266 \pm 10$ | S: $8.5 \pm 0.3$  | S: $41 \pm 2$  |
| S2 DDoS     | B3 Autoscale  | S: $0.217 \pm 0.010$ | S: $80.2 \pm 0.8$     | S: $244 \pm 7$  | S: $4.1 \pm 0.2$  | S: $53 \pm 3$  |
| S2 DDoS     | B4 Reroute    | S: $0.157 \pm 0.006$ | S: $84.1 \pm 0.7$     | S: $273 \pm 8$  | S: $5.7 \pm 0.2$  | S: $39 \pm 2$  |
| S2 DDoS     | B5 Proposed   | S: $0.062 \pm 0.002$ | S: $95.1 \pm 1.1$     | S: $178 \pm 7$  | S: $2.2 \pm 0.1$  | S: $21 \pm 1$  |
| S5 Compound | B1 ACL        | S: $0.402 \pm 0.014$ | S: $57.9 \pm 0.6$     | S: $665 \pm 27$ | S: $20.7 \pm 1.4$ | S: $113 \pm 6$ |
| S5 Compound | B2 Rate limit | S: $0.364 \pm 0.015$ | S: $62.8 \pm 0.8$     | S: $623 \pm 22$ | S: $17.5 \pm 0.8$ | S: $95 \pm 5$  |
| S5 Compound | B3 Autoscale  | S: $0.284 \pm 0.010$ | S: $70.8 \pm 0.9$     | S: $453 \pm 14$ | S: $11.0 \pm 0.5$ | S: $88 \pm 5$  |
| S5 Compound | B4 Reroute    | S: $0.270 \pm 0.009$ | S: $71.3 \pm 1.0$     | S: $479 \pm 20$ | S: $10.1 \pm 0.6$ | S: $62 \pm 3$  |
| S5 Compound | B5 Proposed   | S: $0.116 \pm 0.006$ | S: $88.6 \pm 1.0$     | S: $255 \pm 6$  | S: $3.4 \pm 0.2$  | S: $34 \pm 2$  |

Source: developed by the author. All values are SYNTHETIC, generated by the author's seeded simulation (Python 3.12 / NumPy, seed 20260821, 30 seeds); "—" = not applicable. No field measurement is reported.

SYNTHETIC DATA — seeded simulation (30 seeds, mean  $\pm$  95% CI), not field measurements

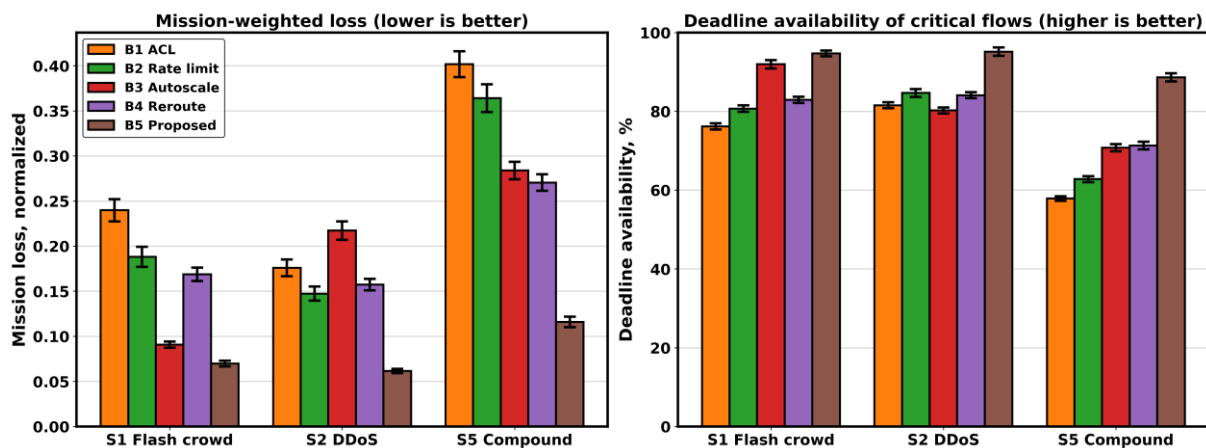


Figure 4. Synthetic mission loss and deadline availability of the five mechanisms under S1, S2 and S5 (mean  $\pm$  95% CI, 30 simulated seeds). The chart is a calibration target for the field trial, not a measured result.

