

Original software publication



TopoHub: A repository of reference Gabriel graph and real-world topologies for networking research

Piotr Jurkiewicz

Institute of Telecommunications, AGH University of Krakow, Poland

ARTICLE INFO

Keywords:

Network topologies
Optical networks
Sndlib
Topology zoo

ABSTRACT

Networking research often requires realistic topologies to study performance and resilience. This project introduces TopoHub, an open repository of reference network topologies of variety of network sizes based on the Gabriel graph model. The accompanying Python package offers functionalities for topology generation, analysis, and integration with the Mininet network emulator. A web interface allows users to explore topologies, including visualization of link utilization under various traffic demands. The project aims to provide a comprehensive topology data source for networking researchers, enabling standardized benchmarking and reproducible research.

Code metadata

Current code version
Permanent link to code/repository used for this code version
Legal Code License
Code versioning system used
Software code languages, tools, and services used
Compilation requirements, operating environments & dependencies
Link to developer documentation/manual
Support email for questions

1.0
<https://github.com/ElsevierSoftwareX/SOFTX-D-23-00539>
MIT
Git
Python
NetworkX
<https://topohub.readthedocs.io>
piotr.jurkiewicz@agh.edu.pl

1. Motivation and significance

The Internet's topology has undergone extensive study for many years. However, the majority of research in this field has been focused on the logical level. Nevertheless, the usage of realistic physical-level topologies is necessary for research on several network aspects, starting with resilience and survivability, and ending with traffic engineering and multi-path routing. For example, in research on optical networks, it is crucial to provide realistic link lengths as they determine feasible modulations and, thus, link capacities. On the other hand, in traffic engineering and multi-path routing experiments, it is important to ensure a realistic node degree distribution as it significantly impacts the number of available transmission paths between nodes and, consequently, the location of congestions and achievable throughput.

Currently, there are two publicly available sources of network topologies used in networking research. The Internet Topology Zoo [1] was an attempt to create a database of real network topologies based on information provided by network operators on their websites and

in corporate reports. Particular attention was given to the correctness of node locations, successfully geocoding over 90% of nodes in 83% of the networks [2]. The database contains 261 network topologies. The second source is SNDLib [3], initially designed to provide network optimization problem test instances but later used as a source of network topologies as well [4–6]. It includes 26 topologies based on real physical networks and also provides information about traffic demands between nodes. Both projects were abandoned more than 10 years ago.

Despite the existence of these databases, there is also a demand for synthetic, but realistic topologies. In particular, there is a need for a homogeneous set of topologies with varying node counts to assess algorithms or systems performance across different network sizes. However, these synthetic topologies must closely mimic real topologies. In [7], the authors examined five real networks to determine which synthetic network model best approximates real networks. Their investigation

E-mail address: piotr.jurkiewicz@agh.edu.pl.

<https://doi.org/10.1016/j.softx.2023.101540>

Received 12 August 2023; Received in revised form 27 September 2023; Accepted 28 September 2023

Available online 10 October 2023

2352-7110/© 2023 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

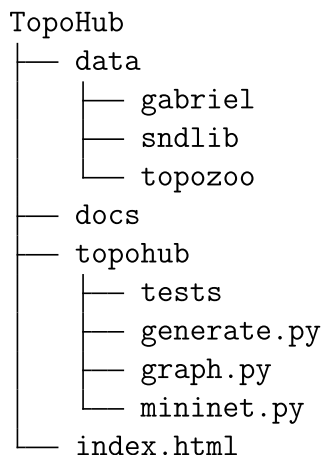


Fig. 1. The project structure.

revealed that Gabriel graphs [8] exhibit the closest resemblance to the structure of physical networks. As a result, Gabriel graphs have become increasingly used, especially in optical networking research [9–18].

However, there is currently no library of reference Gabriel graphs for networking research purposes. Each researcher generates their own Gabriel graph topologies as required for their specific research needs. This individualized approach introduces variations in parameters, making it difficult to compare research findings across different studies. The goal of this project is to fill this gap and provide an open repository of reference network topologies of various sizes based on the Gabriel graph model.

