IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/45AI/2025/4540625/
Vol. 2025 (20), Article ID 4540625

Method for selecting strategy of communication of multiple simulators in an environment based on the HLA standard*

Krzysztof PANUFNIK and Tadeusz NOWICKI

Faculty of Cybernetics, Military University of Technology,
00-908 Warsaw, Kaliskiego 2 Street, Poland

Correspondence should be addressed to: Krzysztof PANUFNIK, kpanufnik@gmail.com

* Presented at the 45th IBIMA International Conference, 25-26 June 2025, Cordoba, Spain

Abstract

The paper presents a new method of selecting communication strategy for simulators connected in the HLA architecture (high-level architecture). The HLA architecture is fundamental for communication between different simulators. It is the modern standard for communication in this area. Simulators connected in this architecture, called federates, often use common data, the values of which can change. Information about changes is sent to other federates interested in these changes. Communication usually places a heavy burden on the network with federates. It turns out that depending on the method of sending messages between federates, the load on the entire system can vary greatly. The motivation for this work was to develop a strategy for broadcasting information about changes that cause the lowest communication load in the network. There are various strategies for transferring data in HLA reported in literature. Unfortunately, there are only a few adequately effective methods to solve this problem. Moreover, they are not efficient enough. The paper presents a new method to solve that problem. The motivation for this work was therefore to develop a new, more efficient, information broadcasting strategy that causes the lowest communication load in the network. Comparative studies in a simulation environment have shown that the new method is more effective than those previously proposed.

Keywords: simulation, standard HLA, data distribution service, data communication management

Introduction

In distributed simulation, one of the most important problems is the time synchronization of the simulation processes, called Time Management. The problem is to synchronize the changes in simulation time, because the computational processes are performed at different speeds in individual simulators. The growing size of simulation experiments has made the exchange of data within the experiments a significant problem. Simulations often reach the size of tens or hundreds of thousands of independent objects, distributed between many simulators. In the case of simulations of this size, it is extremely important for the speed of execution to limit the amount of information that is transmitted via the computer network. Over the years, many dedicated simulation protocols and standards have been created to support solving the above problems.

Cite this Article as: Krzysztof PANUFNIK and Tadeusz NOWICKI, Vol. 2025 (20) “Method for selecting strategy of communication of multiple simulators in an environment based on the HLA standard" Communications of International Proceedings, Vol. 2025 (20), Article ID 4540625,
https://doi.org/10.5171/2025.4540625

At the beginning of the 2000s, a new standard for connecting simulators was implemented, called High Level Architecture (HLA) [1]. HLA has now become the dominant standard for connecting simulators. Simulators that create a simulation environment are called federates. Although HLA is a description of simulation architecture, it assumes the existence of a central software called Runtime Infrastructure (RTI) [2].

The HLA standard assumes that a common data model is used, described by the simulation object model (SOM) [3]. SOM is a formal description of simulation objects and their attributes and interactions that can be distributed and/or received by a federate. The set of all objects and interactions described in the SOM, together with additional information such as data representation, forms a federation object model (FOM). Objects have attributes that reflect their state and are important from the point of view of the simulation experiment. The values of some attributes may change frequently, others less frequently, and still others do not change. Management of activities related to objects and interactions used during simulation is possible thanks to the object management module (OM). The federate that owns an object attribute, when making changes to it, notifies other interested federates via RTI. Transferring ownership of an attribute is possible using the RTI module in the scope of the so-called ownership management.

HLA provides a mechanism for managing interests in the form of a data distribution management (DDM) service. While the OM service allows the expression of interest in data at the object class or interaction level, DDM allows the expression of interest in data at the object attribute level.

In simulation processes based on HLA, a data space is defined, which is usually intensively modified during a simulation experiment. The entire data space consists of attributes of all simulation objects. This space can be divided into separate subsets called regions, which will be called data subareas in the following. The division of the space depends, among other things, on the type of simulation experiment and the needs of the federates participating in it.

HLA provides the possibility of limiting the information about changes received by the federate. This is done thanks to the federate declaring the data subareas about the changes of which the federate wants to be informed. Modification of a data subareas by a federate means that the modified data must be sent to all federates interested in those subareas. It is assumed that changes can be sent by the federate using one of two types of communication channels: one-to-one or many-to-many.

Using a one-to-one channel when many recipients are interested in each data subarea results in the necessity to send a given change multiple time. The situation in which the federate is forced to send information about a change multiple times is called the burden of the sending federate.