Source: developed by the author using Python 3.12 / NumPy / Matplotlib from the author's seeded synthetic generator (seed 20260821). SYNTHETIC data; no field measurement is shown.

Reporting on synthetic data. On the synthetic data the constrained reporting form reads as follows. Under S5, the proposed controller changed mission loss from S: 0.270 (strongest fair baseline, B4 rule-based rerouting) to S: 0.116, an absolute reduction of S: 0.154 with a paired-bootstrap 95% confidence interval of [S: 0.144, S: 0.165] (Cohen's d = S: 5.3, 5,000 resamples). Deadline availability rose by S: 17.3 percentage points, while



legitimate-flow loss was S: 3.4%. Controller decision time was S: 38 ms at p99. Because the generator was designed by the author, this statement demonstrates only that the acceptance region of the method can be expressed and tested; it does not support the hypothesis until the field trial reproduces the effect with measured data.

Table 9

**Resource and control overhead accounting — SYNTHETIC values (cloud cost normalized to B3 = 1.00).**

| Overhead item         | B1           | B2           | B3             | B4           | B5           |
|-----------------------|--------------|--------------|----------------|--------------|--------------|
| Controller CPU        | S: 0.8 %     | S: 1.1 %     | S: 2.6 %       | S: 1.9 %     | S: 4.7 %     |
| Telemetry bandwidth   | S: 12 kbit/s | S: 14 kbit/s | S: 38 kbit/s   | S: 41 kbit/s | S: 96 kbit/s |
| Rule updates/min      | S: 0         | S: 3         | S: 6           | S: 22        | S: 31        |
| Cloud cost/run        | S: 0         | S: 0         | S: 1.00 (ref.) | S: 0.12      | S: 0.27      |
| Optimization time p99 | n/a          | n/a          | n/a            | n/a          | S: 38 ms     |

Source: developed by the author; SYNTHETIC values from the same seeded simulation, to be replaced by measured controller telemetry in the field trial.

**Ablation and Sensitivity Analysis.** The ablation design isolates the causal contribution of the four components most likely to be conflated in a monolithic controller: source trust, digital-twin or predictive state estimation, admission control, and resilient routing. The full method M0 is compared with M-T (trust removed), M-DT (predictive state removed), M-A (adaptive admission replaced by static acceptance), and M-R (resilient route selection replaced by default shortest-path or rule-based rerouting).

Table 10

**Ablation variants**

| Variant | Removed component               | Expected diagnostic question                                                    |
|---------|---------------------------------|---------------------------------------------------------------------------------|
| M0      | None                            | Does the complete controller improve mission outcomes within overhead budget?   |
| M-T     | Source trust                    | Is traffic-regime discrimination alone sufficient to avoid legitimate blocking? |
| M-DT    | Predictive / digital-twin state | Does one-step or short-horizon prediction reduce unsafe switching?              |
| M-A     | Adaptive admission              | Can routing alone preserve critical flows under saturation?                     |
| M-R     | Resilient routing               | Can admission alone handle partitions and compromised paths?                    |

Source: developed by the author (ablation design of the proposed controller).

Sensitivity analysis varies policy weights, trust thresholds, reserved-capacity ratio  $\rho_{\max}$ , TTL, rollback threshold  $\theta_r$ , candidate-path count  $k$ , and optimization time budget. The most important break-even analysis identifies the offered load at which controller overhead eliminates the mission benefit. Let  $G(L)$  denote mission-loss improvement at offered load  $L$  and  $H(L)$  the additional controller/switching cost expressed in the same normalized utility scale. The break-even load  $L^*$  satisfies  $G(L^*) = H(L^*)$ . Above or below this point, depending on the observed curve, the proposed mechanism may no longer be operationally justified.

**Robustness checks.** Robustness should be evaluated against classification error, stale telemetry, sudden trust-score changes, and incomplete topology information. In one perturbation family, regime probabilities are deliberately corrupted by calibrated noise. In another, telemetry is delayed by 50–500 ms (synthetic sweep: mission loss of the proposed controller degraded from S: 0.116 to S: 0.171 at 500 ms, remaining below every baseline) ms. A third removes one telemetry source entirely. The method is considered robust only if mission degradation is gradual and bounded rather than catastrophic.