2. Software architecture

The project is stored on the GitHub platform¹ and consists of three main components:

- The topology repository, located in the `data` directory, containing topology definitions in the `networkx` JSON node-link graph format and their SVG visualizations.
- The Python package `topohub`, which includes scripts for generating topology graphs and analyzing their properties, along with the `mininet` module, which can be imported into Mininet to automatically use topologies from the repository.
- The web interface, built as a single-page web app in the `index.html` file, which allows browsing, visualization, and downloading of topologies.

Additionally, the project repository contains (see Fig. 1):

- Python API reference documentation located in the `docs` directory, accessible at: <https://topohub.readthedocs.io>
- Automated testing scripts located in the `topohub/tests` directory, which run on each commit and pull requests through GitHub actions.

3. Software functionalities

3.1. Topology repository

The topology repository contains 200 Gabriel graph topologies, with linearly increasing sizes ranging from 25 to 500 nodes, in steps of 25 nodes. For each network size, ten different topologies were

generated, each with a different, but controlled seed. This ensures their reproducibility.

Two constraints were applied to the generated Gabriel graphs. The positions of nodes (and thus distances between them) were scaled in such a way as to ensure the average link length of approximately 100 km. Additionally, we imposed a limitation on the minimal distance between nodes, which was set to 25 km. The number of graph edges was not considered as an input parameter because, in Gabriel graphs, it depends on the location of vertices and cannot be directly controlled.

The definition of each topology is stored in a JSON file constructed according to the `networkx` node-link format. This JSON object contains the following fields:

- The `graph` field, consisting of: the topology name, the `demands` object holding demands between pairs of nodes (if specified by the original source), and the `stats` object, containing pre-calculated topology statistical properties.
- The `nodes` field, containing a list of nodes in the topology, along with their positions, stored as latitude and longitude pairs in the `pos` arrays.
- The `links` field, containing a list of links in the topology, defined by `source` and `target` node names. The links objects also contain the `dist` field providing the link length in kilometers, and `ecmp_fwd` and `ecmp_bwd` fields. They contain pre-calculated links percentage utilization in forward and backward directions for the case of ECMP shortest path routing being used in the network for different demands models.

In addition to the synthetic Gabriel graph topologies, we also generated JSON definitions and SVG visualizations for topologies from the Internet Topology Zoo [1] and SNDLib [3] and also provide them in the repository. This gives the possibility to explore their properties using the web interface and allows using them interchangeably with the synthetic Gabriel graph topologies, for example in Mininet with the help of the automatic import feature.

In Table 1 we present a comparison of properties between real physical transport networks from the SNDLib repository and artificially generated Gabriel graph topologies of similar sizes. For Gabriel graphs, we provide the mean values of parameters calculated across all 10 graphs of the selected size provided in the repository, along with the calculated standard deviation. It can be observed that the parameters between respective pairs (nobel-eu and Gabriel 25) and (germany50 and Gabriel 50) of topologies are very similar.

3.2. Python package

The Python package consists of three Python modules: `generate`, `graph`, and `mininet`. The `generate` executable module can be used to make topology JSON definition and SVG visualization files. It can generate synthetic Gabriel graphs of a given size, or create topology definitions from the original topology files downloaded from the SNDLib or Topology Zoo.

The `graph` module contains functions for performing operations and calculations on network graphs. In particular, it allows determining all shortest paths and all disjoint paths between any node pair in a network. It uses Dijkstra and Edmonds-Karp algorithms implementations provided by the `networkx` library [19] for these purposes, respectively. Based on the computed disjoint paths, it calculates statistics of paths between all node pairs in the network, including the number of disjoint shortest paths and the number of all disjoint paths. The `graph` module also offers functionalities for calculating topology statistical properties and generating their SVG visualizations.

The package also includes a `mininet` module that can be imported into the popular network emulator Mininet [20], enabling the automatic usage of the topologies from the repository. The package, along with JSON topology definitions, can be installed from the Python Package Index (PyPI) using the `pip install topohub` command.

¹ <https://github.com/piotrjurkiewicz/topohub>

Table 1
Comparison of topologies properties.

