GeNESIS: Generator for Network Evaluation Scenarios of Industrial Systems

Lukas Bechtel*†, Samuel Müller*, Michael Menth§, Tobias Heer*†
*University of Applied Sciences Esslingen, Germany, {lukas.bechtel,samuel.mueller,tobias.heer}@hs-esslingen.de
§Chair of Communication Networks University of Tuebingen, Germany, menth@uni-tuebingen.de
†Belden Inc., Neckartenzlingen, Germany

***Abstract*—The lack of standardized, realistic industrial network scenarios hinders the comparative evaluation of scientific research for industrial networks. Current evaluations often use non-public or synthetic scenarios, making it difficult to compare results across different studies. This paper introduces GeNESIS, a tool designed to generate and exchange realistic, reproducible industrial network evaluation scenarios. GeNESIS produces comprehensive topologies formatted according to IETF standards, including network devices and connections. Additionally, it generates configurations for network devices, supporting evaluations such as algorithm comparisons or network simulations. GeNESIS was created to evaluate firewall configurations but is designed to further support other use cases, such as QoS or reliability. By providing a standardized exchange format, GeNESIS ensures the simple availability of evaluation scenarios, promoting comparability and reproducibility in industrial network research.**

***Index Terms*—Industrial Networks, Generation, Evaluation**

I. Introduction

Modern industrial networks require new mechanisms and configurations to fulfill requirements of, e.g., Quality of Service (QoS), secure firewall configurations, or failure tolerance. Hence, many researchers develop algorithms to improve the configuration of industrial networks. The selection of evaluation scenarios for such algorithms has a strong impact on the evaluation results. For industrial networks, there are only few fully disclosed scenarios of real networks and many synthetic ones that are not realistic or cover only specific characteristics with fixed arbitrary sizes. As a result, evaluations are difficult to compare between different research groups, as each group uses their own network models, or limits the evaluation to specific and tailored scenarios. Research lacks a simple method for sharing a large set of tunable, reproducible, comparable, and close-to-reality scenarios to evaluate contributions.

This paper presents the design of *GeNESIS*, a tool to generate and exchange sets of realistic and reproducible industrial network evaluation scenarios. An evaluation scenario, i.e., the output of *GeNESIS*, is a set of complete network layouts for an industrial factory network, including network devices, such as controllers, sensors, or switches, and the connections between them in the format of the IETF Network Topology Model [1]. Additionally, *GeNESIS* generates configurations for all network devices to serve as input for research evaluations, e.g., comparison of optimization algorithms or network simulations. Specifically, *GeNESIS* creates firewall configurations based on self-generated communication relationships in the network. The design of *GeNESIS* also allows for the generation of other device configurations, such as QoS or reliability, in the future. To this end, *GeNESIS* combines all relevant parameters in a short and simplified exchange format, i.e., the *GeNESIS-TAG*[1], to be included in evaluations and reused by other researchers. Based on this short representation, *GeNESIS*, which is publicly available[2], can regenerate exactly the same set of evaluation scenarios to ease the comparison of evaluations without an additional dataset repository for topologies and configurations.

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 528745080 - FIP 68. The authors alone are responsible for the content of the paper.

II. Related Work

This section presents an overview of the related work in the field of industrial networks. First, we discuss other network generators. Second, we highlight related work evaluating with self-generated topology layouts. Finally, we address related work using firewall configurations in their evaluation.

Currently, available generators for network topologies focus on internet topologies or do not generate reproducible scenarios [3]–[5]. Medina et al. [3] present *BRITE* as a foundation of network generators without the capabilities to generate industrial networks. Similarly, Cheng et al. [4] develop *RealNet* addressing the lack of realistic generated networks but focusing only on BGP-based Internet routing topologies. Alrumaih et al. [5] present *GENIND* for generating industrial network topologies. It uses hierarchical network architectures but is limited to three fixed layers. Further, the output is not reproducible, as each execution uses new seeds.

In literature, optimizations for time-critical networks require topologies to evaluate the benefit of optimizations [6]–[9]. The generation of reference topologies is often described insufficiently or is not publicly available. Thus, the generated topologies cannot be exchanged since the interfaces are too case-specific. Hence, the evaluations are difficult to compare.

Similarly, some evaluations require knowledge of the topology and device configurations for optimizations, e.g., the optimization of firewall configurations. Related work uses either closed-source firewall rulesets and requires the implementation of other algorithms to achieve comparable results [10]–[12], or use open-source ruleset collections without information about associated network topologies [13].