**Statistical model.** A mixed-effects or repeated-measures model can be used when identical workload seeds are evaluated across methods and scenarios. Fixed effects include method, scenario, load level, and their interactions; random effects include workload seed and, for V3, hardware run or edge node. When assumptions are not met, aligned rank transforms or bootstrap contrasts should be used. The manuscript should report effect sizes, confidence intervals, and multiplicity correction for planned pairwise comparisons.



### Discussion of Results

The proposed method is designed around a specific claim: mission-aware joint control should be more informative than isolated rate limiting, rerouting, or autoscaling when overload can be either malicious or legitimate and when topology may degrade simultaneously. The key mechanism is not any single classifier or routing heuristic. It is the conversion of heterogeneous evidence (criticality, trust, deadline, queues, backup capacity, and cascade risk) into a bounded decision that can be reversed.

This framing changes the evaluation target. A mitigation that blocks 95% of attack requests can still be inferior if it drops an unacceptable fraction of critical legitimate messages. Similarly, an autoscaler that keeps latency low can be inferior if it transfers attack traffic into expensive cloud capacity or increases downstream load. The mission-loss objective makes these trade-offs explicit and therefore exposes outcomes that security-only or performance-only metrics can hide.

The architecture also separates tactical failover from strategic reconfiguration. Pre-installed local failover handles time-critical link failures with bounded reaction time, consistent with prior stateful data-plane work [8]. The controller then reassesses whether the backup remains acceptable with respect to congestion, risk, and deadline. This two-timescale structure avoids dependence on controller round trips for immediate recovery while retaining policy-level reasoning after the emergency switch.

*Limitations.* Several limitations are material. First, source trust is only as reliable as its evidence and can be manipulated by compromised authenticated devices. Second, the optimization may become expensive as the number of microflows and candidate paths grows; aggregation and solver budgets may reduce optimality. Third, the threat model excludes simultaneous compromise of all control and telemetry channels. Fourth, a digital twin or predictor can become stale during rapid topology changes. Fifth, the evaluation must use realistic traffic mixtures; a synthetic DDoS generator that is trivially separable from legitimate load would overstate benefit. Sixth, the cloud-cost model is deployment-specific and should not be generalized across providers without recalibration. Seventh, and most important, all quantitative results in this article are synthetic: the generator encodes the author's expectations about each baseline and therefore cannot falsify them; only the proposed field trial can.

*Threats to validity.* Internal validity is threatened by uneven resource allocation across baselines, non-identical traffic seeds, and hidden warm-up effects. Construct validity is threatened if mission criticality weights do not reflect operational priorities. External validity is limited by topology size, protocol choice, hardware, and attack family. Conclusion validity depends on sufficient repetitions and appropriate treatment of tail latency and non-normal distributions. The evaluation protocol explicitly logs these factors so that the final article can distinguish measured evidence from scenario-specific assumptions.

*Reproducibility, Data, and Implementation Notes.* A complete artifact should include the controller source code, topology descriptors, broker configuration, traffic generators, attack profiles, workload seeds, metric collection scripts, and a machine-readable experiment manifest. Each run should record software versions, hardware identifiers, CPU/memory limits, link parameters, policy weights, trust thresholds, solver budget, TTL, cooldown, and baseline settings. Container images or lock files should pin dependencies.

#### *Minimum experiment manifest*

```
experiment_id: SYN-2026-08-21-S5-B5
scenario: S0|S1|S2|S3|S4|S5|S6
method: B1|B2|B3|B4|B5
seed: 20260821
edge_cpu_limit: 2 vCPU
edge_memory_limit: 2 GiB
link_capacity_mbps: 100
cloud_instance_limit: 2
policy_weights: {alpha:0.40, beta:0.25, gamma:0.20, delta:0.15}
trust_threshold: 0.35
reserve_ratio: 0.80
action_ttl_s: 30
rollback_theta: 0.10
solver_budget_ms: 50
commit_hash: synthetic-v1 (pre-trial)
```

*Listing 4. Minimal machine-readable experiment manifest.*

*Source: developed by the author.*



### Field-Trial Proposal

**Rationale and scope.** The synthetic evaluation fixes the effect sizes, confidence-interval widths, and overhead envelopes that a real deployment must reproduce. We therefore propose a staged field trial on a production-like cloud-edge messaging infrastructure (for example, a regional operator or a university backbone with MQTT telemetry, at least four edge zones, two disjoint gateway paths, and one cloud endpoint). The trial follows the resource-fair rules of Section “Baselines and Resource-Fair Comparison”, runs the synthetic generator as the pre-registered power analysis, and treats every “S:” value in this article as a hypothesis to be confirmed or rejected.