Topology source	Gabriel	SNDLib	Gabriel	SNDLib
Topology name	25 (all)	nobel-eu	50 (all)	germany50
Number of nodes	25	28	50	50
Number of links	41.10 ± 1.97	41	88.10 ± 5.26	88
Avg. vertex degree	3.29 ± 0.16	2.93	3.52 ± 0.21	3.52
Max. vertex degree	5.40 ± 0.52	5	6.00 ± 0.82	5
Avg. link length	96.95 ± 9.45	416.11	97.04 ± 5.42	100.71
Diameter (by hops)	7.30 ± 0.82	8	10.40 ± 0.84	9
Avg. number of disjoint shortest paths	1.19 ± 0.05	1.20	1.16 ± 0.05	1.23
Max. number of disjoint shortest paths	2.40 ± 0.52	2	2.70 ± 0.48	3
Avg. number of all disjoint paths	2.54 ± 0.18	2.61	2.80 ± 0.25	3.19
Max. number of all disjoint paths	4.60 ± 0.52	4	5.50 ± 0.71	5

3.3. Web interface

The web interface was built as a single-page web application and is contained in the `index.html` file. This interface facilitates browsing, visualization, and downloading of topologies. It is automatically deployed on each commit using GitHub Pages infrastructure and can be accessed at <https://www.topohub.org>.

A distinctive feature of the web interface is the visualization of link utilization for scenarios where ECMP (Equal-Cost Multi-Path) shortest path routing is utilized in the network. This reflects the current state of the art in networking, as most interior gateway routing protocols (e.g., OSPF or IS-IS) inherently provide multiple equal-cost shortest paths. The web interface offers visualization for three different traffic demand models:

- **uniform**, which assumes a constant traffic demand between all pairs of nodes,
- **degree**, in which the demand between nodes is proportional to the product of the degrees of these nodes, and
- **original**, based on the original traffic demand matrices provided by the SNDLib project (applicable only for SNDLib topologies).

Users have the capability to select their desired traffic model for visualization from a dropdown menu. The link utilization is represented on the topology graph by varying the width, color, or opacity of the links, with these visual attributes adjustable by the user. By hovering the mouse over a link, users can obtain the exact percentage utilization values. Furthermore, a histogram illustrating the 20 most utilized links is provided, and users can explore the details by hovering the mouse over the bars.

4. Illustrative examples

The TopoHub is publicly available at <https://www.topohub.org>. However, we are providing two screenshots of the web interface for the reference.

In Fig. 2, an example Gabriel graph topology is displayed. Users can navigate through the left menu to select the type of topology they want to explore. Currently, the options include Gabriel, SNDLib, and Topology Zoo. For Gabriel graphs, users can choose the desired topology size from the second dropdown menu. Subsequently, they can select a specific topology from the list.

In Fig. 3 we present the visualization of link utilization for the `germany50` topology sourced from the SNDLib library. The color intensity of the links corresponds to their utilization levels, with hotter colors indicating higher utilization.

We also provide a minimal example of package usage in Mininet to automatically import topologies from the repository. The 1 shows how to build and initialize a Mininet network using a Gabriel graph topology from the repository.

```
import mininet.net
import topohub.mininet

topo_cls = topohub.mininet.TOPO['gabriel/25/0']
topo = topo_cls()

net = mininet.net.Mininet(topo=topo)
net.interact()
```

Listing 1: Minimal example of the Mininet automatic topology import feature

5. Impact

Gabriel graphs are capable of representing the characteristics of geographical transport network topologies as they are planar and can effectively model various factors such as fiber cable lengths, node degrees, and the biconnected property. The repository of reference network topologies based on Gabriel graphs will provide researchers with a standardized benchmark for testing and comparing network design models and algorithms. It will enhance the ability to conduct studies on network performance and algorithm complexity based on different network sizes and facilitate reproducible research in the field of transport networks. Below, we provide examples of use case scenarios where we believe the provided repository can be particularly useful and applicable:

- **Investigation of network algorithm computational complexity.** Sometimes, it is very challenging or even impossible to analytically derive the average computational complexity of an algorithm. In such cases, empirical computational complexity can be calculated by executing the algorithm on inputs of different sizes and then measuring the execution time. By providing a homogeneous set of topologies with linearly increasing node numbers, the presented repository is perfectly suited for such a use case. The published Gabriel graph topologies were already used in [17] to investigate the computational complexity of the Generic Dijkstra optical network routing algorithm [11].
- **Graph Neural Networks generalization research.** GNNs represent a novel type of neural networks able to understand the complex relationship between topology, routing and input traffic in a computer networks. They enable the generation of accurate estimates for mean delay and jitter per source/destination pair. GNNs exhibit the potential for generalization across various topologies, routing schemes, and variable traffic conditions [21]. They should be capable of learning from datasets comprising small network testbed samples and predict metrics for considerably larger networks, possibly by a factor of 10-100x. However, as emphasized in [22], achieving such generalization remains an ongoing research challenge. The utilization of the topology set provided in this paper would significantly facilitate that research.