[1]genesis:v1.0#894::2·726:3·833:1·795::1·100·0::1·0·0·0:0·1·1·2:0·1·1·1::0·1·5·2:0·1·0·0::1·1·5·1:1·0·0·0::2·4·5·1:1·0·0·0# (e.g., EPIC dataset [2])

[2]https://github.com/hs-esslingen-it-security/hses-GeNESIS

Accepted for publication in Proceedings of *29th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Padova, Italy, September 10-13, 2024

III. GeNESIS Algorithm Design

This section introduces the design of *GeNESIS*, a **Ge**nerator for **N**etwork **E**valuation **S**cenarios of **I**ndustrial **S**ystems. First, we give a general overview of *GeNESIS*. Second, we present the network architecture *GeNESIS* utilizes. Third, we introduce common traffic profiles supported by the generator. Finally, we highlight the configuration generation based on the example of firewall rulesets.

A. Overview

The algorithm to generate reference scenarios for industrial networks consists of three steps: 1) topology generation, 2) communication generation, and 3) configuration generation. Each of these steps is designed modular, allowing extensions to the generator in the future. The topology generation follows a hierarchical model. For each network layer of the hierarchical model, the generator selects from different topology patterns, e.g., ring or daisy chain, which are typical for industrial networks. In each network segment, *GeNESIS* creates typical end devices in industrial networks, e.g., sensors, controllers, or servers. Next, the generator uses predefined communication profiles, such as strict isolation in historical networks or converged networks for TSN and DetNet topologies. Based on these profiles and the types of the generated devices, *GeNESIS* creates a customizable number of communication relationships between devices to depict realistic communication. In the last step, the generator creates configurations for devices in the network. Specifically, in this first version, *GeNESIS* generates firewall rulesets. The implementation of the generator can be extended to generate configurations for, e.g., TSN or redundancy. Finally, *GeNESIS* encodes all parameters and seeds required to generate the complete set of topologies and configurations in a short and exchangeable format to enable a reconfiguration of exactly the same output. The format is short enough to be pasted as a footnote in a scientific paper, allowing the exchange of topologies without additional repositories. In the following, we will detail the configuration possibilities and implications for the generated industrial network scenarios.

B. Industrial Network Architecture

This section details the basics of industrial network architectures and their parameterization in *GeNESIS*. We derive the input for the design and capabilities of *GeNESIS* from industrial standards and industrial projects in factory and process automation. First, we introduce the hierarchical concept of industrial networks. Second, we present the different topology layouts used in individual networks. Finally, we present the utilized device roles and their communication capabilities.

1) Hierarchical Network Topologies: The typical industrial network structure follows a hierarchical model to facilitate operational control and data acquisition from the plant floor to the enterprise level (cf. Figure 1). This architecture follows the principles described in the Purdue model for secure network architectures and is required by industrial security standards [14]. The Purdue model defines separating network segments with specific tasks through firewalls or gateways

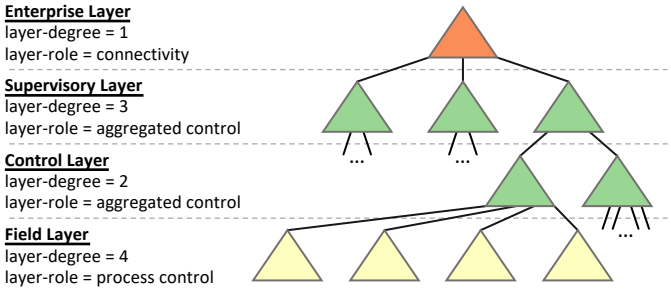


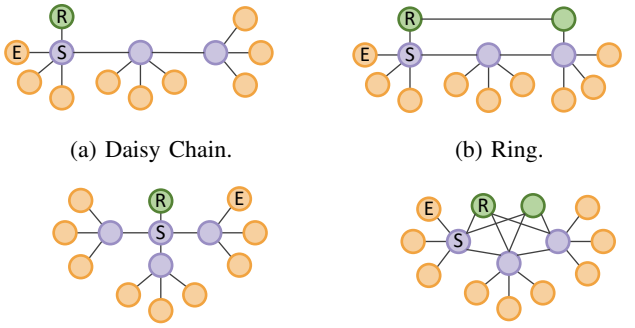
Fig. 1: Hierarchical industrial network architecture.

limiting traffic. Hence, industrial protocols defined in the IEC 61784-1 [15], such as PROFINET or EtherCAT, are tailored to this hierarchical approach. Each layer has specific transmission bandwidth, data volume, and security requirements, guiding the selection of appropriate technologies and protocols.

