

# Towards more realistic network models based on Graph Neural Networks

Arnau Badia-Sampera

Barcelona Neural Networking Center  
Universitat Politècnica de Catalunya

Paul Almasan

Barcelona Neural Networking Center  
Universitat Politècnica de Catalunya

Pere Barlet-Ros

Barcelona Neural Networking Center  
Universitat Politècnica de Catalunya

José Suárez-Varela

Barcelona Neural Networking Center  
Universitat Politècnica de Catalunya

Krzysztof Rusek

Barcelona Neural Networking Center  
AGH University of Science and Technology

Albert Cabellos-Aparicio

Barcelona Neural Networking Center  
Universitat Politècnica de Catalunya

## ABSTRACT

Recently, a Graph Neural Network (GNN) model called RouteNet was proposed as an efficient method to estimate end-to-end network performance metrics such as delay or jitter, given the topology, routing, and traffic of the network. Despite its success in making accurate estimations and generalizing to unseen topologies, the model makes some simplifying assumptions about the network, and does not consider all the particularities of how real networks operate. In this work we extend the architecture of RouteNet to support different features on forwarding devices, specifically we focus on devices with variable queue sizes, and we experimentally evaluate the accuracy of the extended RouteNet architecture.

## CCS CONCEPTS

• **Networks** → **Network performance evaluation**; • **Computing methodologies** → *Machine learning*.

## KEYWORDS

Graph Neural Networks, Network Modeling

## 1 INTRODUCTION

Network modeling is crucial for building solutions that efficiently optimize and manage computer networks, especially in the context of the knowledge-defined networking paradigm [5]. Traditional methods like Queueing Theory often fail to provide accurate models for complex real-world scenarios [8]. On the other hand, packet-level simulators are able to produce accurate performance predictions but at a high computational cost.

In the Deep Learning field, Graph Neural Networks (GNNs) have been recently proposed as effective models for graph-structured data. In the context of networking, RouteNet [6] is a pioneering GNN architecture tailored for computer network modeling, which is able to model the relations between an input topology, a (source-destination) routing scheme and an end-to-end traffic matrix, and estimate the resulting per-source/destination performance (e.g., delay, jitter). In short, RouteNet achieves an accuracy comparable to packet-level network simulators with a very low computational cost.

However, RouteNet makes some simplifying assumptions about how actual networks operate and, at the time of this writing, it cannot model all the particularities of real-world networks (e.g. different forwarding behaviors, queues sizes). In this work, we extend the architecture of RouteNet to model different features on forwarding devices. Particularly, we evaluate the capacity of our model to make predictions in scenarios where forwarding devices have different queue sizes. This work represents the foundation for modeling any node-related features in RouteNet, with the ultimate goal of developing more realistic GNN-based network models.

## 2 INTRODUCING THE *NODE* ENTITY

RouteNet is a GNN-based model that produces estimations of source-destination network performance metrics. The main advantage of GNN models versus other neural network architectures (fully-connected, convolutional, recurrent) is that they can understand and exploit the relational patterns between the elements of a graph [2], and the fundamental way to represent network-related data is in the form of graphs. This explains that RouteNet is able to generalize to network topologies, routing and input traffic unseen during training.

The fundamental idea behind RouteNet relies on the reciprocal relationships between the state of paths and links in networks. Intuitively, a path is influenced by the links it traverses, and a link, in turn, is influenced by all the paths that cross that link. Thus, RouteNet is fed with some initial path and link states (encoded as fixed-size vectors) and executes an iterative message-passing algorithm among these states according to the input topology and routing scheme. At every iteration the states of links and paths are updated with the new information combined according to graph structure. After a certain number of iterations, the paths' states are used to estimate end-to-end network performance metrics [6].

However, with this architecture, RouteNet is not able to model the particular characteristics of different forwarding devices (e.g., scheduling, queue sizes), which can have an important impact on the network performance. A natural way to introduce the concept of network node (i.e., forwarding device) is to incorporate a node entity into RouteNet's architecture. Similarly to the states associated to every path and link, we can introduce a state associated to each network node. In Fig. 1, we provide a visual representation of the

message passing in the extended RouteNet architecture. In this new message passing architecture, the state of every node (encoded in a fixed-size vector) is updated based on information from all the paths that traverse that node, first performing an element-wise summation of all the path states associated to the node and then using the result as an input to a recurrent neural network<sup>1</sup>. Also, instead of updating the path states with only link information, we leverage both the state of links and nodes. In the original RouteNet design, the path states are updated by a Recurrent Neural Network taking as input a sequence of link states related to every path. In contrast, in the new architecture we insert the node states in such sequences and interleave them with the link states related to every path (i.e., node1-link1-node2-link2...)<sup>2</sup>.

As in the original RouteNet, when the message passing ends we use the final path states to estimate end-to-end network performance metrics. This is done by applying a *readout function*, implemented as a feed-forward neural network, to each path state. Thus, the learning problem in the extended RouteNet consists in learning the optimal parameters of four functions:  $RNN_N$ ,  $RNN_P$ ,  $RNN_L$  (shown in Fig. 1), and the *readout function*.

### 3 EVALUATION WITH DIFFERENT QUEUE SIZES

For our experiments, we generated datasets with an in-house packet-level simulator in OMNeT++. We made simulations in two network topologies: the 14-node NSFNET topology [3] and the 24-node GEANT2 topology [1]. Our datasets include diverse combinations of forwarding devices with variable queue sizes, routing schemes and end-to-end traffic matrices.

