Borysov, Ruslan

Article

Development of functionalities extension approach and implementation of address routing for IFogSim based simulators

Reference: Borysov, Ruslan (2021). Development of functionalities extension approach and implementation of address routing for IFogSim based simulators. In: Technology audit and production reserves 3 (2/59), S. 13 - 18.
http://journals.uran.ua/tarp/article/download/232810/234421/539636.
doi:10.15587/2706-5448.2021.232810.

UDC 004.94
DOI: 10.15587/2706-5448.2021.232810
Article type «Reports on Research Projects»

Ruslan Borysov

DEVELOPMENT OF FUNCTIONALITIES EXTENSION APPROACH AND IMPLEMENTATION OF ADDRESS ROUTING FOR IFOGSIM BASED SIMULATORS

The object of research is an approach of functionality extension for simulation toolkits based on iFogSim. It is assumed by the native approach that enhancement of functionalities should be achieved by inheriting the fog device class and defining new features in its body. However, this approach makes it impossible to use inherited simulators together and significantly decreases flexibility even when utilizing a single simulator. Another problem related exclusively to iFogSim is a specific communication scheme between application modules, which results in data routing limitations in fog architectures and odd data streams taken into account.

This paper introduces an alternative extension approach incorporating a peculiar inheritance scheme which tries to reconsider the standard approach from a behavioral design patterns point of view. The key feature of the suggested approach is an extraction of fog device features from the native class into separate behavioral classes. Meanwhile, the designed inheritance scheme allows to flexibly override and combine behaviors. According to the approach principles the developed simulator extends iFogSim with application modules addressing capabilities solving limitations, along with implementing users' mobility and dynamic wireless connectivity as it is done in MobFogSim. With the aim to check its correctness, the designed toolkit was validated with the standard for iFogSim case study of «EEG Tractor Beam game» application. The validation included four scenarios. In the first two scenarios the features of users' mobility and dynamic base station connectivity were validated. And in the next scenarios that utilized address routing the obtained delay and network usage values were compared with theoretically calculated ones. The validation results indicated the correct simulator behavior, and introduced functionalities extension approach, being more complex in comparison with the inative one, can significantly improve flexibility of the simulator.

Keywords: *iFogSim extension, address routing, MobFogSim, fog computing simulation, behavioral design patterns.*

Received date: 15.02.2021
Accepted date: 25.03.2021
Published date: 30.06.2021

How to cite
Borysov, R. (2021). Development of functionalities extension approach and implementation of address routing for iFogSim based simulators. Technology Audit and Production Reserves, 3 (2 (59)), 13–18. doi: http://doi.org/10.15587/2706-5448.2021.232810

1. Introduction

The simulation process is not a necessary stage in information system design and implementation. However, significant benefits can be achieved while utilizing simulation [1], e. g. ability to control internal and external system parameters, repeat experiment at any time, discover and handle system bottlenecks in advance, and save money reducing physical hardware utilization. These benefits are topical for designing both cloud and fog computing systems. Preparation of a real testing environment would require enlisting a huge amount of hardware resources while the achieved results could not have been accurately reproduced because of stochastic nature. This was the driving force of the development of cloud and fog computing simulation toolkits. In the context of the paper only three of them would be discussed:

– CloudSim [1], which models cloud data center, manages its hardware (hosts) and software (virtual machines) resources, controls users' tasks execution;
– iFogSim [2] based on CloudSim and models fog computing nodes, IoT devices (sensors and actuators) and distributed applications;
– MobFogSim [3] extends iFogSim with functionalities of users' mobility, wireless network connectivity and enables application migrations between devices.

These simulators are commonly used by researchers in cloud, fog and edge computing. For example, iFogSim was used in [4] considering the impact of user mobility in fog systems. Also, the «edgeward» strategy from iFogSim was adopted in [5] for the multiobjective module placement strategy research. Authors of [6] designed an IoT simulation platform based on CloudSim and tested it with two real-world applications data. Finally, MobFogSim was used in [7] to analyze migration techniques for the mobile clients use case.

