Network Digital Twin for IGP Weight Optimization Demo

Mohamed Zalat
Systems and Computer Engineering
Carleton University
Ottawa, Canada
mohamedzalat@sce.carleton.ca

Maede Davoudzade
Systems and Computer Engineering
Carleton University
Ottawa, Canada
maededavoudzade@sce.carleton.ca

Chris Barber
Blue Planet
Ciena
Ottawa, Canada
cbarber@ciena.com

David Krauss
Advanced Development
Visionary Technologies, Inc.
Baltimore, USA
dkrauss@visionaryinc.net

Babak Esfandiari
Systems and Computer Engineering
Carleton University
Ottawa, Canada
babak@sce.carleton.ca

Thomas Kunz
Systems and Computer Engineering
Carleton University
Ottawa, Canada
tkunz@sce.carleton.ca

***Abstract*—Current research on Network Digital Twins (NDT) focuses on building models that predict the behavior of the network on simulations rather than building a complete NDT that automatically monitors a physical network and optimizes it according to its current state and traffic flows. In this proposal, we demo a NDT for Interior Gateway Protocol (IGP) weight optimization on a physical network of 7 Cisco routers and an IXIA generator connected via Ethernet links. We will show how a link failure in this network results in a large packet loss, and how our NDT automatically optimizes the IGP weights of the network to reduce packet loss. Our demo presents a practical implementation of a NDT, where the audience can see a self-adapting network without human interference.**

***Index Terms*—Network Digital Twins (NDT), Traffic Engineering, IGP Weight Optimization, RouteNet-Fermi, Fine-tuning**

I. Introduction

Digital Twins (DT) of communication networks, known as Network Digital Twins (NDT), are an idea that has been theorized in recent years with a current work-in-progress Internet Research Task Force (IRTF) draft to standardize them [1]. Research efforts on NDTs have mostly focused on developing models that predict the behavior of communication networks rather than creating a NDT that monitors a physical network and optimizes it in real-time [2]–[7]. Creating such a NDT poses many challenges, such as continuously monitoring the topology and interface information from the network and applying configurations automatically. Furthermore, a NDT that relies on Machine Learning (ML) models can sometimes inaccurately predict the behavior of the network resulting in undesired effects that can negatively impact the physical network. Using a NDT to optimize a network allows us to adapt the network to changes, such as link failures and traffic pattern changes. At the time of writing, no published study has implemented a real-time NDT for Interior Gateway Protocol (IGP) weight optimization on a physical network.

We acknowledge the support of Ciena and the Natural Sciences and Engineering Research Council of Canada (NSERC).

We demo a NDT for IGP weight optimization based on average traffic delay [8] on a physical network consisting of 7 Cisco routers and an IXIA generator. We run our NDT on a physical network to observe how it adapts the IGP weights of the network after a link failure to minimize the average delay and packet loss of the traffic flows.

II. Online Learning NDT for IGP weight optimization

A NDT is a DT of a communication network [2]. It encompasses the virtual model of the network, its optimization algorithm(s), and all interfaces involved in the bidirectional data flow between the virtual model and the physical model. The virtual model in a NDT provides a framework for testing different traffic flow and network configuration scenarios without affecting the physical network. Statistics and metrics such as the average delay, loss, and jitter per flow can be predicted using the virtual model for a given scenario. The optimizer component of the NDT collects data from the physical network to try different network configurations using the virtual model and can apply a configuration to the network once a better configuration is found.

In Figure 1, we present the architecture of our online learning NDT for IGP weight optimization which allows us to test IGP weight configurations without affecting the traffic of the real network. Details on the architecture of our NDT for IGP weight optimization can be found in [8]. We use RouteNet-Fermi as the virtual model of our NDT to predict the traffic delays and losses, since it was shown to have fast and accurate predictions [4]. Our NDT continuously polls the network devices using Secure Shell Protocol (SSH) for information on the current topology of the network including the bandwidths, IGP weights and queue sizes of the interfaces in the network. It also collects information on the traffic flows going through the network and their respective delays and losses from IXIA. Then, it compares the delay and loss measurements from

