XNDT: Towards Explainable Network Digital Twins using a High-Fidelity Hardware Testbed

Klaus J. Bertsch, Oliver P. Waldhorst

Institute of Data-Centric Software Systems, Karlsruhe University of Applied Sciences, Germany
{klaus.bertsch,oliver.waldhorst}@h-ka.de

Abstract. Growing traffic volumes and dynamic operating conditions complicate the management of modern communication networks. *Machine Learning* (*ML*)-based *Network Digital Twins* (*NDTs*) can support performance prediction and what-if analysis, with *Graph Neural Networks* (*GNNs*) capturing topology-dependent network behavior. Yet existing *explainable AI* (*XAI*) methods typically return attribution scores, feature rankings or graph structures that remain difficult to interpret operationally. We present *XNDT*, a research vision for operator-facing explainability in GNN-based NDTs. XNDT links model-level attribution evidence to the active network state and independent telemetry to produce traceable network-semantic explanations of predicted performance degradations. We instantiate this explanation chain in a controlled congestion scenario on a reconfigurable hardware testbed. The proof of concept traces predicted packet loss to the relevant traffic demands, their paths through a shared bottleneck and independent hardware evidence from interface throughput and queue-drop counters. The results provide initial feasibility evidence and motivate broader evaluation across network scenarios and explanation methods, including technical evaluation of operator-facing explanations.

Keywords: Network Digital Twin, Explainable AI, Hardware Testbed

1 Introduction

The increasing traffic volume and operational complexity of modern communication networks create a growing need for accurate and explainable decision support. *Network Digital Twins* (*NDTs*) provide continuously updated representations of physical networks and support tasks such as state estimation, performance prediction and what-if analysis [1, 2]. Within *Machine Learning* (*ML*)-based NDTs, performance models based on *Graph Neural Networks* (*GNNs*) are particularly promising because they can represent topology-dependent interactions between traffic demands and the resources encountered along their forwarding paths. The *RouteNet* family, which comprises GNN-based models for predicting network performance, has demonstrated that such models can predict topology-dependent performance metrics from traffic matrices, which encode the offered rates of these demands, and from routing configurations [3–6].

Accurate predictions alone, however, are not sufficient for operational use. Before acting on a model prediction, network operators need to understand which aspects of the current network state influenced the result and whether the explanation is consistent with observable network behavior. Existing *explainable AI* (*XAI*) methods for GNNs identify prediction-relevant features or graph structures using a range of architecture-based and post-hoc techniques [7–12]. Networking-specific studies have further applied feature-attribution methods to learned NDTs and traffic-matrix-related tasks [13, 14].

However, the resulting explanations typically remain model-centric. Although these outputs indicate which model inputs or entities influenced a prediction, they do not by themselves provide an operator-facing network explanation. For example, an attribution associated with a model entity does not necessarily reveal which traffic demands traverse the corresponding network resource, why their forwarding paths converge there or whether the resource exhibits the expected queueing or congestion behavior. Model attribution and network-semantic interpretation must therefore be treated as distinct but connected tasks. Network-semantic interpretation additionally requires network-state information and supporting telemetry.

This paper presents *XNDT*, a research vision for operator-facing explainability in GNN-based NDTs. XNDT augments a RouteNet-compatible network-performance model with an explanation pipeline that links model-level attribution evidence to the active network state and supporting telemetry. Instead of presenting raw attribution scores as final explanations, XNDT aims to construct traceable explanation chains from a predicted performance degradation through the relevant forwarding dependencies and affected resources, with supporting evidence from telemetry. XNDT is designed as human-in-the-loop decision support, enabling operators to inspect the evidence underlying a model prediction rather than treating attribution as a final explanation.

To investigate the feasibility of this vision, we instantiate the central explanation chain in a controlled congestion scenario on an existing reconfigurable hardware testbed. Traffic demands are configured to converge on a selected capacity-constrained link under a known forwarding state. GNN predictions and model-level attribution evidence are then compared with the forwarding dependencies implied by the configured state and with independent hardware telemetry. The proof of concept is intended to provide initial feasibility evidence rather than a comprehensive validation beyond the controlled testbed setting.