However, these simulation toolkits have both distinct disadvantages and a common conceptual problem which is related to their functionalities extension. Its first aspect consists in potential changes of the toolkit's source code: while iFogSim uses the original source code of CloudSim, the source code of iFogSim itself was amended in MobFogSim. The latter brings in the risk of incompatibility between different toolkits: although it is not accepted to use different simulation environments together, this opportunity should not be discarded. But a more serious aspect of the problem is the utilization of the simplest inheritance method. This leads to entity class definitions that have features, which are unnecessary for the user but still demand initialization and control during the simulation process. In addition, one of the iFogSim disadvantages, as it is claimed by authors of [2], is a routing limitation of data streams (tuples) which is caused by specific communication between application modules. On the other hand, although MobFogSim implements some advanced application migration techniques and base stations handoff models, it has opaque control mechanisms and bugs in the source code that decrease reliability.

Thus, *the object of research* is an approach of functionality extension for simulation toolkits based on iFogSim. *The aim of this research* is to introduce a new simulator on the basis of iFogSim and alternative to MobFogSim, which adopts the behavioral extension approach, solves routing problems of iFogSim by implementing addressing, and is checked for validity.

2. Methods of research

As an attempt to improve the extension problem the simulator utilizes a peculiar class inheritance scheme within the object-oriented concept of Java programming language. Furthermore, the simulator overcomes the communication limitations by providing tuple address routing. In addition, the validation of the developed simulation toolkit was performed with the standard «EEG Tractor Beam game» case study, which was described in [2].

2.1. Functionalities extension approach. The main source of flexibility insufficiency in implementing new features and extending the existing ones is a simple and typical inheritance scheme, which is illustrated in Fig. 1. To demonstrate the problem the case of implementing a new feature is considered, which includes fog devices sending tuples not along a single vertical direction (uplink or downlink) but choosing direction dynamically depending on the recipient module. According to the existing approach the hypothetical class AddressingFogDevice must be inherited from the FogDevice class from the iFogSim library. However, in this case the feature of target class can't be easily combined with features of device mobility or dynamic wireless connectivity of such MobFogSim classes like MobileDevice, ApDevice and FogDevice. In the case of inheriting AddressingFogDevice from FogDevice from MobFogSim library, it becomes necessary to initialize and control each aspect of mobility and dynamic connectivity which is undesired. Thus, a simple inheritance scheme obstructs toolkits' flexibility.

The alternative extension approach draws on ideas of behavioral design patterns [8] and involves the «encapsulation over inheritance» programming concept [9]. Considering extensible features as behavior entities which FogDevice class can obtain it is possible to achieve desired flexibility. In case of unrelated features such as mobility and dynamic network connectivity, two behavior entities can be easily integrated together. More complex issue arises in case of one feature extending another, which would demand inheritance between two behavior classes bringing difficulties for combining two different extensions. However, replacing inheritance relationship with composition and providing abstraction of the behavior being extended the issue can be overcome. Thus, the suggested inheritance scheme, while being more complex and difficult, can increase flexibility of functionality extensions for iFogSim based toolkits.

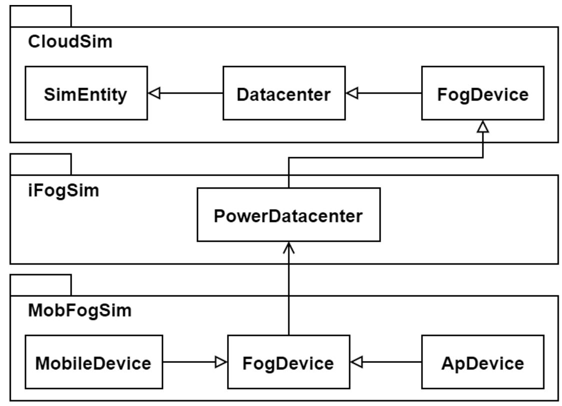

Fig. 1. Typical inheritance scheme of simulation toolkits

2.2. Data routing in iFogSim. The routing limitation consists in the fact that direction of tuple route must be constant in iFogSim. It means that building hierarchical architecture it is impossible to ensure communication between two child devices (e. g. two base stations) even if they have a common parent device (e. g. Internet service provider gateway). Another feature of iFogSim routing is a broadcasting nature of downlink tuples, which sometimes is undesired behavior resulting in odd data streams taken into account. Both these problems are illustrated in Fig. 2 (*a* and *b*, respectively).