The topology of the hierarchical networks is similar to a tree. The configuration of *GeNESIS* enables the definition of the number of hierarchical layers and the degree to the next higher layer. Typically, industrial deployments have four layers, i.e., *Field Layer*, *Control Layer*, *Supervisory Layer*, and *Enterprise Layer* (cf. Figure 1). *GeNESIS* assigns a role to each of these layers to define the types of end devices per layer and their purpose of communication. The field layer has the role *process control* with data collection and actuation. The control and supervisory layers have the role *aggregated control* with process visualization in operation centers or at smaller dashboards. Finally, the enterprise layer has the role *connectivity* to enable remote access and data analysis. The root has the degree one, consisting of only one network segment. For all other layers, the degree defines the number of network segments attached to one network segment in the layer above. This degree depends on the actual production process but typically is within a range of 5 to 20. Industrial protocols typically operate in isolated subnetworks. Hence, *GeNESIS* creates each network segment as an individual subnetwork and uses routers to connect these subnetworks.

2) Topology Patterns in Industrial Ethernet Networks: Typically, industrial networks follow template designs. Hence, network segments are structured similarly if they have the same purpose. This allows simplified installation and maintainability. There are four common topology patterns in industrial networks: 1) daisy chain, 2) ring, 3) star, and 4) mesh. *GeNESIS* implements all four topology patterns, with the configuration specifying a distribution per layer between these four patterns. The configuration file of *GeNESIS* enables specification for the number of switches in a network segment and end devices per switch. These two parameters are configurable per network layer as a range of values, such that *GeNESIS* can build similar configurations in the specified ranges. In the following, we detail the specificities of all patterns.

a) Daisy Chain / Linear Topology: GeNESIS models the daisy chain as a line of switches, where the first one has routing capabilities for the connection to the higher network layer (cf. Figure 2a). Each switch connects one to multiple



(a) Daisy Chain. (b) Ring.

(c) Star. (d) Mesh.

Fig. 2: Topology patterns with routers (R), switches (S), and end-devices (E).

end devices. For special cases of *switched end-devices*, i.e., one device combines switch and end device, *GeNESIS* must be configured to connect only one end device per switch.

b) Ring Topology: Media-Redundancy Protocol (MRP) or Device Level Ring (DLR) are typical redundancy protocols for rings in industrial deployments. Two of these switches have routing functionality and connect the ring redundantly to the higher-layer network segment. The generation of end-devices and switched end-devices is equivalent to the linear topology.

c) Star Topology: The star topology has one central switch which connects adjacent switches. All end devices connect to these adjacent switches (cf. Figure 2c). One of these adjacent switches has routing functionality to connect the network segment to the higher-layer network segment.

d) Mesh Topology: In a mesh topology, *GeNESIS* connects the switches in a web-like structure (cf. Figure 2d), typically with Rapid Spanning Tree Protocol (RSTP) as redundancy protocol. Two switches have routing capability to connect the network segment to the higher layer.

C. Industrial Network Communication

This section first introduces the different device types and the protocols they support. Second, we present the three typical traffic profiles in industrial networks.

1) Device Categories: In this first version, *GeNESIS* differentiates the devices into one of seven following categories: 1) operational technology (OT) end devices, such as sensors or actuators, 2) controllers, also known as PLC, 3) workstations and dashboards for visualization and operation, i.e., SCADA, 4) IT end devices such as IP cameras, 5) servers for data collection and analysis, 6) switches, and 7) routers.

For each role assignable to a network layer, *GeNESIS* uses different distributions for device types among generated end devices. Typically, in a *field* network segment, *GeNESIS* does not create any servers but many OT end devices. Similarly, a network segment with the *enterprise* role has many servers but no OT end devices. For each device category, *GeNESIS* uses a fixed number of services, as listed in the project repository of *GeNESIS*. For example, a controller has a web server and uses industrial control protocols like PROFINET, EtherCAT, OPC UA, or EtherNet/IP [15], [16]. Switches and routers have an SSH and NETCONF server.

2) Traffic Profiles: Industrial networks have general profiles for their traffic, which define the types of communication relations available. *GeNESIS* implements the following profiles (cf. Figure 3): 1) strict isolation, 2) converged networks 3) distributed control. In strict isolation, only controllers may communicate with controllers in neighboring network segments. All other traffic is limited strictly to the same network segment. Converged networks enable communication from all end devices to the enterprise level for use cases like predictive maintenance or video surveillance. Distributed control is the most advanced traffic profile, allowing traffic between all controllers in the network. Additionally, all OT and IT end devices may communicate with any controller in the same branch of the hierarchical network architecture.