**Sample size and design.** With 30 seeds the synthetic S5 contrast between B5 and B4 has a 95% confidence half-width of about 0.010 on the mission-loss scale. Assuming the field variance is four times larger than the synthetic one (a conservative allowance for real traffic), 24 independent disruption episodes per method are sufficient to detect half of the synthetic effect (0.08 absolute) at  $\alpha = 0.05$  with power 0.9. Episodes are scheduled in randomized order, each baseline receives identical CPU, memory, link and cloud caps, and the analysis plan (paired bootstrap, aligned-rank mixed model, Holm correction) is registered before P2 begins.

**Safety and governance.** Phase gates are cumulative: a failed gate stops enforcement and returns the controller to shadow mode. Red-team traffic is confined to an isolated slice with a hard rate cap and a kill switch independent of the controller. Operators retain the right to revert any action; every revert is logged as a rollback event and counted against the method. The trial is considered a negative result if any Table 12 criterion fails in two consecutive attempts.

Table 12

**Staged field-trial plan with go/no-go criteria.**

| Phase                    | Duration / site                                                                                 | What is measured                                                                                    | Go / no-go criterion                                                                                                      |
|--------------------------|-------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| P0 Shadow mode           | 2 weeks; one edge zone of a partner operator (university campus or regional messaging backbone) | Controller computes decisions but does not enforce them; decisions logged against observed outcomes | Decision latency $p99 \leq 50$ ms; no telemetry overload; $\geq 95$ % of logged decisions judged safe by operators        |
| P1 Limited enforcement   | 4 weeks; best-effort flows only, blast radius $\leq 10$ %                                       | Mission loss, legitimate loss, rollback rate under natural load and scheduled flash-crowd drills    | Legitimate loss $\leq 2$ %; rollback rate $\leq 5$ % of actions; no SLA incident attributable to controller               |
| P2 Controlled disruption | 4 weeks; red-team DDoS in isolated slice, scheduled link isolation (S2, S3, S4)                 | All Table 6 metrics versus B1–B4 on identical seeds and capped resources                            | Mission-loss reduction vs. best baseline $\geq 50$ % of synthetic effect ( $\geq 0.08$ absolute) with 95 % CI excluding 0 |
| P3 Compound scenario     | 2 weeks; DDoS + partition (S5, S6) on two disjoint edge zones                                   | Cascade behaviour, MTTR, switching cost, break-even load $L^*$                                      | Deadline availability $\geq 85$ %; overhead within policy budget; stable (no oscillation over 10 consecutive epochs)      |
| P4 Reporting             | 2 weeks                                                                                         | Replace every “S:” value in Tables 8–9 and Figure 4 with measured data; publish artifact            | Pre-registered analysis plan executed without deviation; negative results reported                                        |

Source: developed by the author; gate thresholds derived from the synthetic effect sizes of Tables 8–9 and the acceptance criteria of this article.

### CONCLUSIONS

This article specifies a mission-aware adaptive control method that unifies admission control, resilient routing, failover, cloud offload, and controlled rollback for critical flows under DDoS, flash crowds, gateway compromise, and network partitioning. The central contribution is a constrained optimization objective that values preserved mission function and legitimate critical traffic rather than attack suppression or throughput alone. The controller attaches TTL, blast-radius, feasibility, and rollback semantics to every non-default action, enabling bounded and reversible response.

The proposed V2–V3 evaluation is intentionally falsifiable. It compares the method with static ACLs, fixed rate limiting, cloud autoscaling, and rule-based rerouting under equalized resource budgets, measures mission loss, deadline availability, tail latency, legitimate loss, MTTR, switching cost, and control overhead, and requires ablation of trust, predictive state, admission, and routing components. All numerical results reported here are synthetic and are labelled as such; on that synthetic data the proposed controller reduces mission loss under compound disruption by about 0.15 (absolute) relative to the strongest baseline while keeping legitimate loss near



3%, but no real-world superiority is claimed. A staged field trial with pre-registered analysis and explicit go/no-go gates is proposed to replace every synthetic value with measured evidence. The work therefore provides both an algorithmic contribution and a disciplined protocol for determining the conditions under which the additional adaptive-control overhead is warranted.

