Comparative analysis of Kathara and Containerlab as full-scale modeling tools for computer virus propagation

Irina O. Ishchenko and Anastasiya A. Mishina

RUDN University, 6 Miklukho-Maklaya St, Moscow, 117198, Russian Federation

Background SIR, SEIR, and SEIRS epidemiological models are widely used in research. The use of theoretical infection models in this work allows us to focus on studying the system parameters of the Kathara emulator rather than on the mathematical accuracy of the emulation results. Studying the simulation tool is necessary for its use in future research. **Purpose** To analyze the technical performance indicators of the Kathara real-world simulation tool using the example of a computer virus outbreak. **Methods** Compartmental epidemiological models SIR, SEIR, and SEIRS are used to analyze the dynamics of infection spread and evaluate system parameters in a reproducible real-world experiment in Kathara. Discrete-time implementations of the models with parameters β, σ, and γ are constructed. The computational experiment is organized as a network of interacting nodes, one of which is the index patient and initiates the infection process. **Results** Graphs of three epidemiological models were constructed and analyzed. System metrics showed faster simulation dynamics on the Conteinerlab simulation tool. **Conclusion** Containerlab proved to be technically superior for the processes under study.

Keywords: Network emulation; Kathara; Containerlab; epidemiological model; SIR; SEIR; SEIRS

1 Introduction

This study is devoted to conducting a field experiment on modeling computer virus infection in the Kathara emulator. Studying the mechanisms of virus propagation allows us to analyze the capabilities of the system in which the research is conducted and its resistance to malicious traffic. Traditionally, mathematical models borrowed from epidemiology [1], such as SIR (Susceptible-Infected-Recovered), SEIR (Susceptible-Exposed-Infected-Recovered), and SEIRS (Susceptible-Exposed-Infected-Recovered-Susceptible) models. These models allow us to describe the processes of infection, incubation, and recovery in a population of network nodes.

The Kathara simulation tool makes it possible to create virtual network topologies and study their behavior under controlled conditions. Despite the growing popularity of this emulator in the academic community [2] [3], systematic analysis of its technical characteristics and performance in modeling the spread of computer viruses remains an understudied issue. This work aims to evaluate the system's capabilities when working with various epidemiological models, as well as to understand the platform's limitations in terms of scalability and accuracy of results.

Using well-studied epidemiological models SIR, SEIR, and SEIRS as a theoretical basis allows us to focus specifically on the system parameters of the emulator: memory usage, memory consumption, and network traffic metrics.

Irina O. Ishchenko: 1132226529@rudn.ru
Anastasiya A. Mishina: 1132226532@rudn.ru

2 Main part

2.1 Эпидемиологические модели

Compartment models are mathematical models that describe the processes of interaction between different entities (e.g., individuals) from different groups (compartments) over time. Each entity belongs to one group, and within each group, the entities are indistinguishable.

One of the simplest compartmental models is the SIR epidemic spread model. This model was created by William Kermack and Anderson McKendrick in 1927. All individuals in the population in this model are divided into three groups:

- S (susceptible) - healthy individuals who are susceptible to the disease;
- I (infectious) - infected individuals who spread the disease;
- R (recovered) - individuals who have recovered from the disease and acquired immunity.

However, many diseases have an incubation period during which an individual is infected but not yet contagious. This led to the SEIR model, which includes a fourth group, E (exposed), consisting of individuals in the latent period of the disease [4]. This model increases the accuracy of modeling infectious diseases such as COVID-19.

We will consider a closed population where there are no birth and death processes. Then the evolution of an individual can be described by the following diagram:

$$S \overset{\beta}{\to} E \overset{\sigma}{\to} I \overset{\gamma}{\to} R$$

The model is described by a system of differential equations:

$$\begin{aligned}
\frac{dS}{dt} &= -\frac{\beta SI}{N},\\
\frac{dE}{dt} &= \frac{\beta SI}{N} - \sigma E,\\
\frac{dI}{dt} &= \sigma E - \gamma I,\\
\frac{dR}{dt} &= \gamma I,\\
N &= S(t) + E(t) + I(t) + R(t),
\end{aligned}$$

where

- β - infection coefficient (the probability that contact between a susceptible and an infected individual will result in a new infection) ($S \to E$);
- σ - incubation transition coefficient (the probability that an infected individual will become contagious) ($E \to I$);
- γ - recovery coefficient (probability that an infected individual recovers) ($I \to R$).