From the point of view of the sending federate, the disadvantage of the one-to-one channel can be eliminated by using a many-to-many channel. By using such a channel, the change is sent once and then it will be delivered to all recipients who have declared their willingness to listen to this channel. If the many-to-many channel is used to send multiple subareas (to one or more recipients), a situation may arise when not all listeners of the channel data may be interested in changes in all subareas that are sent using such a channel. The situation in which the federate receives information about a change for a subarea in which it was not interested is called the burden of the receiving federate.

Due to the limited number of many-to-many channels, the potentially large number of possible sub-areas and the different interests of the federates, it is assumed that the number of many-to-many channels available for use is insufficient for complex experiments. As a result, a simulation experiment may lead to a situation where there is no such assignment of individual federates to channels that would avoid excessive load on the federates. The aim of the research was to develop a new and effective method for selecting a communication strategy for simulators in an HLA-based environment. The selection of a communication strategy has a significant impact on the performance of a complex simulation experiment. The strategy affects the load of federates and, consequently, the time between the generation of a change by a federate and the update of information by all federates interested in each change. The use of a communication optimization method between federates connected via HLA will allow for the improvement of one or more of the following parameters: the load of sending federates, the load of receiving federates or the average time of sending information between federates. The most advanced results on this subject published in literature are the works [4] and [5].

A new method that routes communication via available multicast and unicast channels was developed. The article contains a comparison of selected methods according to mathematical model developed by authors [6]. This method can be used in the organization of distributed simulation exercises [7], [8].

Weighted Input Restricted Largest Outgoing Connection

In [4] authors presented concept of connection which represents tuple of sending federate, subarea and receiving federates. Connection has size characteristic which is understood as the expected number of changes generated by a federate. An example of a connection graph is shown in Figure 1.

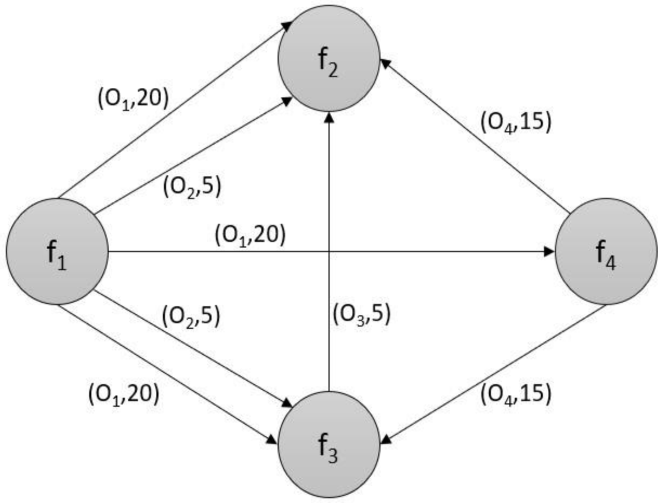


Fig. 1. Connection graph containing information about which federate sends changes in each subarea and who are the recipients of these changes.

All existing methods analyze list of connections and assign them to either many-to-many or one-to-one channels based on criteria selected by methods authors. Up to date there are three main methods of selecting communication strategy Largest Outgoing Connection (LOC), Input Restricted Largest Outgoing Connection (IRLOC) and Adaptive Communication Protocol Selection (ACPS). The proposed method will be compared to

IRLOC and ACPS methods, as previous works [4] and [5] showed that IRLOC outperforms LOC in most cases. We call our new method Weighted Input Restricted Largest Outgoing Connection (WIRLOC) as it is generalization of IRLOC method. Each federate has two numerical characteristics sending weight and receiving weight. Those values allow federate comparison and examination which federates will be sending and receiving more changes than others. Based on that algorithm may favor some federates over others, when deciding which federates will be excessively loaded. If all weights are set to 1 WIRLOC would become IRLOC method. There is no strict method of defining federate weights, but we found that good starting point is number of send and received messages. During method development we observed that small changes to federates weight may give different results. In one of our experiments, change of receiving weight of one federate from 2 to 3 resulted in significantly lower excessive load.

We define positive effect as product of size of connection, number of receiving federates minus one and weight of sending federate. Similarly, we can define the negative effect of connection being added to many-to-many channel as sum of additional excessive loads, caused by added connection, multiplied by receiving weights of federate that received excessive load. This is similar to definitions given in [4], besides that we additionally multiply by federate weight.

WIRLOC method works as follows:

1. For available many-to-many channel:
2. For each connection that was not assigned yet, starting from one with largest positive effect
3. If the positive effect of connection is greater than the negative effect of adding connection to channel, assign connection to channel and remove it from list.

4. After all connections have been analyzed remove the channel from the list.
5. If there are still available channels and unassigned connections go to step 1.
6. For all unassigned connections use one-to-one channels.

