International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Beyond Link-Up: Detecting Silent Network Failures Through Performance-Aware Routing

Harish Goud B[1], K. Tejitha Reddy[2], P. Sadhika[3]

Department of Information Technology, Chaitanya Bharathi Institute of Technology, Hyderabad, India

Abstract

As far as the management system is concerned the network may appear fine, and the users may have poor response times, drop packets, poor video quality, or may not be able to complete transactions. A router might still be available to serve control messages, an interface might still be up, and routing protocols might be still be advertising a path that does not provide acceptable service. This case study deals with reachability and performance. These failures are also known as 'silent' or 'grey' failures. They can impact only packets, flows, destinations or traffic conditions and are therefore difficult to detect just by monitoring link state.

This paper introduces a conceptual routing framework that is performance aware as well as passive device telemetry and active end-to-end measurement based. The first part of the workflow is to make sure that the observations it makes are accurate and reliable, followed by identifying anomalies with continuity, identifying the best possible path segments to be degraded, and then considering the spectrum of solutions. A change in a route is only considered if there is evidence of it and expected improvement is above set thresholds. We will be doing this so we can examine how these decisions are put into practice in a mock-up enterprise network. Furthermore, a repeatable evaluation plan and a decision logic is presented in the paper and implemented on a virtual SDN testbed, along with a system model and diagrams, tables. The framework is a suggested design and not implemented/experiments each value of each scenario will be illustrative.

Index Terms: **Gray failure, silent network failure, performance-aware routing, network monitoring, network tomography, software-defined networking, fault localization, path selection**

I. Introduction

A network operator asking himself the question: Is the Link Up or Down, is one of the most common situations when troubleshooting a network. This is a very important check, but it can't be used to determine whether a network route is usable. A link can run in operation even in the presence of lost packets, delayed packets, and/or reduced throughput. For instance, a path to a reacher can take a long time, an a user may still may be a reacher and the path may be choppy or a transaction may take several seconds to execute. The network is reachable, but its service quality has degraded. The hard failure will be detected by the normal control systems.An interface or a route might be withdrawn due to a cable disconnect,powered off switch,or failed neighbor session.Some traffic continues to run without any failure being quietly detected by so the routing system which could not get a clean message that the path is deemed unhealthy.A faulty transceiver can introduce errors,a queue may build under load,or a forwarding problem may affect only a subset of packets.The partial failures are hard to deal with since they can occur repeatedly and intermittently, without being noticed in the periodicping and average utilization value[1, 2]. The problem becomes more important in multi path networks. If there are multiple paths between a client and a server the controller will have to decide whether to take the degraded path or change the path of the traffic.When a user moves late, the impact of the movement is extended in time.Moving too quickly can create route

oscillation, overload an alternative path,or move traffic on to a route with worse end-to-end delay.It is noted however, that a solution is more than a detector, it is a decision process that takes into account measurement confidence, application requirements as well as alternative path capacity and route stability. This paper proposes a modeling of an approach for detecting the silent degradation and routes selection, based on the quality of the observed services. It is designed to be used in an enterprise or campus network that has multiple paths and an SDN-capable controller. The scale and operational requirements are different for data-center and wide area networks, but the principles could be different for these networks.

A. Objectives and Scope

The objectives outlined in this case study are to: 1) Determine why reachability analysis of interface and next hop nodes is not enough to ensure good application performance; 2) Identify measurable signals of persistent path degradation; 3) Illustrate what is called quality and capacity aware route selection; 4) Propose a testbed for the validation of the above using a baseline and controlled fault conditions. This case study is descriptive in the sense that it is based on design and published literature and suggestions for methodology. There is no guarantee that the proposed methods will work for training, implementation, and improvements in performance. However, the use of numerical examples or tables has illustrative purposes only.