Another modification of the SIR model is the SEIRS model [5]. In real life, the immunity of a certain proportion of recovered individuals weakens over time. The SEIRS model allows individuals to transition from a recovered state to a susceptible state.

The individual evolution diagram looks like this (Fig. 1).

The system of differential equations for the SEIRS model is as follows:

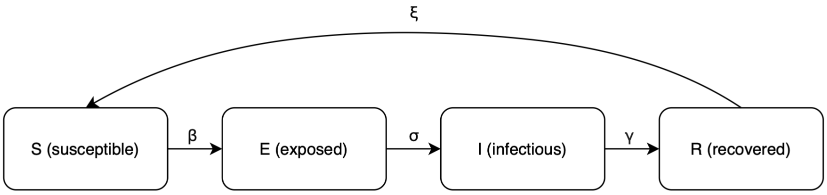

Figure 1: Individual evolution diagram in the SEIRS model

$$\begin{aligned}
\frac{dS}{dt} &= -\frac{\beta SI}{N} + \xi R, \\
\frac{dE}{dt} &= \frac{\beta SI}{N} - \sigma E, \\
\frac{dI}{dt} &= \sigma E - \gamma I, \\
\frac{dR}{dt} &= \gamma I - \xi R \\
N &= S(t) + E(t) + I(t) + R(t),
\end{aligned}$$

where

- ξ - probability of immunity loss (probability that a recovered individual will eventually return to the susceptible category) ($R \to S$).

If the influx of susceptible individuals into the population is large enough, the system in a steady state transitions to endemic equilibrium (a stable state of the system in which the infection remains at a constant level: it does not disappear or fade away), accompanied by damped fluctuations in the number of infected individuals.

2.2 Modeling

Two network emulation tools were used to conduct the experiment: Kathará [6] and Containerlab [7]. Kathará is an open-source network emulation system based on Docker containers, supported on all major operating systems (Linux, macOS, Windows). It is designed to demonstrate the interaction of network nodes in an isolated environment and to develop new network protocols.

Containerlab is a tool that provides a command-line interface for orchestrating and managing container-based network labs. To create a lab topology, Containerlab launches containers, establishes virtual connections between them, and independently manages the lifecycle of the entire lab. The emulator uses the YAML format to describe the topology and supports integration with Docker.

This work used a standard Debian-based image with pre-installed network utilities (kathara/base) in Kathará and a custom debian:bookworm-slim-based image with openssh-server, openssh-client, python3, and iproute2 in Containerlab.

The network topology is the same for both tools: a flat network with 100 nodes. The first host node (pc1) is patient zero, i.e., it is infected from the outset. This node initiates the spread of the virus. Transitions are implemented as stochastic processes [8]: every t seconds, the host checks the probability of transitioning to the next state. The work uses the Morris worm principle: if the host is vulnerable, further propagation occurs by copying the Python script via SCP and remotely launching the "virus" via SSH. Each node in turn listens to TCP port 4000 to exchange information about its current state with its neighbors. At the same time, events, namely transitions between states, are logged in a CSV file. To monitor the process, an external script was written that polls all nodes and collects general statistics on the states S, E, I, R (S, I, R for the SIR model). The emulation parameters are reflected in the tables (Table 1, Table 2, Table 3).