2 Background and Related Work

2.1 GNN-Based Network Performance Modeling

GNN-based performance models for NDTs can represent topology-dependent interactions between traffic demands and network resources to estimate network behavior under changing network conditions [1, 2, 15].

The RouteNet family comprises GNN models that represent network performance through message passing across graph entities encoding traffic demands and their forwarding resources, including queues in some variants. RouteNet demonstrated that GNNs can predict end-to-end performance metrics from traffic matrices and routing configurations [3]. RouteNet-Erlang extended this approach to different queueing configurations, while RouteNet-Fermi improved generalization across network scenarios [4, 5]. RouteNet-Gauss further investigated the use of measurements obtained from physical networking hardware [6].

These works primarily focus on network-performance prediction, whereas XNDT focuses on the network-semantic interpretation of individual predictions.

2.2 Explainability for GNN-Based Network Performance Models

Graph-XAI methods investigate which features, graph entities or structural relationships contribute to a GNN prediction [7, 12]. Architecture-based approaches, such as Graph Attention Networks, expose internal coefficients that may provide indications of which neighboring entities are emphasized during message passing [10]. However, these coefficients do not necessarily constitute complete operator-facing explanations.

Post-hoc methods analyze trained models without changing their architecture. Gradient-based approaches estimate the relevance or sensitivity of a prediction with respect to graph entities or features [8, 16]. PGExplainer learns masks that identify prediction-relevant subgraphs [9], whereas CF-GNNExplainer generates counterfactual explanations by identifying graph changes that alter a prediction [11]. These methods differ in their assumptions and outputs, but commonly express explanations as attribution values, feature rankings, masks, subgraphs or counterfactual modifications.

Initial work has applied XAI techniques in networking contexts. Chowa et al. use surrogate-based XAI methods to identify influential input features in a GNN-based NDT [13]. Zilli et al. investigate influential inputs for traffic-matrix inference [14]. Concept-based approaches provide a complementary perspective by expressing model behavior through domain-relevant concepts. Agua, for example, investigates concept-based explanations for learning-enabled systems [17].

These approaches expose model-relevant inputs or concepts, but their outputs do not necessarily describe the state-dependent forwarding relationships of a concrete network. An attribution associated with a model entity does not by itself identify which traffic demands use the corresponding network resource, why their paths converge or whether the resource exhibits the expected congestion behavior.

2.3 Research Gap and Positioning of XNDT

XNDT addresses the gap between model-level attribution evidence and network-semantic interpretation. We use the term model-level attribution evidence for the output of a selected Graph-XAI method. A network-semantic interpretation additionally relates this evidence to identifiable network entities, traffic demands, active forwarding dependencies and telemetry.

To the best of our knowledge, existing work has not yet systematically combined these elements into a traceable explanation pipeline for GNN-based NDTs. XNDT is therefore not intended as a new general-purpose GNN explainer. Instead, it introduces a semantic layer between an attribution method and an operator-facing explanation. This layer transforms model-level attribution evidence into traceable network-semantic evidence for an operator-facing explanation.

3 The XNDT Vision and Architecture

XNDT follows a modular architecture that connects network-state acquisition, performance prediction and network-semantic explanation (Fig. 1). A physical testbed provides controlled network states and independent telemetry for evaluating the prediction-to-explanation chain.

3.1 System Overview

Figure 1 summarizes XNDT as six functional components. The labels within components (2) and (3) indicate their proof-of-concept instantiation. (1) Network State Acquisition provides the traffic matrix, topology and forwarding state to (2) NDT Performance Model, a RouteNet-compatible GNN that predicts metrics such as delay or packet loss [3–6]. The forwarding state captures the active forwarding information that determines the path each demand takes. The required network-state information can be obtained from different monitoring and control-plane sources, such as SNMP, streaming telemetry or SDN controller information, thereby avoiding dependence on a specific collection mechanism. The three purple components, (3) Model XAI, (4) XNDT Semantic Alignment and (5) XNDT Evidence Integration, constitute the XNDT

explanation pipeline. The prediction from (2) is passed to (3), which produces model-level attribution evidence. Components (4) and (5) form the XNDT-specific semantic layer. Independent hardware telemetry from (1) bypasses the performance model and Model XAI and is incorporated only in (5) XNDT Evidence Integration. The resulting evidence supports the operator-facing explanation for the envisioned (6) Operator Dashboard.