The proposed method aims to minimize load of largest sending and receiving federates, and as a consequence improve transmissions times in simulation as a whole.

Experimental Methods Comparison

Given set of 12 federates and 24 subregions we compared performance of WIRLOC, IRLOC and ACPS methods. Each federate sends and receives changes in time of 1 ms. All changes have size determined by uniform distribution between 8 and 16 kB. We have run multiple experiments with different connections speed averaging with 1, 1.5 and 2 ms transmission times. Finally, we evaluated the impact of available numbers of many-to-many for 4, 7, 10, 13 and 16 such channels available. Each experiment run for 100s and federates were generating 156 changes per 100ms. The largest sending federate was expected to send 30 changes per 100ms, and the largest receiving federates were expected to receive 63 and 60 changes per 100ms.

For our experiment ACPS had worst results in generating excessive load for sending federates. Out of 15 experiments only twice it achieved lowest excessive load generated. Opposite can be observed when analyzing receiving federates excessive load, where besides 3 experiments ACPS did not generate additional load. Excessive load comparison is presented in Tables 1 and 2.

Tab. 1 Excessive load of sending federates depending on number of channels and average transmission time.

		1ms			1,5ms			2ms	
Channels	IRLOC	ACPS	WIRLOC	IRLOC	ACPS	WIRLOC	IRLOC	ACPS	WIRLOC
4	1317	1867	1724	1311	1895	1727	1473	1872	1730
7	835	1164	1009	896	1158	1013	988	1163	1161
10	519	841	572	635	836	621	709	839	719
13	206	512	279	534	521	356	422	521	430
16	71	199	70	532	201	291	373	198	223

Tab. 2 Excessive load of receiving federates depending on number of channels and average transmission time.

		1ms			1,5ms			2ms	
Channels	IRLOC	ACPS	WIRLOC	IRLOC	ACPS	WIRLOC	IRLOC	ACPS	WIRLOC
4	469	0	120	475	0	119	370	0	120
7	470	50	171	468	50	170	369	50	121
10	470	0	171	471	0	170	372	0	120
13	471	0	168	472	0	170	371	0	120
16	470	0	171	469	0	170	364	0	120

Based on federates characteristics and connection speeds we could expect on average 3, 3.5 and 4ms of delay between change being generated in subregion and received by the federate. That assumes that change would not spend any time in either sending or receiving queue. Such results are unachievable, unless experiment is trivial, hence minimalization of this delay is important. ACPS, despite having the lowest excessive load, in some cases, out of the three methods is generating the largest delays. This is because ACPS is mainly burdening sending federates with excessive load and thus affecting more federates. Results of our experiments are presented in Table 3 and Figures 2-4.

Tab. 3 Average change delay (ms) depending on method and number of available many-to-many channels.

Number of channels	4		7	10	13	16
		1 ms average transmission time				
IRLOC	**5,03**		4,67	4,56	4,37	4,3
ACPS	6,18		4,73	4,53	4,37	4,17
WIRLOC	5,92		**4,52**	**4,24**	**4,18**	**4,12**
		1.5 ms average transmission time				
IRLOC	**5,53**		5,24	5,15	5,12	5,11
ACPS	6,80		5,22	4,99	4,88	**4,67**
WIRLOC	6,50		**5,04**	**4,86**	**4,74**	4,73
		2 ms average transmission time				
IRLOC	**6,34**		5,97	5,9	5,72	5,66
ACPS	7,15		**5,73**	**5,51**	**5,38**	**5,15**
WIRLOC	6,93		5,81	5,56	5,46	5,42

The proposed WIRLOC method gave overall the best results for two cases of connection speed (1ms and 1.5 ms). In that case it achieved best delay times in 7 out of 10 cases. IRLOC method gave the best performance with all three-connection speed when number of many-to-many was lowest. It is worth mentioning that results of ACPS and WIRLOC were much worse than that of IRLOC method in that case. For the slowest transmissions speed ACPS gave the best results in 4 out of 5 cases. Out of all cases only once two methods achieved the same results. This happened for 13 many-to-many channels, fastest connection and methods IRLOC and ACPS.