GeNESIS chooses the communication partners based on the communication profile and protocols available on the devices. Based on the configuration for the number of all connections, *GeNESIS* randomly samples from this list. The generator associates the required bandwidth and communication intervals for each sampled connection.

D. Industrial Network Configuration

The configuration of industrial reference scenarios is based on the topology itself and the resulting traffic in the network. In this first version of *GeNESIS*, the generator configures only the addresses and firewalls in the network.

1) Addressing: The algorithm configures each network segment as an independent IP subnetwork. The subnets are enumerated sequentially and enable the implementation of any device-to-device communication.

2) Firewall Configuration: *GeNESIS* places routers capable of filtering network packets, i.e., firewalls, between network segments. Industrial standards require applying the *Zones and Conduits* concept [14], which restricts firewalls to forward only traffic accepted by specified rules in the firewall ruleset. The ruleset size mainly depends on the traffic patterns and the number of sampled communication relations. The generator configures all rulesets in the network, i.e., outputs the firewall rulesets in the *iptables* format. For ruleset optimization algorithms, *GeNESIS* can generate rulesets with anomalies, i.e., not optimal configurations, as described by Al-Shaer in [17]. This network-wide configuration parameter defines the average number of dependencies in every generated ruleset.

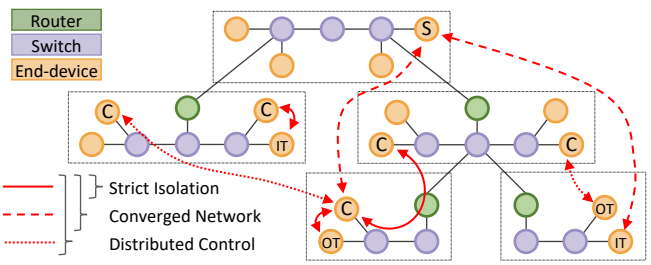


Fig. 3: Traffic profiles in hierarchical network architecture.

IV. Application of GeNESIS

The main purpose of *GeNESIS* is the simplified exchange for large numbers of realistic industrial network configurations. This section details the application of *GeNESIS* to create, share, and reuse the generated configurations. We provide more details in the publicly available repository of *GeNESIS*.

The first step to utilize *GeNESIS* is creating the configuration file in a *json* format. As a reference scenario, we use the SWaT dataset [18] for a water treatment facility in Singapore, which is available in the resource folder of the *GeNESIS* repository, with a second example of the EPIC dataset [2]. The SWaT network has four layers: one *process control* network with a star topology, two *aggregated control* networks, utilizing a star and a mesh topology, and one *connectivity* network with a star topology.

In the next step, *GeNESIS* outputs the *GeNESIS-TAG*[3] and generates the specified topologies, communication relationships, and device configurations. *GeNESIS* implements these three stages as nested for loops and adjusts the seeds with each iteration of the loops. This enables the generation of industrial network scenarios with similar characteristics as all non-random parameters stay the same. Each generated topology is stored in the standardized IETF format [1], including all communication relations and device configurations.

Researchers aiming to compare their research to previous work can execute *GeNESIS* with a *GeNESIS-TAG* as provided in the footnote[3]. The *GeNESIS-TAG* encodes the complete configuration to reproduce the same output of topologies, communication relationships, and device configurations. Hence, all topology and configuration files are available for comparable evaluation runs without requiring a dataset repository.

V. Future Work

GeNESIS is designed to generate configurations used in network simulators, not simulating the network, e.g., redundancy or congestion, itself. It has a modular approach, allowing for additional topology patterns or network configurations. The current roadmap includes the following additions to *GeNESIS*.

First, *GeNESIS* shall support the connection between buildings through a routed IP or MPLS-TP network to connect larger facilities. Second, the topology patterns will be extended to support data center topologies, as well as electrical substation topologies [19]. Third, we will extend *GeNESIS* to generate QoS configurations and message scheduling to enable the application of TSN and DetNet models as defined by Wüsteney et al. [7]. Finally, we will enhance the generation of communication relationships by *GeNESIS*.