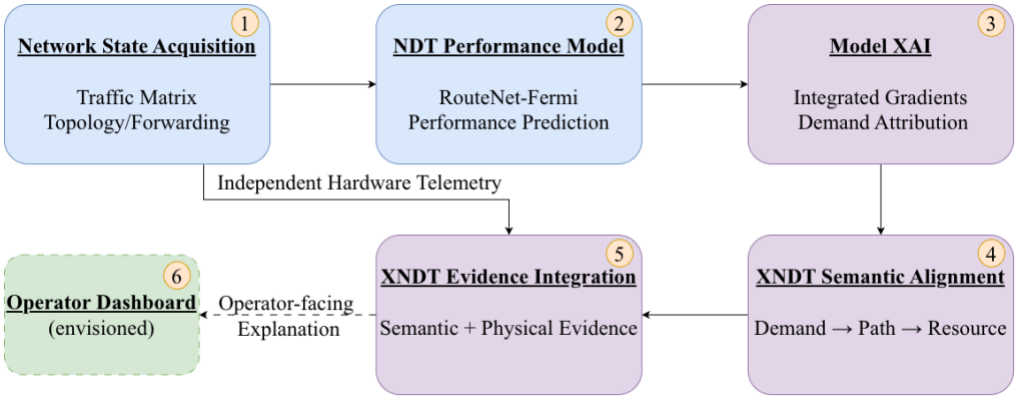


Fig. 1. XNDT architecture and its relation to the physical evaluation environment. The purple components (3)-(5) constitute the XNDT explanation pipeline, component (6) remains envisioned.

3.2 XNDT Explanation Pipeline

The explanation pipeline comprises three steps corresponding to components (3)-(5). First, (3) Model XAI identifies prediction-relevant features or entities. Second, (4) XNDT Semantic Alignment associates attributed model features or entities with identifiable network entities and reconstructs the relevant forwarding dependencies. Graph-XAI attribution methods identify features or graph structures that are relevant to a model prediction rather than establishing causal relationships in the underlying physical system [7, 8]. XNDT therefore does not treat model-level attribution as causal proof. Third, (5) XNDT Evidence Integration cross-references the aligned evidence with independent network telemetry while preserving the provenance of each evidence item.

In the proof of concept, (4) XNDT Semantic Alignment maps the demands attributed by (3) Model XAI to their installed directed forwarding paths. Overlapping paths are then used to identify network resources shared by the attributed demands. For these resources, (5) XNDT Evidence Integration retrieves the corresponding independent hardware telemetry, such as interface throughput and queue-drop counters, and combines it with the attribution and forwarding evidence. The proof of concept evaluates this structured explanation evidence; operator-facing natural-language rendering remains outside the current scope.

4 Physical Evaluation Environment

The evaluation uses an existing high-fidelity hardware testbed with eight Juniper QFX5110-48S data-center switches interconnected as a physical full mesh by 10 Gb/s links. Four traffic-generation servers use Cisco TRex [18] over 100 Gb/s links to generate reproducible source-destination demands at configurable rates. Selectively enabling links realizes different logical topologies and forwarding configurations. In this work, high fidelity denotes physical switching

hardware, controlled traffic at physical link rates and hardware-level interface and queue telemetry, rather than a complete reproduction of a production network.

The testbed supports *Simple Network Management Protocol* (*SNMP*), Junos streaming telemetry and *Software-Defined Networking* (*SDN*) control-plane information. In the proof of concept, traffic demands are taken directly from the TRex workload configuration, while the active forwarding state is derived from the SDN controller and installed switch rules. For experimental evaluation, the recorded network state and performance observations are combined with GNN predictions and model-level attribution evidence. Performance observations include interface and queue telemetry, including *Random Early Detection* (*RED*) drop counters, together with packet-loss measurements. The RED counters record queue-management packet drops and are collected through Junos streaming telemetry.

These measurements provide an empirical reference for assessing semantic agreement and capture device-specific queueing and buffering effects that may not be represented in the simulation-based datasets commonly used to develop RouteNet models [3–5].