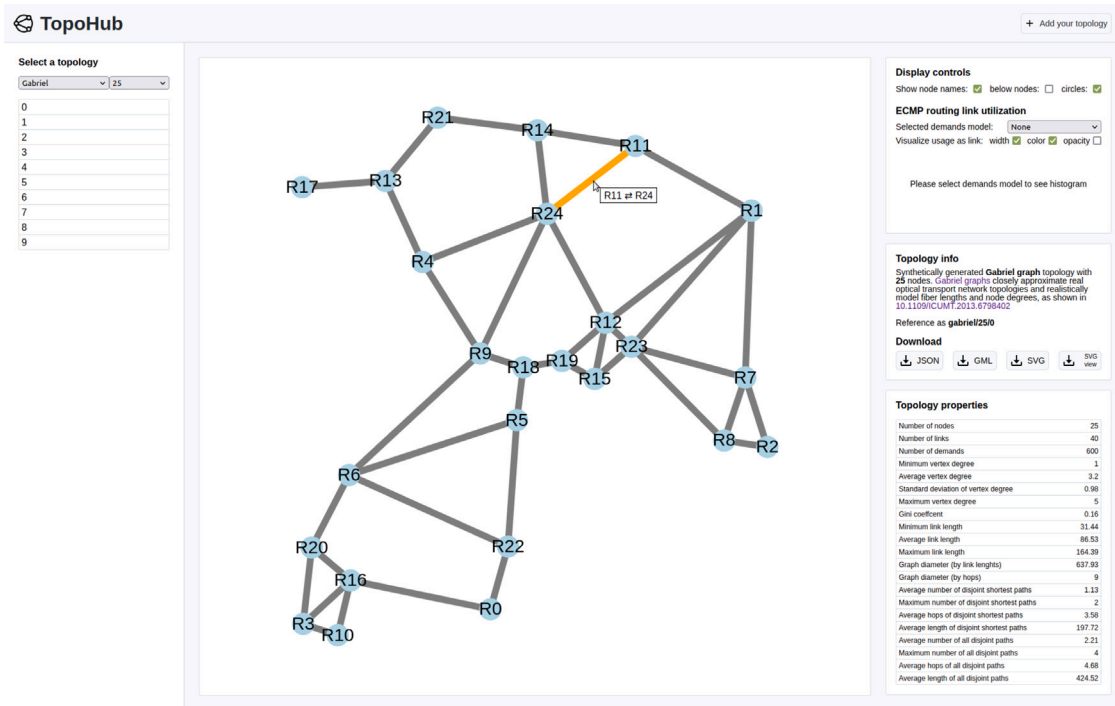


Fig. 2. The web interface displaying a 25-node Gabriel graph topology.