TABLE I
REPRESENTATIVE NETWORK FAILURE CONDITIONS

Condition	Typical symptom	Why basic checks may miss it
Hard link failure	Interface down; route removed	Usually visible to link-state mechanisms
Intermittent loss	Occasional packet drops	A short probe may succeed
Queueing degradation	Rising RTT and jitter	Link remains operational
Partial forwarding fault	Some flows fail or slow	Other traffic may still pass
Load-dependent issue	Degradation only at busy times	Idle-time checks look normal

B. Contributions

The main contributions to be set out: (1) a full explanation of the reachability-performance gap; (2) a layered monitoring model with active and passive signals; (3) a detection to routing process with persistence, feasibility, and hystereses; (4) pre-defined enterprise case study with three connecting paths; and (5) a feedback loop system that will help eliminate route churn and useless routing.

II. TECHNICAL BACKGROUND

A. Hard Failures and Silent Failures

An actual fault leads to a complete interruption in communication. For instance, a broken wire, a shut down interface, or a routing neighbor that stops responding. A silent, or gray, failure allows for some portion of the communication to still function. Symptoms may include minor packet loss, high latency, jitter, or impairment in only certain packets. Various issues can lead to gray failures. For instance, physical degradation of layers might lead to more frame errors but no interruption in services. It is possible that in case of traffic jams, the delay in queuing is increased, and therefore packets are lost only during specific events.

B. Why Reachability Is Not Enough

A successful ping means that one successfully exchanged one probe packet and its response. However, it does not indicate that packets relating to the application will go through, the route will remain stable, or the delay budget of the application will be respected. Averages might mask tail behaviors when most packets might go through in time while some of them could have delays. Interactive applications are sensitive to those outliers.

C. Performance-Aware Routing

Performance-aware routing is the comparison of candidate paths based on the quality of the service (QoS) predicted and monitored. The parameters affecting the choice are round trip time (RTT), packet dropping, jitter, throughput, queue occupancy, utilization of a line, etc. Sometimes, the shortest path is not the best path. While the route has fewer hops and has the degrading link in the current moment, there might be a longer path with less delay and enough capacity.

In Network tomography and end-to-end probing, one is able to determine the quality of the path when there is a lack of internal insight.

III. Related Work

The work done on silent failures addresses the issues of detecting them, testing them, finding their location, and taking action regarding them. Costa Molero, Vissicchio, and Vanbever are responsible for FANcY, a system that can detect and locate gray failures in ISP networks, as well as participate in their mitigation [1]. The authors of this study show that one should go beyond standard techniques when dealing with partial failures. Studies carried out on probing often focus on how to choose probes to maximally achieve fast detection without high cost of measuring [2]. Continuous probing of every path can generate big expenses, while using probes too rare may lead to missing the failures that happen for a short period of time. Tomography-based monitoring assumes that there is no need to measure every path directly but instead the agents should get the knowledge about the paths based on a fixed number of observations they made [3]. Different researchers studied the issue of adaptive path selection to make sure that it is necessary to choose the paths carefully to get accurate information about localized failures [4]. One of the case studies is represented by NetSonar that elaborated a gray-box system for troubleshooting multipath networks [5].

The papers confirm that gray-failure detection and path-aware mitigation are subjects of ongoing research. This paper does not claim to present either concept. The emphasis is on providing a well defined, structured approach for a small testbed which involves collecting trace, confirming steady mean degradation, determining likely defective segment, considering possible paths, performing a controlled rerouting, and checking the results.

TABLE II
Selected Related Work and Its Relevance to the Proposed Case Study

Work	Focus	Relevance	Scope consideration
FANcY [1]	ISP gray-failure detection and localization	Shows why partial failures need specialized detection and mitigation	Large ISP setting; a student prototype may use a smaller topology
Probe scheduling [2]	Efficient probe placement and timing	Motivates balancing detection speed against probe overhead	Probe policy must be tuned to topology and traffic
Tomography-based monitoring [3]	End-to-end path inference	Supports using overlapping paths to infer likely degraded segments	Inference may be ambiguous when paths share many links
Adaptive path selection [4]	Loss inference through selected paths	Highlights that measurement-path choice affects diagnosis	Requires a suitable set of candidate paths
NetSonar [5]	Gray-box troubleshooting in multipath networks	Supports combining topology knowledge with targeted tests	Operational deployment assumptions may differ from a lab
Performance-aware ranking [6]	Ranking mitigation options by impact	Motivates comparing alternatives instead of rerouting blindly	Ranking must include capacity and route-stability constraints

IV. Problem Statement and Research Gap

In a network containing various routes from a client to a server, even though one particular path is still in use, a part of it may start dropping a fraction of packets or adding delay. At the same time, the respective routing protocol announces the availability of the path for use. In this case, the application gets the effect of bad performance. The operator needs to figure out whether the issue appears to be permanent, where it

can come from, and whether there is another route suitable for solving the existing problems without creating another one.