5 Proof-of-Concept Evaluation

5.1 Experimental Scenario and Implementation

The proof of concept follows a controlled-intervention design using a logical dumbbell topology instantiated on the physical full-mesh testbed by selectively configuring the forwarding paths. Two forward demands, d1 and d2, share a 10 Gb/s bottleneck link (Switch A → Switch B). A third demand, d3, traverses the same physical connection in the reverse direction and serves as a reverse-direction control. Because the two directions use separate output queues, changes in d3 are not expected to affect forward-direction congestion. Figure 2 summarizes the resulting logical topology and traffic directions.

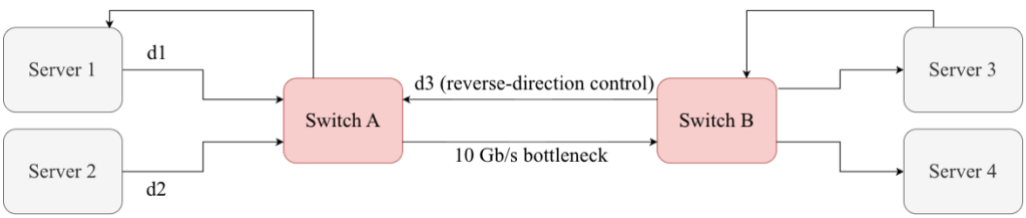


Fig. 2. Logical dumbbell scenario used for the proof of concept. Forward demands d1 and d2 share a 10 Gb/s bottleneck, while control demand d3 traverses the reverse direction.

We use RouteNet-Fermi [5] for per-demand packet-loss prediction. *Integrated Gradients* (*IG*) [16] attributes predictions to offered demand loads. We use 512 trapezoidal intervals, a setting fixed on the validation data to satisfy the pre-specified 10^{-3} completeness tolerance for all validation explanations. Hardware telemetry is excluded from model and attribution inputs and used only as independent evaluation evidence.

The 126 scenarios vary aggregate offered load from 20% to 140% of bottleneck capacity together with the d1/d2 load split and the d3 control rate, with three 60-s repetitions each (378 runs). A scenario-level split gives 98/14/14 training, validation and test scenarios (294/42/42 runs), keeping all repetitions together. Model and XAI configuration are fixed before evaluating the test split. Each run records TRex workload counters together with the corresponding forwarding state and hardware telemetry.

Predictive metrics are computed at the run-demand level. The test split therefore contains (42×3=126) run-demand observations, and the positive-loss subset is defined by strictly positive measured loss. Within each scenario, model inputs are identical across repetitions. Attribution is therefore evaluated once per test scenario using a deterministic representative run. For each of the 14 test scenarios, we explain the two forward prediction targets d1 and d2, resulting in 28 scenario-target pairs. Each pair is evaluated using two IG reference states, zero traffic and a matched loss-free low-load state selected from the validation split, yielding 56 scenario-target-reference explanations. Occlusion is evaluated for the same 56 combinations.

5.2 Evaluation Criteria

The evaluation addresses three questions: (i) Predictive fidelity: Does the GNN predict the induced performance degradation? (ii) Attribution quality: Does the explanation method identify the traffic demands associated with the bottleneck? (iii) Semantic agreement: Does XNDT relate this model evidence to the corresponding forwarding dependencies and measured network state?

These criteria are evaluated separately rather than collapsed into a single explanation score. Predictive fidelity is quantified by *mean absolute error* (*MAE*) and *root mean squared error* (*RMSE*), with positive-loss MAE additionally compared against an always-zero and a capacity-based overload baseline. Attribution quality is assessed using d1 and d2 as the known forward-demand set and d3 as the reverse-direction control. Robustness is assessed across zero-traffic and low-load IG reference states and independently checked using occlusion. Finally, semantic agreement is assessed by mapping attributed demands to the installed directed paths and cross-referencing the resulting shared resource with independent interface throughput and RED drop counters.