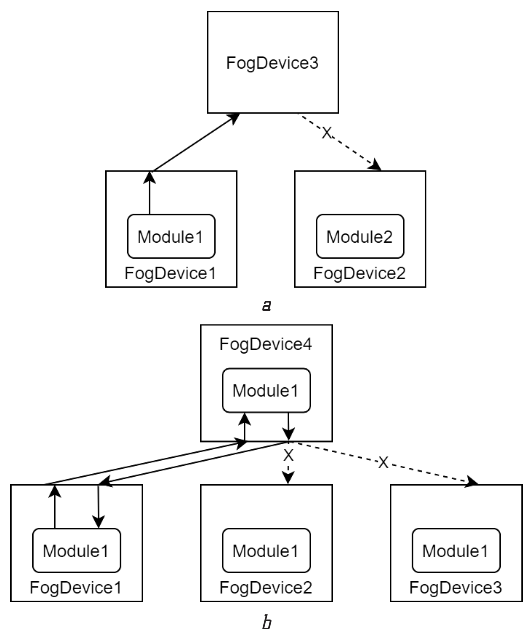

Fig. 2. Features of tuples routing in iFogSim: *a* – constant directionality limitation; *b* – odd data streams accounting

If undesired, both these behaviors can be fixed by address routing implementation. In the process of simulation each fog device can dynamically decide whether the received tuple should be routed uplink or downlink to a specific child. At each tuple incoming the device can determine the location of the target module, build the current network graph, search for the path with the least latency, and send the tuple to the next hop device. Although this addressing scheme is not suitable for real-world networks, it is acceptable to be utilized in a simulation toolkit.

2.3. Validation scenarios. While introducing their novel toolkit, authors of [2] applied iFogSim in two test scenarios. The first one was «EEG Tractor Beam game» which was adopted for validation of the toolkit from this research. The scenario includes three application modules, sensor and actuator, and all possible tuple directions: single requests to uplink modules, single responses to downlink modules, broadcasting uplink and downlink queries. The system topology is depicted in Fig. 3 where configuration (Config) number (1–5) determines the number of gateways (GW) in the system: 1, 2, 4, 8 and 16. For each configuration two types of sensors were used: type A with period of 10 ms, type B with period of 5 ms. And two distinct module placement strategies were applied: cloud strategy with two modules placed in the cloud datacenter and edgeward strategy with two modules placed either on mobile device (Mob) or on GW. It is important to notice that 10 % of sensor recordings are discarded and other 90 % are processed in a normal mode. At the end of the simulation scenario the values of average delay (ms) and network usage (Megabytes) are determined.

With the aim to validate the designed simulation toolkit three simulation scenarios were executed. In the first scenario the original classes from iFogSim were used. In contrast, the second scenario includes mobile devices with the ability to dynamically connect to the closest base station. In the topology for second scenario (Fig. 4) four base stations (BS) were connected to each gateway. The latency of the new links was set to 0 and bandwidth was set to infinity. In this way the link will not affect the simulation results.

The users' mobility pattern involves grouped counterclockwise movement along the square of base stations. The validation will be considered successful, if the results in scenario 1 and 2 coincide.