A single threshold is not enough for all cases. RTT may increase due to real traffic demands while a small loss burst might not cause any problem for one application but may disturb the operation of another one. A system that reacts to every spike can change the routes at an uncontrollable rate. On the contrary, a system that waits for a particular outage will have old traffic on a bad path for too long. Thus, the problem solved in this case is about the demand of a transparent control loop that can combine many different metrics with persistence checks, path feasibility checks

The workflow presented is intentionally modular in nature. A basic version can, for instance, employ moving averages and rules while in subsequent experiments, the detector or ranking function can be replaced by statistical or machine-learning-based techniques. The segregation of the detection and routing interfaces permits a straightforward comparison of their performance under identical traffic and fault circumstances.

V. Proposed Framework

The architecture comprises 6 components: the collection of telemetry, processing of features, anomaly identification, analysis of paths, evaluation of routes, and implementation of methods. The graphic illustrates the theoretical design of the system.

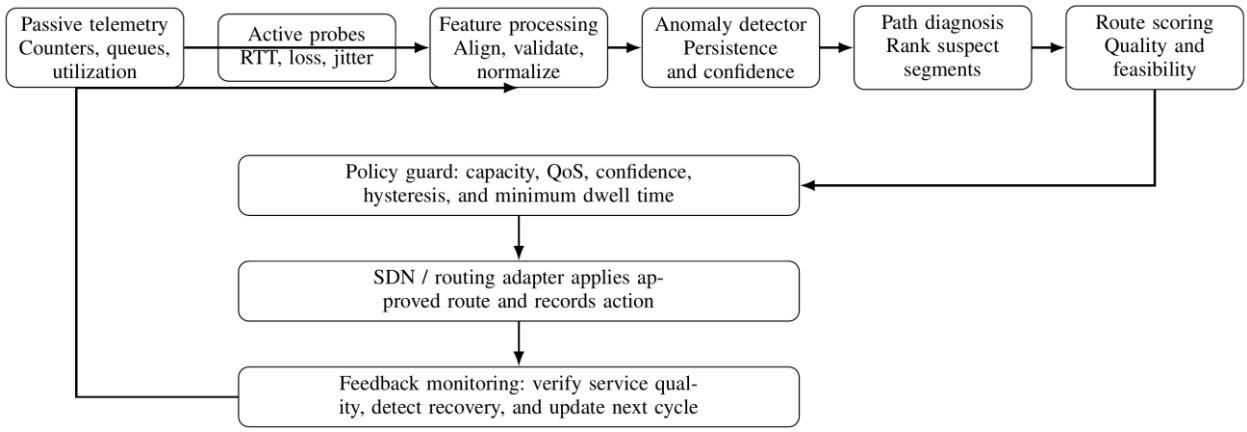


Fig. 1. Architecture of the proposed performance-aware routing system, combining passive and active measurements to detect network issues and select a suitable alternative path

A. Telemetry Collection

Telemetery gathering is achieved through the use of two disparate data sources. Passive telemetry talks about interface counters, queue occupancy, transmission rates, error rates, and flow metrics for the place at hand. Active telemetry takes low-bit measurements on the weapons fitted at given endpoints to determine RTT, loss, and jitter. Both methods are necessary to attain effective implementation of all possible topologies. Any sample must have a time mark, the name of the interface, and some kind of measurement data source. The equipment should be able to mark lost signals, reset devices, or detect that data has become out of date.