5.3 Results

Regarding (i), predictive fidelity, RouteNet-Fermi predicts per-demand packet loss with a run-level MAE of 0.0017 and an RMSE of 0.0050. For the strictly positive measured loss subset (32 of 126 run-demand observations), the MAE is 0.0068, compared with 0.0184 for a capacity-based overload heuristic and 0.1190 for an always-zero baseline. At or below the bottleneck capacity, measured and predicted forward loss remain negligible. For the five overloaded scenarios, the mean predicted forward-loss ratio is 0.1263 compared with a measured ratio of 0.1269.

For (ii), attribution quality, IG identifies the two demands associated with the bottleneck. At or below the bottleneck capacity, absolute demand attribution is negligible. Averaged across the five congested test scenarios, both prediction targets, and both IG reference states, d1 accounts for 45.2% and d2 for 54.8% of relative attribution, while the reverse-direction control demand d3 receives no attribution. Across all 28 test scenario-target pairs, both IG reference states preserve the separation between the forward-demand set {d1, d2} and control demand d3. The corresponding occlusion checks identify the same relevant-demand set in all 56 scenario-target-reference comparisons.

For (iii), semantic agreement, XNDT maps these demand-level attributions to the installed forwarding state. In all test scenarios, d1 and d2 converge on the shared directed 10 Gb/s bottleneck Switch A → Switch B, whereas d3 traverses the reverse direction. Independent hardware counters show a stable forward octet-counter throughput plateau and increasing RED drop counts under overload, while no RED drops are recorded below the capacity threshold or in the reverse direction. This physical evidence is consistent with the resource identified by the attribution-to-forwarding mapping.

6 Limitations

The proof of concept provides initial feasibility evidence for the proposed explanation pipeline but does not constitute a comprehensive validation of XNDT. The eight-switch testbed enables controlled and reproducible experiments but cannot reproduce the scale and operational diversity of production networks. The observed queueing, buffering and congestion behavior is also specific to the used hardware and software configuration and may not generalize across vendors or device generations. Accordingly, the measurements provide an empirical reference for the controlled testbed scenario but should not be interpreted as vendor-independent ground truth.

The experimental design also simplifies the explanation task. Traffic demands are specified by the Cisco TRex configuration, while the forwarding state is controlled and observed through SDN. Production environments may provide imperfect network-state information and may involve dynamic routing or multipath forwarding.

Finally, model-level attribution identifies features or entities relevant to a prediction but does not by itself establish causal relationships [7, 8]. Explanation quality remains dependent on the fidelity of the GNN prediction, the selected explanation method and the accuracy of the available network-state information. The operational usefulness of XNDT has not yet been assessed through systematic studies with network operators. XNDT should therefore be understood as human-in-the-loop decision support rather than an autonomous control mechanism.

7 Future Work

Future work will extend the initial proof of concept along three dimensions. First, we will evaluate the GNN and explanation pipeline across more diverse network scenarios, assessing predictive generalization, attribution stability and semantic agreement beyond the controlled single-bottleneck setting.

Second, different Graph-XAI paradigms will be compared with respect to their fidelity to the model prediction, robustness and practical suitability for network-semantic interpretation [9–11, 17]. XNDT Semantic Alignment and XNDT Evidence Integration will be extended to accommodate different model entities and explainer outputs.

Third, operator-facing explanation representations will be technically evaluated with respect to traceability, interpretability and actionability. In later stages, XNDT may support explanation-informed network adaptations through the SDN control plane, initially subject to operator review before any progression toward partially automated or closed-loop operation.

8 Acknowledgements

This work is supported by the Ministry of Science, Research and the Arts of Baden-Württemberg (MWK) within the BWNet2.0 project.

References

[1] M. Ferriol-Galmés *et al.*, “Building a Digital Twin for network optimization using Graph Neural Networks,” *Comput. Netw.*, vol. 217, p. 109329, Nov. 2022, doi: 10.1016/j.comnet.2022.109329.

[2] C. Zhou *et al.*, “Network Digital Twin: Concepts and Reference Architecture.” Accessed: Aug. 12, 2026. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch, draft-irtf-nmrg-network-digital-twin-arch-13, work in progress.

[3] K. Rusek, J. Suárez-Varela, P. Almasan, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 10, pp. 2260–2270, Jun. 2020, doi: 10.1109/JSAC.2020.3000405.