The third validation scenario has the same topology as in scenario 2; however, address routing was added to fog devices' behavior. To estimate its validity the theoretical calculations of network usage were performed and illustrated in Fig. 5. The theoretical model is simplistic as it ignores tuple processing time, but it considers all link types in the scenario. Thus, experimentally achieved network usage values are expected to be slightly lower than ones obtained theoretically and preserve a corresponding trend depending on the configuration. Furthermore, applying address routing in scenario 3 should decrease network usage in comparison with values in scenarios 1 and 2. Configurations 4 and 5 for cloud placement strategy are exceptional: as it can be stated from [2], the cloud center for Config 4 and 5 works in overloaded mode increasing the delay value significantly. Since the theoretical model does not consider this parameter, experimental network usage in these cases will be much lower than theoretically expected.

Additional scenario 4 aims not to validate the simulator, but to demonstrate its advantage. The devices of this scenario are the same as in scenario 3, but they are interconnected into different topology as it is depicted in Fig. 6.

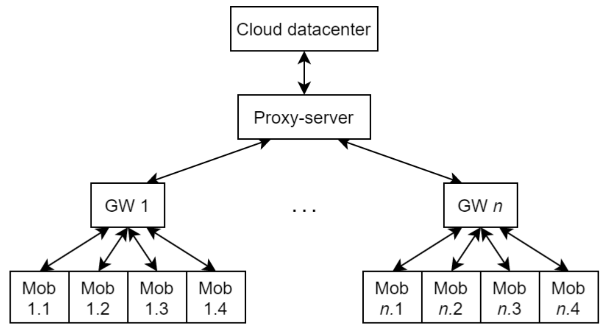

Fig. 3. Scenario 1 topology

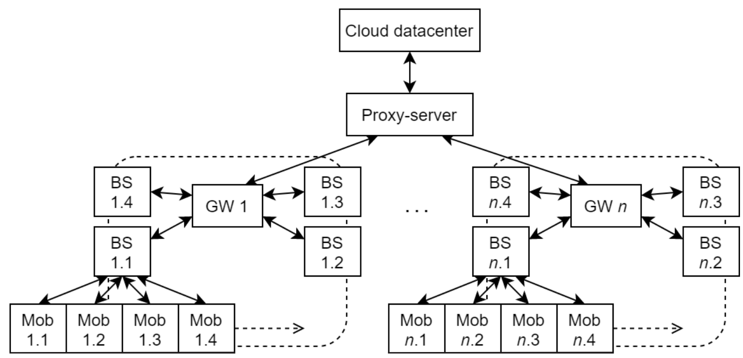

Fig. 4. Scenario 2 and 3 topology

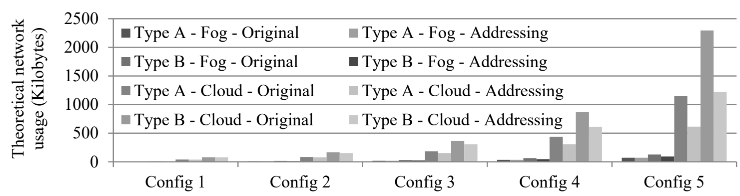

Fig. 5. Theoretical calculation of network usage for scenarios 2 and 3

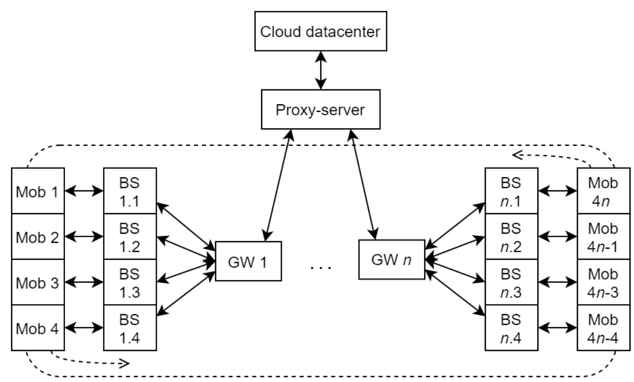

Fig. 6. Scenario 4 topology

In the fourth scenario mobile devices move independently along the common trajectory connecting to different base stations. The key difference of the system behavior is the data exchange between gateways in the case, when the mobile user's tuple has been finished processing by one gateway and the user is now connected to another. In such a way tuple dynamic directionality is demonstrated.