B. Feature Processing and Detection

For each candidate path p, the controller keeps track of measurements over a rolling time window. Let $R_p(t)$ represent the round-trip time (RTT), $L_p(t)$ the packet-loss ratio, $J_p(t)$ the jitter, $T_p(t)$ the achieved

throughput, and $U_p(t)$ the link utilization. To identify possible performance problems, the controller calculates a degradation score as follows:

$$S_p(t) = w_r\hat{R}_p(t) + w_l\hat{L}_p(t) + w_j\hat{J}_p(t) + w_t\hat{D}_p(t), \quad (1)$$

The hat symbol represents a normalized feature, while $\hat{D}_p(t)$ represents the normalized throughput deficit. The weights w_r, w_l, w_j, w_t are chosen based on the application's requirements and should be adjusted using data collected from both healthy and degraded network conditions. The score provides a quick overview of the path's performance, but it should not replace checking the individual metrics to understand the actual problem. A basic detection method can flag a path when a particular metric crosses a predefined threshold for k consecutive samples. Another approach is to consider multiple metrics together. For example, an increase in RTT along with rising packet loss, or packet loss combined with more interface errors, may indicate a possible problem. Checking whether the issue continues across several samples helps avoid false alarms caused by temporary spikes. The detector should also maintain a confidence value based on the number of samples collected, how recent the measurements are, and whether the different signals support the same conclusion. In a future implementation, different detection methods, such as moving averages, robust statistical techniques, and supervised machine learning, can be compared. If machine learning is used, the training labels and the time period for making predictions should be clearly defined before training. The evaluation should use separate training and testing traces and include traffic patterns that were not part of the training data. This helps determine how well the method works under different network conditions and reduces the chance of getting overly optimistic results.

C. Path Diagnosis

While detection determines if a path has degraded, diagnosis defines the location of the problem. When two routes with a common part show the same degradation, this common route is an obvious suspect. In case when one route is affected while a different route remains unaffected, it indicates that the problem lies on the part unique to the affected route. This approach can be connected to the concept of network tomography [3], [4].

Uncertainty in evidence should mean that diagnosis results in different ranks rather than a clear statement. The system can introduce specific tests along the pre-defined pairs of paths. In practice, it has to note which measurements may support each segment under suspicion and avoid identifying a component as defective if it simply appears on one faulty track.

D. Candidate Path Scoring

For each flow that is impacted, the routing engine lists possible options. Each option is feasible only when it meets the requirements of the routing policy and has enough available capacity, as well as satisfies the criteria of any strict QoS requirements. Finally, the viable options are assessed using their expected delays, losses, jitters, and costs of route churn:

$$C(p) = \alpha R_p + \beta L_p + \gamma J_p + \delta Q_p + \eta M_p, \quad (2)$$

where Q_p represents a capacity or utilization penalty and M_p represents a route-change or instability penalty. Lower cost is preferred, provided the path passes feasibility checks. Weights must be selected for the intended application rather than treated as universal values.

E. Guarded Execution and Recovery

A new route must not be taken up by the controller solely based on the fact that one candidate had a score that was slightly better than the rest. The controller must be adequately confident and expect some minimum level of improvements to take up a new route. The concept of hysteresis allows the setting of two separate thresholds for switching to a different route and for going back to the one that was left, thus

avoiding repeated switching of the routes in case of fluctuations in readings near the limits of a boundary. Furthermore, a minimum dwell time can stop the flow from switching back and forth many times in a short period of time.

The controller follows the new path when a change in the route has occurred. If the quality of service improves, the controller confirms the action and continues monitoring. But once the quality of the new route becomes worse, it re-evaluates the options. If the old route is back at its normal state and remains steady for the required period of time, the controller can think about the return to the original route.

VI. Illustrative Case Study: Enterprise Network

A campus or branch-office network can be depicted where a group of users connects to one application server using three possible routes. The first route A is made up of two segments, however, it is in use along with other traffic on a heavy transmission line. Route B consists of three hops and has plenty of spare capacity. Route C is also made up of three hops, but it goes a longer physical route. It is assumed that the case under consideration will include a video-meeting that requires low latency and a file transfer that does not require this characteristic.