978-3-903176-66-9 ©2024 IFIP

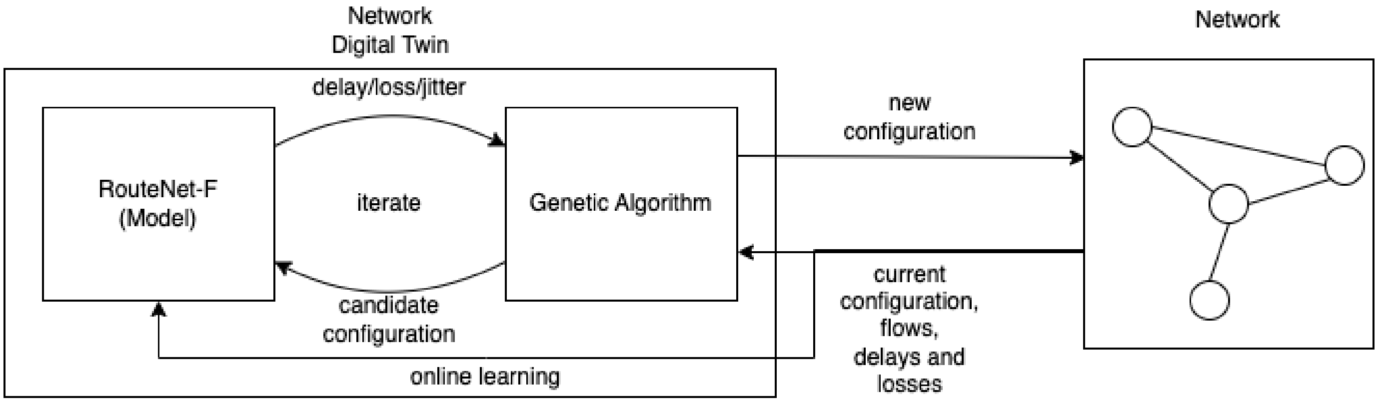
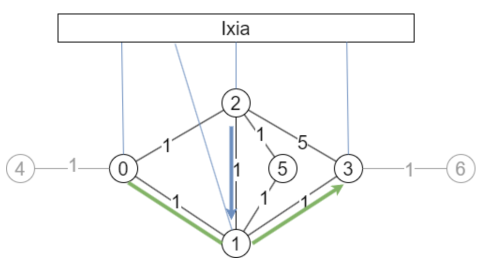


Fig. 1: Online learning NDT for Quality of Service (QoS) based IGP weight optimization.

the network to the predictions of RouteNet-Fermi to decide whether it should train RouteNet-Fermi or optimize the IGP weights of the physical network. If the measurements from the network are not positively correlated with the predictions of RouteNet-Fermi, with a statistical significance α, it continuously trains RouteNet-Fermi on new samples collected from the network until it is positively correlated (refer to [8] where we discuss the speed at which the model converges and we begin optimizing IGP weights). Each new sample is collected when a change in the network occurs (including traffic flow changes). Once the measurements from the network and the predictions of RouteNet-Fermi are sufficiently correlated, our NDT uses the genetic algorithm described in [9] to try different IGP weight configurations on RouteNet-Fermi to find a better configuration for the current scenario. Once a better IGP weight configuration is found, it is then applied to the physical network by connecting to the routers through SSH and applying the weights to the interfaces.

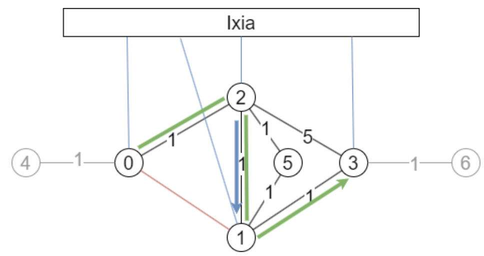
III. Demo Setup



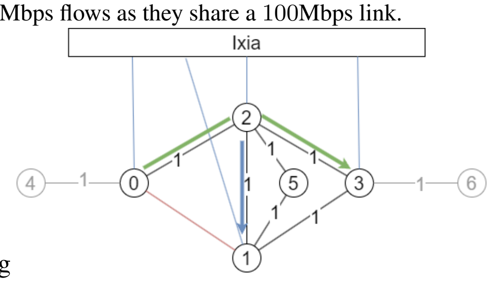
(2).png

Fig. 2: The topology of the physical testbed. The numbered nodes here are routers and the links between them are L3 links. The numbers on each link denote the IGP weight of the link. The blue and green arrows are 80Mbps constant bit-rate flows.