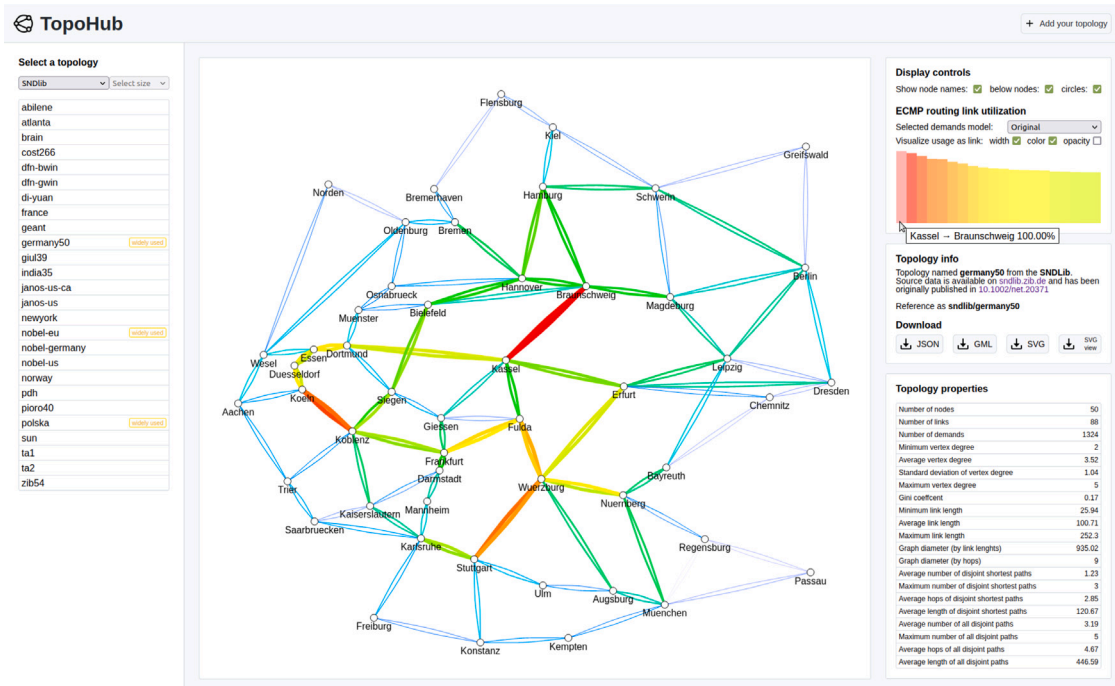


Fig. 3. ECMP traffic visualization for the germany50 SNDLib topology.

- **Rapid SD-WAN prototyping.** The Mininet integration allows to automatically import any topology from the repository and run emulated networks of different topologies within a seconds. Combined with the provided precalculated ECMP routing links utilizations it would allow to rapidly test SDN-based WAN traffic engineering systems and compare their performance with the current state of the art ECMP routing.

In addition to the synthetic Gabriel topologies, the repository incorporates established topologies from The Internet Topology Zoo and SNDLib. As a result, it can function as a comprehensive hub of physical network topology data for networking research, also preserving data from these projects that have been abandoned. The repository, due to its open source character, remains also open for topology contributions from fellow researchers. To contribute, they can submit pull requests on GitHub.

6. Conclusions

The TopoHub project addresses a significant gap in networking research by providing a comprehensive repository of reference network topologies based on the Gabriel graph model. By offering synthetic topologies with increasing node counts that closely resemble real networks, this project enables researchers to conduct experiments and evaluations in networks of various sizes in a comparable and reproducible manner. The integration of existing topologies from The Internet Topology Zoo and SNDLib further enriches the repository's utility. A web interface allows users to visualize and explore topologies, including link utilization under various traffic models. The TopoHub project provides a valuable resource of topology data for networking researchers, enabling standardized benchmarking and reproducible research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the data is openly available in the GitHub repository.