TABLE III
Illustrative Path Snapshot (Hypothetical Values)

Path	Hops	RTT	Loss	Decision
A	2	78 ms	2.1%	Flag; avoid
B	3	29 ms	0.1%	Candidate selected
C	3	94 ms	0.1%	Retain as fallback

Under conventional routing, route A is likely to be chosen because it has fewer hops. During a traffic burst, its queue begins to grow and packet loss appears intermittently. The interface remains up and a low-rate reachability check may still succeed. The monitoring system observes a combination of increasing RTT, jitter, and loss over multiple intervals. After the persistence condition is met, the detector marks Path A as degraded and the diagnosis module compares measurements from the other paths.

The values in Table III are hypothetical and serve only to explain the decision. Path B is selected if it has sufficient spare capacity and meets the flow's delay requirement. Path C remains a fallback but may be less suitable for a delay-sensitive flow because of its higher measured RTT. The controller then verifies whether the route change improves end-to-end service. It does not assume that the route is fixed forever: if Path B becomes congested, the same procedure runs again.

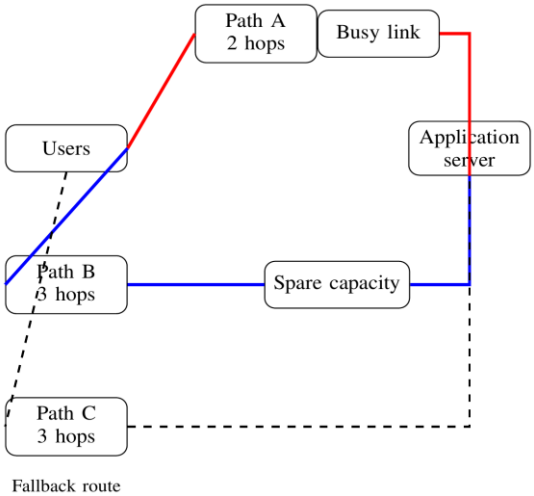


Fig. 2. Illustrative three-path enterprise topology. The highlighted route conditions are hypothetical and are not measured results.

There are three points to consider as part of the case. The first is that the trigger uses performance degradation instead of a link-down occurrence. The second is that the routes are chosen based on both bandwidth and application delay and not simply on the number of hops. The third point is that the controller verifies the result after implementing it and keeps a back-up plan to deal with the situations in which the decisions were made in a disappointing or unreliable manner.

VII. Detection and Routing Workflow

Fig. 3 shows the control cycle that is suggested. The process intentionally has a path for measurements that are not very confident and a check, for whether something is possible before any route is put in place.

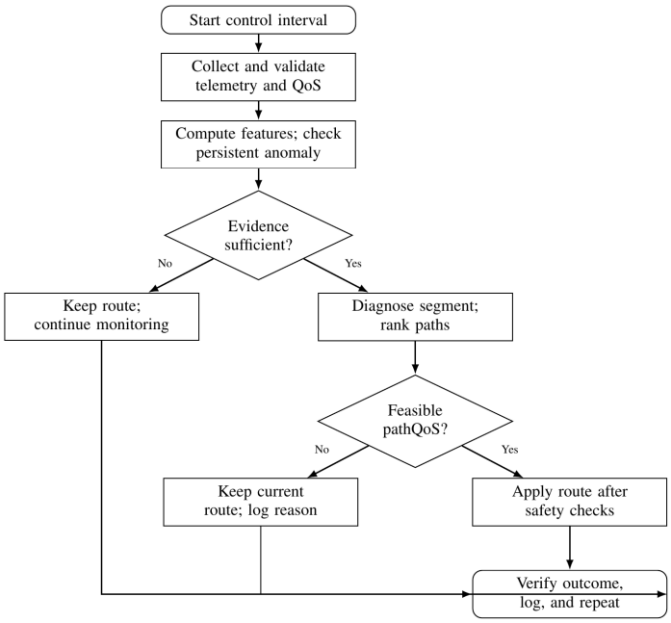


Fig. 3. Compact guarded detection and routing flowchart. Weak evidence or no feasible alternative results in conservative routing.