3. Research results and discussion

3.1. Results on designed extension approach. Designed behavioral-oriented extension approach for simulators based on iFogSim is illustrated in Fig. 7.

 ISSN 2664-9969

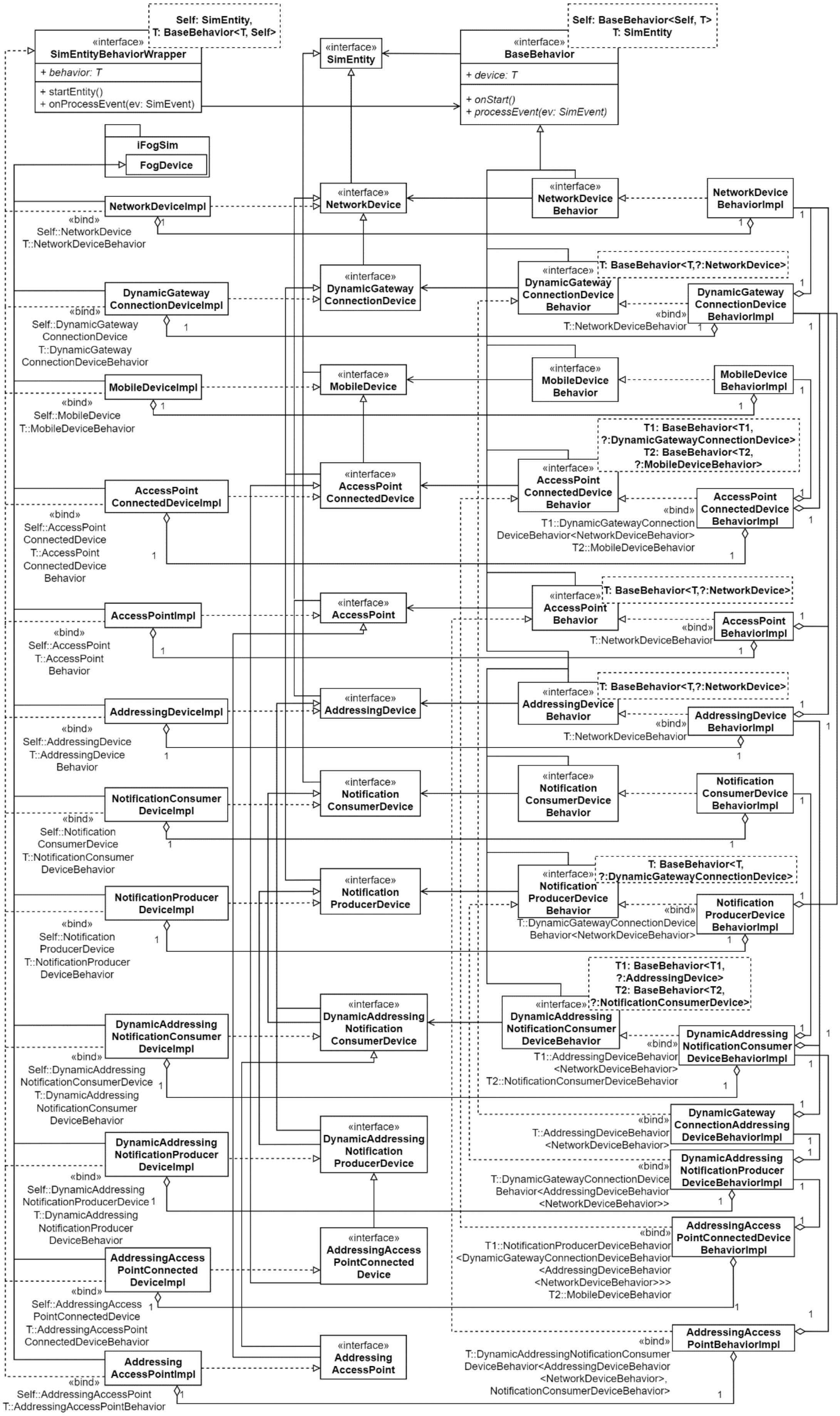

Fig. 7. Inheritance scheme of designed extension approach

The key principles of the approach are:

– all behavioral logic is implemented in Java-interfaces;

– behavioral classes do not contain any logic and only specify and combine necessary behavioral interfaces;

– stateful properties are located in entity interfaces while behavioral interfaces are stateless;

– simulation events must be handled exclusively in behavioral interfaces;

– extended behaviors can interact with base behaviors either by delegation of accepted event handling or by generating new events.