Table 1: SIR model parameters

Parameter	**Symbol**	**Value**
Probability of infection	β	0.8
Probability of recovery	γ	0.1
Interval between checks $I \to R$	—	5 s
Delay between scans	—	0.2 s

Table 2: SEIR model parameters

Parameter	**Symbol**	**Value**
Infection probability	β	0.4
Activation probability	σ	0.4
Recovery probability	γ	0.3
Check interval $E \to I$	—	3 s
Check interval $I \to R$	—	5 s
Delay between scans	—	2 s

Table 3: SEIRS model parameters

Parameter	**Symbol**	**Value**
Infection probability	β	0.7
Activation probability	σ	0.5
Probability of recovery	γ	0.15
Probability of immunity loss	ξ	0.25
Check interval $E \to I$	—	3 s
Check interval $I \to R$	—	8 s
Check interval $R \to S$	—	15 s
Maximum number of reinfection cycles	—	3
Delay between scans	—	1 s

3 Results

Based on the emulation results, graphs were constructed for each model. For this purpose, a log analysis script was created, which constructed graphs using the well-known matplotlib library. The values on the x-axis reflect the time of the experiment, and the values on the y-axis reflect the number of hosts.

Let's take a closer look and compare the graphs of both tools for the SEIRS model (Fig. 2), (Fig. 3).

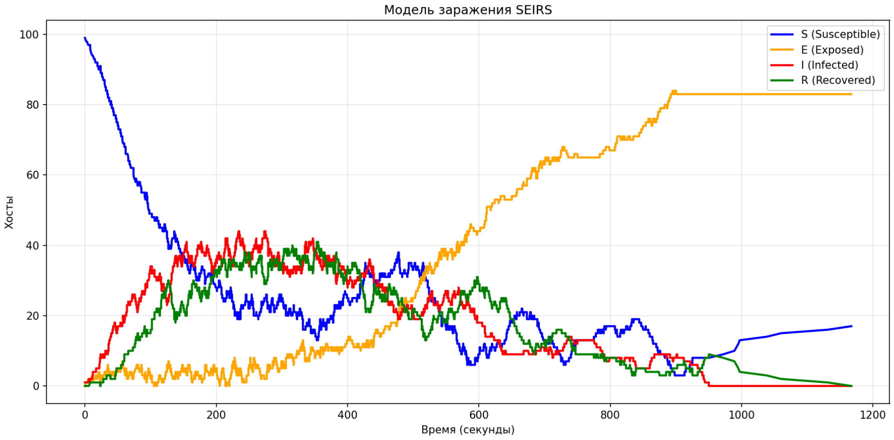

Figure 2: Graph of the propagation according to the SEIRS model on Kathará

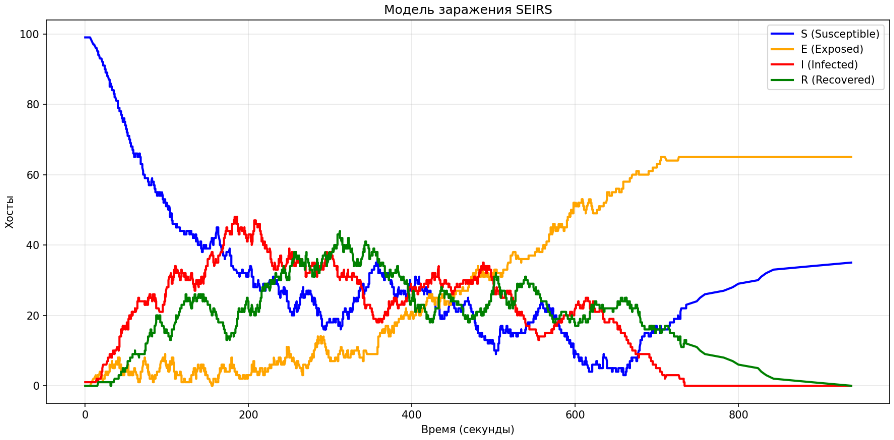

Figure 3: Graph of the propagation according to the SEIRS model on Containerlab

The graphs show differences in the dynamics of the spread of infection. In Containerlab, the epidemic process develops faster, which is reflected in an earlier peak of infected nodes and accelerated redistribution of states between compartments S, E, I, and R. In contrast, Kathará shows a more protracted dynamic. Differences in the emulation rate may indicate differences in the docker containers used. However, it should be noted that the spread of the virus is modeled stochastically, so let's take a closer look at the system parameters (Table 4) rather than infection trends.

According to the table, Containerlab used 4.7 times less memory than Kathará. It exchanged traffic more actively with less CPU load. Kathará, on the other hand, has a lower container load.