A. Conceptual Algorithm

The controller can be defined as a high-level process that includes the following steps:

1) The initialization of the path inventory, the window of monitoring, the limits of thresholds, and the active routes.
2) The collection of the active probes with stamped time that are passive telemetry of all monitored paths.
3) The validation of the freshness of the sample, thus calculating the normalized performance characteristics.
4) The observation of whether the level of degradation exceeds set thresholds in number of samples.
5) In case of weakness of evidence, keeping the old route.
6) If the level of degradation has a permanent nature, then overlapping path observations are utilized, alongside the topological data, to rank the various segments suspected.
7) Finding various alternative routes and excluding those that breach capacity constraints or violate policy or application constraints.
8) Scoring remaining paths and necessitating the minimum expected improvement and minimum level of confidence.
9) Choosing the route through the controller adapter and verifying the results.

VIII. System Model and Decision Formulation

The network is represented as a directed graph $G = (V, E)$, where V contains hosts, switches, routers, and other forwarding devices, and E contains communication links. Each link e has capacity B_e, utilization $u_e(t)$, error information, and possibly queue occupancy $q_e(t)$. A flow f has source s_f, destination d_f, requested rate r_f, traffic class, and maximum acceptable delay D_f. The candidate path set is denoted by $\mathcal{P}_f$.

For each path, the controller estimates a quality vector:

$$\mathbf{z}_p(t) = \left[R_p(t), L_p(t), J_p(t), T_p(t), U_p(t)\right].$$

The route-selection problem can be expressed as:

$$p^* = \operatorname{argmin}_{p \in \mathcal{P}_f^{\text{feasible}}} C(p), \tag{3}$$

where $C(p)$ is the path cost in (2), and $\mathcal{P}_f^{\text{feasible}}$ excludes paths that violate capacity, policy, or hard QoS requirements. This formulation is intentionally simplified; a production system would also account for interactions among multiple flows and shared link resources.

A. Stability and Hysteresis

Let C_{current} be the cost of the current path and $C_{\text{candidate}}$ the cost of an alternative. A route change is considered only when:

$$C_{\text{current}} - C_{\text{candidate}} > \Delta_{\min}, \tag{4}$$

where $\Delta_{\min}$ is a configured improvement margin. The controller also enforces a minimum dwell time T_{dwell} between changes, except where an operator-defined emergency policy applies. A recovery threshold different from the degradation threshold prevents immediate route reversal.

B. Policy Constraints

The design separates hard constraints from soft objectives. Hard constraints include route policy, link capacity, access-control requirements, and application-specific delay limits where they are treated as mandatory. Soft objectives include reducing average delay, limiting route churn, and minimizing monitoring overhead. This separation helps prevent the scoring function from selecting a low-latency route that cannot carry the flow safely.

IX. Evaluation Methodology

There must be a providing need for using an assessment framework when claims of improvement. A small virtual topology can be formed through Mininet and SDN controller. Generators of traffic must create steady flow and require different traffic mixes. Linux traffic controller must simulate in delays, losses, rate restrictions, and queue behavior. There is a need to consider a platform and controller enabling other repetitions of the study.

A. Baselines

The basis of comparison must be made out of different aspects including only reachability monitoring that implies changes are activated only after the hard accident or unavailability of the route.

Fixed threshold check-up is another method when at least one parameter must exceed the limit.

Performance-based routing without localization is another method that allows using end-to-end metrics. However, the controller is not capable of identifying a suspicious segment.

First method to be tested is the aforementioned method combining multiple measurements, persistence, diagnosis tests, feasibility check, and hysteresis measures in conjunction with each other.

All methods must work with the same types of traffic traces and failure schedules.

B. Fault Scenarios

The table IV presents a set of scenarios that are aimed to control the errors present in the experiments. Every single error can be introduced and removed in a fixed way, which allows measuring the time of detecting the error and how fast it is recovered.