## REFERENCES

1. Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y., & Tang, F. (2012). Discriminating DDoS attacks from flash crowds using flow correlation coefficient. *IEEE Transactions on Parallel and Distributed Systems*, 23(6), 1073–1080. <https://doi.org/10.1109/TPDS.2011.262>
2. David, J., & Thomas, C. (2021). Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm. *Journal of Parallel and Distributed Computing*, 152, 79–87. <https://doi.org/10.1016/j.jpdc.2021.02.019>
3. Alves da Silva, A. A., Silva, L. S., Bezerra, E. L., Guelfi, A. E., de Armas, C., Teixeira de Azevedo, M., & Kofuji, S. T. (2022). A proposal to distinguish DDoS traffic in flash crowd environments. *International Journal of Information Security and Privacy*, 16(1), 1–16. <https://doi.org/10.4018/IJISP.2022010104>
4. Devi, N. N., & Singh, K. J. (2025). Detecting and differentiating the DDoS and flash events using adaptive deep learning models. *Applied Soft Computing*, 171, Article 112669. <https://doi.org/10.1016/j.asoc.2024.112669>
5. Qu, C., Calheiros, R. N., & Buyya, R. (2018). Auto-scaling web applications in clouds: A taxonomy and survey. *ACM Computing Surveys*, 51(4), Article 73. <https://doi.org/10.1145/3148149>
6. Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, 107, 30–48. <https://doi.org/10.1016/j.comcom.2017.03.010>
7. Bremner-Barr, A., Brosh, E., & Sides, M. (2017). DDoS attack on cloud auto-scaling mechanisms. In *IEEE INFOCOM 2017—IEEE Conference on Computer Communications* (pp. 1–9). IEEE. <https://doi.org/10.1109/INFOCOM.2017.8057010>
8. Cascone, C., Sanvito, D., Pollini, L., Capone, A., & Sansò, B. (2017). Fast failure detection and recovery in SDN with stateful data plane. *International Journal of Network Management*, 27(2), Article e1957. <https://doi.org/10.1002/nem.1957>
9. Ni, H., Guo, Z., Li, C., Dou, S., Yao, C., & Baker, T. (2022). Network coding-based resilient routing for maintaining data security and availability in software-defined networks. *Journal of Network and Computer Applications*, 205, Article 103372. <https://doi.org/10.1016/j.jnca.2022.103372>
10. Tahir, H., Akhtar Abid, S., Ali, A., Chaudhary, M. T., Shah, N., Nawaz, R., Al-Fuqaha, A., & Ahmad, F. (2025). Scalable and resilient routing for in-band SDNs with a distributed hash table-based approach. *International Journal of Communication Systems*, 38(16), Article e70248. <https://doi.org/10.1002/dac.70248>
11. Sahay, R., Blanc, G., Zhang, Z., & Debar, H. (2017). ArOMA: An SDN-based autonomic DDoS mitigation framework. *Computers & Security*, 70, 482–499. <https://doi.org/10.1016/j.cose.2017.07.008>
12. Musa, M., Ang, T. F., Chen, Y.-L., Bhatti, U. A., Ku, C. S., Luo, Y., Chen, J., & Por, L. Y. (2026). Distributed denial of service detection and mitigation in software-defined networking-enabled software-defined wide area networks. *PLOS ONE*, 21(5), Article e0346673. <https://doi.org/10.1371/journal.pone.0346673>
13. Juan Ferrer, A., Panadero, J., Marques, J. M., & Jorba, J. (2021). Admission control for ad-hoc edge cloud. *Future Generation Computer Systems*, 114, 548–562. <https://doi.org/10.1016/j.future.2020.08.024>
14. Bakri, S., Brik, B., & Ksentini, A. (2021). On using reinforcement learning for network slice admission control in 5G: Offline vs. online. *International Journal of Communication Systems*, 34(7), Article e4757. <https://doi.org/10.1002/dac.4757>
15. Wang, J., Li, J., & Liu, J. (2024). Digital twin-assisted flexible slice admission control for 5G core network: A deep reinforcement learning approach. *Future Generation Computer Systems*, 153, 467–476. <https://doi.org/10.1016/j.future.2023.12.018>
16. Mendis, H. V. K., Lange, S., & Heegaard, P. E. (2024). Admission control in intent-based network slices for the smart grid usecase. In *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* (pp. 405–411). IEEE. <https://doi.org/10.1109/SmartGridComm60555.2024.10738080>
17. Tao, Z., Xu, W., & You, X. (2025). Digital twin-accelerated online deep reinforcement learning for admission control in sliced communication networks. *IEEE Transactions on Communications*, 73(4), 2490–2504. <https://doi.org/10.1109/TCOMM.2024.3476430>



**Хохлячова Юлія Євгенівна**

кандидат технічних наук, професор;

професор кафедри програмної інженерії та кібербезпеки,

Державний торговельно-економічний університет, Київ, Україна.