VI. Conclusion

Research evaluation for industrial networks is difficult due to missing realistic and fully disclosed scenarios. Specifically, the configuration and performance of firewalls are difficult to compare between related work, as they depend on the ruleset and the traffic in the network. Hence, we developed *GeNESIS* to provide reproducible and exchangeable industrial network scenarios. The generated topologies follow industrial standards for hierarchical networks and use common topology patterns. *GeNESIS* creates realistic firewall configurations based on typical protocols and communication relations. We document all configuration possibilities in the open-source repository.

[3] genesis:v1.0#459::3·36:1·465:2·83::0·100·0::1·0·0·0:0·1·1·2:0·1·1·1::0·1·1·1:0·0·1·0::1·1·3·2:1·0·0·0::1·6·2·2:0·1·0·0::2·1·3·10:0·1·0·0# (SWaT [18])

References

[1] A. Clemm, J. Medved, R. Varga, N. Bahadur, H. Ananthakrishnan, and X. Liu, "A YANG Data Model for Network Topologies," RFC 8345, Mar. 2018.

[2] S. Adepu, N. K. Kandasamy, and A. Mathur, "EPIC: An Electric Power Testbed for Research and Training in Cyber Physical Systems Security," in *Computer Security: ESORICS International Workshops, CyberICPS and SECPRE*, Barcelona, Spain, Sep. 2018.

[3] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2001.

[4] L. Cheng, N. C. Hutchinson, and M. R. Ito, "RealNet: A Topology Generator Based on Real Internet Topology," in *International Conference on Advanced Information Networking and Applications (AINA)*, Okinawa, Japan, Mar. 2008.

[5] T. N. Alrumaih and M. J. Alenazi, "GENIND: An Industrial Network Topology Generator," *Alexandria Engineering Journal*, vol. 79, 2023.

[6] D. Hellmanns, A. Glavackij, J. Falk, R. Hummen, S. Kehrer, and F. Dürr, "Scaling TSN Scheduling for Factory Automation Networks," in *IEEE International Conference on Factory Communication Systems (WFCS)*, Porto, Portugal, Apr. 2020.

[7] L. Wüsteney, D. Hellmanns, M. Schramm, L. Osswald, R. Hummen, M. Menth, and T. Heer, "Analyzing and Modeling the Latency and Jitter Behavior of Mixed Industrial TSN and DetNet Networks," in *Proceedings of the International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Roma, Italy, Dec. 2022.

[8] T. Stüber, M. Eppler, L. Osswald, and M. Menth, "Performance Comparison of Offline Scheduling Algorithms for the Time-Aware Shaper (TAS)," *IEEE Transactions on Industrial Informatics*, 2024.

[9] E. B. Schweissguth, P. Danielis, D. Timmermann, H. Parzyjegla, and G. Mühl, "ILP-based joint routing and scheduling for time-triggered networks," in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, Grenoble, France, Oct. 2017.

[10] E. W. Fulp, "Optimization of Network Firewall Policies Using Directed Acyclical Graphs," in *Proceedings of the IEEE Internet Management Conference*, Jan. 2005.

[11] H. H. Hamed and E. Al-Shaer, "Dynamic Rule-ordering Optimization for High-speed Firewall Filtering," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Taipei, Taiwan, Mar. 2006.

[12] R. Mohan, A. Yazidi, B. Feng, and B. J. Oommen, "Dynamic Ordering of Firewall Rules Using a Novel Swapping Window-based Paradigm," in *Proceedings of the International Conference on Communication and Network Security (ICCNS)*, Singapore, Nov. 2016.

[13] C. Diekmann, L. Hupel, J. Michaelis, M. W. Haslbeck, and G. Carle, "Verified iptables Firewall Analysis and Verification," *Journal of Automated Reasoning*, vol. 61, Jun. 2018.

[14] International Electrotechnical Commission (IEC), "IEC 62443-3-2," *Security for industrial automation and control systems – Part 3-2: Security risk assessment for system design*, 2020.

[15] ——, "IEC 61784-1," *Industrial networks - Profiles - Part 1-1: Fieldbus profiles - Communication Profile Family 1*, 2023.

[16] ——, "IEC 62541-1," *OPC Unified Architecture – Part 1: Overview and Concepts*, 2020.

[17] E. S. Al-Shaer and H. H. Hamed, "Modeling and Management of Firewall Policies," *IEEE Trans. Netw. Serv. Manag.*, vol. 1, no. 1, 2004.

[18] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, Vienna, Austria, Apr. 2016.

[19] International Electrotechnical Commission (IEC), "IEC 61850," *Communication networks and systems for power utility automation*, 2024.