Table 4: Comparison of Kathará and Containerlab metrics

Metric	**Kathará (avg / max)**	**Containerlab (avg / max)**
Total number of samples	267	183
Experiment duration	22 min 16 sec	16 min 30 sec
Host: CPU load (%)	50.2% / 100.0%	23.7% / 91.6%
Host: Memory usage (%)	61.6% / 62.7%	25.2% / 30.2%
Host: Memory usage (GB)	7.89 / 8.02 GB	1.69 / 2.06 GB
Host: transfer rate (MB/s)	0.02 MB/s	1.35 MB/s
Host: receive rate (MB/s)	0.03 MB/s	1.09 MB/s
Containers: count	101.0	99.5
Containers: CPU load (%)	128.2% / 348.8%	282.7% / 1158.7%
Containers: memory usage (GB)	0.66 / 1.18 GB	0.73 / 1.10 GB

4 Discussion

Experiments have shown that Containerlab uses the host more efficiently and has higher network activity. This allows you to set a large number of nodes and scale the network horizontally. The tool is good for exploring topologies where the emphasis is on network exchange, including cybersecurity tasks.

The advantage of Kathará is the stability of container loads. Consequently, the emulator is useful for isolated computing or systems with low network traffic. If the goal of the research is to monitor trends, Kathará may also be chosen for the task due to its longer data collection time.

5 Conclusion

The study focused on modeling epidemiological models in Kathara and Containerlab emulators for the purpose of further comparison and analysis of system parameters. We were able to create reproducible experiments in two environments, the results of which were comparable and similar. The behavior of theoretical models did not produce unpredictable results. For the processes under study, the emulator Containerlab proved to be technically stronger, so it will be prioritized for cybersecurity tasks in the future.

Acknowledgments

The authors are deeply grateful to their scientific supervisor Anna Vladislavovna Korolkova for valuable guidance and support throughout this research. Special thanks go to Professor Dmitry Sergeevich Kulyabov for posing the problem and for his insightful suggestions.

References

1. Hoang, M. T. A simple approach for studying stability properties of an SEIRS epidemic model. *Journal of Applied Analysis* **31,** 143–156. DOI: doi:10.1515/jaa-2024-0019 (2025).

2. Huu, V. L. N. & Chau, K. K. *Enhancing Computer Network Learning with Hands-on Lab Exercises* in *Software Engineering: Emerging Trends and Practices in System Development* (eds Silhavy, R. & Silhavy, P.) (Springer Nature Switzerland, Cham, 2025), 327–338.

3. Usmanov, O., Muthanna, A. & Paramonov, A. *Analysis of SDN Traffic using Full-scale Modeling* in (IEEE, Moscow, Russia, 2018), 1–4. DOI: 10.1109/ICUMT.2018.8631205.

4. Biswas, M. H. A., Paiva, L. T., de Pinho, M. D. R., *et al.* A SEIR model for control of infectious diseases with constraints. *Mathematical Biosciences and Engineering* **11,** 761–784 (2014).

5. Butler, B. A., Stern, R. & Paré, P. E. Analysis and Applications of Population Flows in a Networked SEIRS Epidemic Process. *IEEE Transactions on Network Science and Engineering* **11,** 6664–6677. DOI: 10.1109/TNSE.2024.3468991 (2024).

6. Scazzariello, M., Ariemma, L. & Caiazzi, T. *Katharà: A Lightweight Network Emulation System* in (IEEE, Budapest, Hungary, 2020), 1–2. DOI: 10.1109/NOMS47738.2020.9110351.

7. Muslim, U. & Recker, S. *A Comparative Analysis of Digital Twins for Advanced Networks* in *2024 IEEE 7th International Conference and Workshop Óbuda on Electrical and Power Engineering (CANDO-EPE)* (2024), 281–286. DOI: 10.1109/CANDO-EPE65072.2024.10772762.

8. Kulyabov, D. S., Korolkova, A. V. & Sevastianov, L. A. Two Formalisms of Stochastization of One-Step Models. *Physics of Atomic Nuclei* **81,** 916–922 (2018).