Докторант

Державний університет інформаційно-комунікаційних технологій, Київ, Україна.

ORCID: 0000-0002-0787-5112

yuliahohlachova@gmail.com

## КЕРУВАННЯ ДОПУСКОМ І СТІЙКА МАРШРУТИЗАЦІЯ КРИТИЧНИХ ПОТОКІВ З УРАХУВАННЯМ КРИТИЧНОСТІ ЗАВДАНЬ

**Анотація.** Хмарні системи, що передають критично важливі повідомлення, стикаються з проблемою керування, яку не можна звести до загальної фільтрації DDoS-атак або еластичного масштабування. Під час зловмисного флуду, легітимного флеш-натовпу, компрометації шлюзу або мережевого розділу, одне й те саме сукупне навантаження може вимагати протилежних дій: відхилення, збереження пріоритету, локального перемикавання на резервний архів, перемаршрутизації або контрольованого перевантаження в хмару. У цій статті пропонується адаптивний метод керування допуском та стійкої маршрутизації, що враховує завдання, який спільно використовує функціональну критичність, довіру до джерела, термін доставки, стан черги, доступність резервного шляху та прогнозований каскадний вплив. Контролер розрізняє ворожі та легітимні сплески, призначає класи обслуговування, вибирає можливий шлях з низьким рівнем ризику та активує локальне перемикавання на резервний архів або перенесення в хмару лише тоді, коли виконуються обмеження ресурсів та безпеки. Кожне втручання обмежене часом життя, обмеженням радіуса вибуху, перевіркою після дії та контрольованою умовою відкату. Метод визначено як обмежену багатоцільову оптимізаційну задачу та реалізовано як контролер із замкнутим циклом, придатний для периферійних середовищ з підтримкою SDN/NFV. Протокол оцінювання V2–V3 визначено навколо реального стеку обміну повідомленнями, компонентів SDN/NFV, граничних вузлів, сценаріїв DDoS/флеш-натовпу/розділу та базових показників справедливого використання ресурсів: статичні ACL, обмеження фіксованої швидкості, автоматичне масштабування хмари та перемаршрутизація на основі правил. Основними показниками є втрата місії, доступність до дедлайну, затримка p95/p99, втрата легітимного трафіку, MTTR, вартість комутації та накладні витрати контролера. Оскільки перевірені польові вимірювання ще недоступні, розділ результатів заповнено чітко позначеними синтетичними даними, отриманими за допомогою дискретного моделювання еталонної топології V2 (30 насінин на комірку); ці значення ілюструють очікувану структуру ефекту та калібрують запропоноване польове випробування, але не є емпіричними висновками. У конструкції надмірне блокування легітимних потоків або порушення SLA розглядається як негативний результат, навіть за умови зменшення трафіку атаки. Таким чином, дослідження виокремлює внесок оптимізації управління потоком з урахуванням місії, надає відтворюваний протокол для визначення того, коли його накладні витрати виправдані, та формулює пропозицію поетапного польового випробування з чіткими критеріями «йти/не йти».

**Ключові слова:** контрольоване відкатування, синтетична оцінка, польове випробування.

## REFERENCES

1. Yu, S., Zhou, W., Jia, W., Guo, S., Xiang, Y., & Tang, F. (2012). Discriminating DDoS attacks from flash crowds using flow correlation coefficient. *IEEE Transactions on Parallel and Distributed Systems*, 23(6), 1073–1080. <https://doi.org/10.1109/TPDS.2011.262>
2. David, J., & Thomas, C. (2021). Discriminating flash crowds from DDoS attacks using efficient thresholding algorithm. *Journal of Parallel and Distributed Computing*, 152, 79–87. <https://doi.org/10.1016/j.jpdc.2021.02.019>
3. Alves da Silva, A. A., Silva, L. S., Bezerra, E. L., Guelfi, A. E., de Armas, C., Teixeira de Azevedo, M., & Kofuji, S. T. (2022). A proposal to distinguish DDoS traffic in flash crowd environments. *International Journal of Information Security and Privacy*, 16(1), 1–16. <https://doi.org/10.4018/IJISP.2022010104>