[4] M. Ferriol-Galmés *et al.*, "RouteNet-Erlang: A Graph Neural Network for Network Performance Evaluation," in *IEEE INFOCOM 2022 - IEEE Conf. Comput. Commun.*, May 2022, pp. 2018–2027. doi: 10.1109/INFOCOM48880.2022.9796944.

[5] M. Ferriol-Galmés *et al.*, "RouteNet-Fermi: Network Modeling With Graph Neural Networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 3080–3095, Dec 2023, doi: 10.1109/TNET.2023.3269983.

[6] C. Güemes-Palau, M. Ferriol-Galmés, J. Paillisse-Vilanova, A. López-Brescó, P. Barlet-Ros, and A. Cabellos-Aparicio, "RouteNet-Gauss: Hardware-Enhanced Network Modeling with Machine Learning," *IEEE Trans. Netw.*, vol. 34, pp. 3840–3853, Feb 2026, doi: 10.1109/TON.2026.3668972.

[7] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in Graph Neural Networks: A Taxonomic Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5782–5799, May 2023, doi: 10.1109/TPAMI.2022.3204236.

[8] J. He, A. Rafiey, G. Mishne, and Y. Wang, "Explaining GNN Explanations with Edge Gradients," in *Proc. 31st ACM SIGKDD Conf. Knowl. Disc. Data Mining V.2*, New York, NY, USA: Association for Computing Machinery, Aug. 2025, pp. 884–895. doi: 10.1145/3711896.3736947.

[9] D. Luo *et al.*, "Parameterized Explainer for Graph Neural Network," in *Advances in Neural Information Processing Systems, vol. 33*, 2020, pp. 19620–19631.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks", *Int. Conf. Lear. Repr.*, OpenReview, Feb. 2018. Accessed: Jun. 23, 2026. [Online]. Available: https://openreview.net/forum?id=rJXMpikCZ

[11] A. Lucic, M. ter Hoeve, G. Tolomei, M. de Rijke, and F. Silvestri, "CF-GNNExplainer: Counterfactual Explanations for Graph Neural Networks," in *Proc. 25th Int. Conf. Artif. Intell. Stat.*, PMLR, vol. 151, 2022, pp. 4499–4511.

[12] M. Nandan, S. Mitra, and D. De, "GraphXAI: a survey of graph neural networks (GNNs) for explainable AI (XAI)," *Neural Comput. Appl.*, vol. 37, no. 17, pp. 10949–11000, Mar. 2025, doi: 10.1007/s00521-025-11054-3.

[13] S. Chowa, E. Rondeau, M. Maimour, M. S. Hossain, K. Andersson, and S. Kubler, "Explainable AI Framework for Deep Learning-based Network Digital Twins," in *2025 12th Int. Conf. Fut. Inter. of Things and Cloud (FiCloud)*, Aug. 2025, pp. 76–83. doi: 10.1109/FiCloud66139.2025.00019.

[14] C. Zilli, A. Sacco, D. Monaco, O. Okafor, F. Esposito, and G. Marchetto, "Inferring Visibility of Internet Traffic Matrices Using eXplainable AI," in *NOMS-2024 IEEE Netw. Oper. Manag. Symp.*, May 2024, pp. 1–6. doi: 10.1109/NOMS59830.2024.10575173.

[15] J. Suárez-Varela *et al.*, "Graph Neural Networks for Communication Networks: Context, Use Cases and Opportunities," *IEEE Netw.*, vol. 37, no. 3, pp. 146–153, May 2023, doi: 10.1109/MNET.123.2100773.

[16] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of the 34th Int. Conf. Mach. Learn.*, PMLR, vol. 70, 2017, pp. 3319–3328.

[17] S. Patel, D. Han, N. Narodytska, and S. A. Jyothi, "Agua: A Concept-Based Explainer for Learning-Enabled Systems," in *Proc. ACM SIGCOMM 2025 Conf.*, New York, NY, USA: Association for Computing Machinery, Aug. 2025, pp. 329–346. doi: 10.1145/3718958.3754341.

[18] TRex Team, Cisco Systems, "TRex Traffic Generator." Accessed: Jul. 30, 2026. [Online]. Available: https://trex-tgn.cisco.com/