We perform the demo on an existing physical testbed provided by Ciena [10] with 7 Cisco 7200 routers and an *IXIA* traffic generator where we can specify the traffic flowing through the network (refer to Figure 2). The testbed is part of a larger network and we do not use the peripheral routers



(a) We take down the link from node 0 to 1 resulting in a large packet loss and delay for our two 80Mbps flows as they share a 100Mbps link.



(1).png

(b) The predicted IGP weights and the new traffic routes after one cycle of the NDT. The result is that both traffic flows have 0% packet loss.

Fig. 3: An example scenario: after breaking the link from node 0 to 1, the NDT detects the topology change and optimizes the IGP metrics of the network to fix the network state.

4 and 6 in Figure 2. All the interfaces (links) are Gigabit interfaces that are throttled to a bandwidth of 100Mbps and configured with a queue depth of 1000 frames to congest the network easily. We use a pre-trained RouteNet-Fermi model on the Fat-Tree-128 dataset [4] for our NDT.

We generate constant bit-rate flows through our network using IXIA to observe their delay and packet loss, an example scenario is demonstrated in Figure 2. We then perform a change in the network, such as taking down a link as in Figure 3a and see this reflected on the Route Optimization and Analysis (ROA) tool by Ciena [11]. An example screenshot from the ROA tool is provided in Figure 4 with the metrics

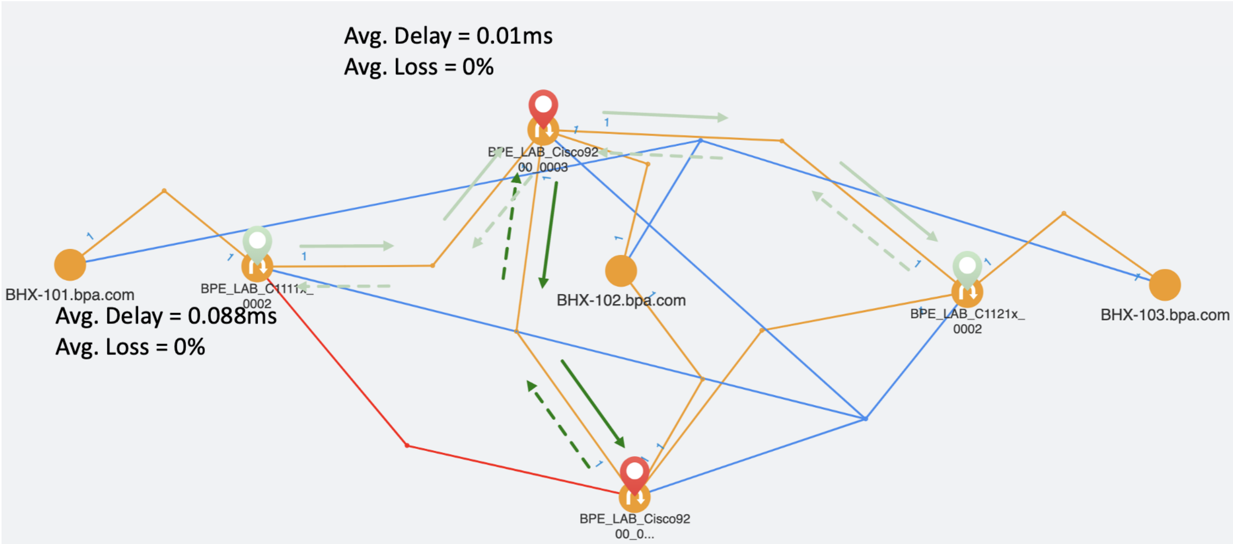


Fig. 4: A screenshot from the ROA tool with the traffic QoS metrics labelled. The orange nodes represent the routers, the orange links are up links, and the red and blue links are down links. There are two flows in this image denoted by the dark and light green arrows, and the numbers on the links are the IGP weights of the interfaces.