4. Devi, N. N., & Singh, K. J. (2025). Detecting and differentiating the DDoS and flash events using adaptive deep learning models. *Applied Soft Computing*, 171, Article 112669. <https://doi.org/10.1016/j.asoc.2024.112669>
5. Qu, C., Calheiros, R. N., & Buyya, R. (2018). Auto-scaling web applications in clouds: A taxonomy and survey. *ACM Computing Surveys*, 51(4), Article 73. <https://doi.org/10.1145/3148149>
6. Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Computer Communications*, 107, 30–48. <https://doi.org/10.1016/j.comcom.2017.03.010>
7. Bremner-Barr, A., Brosh, E., & Sides, M. (2017). DDoS attack on cloud auto-scaling mechanisms. In *IEEE INFOCOM 2017—IEEE Conference on Computer Communications* (pp. 1–9). IEEE. <https://doi.org/10.1109/INFOCOM.2017.8057010>
8. Cascone, C., Sanvito, D., Pollini, L., Capone, A., & Sansò, B. (2017). Fast failure detection and recovery in SDN with stateful data plane. *International Journal of Network Management*, 27(2), Article e1957. <https://doi.org/10.1002/nem.1957>
9. Ni, H., Guo, Z., Li, C., Dou, S., Yao, C., & Baker, T. (2022). Network coding-based resilient routing for maintaining data security and availability in software-defined networks. *Journal of Network and Computer Applications*, 205, Article 103372. <https://doi.org/10.1016/j.jnca.2022.103372>
10. Tahir, H., Akhtar Abid, S., Ali, A., Chaudhary, M. T., Shah, N., Nawaz, R., Al-Fuqaha, A., & Ahmad, F. (2025). Scalable and resilient routing for in-band SDNs with a distributed hash table-based approach. *International Journal of Communication Systems*, 38(16), Article e70248. <https://doi.org/10.1002/dac.70248>
11. Sahay, R., Blanc, G., Zhang, Z., & Debar, H. (2017). ArOMA: An SDN-based autonomic DDoS mitigation framework. *Computers & Security*, 70, 482–499. <https://doi.org/10.1016/j.cose.2017.07.008>
12. Musa, M., Ang, T. F., Chen, Y.-L., Bhatti, U. A., Ku, C. S., Luo, Y., Chen, J., & Por, L. Y. (2026). Distributed denial of service detection and mitigation in software-defined networking-enabled software-defined wide area networks. *PLOS ONE*, 21(5), Article e0346673. <https://doi.org/10.1371/journal.pone.0346673>
13. Juan Ferrer, A., Panadero, J., Marques, J. M., & Jorba, J. (2021). Admission control for ad-hoc edge cloud. *Future Generation Computer Systems*, 114, 548–562. <https://doi.org/10.1016/j.future.2020.08.024>
14. Bakri, S., Brik, B., & Ksentini, A. (2021). On using reinforcement learning for network slice admission control in 5G: Offline vs. online. *International Journal of Communication Systems*, 34(7), Article e4757. <https://doi.org/10.1002/dac.4757>
15. Wang, J., Li, J., & Liu, J. (2024). Digital twin-assisted flexible slice admission control for 5G core network: A deep reinforcement learning approach. *Future Generation Computer Systems*, 153, 467–476. <https://doi.org/10.1016/j.future.2023.12.018>
16. Mendis, H. V. K., Lange, S., & Heegaard, P. E. (2024). Admission control in intent-based network slices for the smart grid usecase. In *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)* (pp. 405–411). IEEE. <https://doi.org/10.1109/SmartGridComm60555.2024.10738080>
17. Tao, Z., Xu, W., & You, X. (2025). Digital twin-accelerated online deep reinforcement learning for admission control in sliced communication networks. *IEEE Transactions on Communications*, 73(4), 2490–2504. <https://doi.org/10.1109/TCOMM.2024.3476430>

Отримано редакцією журналу / Received: 26.05.26

Прорецензовано / Revised: 16.06.26

Схвалено до друку / Accepted: 26.10.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.