We use as a reference RouteNet’s implementation in [7], which includes support for modeling different link capacities. In order to test the effectiveness of the modified architecture, we consider a scenario in which the queue sizes of forwarding devices can be either of standard size or only with support for 1 packet. We introduce the queue size information into the GNN model by encoding it in the states of every node. Similarly, the end-to-end traffic was already encoded in the path states and the link capacities into the states of links.

We train the model to estimate delays with 400,000 samples of GEANT2, and we evaluate it on 100,000 unseen samples of the same topology. Additionally, we also perform the evaluation on 100,000 samples of the NSFNET topology, to test the model’s generalization capabilities to different topologies. In Fig. 2, we show the CDF of the relative error produced by: (i) extended RouteNet on GEANT2, (ii) original RouteNet [7] on GEANT2, (iii) extended RouteNet on NSFNET, and (iv) original RouteNet on NSFNET.

The results show that the extended architecture is able to obtain much more accurate predictions with respect to the original version of RouteNet given that it incorporates information of queue sizes on nodes. Also, we can observe that it generalizes successfully on scenarios of NSFNET, despite it did not see any sample from this topology during training.

<sup>1</sup>A recurrent neural network is used to ease convergence during the message passing process [4].

<sup>2</sup>The source code of the model proposed is publicly available at: <https://github.com/knowledgedefinednetworking/Papers/wiki/Towards-more-realistic-network-models-based-on-GNN>

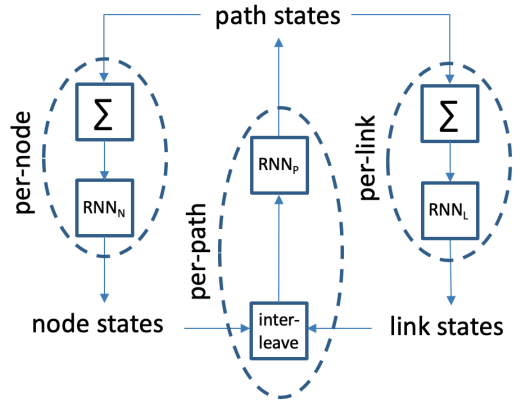


Figure 1: Diagram of Extended RouteNet message passing

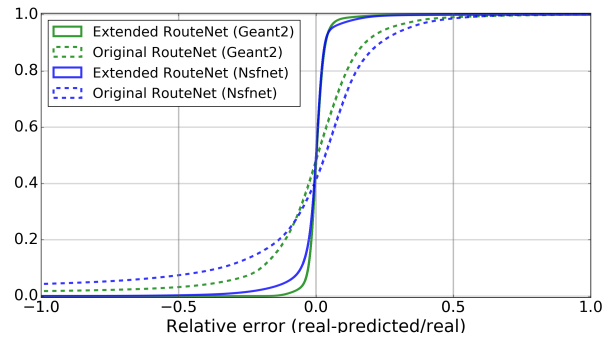


Figure 2: Cumulative Distribution Function (CDF) of the relative error in delay predictions

### 4 CONCLUSIONS

We propose an extension of RouteNet that is able to model different types of forwarding devices. Our experiments show high prediction accuracy even in unseen topologies. Also, this work aims to reveal the feasibility of extending GNN-based models to fit different network scenarios involving specific particularities present in real-world deployments.

### ACKNOWLEDGEMENTS

This work was supported by the Spanish MINECO under contract TEC2017-90034-C2-1-R (ALLIANCE), the Catalan Institution for Research and Advanced Studies (ICREA), FI-AGAUR grant by the Catalan Government and the AGH University of Science and by the Polish Ministry of Science and Higher Education with the subvention funds of the Faculty of Computer Science, Electronics and Telecommunications of AGH University. The research was also supported in part by PL-Grid Infrastructure.

### REFERENCES

- [1] Fernando Barreto, Emilio CG Wille, and Luiz Nacamura Jr. 2012. Fast emergency paths schema to overcome transient link failures in ospf routing. *arXiv preprint arXiv:1204.2465* (2012).
- [2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam

- Santoro, Ryan Faulkner, et al. 2018. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* (2018).
- [3] Xiaojun Hei, Jun Zhang, Brahim Bensaou, and Chi-Chung Cheung. 2004. Wavelength converter placement in least-load-routing-based optical networks using genetic algorithms. *Journal of Optical Networking* 3, 5 (2004), 363–378.
- [4] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).
- [5] Albert Mestres et al. 2017. Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 2–10.
- [6] Krzysztof Rusek, José Suárez-Varela, Albert Mestres, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2019. Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN. In *Proceedings of the 2019 ACM Symposium on SDN Research (SOSR)*. 140–151.
- [7] José Suárez-Varela, Sergi Carol-Bosch, Krzysztof Rusek, Paul Almasan, Marta Arias, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2019. Challenging the generalization capabilities of Graph Neural Networks for network modeling. In *Proceedings of the ACM SIGCOMM 2019 Conference Posters and Demos*. 114–115.
- [8] Zhiyuan Xu et al. 2018. Experience-driven networking: A deep reinforcement learning based approach. In *Proceedings of IEEE INFOCOM*.