Within the inheritance scheme address routing was implemented.

3.2. Validation. Table 1 depicts results of average delay (D) and network usage (NW) measured for each simulation scenario. As it was expected data for scenarios 1 and 2 almost coincide except for three positions. The differences are explained by the fact that the simulator can change the sequence of events that happen simultaneously in a fixed period of model time. The case when two devices at the same have to decide whether to discard the sensor data with 10 % possibility can change the order of random number generator calls. The latter leads to differences in events sequences in scenarios 1 and 2. This effect was confirmed during testing when discard possibility was set to 0 % and absolutely equal results were obtained. Furthermore, address routing from scenario 3 decreased network usage in all configurations because it does not generate odd data streams. In addition, it offloaded the cloud datacenter for configurations 4 and 5, resulting in much lower delays.

Finally, the experimental trends of scenarios 2 and 3 in Fig. 8 are similar, taking into account the remark related to cloud placement strategy in configurations 4 and 5 for scenario 2.

Thus, the designed simulation toolkit can be considered valid as it satisfies the specified requirements. First, users' mobility and base station handoff features work correctly as experimentally obtained delay and network usage values coincide in scenarios 1 and 2 except for justified divergence in three configurations. Second, address routing behavior in scenario 3 works as expected resulting in the similar trend of network usage in comparison with theoretically achieved values.

Table 1

Average delay (ms) and network usage (Megabytes) for validation scenarios

Scenario	Configuration	Type A – Edgeward		Type B – Edgeward		Type A – Cloud		Type B – Cloud	
		D	*NW*	*D*	*NW*	*D*	*NW*	*D*	*NW*
1	1	23.7	5,18	32	8.59	231.2	38.9	225.9	77.1
	2	24.3	9.39	32.3	16.24	231.3	80.76	226.1	161.77
	3	24.1	17.82	32.1	31.49	231.2	177.7	226.2	354.54
	4	23.8	34.68	32.5	62.08	903.3	402.95	2993.4	690.74
	5	22.1	68.39	32.5	123.23	4052.1	690.02	4580.4	1264.84
2	1	23.7	5.18	32	8.59	231.2	38.9	225.9	77.1
	2	24.3	9.39	32.3	16.24	231.3	80.76	226.1	161.77
	3	24.1	17.82	32.1	31.49	231.2	177.7	226.2	354.54
	4	23.8	34.68	32.7	62.08	908.4	403.58	2993.4	690.74
	5	22.1	68.39	32.8	123.34	4052.1	690.02	4580.4	1264.84
3	1	23.7	5.18	35.6	6.5	230.8	37.74	225.6	75.46
	2	24.3	9.39	36.3	12.04	231.2	74.67	225.9	149.08
	3	24.1	17.82	36	23.11	231	148.23	225.9	296.93
	4	23.8	34.68	36.1	45.27	230.9	297.83	226	593.6
	5	22.1	68.39	36	89.56	230.5	594.73	228.2	1186.19
4	1	24.2	5.18	34.8	6.52	231	37.71	225.7	74.63
	2	24.2	9.39	33.9	20.62	231.4	75.4	226.1	148.96
	3	24.4	17.82	34	71.31	231.2	149.3	226.1	298.03
	4	24.0	34.68	34.1	258.02	231.1	297.56	226.3	595.18
	5	22.2	68.39	34	901.23	231.1	594.91	228.3	1188.67

Note: divergent delay and network usage values in scenarios 1 and 2 are marked gray