TABLE IV
SUGGESTED FAULT SCENARIOS FOR A TESTBED

Fault scenario	Injected condition	Expected observation
S1	Random packet loss on one link	Loss rises while the link is still active
S2	Added one-way or round trip delay	RTT rises without using any route withdrawal
S3	Burst loss under high load	Some instances of degradation and jitter can be seen
S4	Rate limiting on the route segment	Throughput reduces with a growing demand
S5	Increase in error rate	Passive probes can show different times
S6	Recovery after removing the faults	Controller should avoid undue recovery

C. Metrics

Assessment ought to encompass detection quality as well as network-wide results. Detection metrics encompass precision, recall, F1-score, false alarm rate, and time between fault injection and detection. Routing metrics encompass end-to-end latency, 95 percentile latency, packet loss, throughput, routing changes, and recovery time. Measurement overhead includes probe traffic and controller processing. Metrics are summarized in Table V.

TABLE V
EVALUATION METRICS AND THEIR PURPOSE

Metric	Definition / measurement	Purpose
Detection delay	Length of time from the beginning of problem to the receipt of the first alarm	Measures responsiveness
Precision	Rate of true positive alarms to total alarms	Measures alarm reliability
Recall	Number of the identified faults out of the total of faults introduced	Measures missed-fault behavior
F1-score	Computation of precision and recall	Summarizes classification quality
False-alarm rate	Rate of false alarms	Captures unnecessary interventions
Mean / p95 RTT	Average / 95-th percentile of “round trip” time	Captures typical and tail delay
Packet loss	Rate of the number of missing probes or packets from total outgoing packets	Measures reliability
Goodput	Rate of useful application data delivered to clients	Measures effective throughput
Route churn	Rate of flow path changes occurred	Measures stability
Recovery time	Time of restoration of service to stable mode after removal of problem	Measures restoration behavior
Probe overhead	Rate of measurement data processed	Measures measurement cost
Controller load	Rate of usage of computer resources during control operations	Measures control-plane cost

D. Experimental Controls

Every case must be repeated under the same conditions, same set of traffic seeds, and for the same amount of time. It is important to note the topology, capacity of links, probe intervals, detection thresholds, version of detector and various traffic scenarios. There also has to be consistency in the duration of popular warm-up period(s). In case eventually a learning-based detector is implemented, the training and testing datasets must be clearly separated and traffic samples including in tests are to be unfamiliar to the training dataset. The technique of an ablation study can help to disable any given component of the system, for example, passive telemetry, active probes, persistence check, diagnostics etc. This approach will help to define whether the component assists in detection despite other metrics. It will be crucial to present variability of results.

X. EXPECTED OBSERVATIONS AND DISCUSSION

The system is intended to discover various performance problems quicker than a system that relies solely on reachability since it takes into account loss, latency, jitter, and throughput even if interfaces work. This expectation is not gathered or evaluated. The effectiveness depends on how well probes cover the area, how often they sample it, how thresholds are determined, and how well the impaired section is isolated from the clear paths.

The combination of multiple signals can bring reductions in false alarms compared to taking the data from one metric, but it can increase reaction time. Its efficiency is defined by the extent to which the proof must be present. It may be problematic to determine the exact location of the connection but still, it has to inform about uncertainty.

While applying performance-aware routing may enhance the quality of service if alternative routes are capable of bearing the traffic and providing better quality, migrating traffic may cause the emergence of new constraints. Thus, route rating should be penalized for its failure to provide sufficient capacity.

XI. Operational Workflow and Prototype Plan

An effective prototype can be built gradually. The initial one should emphasize significant measurement and simple rules rather than a complex model. After establishing an end-to-end control loop, it would be possible to implement new detectors and ranking algorithms without modifying data gathering.

TABLE VI
Prototype Modules and Outputs

Module	Primary output
Telemetry collector	Probes and counters with a timestamp
Feature processor	Validated and aligned features
Detector	Flag showing degradation and output of the detected condition
Diagnosis engine	Ranked list of suspects
Route scorer	Approved routes and alternatives
Policy Guard	Approved route or safe alternative
Controller adapter	Updated forwarding
Logger	Measurements, alarms, actions and results

A. Prototype Modules

For example, the implementation may include Python monitoring service, Mininet topology, SDN controller, and logging database/CSV file. Active probes can be created using basic UDP or ICMP tools, while passive counters may be received using the statistics interface of the controller. The implementation of the route adapter converts the decision made into flow-rule or updating the route. It is preferable to test the prototype in controlled conditions.