References

- [1] Knight S, Nguyen HX, Falkner N, Bowden R, Roughan M. The Internet topology zoo. *IEEE J Sel Areas Commun* 2011;29(9):1765–75. <http://dx.doi.org/10.1109/JSAC.2011.1111002>.
- [2] Knight S, Falkner N, Nguyen HX, Tune P, Roughan M. I can see for miles: Re-visualizing the Internet. *IEEE Netw* 2012;26(6):26–32. <http://dx.doi.org/10.1109/MNET.2012.6375890>.
- [3] Orlowski S, Wessály R, Pióro M, Tomaszewski A. SNDlib 1.0—Survivable network design library. *Networks* 2010;55(3):276–86. <http://dx.doi.org/10.1002/net.20371>.
- [4] Rusek K, Cholda P. Message-passing neural networks learn little's law. *IEEE Commun Lett* 2019;23(2):274–7. <http://dx.doi.org/10.1109/LCOMM.2018.2886259>.
- [5] Kadziolka B, Skała M, Wójcik R, Jurkiewicz P, Domżał J. Employing FAMTAR and AHB to achieve an optical resource-efficient multilayer IP-over-EON SDN network. *IEEE Access* 2022;10:94089–99. <http://dx.doi.org/10.1109/ACCESS.2022.3204290>.
- [6] Biernacka E, Boryło P, Jurkiewicz P, Wójcik R, Domżał J. Handling high and low priority traffic in multi-layer networks. *Bull Pol Acad Sci Tech Sci* 2023;71(3):e145568. <http://dx.doi.org/10.24425/bpasts.2023.145568>.
- [7] Çetinkaya EK, Alenazi MJ, Cheng Y, Peck AM, Sterbenz JP. On the fitness of geographic graph generators for modelling physical level topologies. In: 2013 5th international congress on ultra modern telecommunications and control systems and workshops. 2013, p. 38–45. <http://dx.doi.org/10.1109/ICUMT.2013.6798402>.
- [8] Gabriel KR, Sokal RR. A new statistical approach to geographic variation analysis. *Syst Biol* 1969;18(3):259–78. <http://dx.doi.org/10.2307/2412323>.
- [9] Çetinkaya EK, Alenazi MJ, Cheng Y, Peck AM, Sterbenz JP. A comparative analysis of geometric graph models for modelling backbone networks. *Opt. Switch. Netw.* 2014;14:95–106. <http://dx.doi.org/10.1016/j.osn.2014.05.001>, Special Issue on RNDM 2013.
- [10] Szcześniak I, Gola A, Jajszyk A, Pach AR, Woźna-Szcześniak B. Itinerant routing in elastic optical networks. *J Lightwave Technol* 2017;35(10):1868–75. <http://dx.doi.org/10.1109/JLT.2017.2674182>.
- [11] Szcześniak I, Jajszyk A, Woźna-Szcześniak B. Generic dijkstra for optical networks. *J. Opt. Commun. Netw.* 2019;11(11):568–77. <http://dx.doi.org/10.1364/JOCN.11.000568>.
- [12] Agrawal A, Bhatia V, Prakash S. Network and risk modeling for disaster survivability analysis of backbone optical communication networks. *J Lightwave Technol* 2019;37(10):2352–62. <http://dx.doi.org/10.1109/JLT.2019.2904328>.
- [13] Szcześniak I, Olszewski I, Woźna-Szcześniak B. Towards an efficient and exact algorithm for dynamic dedicated path protection. *Entropy* 2021;23(9). <http://dx.doi.org/10.3390/e23091116>.
- [14] Guillen L, Takahira H, Izumi S, Abe T, Suganuma T. On designing a resilient SDN C/M-plane for multi-controller failure in disaster situations. *IEEE Access* 2020;8:141719–32. <http://dx.doi.org/10.1109/ACCESS.2020.3013323>.
- [15] Guillen L, Izumi S, Abe T, Suganuma T. A resilient mechanism for multi-controller failure in hybrid SDN-based networks. In: 2021 22nd Asia-Pacific network operations and management symposium. 2021, p. 285–90. <http://dx.doi.org/10.23919/APNOMS52696.2021.9562671>.
- [16] Di Cicco N, Mercan EF, Karandin O, Ayoub O, Troia S, Musumeci F, et al. On deep reinforcement learning for static routing and wavelength assignment. *IEEE J Sel Top Quantum Electron* 2022;28(4):1–12. <http://dx.doi.org/10.1109/JSTQE.2022.3151323>.
- [17] Jurkiewicz P, Biernacka E, Domżał J, Wójcik R. Empirical Time Complexity of Generic Dijkstra Algorithm. In: 2021 IFIP/IEEE international symposium on integrated network management. 2021, p. 594–8.
- [18] Hirayama T, Jibiki M, Kafle VP. Saliency-based VNF placement method for evolving multiple network slices. *Comput Netw* 2023;236:110002. <http://dx.doi.org/10.1016/j.comnet.2023.110002>.
- [19] Hagberg A, Swart P, Chult D. Exploring network structure, dynamics, and function using NetworkX. In: Proceedings of the 7th python in science conference. 2008.
- [20] Lantz B, Heller B, McKeown N. A network in a laptop: Rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM SIGCOMM workshop on hot topics in networks. New York, NY, USA: Association for Computing Machinery; 2010. <http://dx.doi.org/10.1145/1868447.1868466>.
- [21] Rusek K, Suárez-Varela J, Mestres A, Barlet-Ros P, Cabellos-Aparicio A. Unveiling the potential of graph neural networks for network modeling and optimization in SDN. In: Proceedings of the 2019 ACM symposium on SDN research. New York, NY, USA: Association for Computing Machinery; 2019, p. 140–51. <http://dx.doi.org/10.1145/3314148.3314357>.
- [22] Ferriol-Galmés M, Paillisse J, Suárez-Varela J, Rusek K, Xiao S, Shi X, et al. RouteNet-Fermi: Network modeling with graph neural networks. *IEEE/ACM Trans Netw* 2023. <http://dx.doi.org/10.1109/TNET.2023.3269983>.