of the flows labelled from IXIA. Once the NDT detects a change in the network, it optimizes the IGP weights of the network to minimize the average delay of the network's traffic flows. The optimal configuration is then applied to the network automatically to improve the average delay in the network (refer to Figure 3 for an example). The IGP configuration of the network is displayed in the ROA tool as it is modified by the NDT, and the traffic QoS metrics are shown through IXIA.

Our current NDT implementation takes about 4 minutes to collect information about the topology from the routers, and 2 minutes to apply IGP weight configurations through SSH on our current network topology. To scale the NDT to a larger production network, the NDT should leverage protocols such as NetFlow [12] and Simple Network Management Protocol (SNMP) to collect traffic flow and topology information.

IV. Conclusion

We demo a NDT for optimizing the IGP weights of a physical network for the QoS metrics of its traffic flows. The NDT collects real-time data from the physical network, compares it with predictions from its virtual model, and adjusts IGP weights through a genetic algorithm. This demo, on a physical testbed, presents a practical implementation of a NDT, where the audience can see a self-adapting network. The NDT detects a link failure, adjusts the IGP weights, and improves the network's traffic packet loss in the demo. This demonstration serves as a showcase of the effectiveness of NDTs in automated real-time network optimization.

References

[1] C. Z. et al., "Network Digital Twin: Concepts and Reference Architecture," Internet Engineering Task Force, Internet-Draft draft-irtf-nmrg-network-digital-twin-arch-05, Mar. 2024, work in progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch/05/

[2] P. A. et al., "Digital Twin Network: Opportunities and Challenges," *CoRR*, vol. abs/2201.01144, 2022. [Online]. Available: https://arxiv.org/abs/2201.01144

[3] M. F. G. et al., "RouteNet-Erlang: A Graph Neural Network for Network Performance Evaluation," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, London, United Kingdom, May 2-5, 2022*. IEEE, 2022, pp. 2018–2027.

[4] ——, "RouteNet-Fermi: Network Modeling With Graph Neural Networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 3080–3095, 2023.

[5] S. Xiao, D. He, and Z. Gong, "Deep-Q: Traffic-driven QoS Inference using Deep Generative Network," in *Proceedings of the 2018 Workshop on Network Meets AI & ML, NetAI@SIGCOMM 2018, Budapest, Hungary, August 24, 2018*. ACM, 2018, pp. 67–73.

[6] 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 Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2260–2270, 2020.

[7] M. F. G. et al., "Building a Digital Twin for Network Optimization Using Graph Neural Networks," *Computer Networks*, vol. 217, p. 109329, 2022.

[8] M. Zalat, M. Davoudzade, C. Barber, D. Krauss, B. Esfandiari, and T. Kunz, "A Practical Network Digital Twin for IGP Weight Optimization," in *2024 20th International Conference on Network and Service Management (CNSM)*, 2024.

[9] M. Zalat, C. Barber, D. Krauss, B. Esfandiari, and T. Kunz, "Network Routing Optimization Using Digital Twins," in *Companion Proceedings of the 16th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling and the 13th Enterprise Design and Engineering Working Conference: DTE 2023, Vienna, Austria, November 28 - December 01, 2023*, ser. CEUR Workshop Proceedings, vol. 3645. CEUR-WS.org, 2023. [Online]. Available: https://ceur-ws.org/Vol-3645/dte4.pdf

[10] C. Corporation, "A networking systems, services, and software company — ciena.com," https://www.ciena.com/.

[11] Ciena, "Route Optimization and Analysis (ROA) - Blue Planet — blueplanet.com," https://www.blueplanet.com/products/route-optimization.html, [Accessed 04-07-2024].

[12] R. H. et al., "Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.