B. Reproducibility and Safety

All the experiments must include the position, along with link capacities, traffic structure, type of fault introduced, time of sampling, threshold levels, software used, and random seed. It is also acceptable to run certain simulations before the real changes are made through a special dry-run mode for navigating proposed changes before they are processed.

XII. Advantages and Limitations

A. Potential Advantages

The key advantage of the presented approach is regarding the service quality as an important factor of path health and not just focusing on interface's status and state. The combination of an active and passive measurement will lead to both end-to-end information and localized status.

By tracking the confidence and carrying out persistence tests, I can decrease the reaction to isolated noise, while feasibility checking and hysteresis will help to make the route transition more controllable. The modular interface also makes it easier to compare different methods, starting from simple rules and going finishing with the usage of learning-based approaches.

B. Limitations

B. Drawbacks There are many drawbacks associated with the design. Active probing requires bandwidth and cannot reflect every application flow. Passive telemetries may not be present and/or may vary between

the devices. End-to-end measurements can signal the existence of degradation in the network, but it cannot identify the faulty component specifically. The thresholds that are successful in one network may not apply in another one. The controller may cause instability because it acts based on stale data/information or does not consider shared capacity. Finally, a small simulated topology cannot resemble with the complexity of the real network without exception.

XIII. Future Scope

Future research could also examine adaptive probing scheduling where the controller raises the frequency of measurements only when evidence indicates a problem. Using graph-based inference might serve in representing the shared connections and the overlapping paths in a more direct manner. In addition, statistical change-point detection or the machine learning models could be compared to white-box thresholding rules as long as there are enough labeled traces. Application-specific routing could also be a direction of research, where each service class has its own constraints with respect to delay, loss and throughput.

A wider assessment could involve more than one fault at the same time, the asymmetric paths, the encrypted traffic, and limited visibility. It could also analyze how the existing routing protocols could work with the controller rather than be replaced by it. Finally, a confidence-based operator interface could present the problematic segment, its proof, alternative routes and the expected impact prior to the action being taken automatically.

XIV. Conclusion

The purpose of the case study is to examine the gap between network reachability and quality of service. Errors that may occur on the network can exist without alerting network managers, causing problems such as packet loss, delay, jitter, and diminished performance. The developed concept uses both passive and active measurement techniques where the degree of QoS impairment is measured, estimated affected parts of the network are identified, alternative paths are analyzed, and changes are made only after preparing the confidence, capacity and quality criteria.

Although useful, the suggested approach is only a purely theoretical concept without testing. Testing the idea in an SDN-enable network can be done comparing it with reachability-based monitoring and simpler performance monitoring presented by various metrics of different parameters such as detection delay, number of false positives, latency, losses, throughput, route churn, and overhead of the monitoring process.

References

[1] E. Costa Molero, S. Vissicchio, and L. Vanbever, “FAst In-network GraY failure detection for ISPs,” in *Proc. ACM SIGCOMM*, 2022, pp. 677–692, doi: 10.1145/3544216.3544242.

[2] “Probe scheduling for efficient detection of silent failures,” *Performance Evaluation*, vol. 79, pp. 73–89, 2014, doi: 10.1016/j.peva.2014.07.005.

[3] Y. Chen, D. Bindel, and R. Katz, “Tomography-based overlay network monitoring,” in *Proc. ACM Internet Measurement Conf. (IMC)*, 2003, doi: 10.1145/948205.948233.

[4] Y. Qiao *et al.*, “Adaptive path selection for link loss inference in network tomography applications,” *PLOS ONE*, vol. 11, no. 9, 2016, doi: 10.1371/journal.pone.0163126.

[5] H. Zeng *et al.*, “Measuring and troubleshooting large operational multipath networks with gray box testing,” Microsoft Research, Tech. Rep. MSR-TR-2015-55, 2015.

[6] P. Namyar *et al.*, “Enhancing Network Failure Mitigation with Performance-Aware Ranking,” in *Proc. ACM SIGCOMM*, 2023.