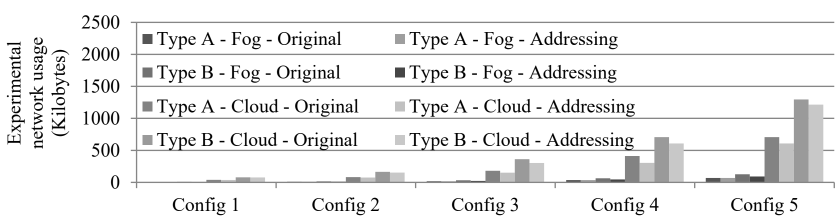

Fig. 8. Experimental values of network usage for scenarios 2 and 3

Finally, similar values of delay and network usage in scenario 4 resembling those in scenario 3 confirm the validity of such an aspect of address routing as tuple dynamic directionality.

4. Conclusions

In this paper the new simulation toolkit based on iFogSim is introduced. Firstly, the adopted extension approach with correspondent behavioral inheritance scheme results in greater complexity of class hierarchy, but significantly enhances the flexibility of the toolkit. Secondly, the feature of address routing, which is implemented within the inheritance scheme, removes the limitation of constant tuple direction and does not take odd data streams into account in comparison with iFogSim toolkit. Finally, the validation of the designed toolkit is carried out with the standard case study. Considering the acquired data from validation scenarios, the toolkit can be stated valid and applicable for further research, and can be accessed via Github link [10].

References

1. Calheiros, R., Ranjan, R., De Rose, C., Buyya, R. (2009). *CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services*. Eprint. Available at: https://arxiv.org/abs/0903.2525
2. Gupta, H., Vahid Dastjerdi, A., Ghosh, S. K., Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Software: Practice and Experience, 47 (9),* 1275–1296. doi: http://doi.org/10.1002/spe.2509
3. Puliafito, C., Gonçalves, D. M., Lopes, M. M., Martins, L. L., Madeira, E., Mingozzi, E. et. al. (2020). MobFogSim: Simulation of mobility and migration for fog computing. *Simulation Modelling Practice and Theory, 101,* 102062. doi: http://doi.org/10.1016/j.simpat.2019.102062
4. Bittencourt, L. F., Diaz-Montes, J., Buyya, R., Rana, O. F., Parashar, M. (2017). Mobility-Aware Application Scheduling in Fog Computing. *IEEE Cloud Computing, 4 (2),* 26–35. doi: http://doi.org/10.1109/mcc.2017.27
5. Mehran, N., Kimovski, D., Prodan, R. (2019). MAPO: A Multi-Objective Model for IoT Application Placement in a Fog Environment. *Proceedings. 9th International Conference on the Internet of Things*. doi: http://doi.org/10.1145/3365871.3365892
6. Salama, M., El-khatib, Y., Blair, G. (2019). IoTNetSim: A Modelling and Simulation Platform for End-to-End IoT Services and Networking. *Proceedings. 12th IEEE/ACM International Conference on Utility and Cloud Computing,* 251–261. doi: http://doi.org/10.1145/3344341.3368820
7. Puliafito, C., Vallati, C., Mingozzi, E., Merlino, G., Longo, F., Puliafito, A. (2019). Container Migration in the Fog: A Performance Evaluation. *Sensors, 19 (7),* 1488. doi: http://doi.org/10.3390/s19071488
8. Brief introduction to Design Patterns. *Kariera Future Processing*. Available at: https://kariera.future-processing.pl/blog/design-patterns/
9. Weisfeld, M. (2005). Encapsulation vs. Inheritance. *Developer.com*. Available at: https://www.developer.com/design/encapsulation-vs-inheritance/
10. *Assassin4i4ek/altmobfogsim: Extension Approach for iFogSim Based Simulators*. Available at: https://github.com/assassin4i4ek/altmobfogsim

Ruslan Borysov, *Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, e-mail: yaneoyeah@gmail.com, ORCID: https://orcid.org/0000-0002-9072-4920*