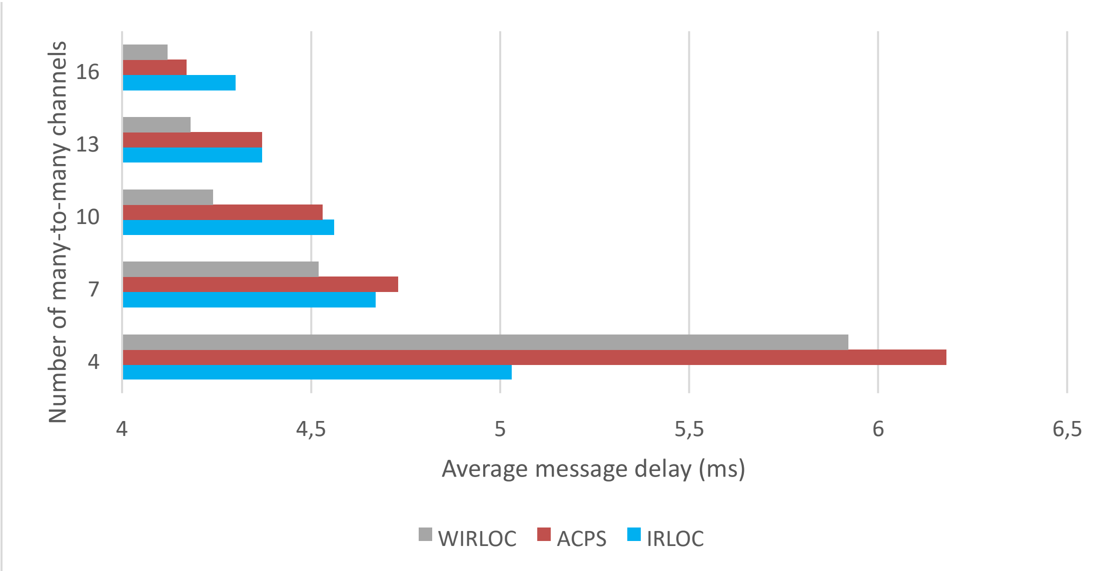


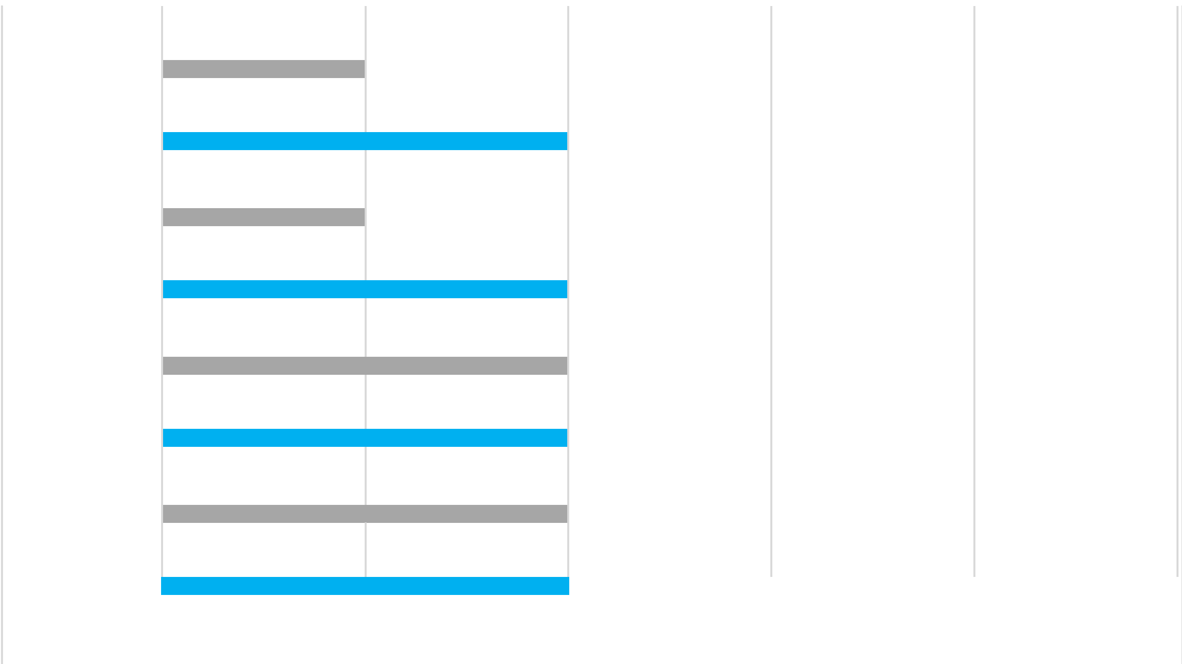
Fig. 2. Average message delay for average transmission time 1ms.

5

We analyzed places which each method took in each experiment and results are presented in Table 4. IRLOC method despite having best results given least amount of many-to-many channels achieved the worst results overall, being worst performing method in 10 out of 15 cases.

Tab. 4 Number of times each method took place compared to others

Method\Place	1	2	3
IRLOC	3	2	10
ACPS	5	6	4
WIRLOC	7	8	0



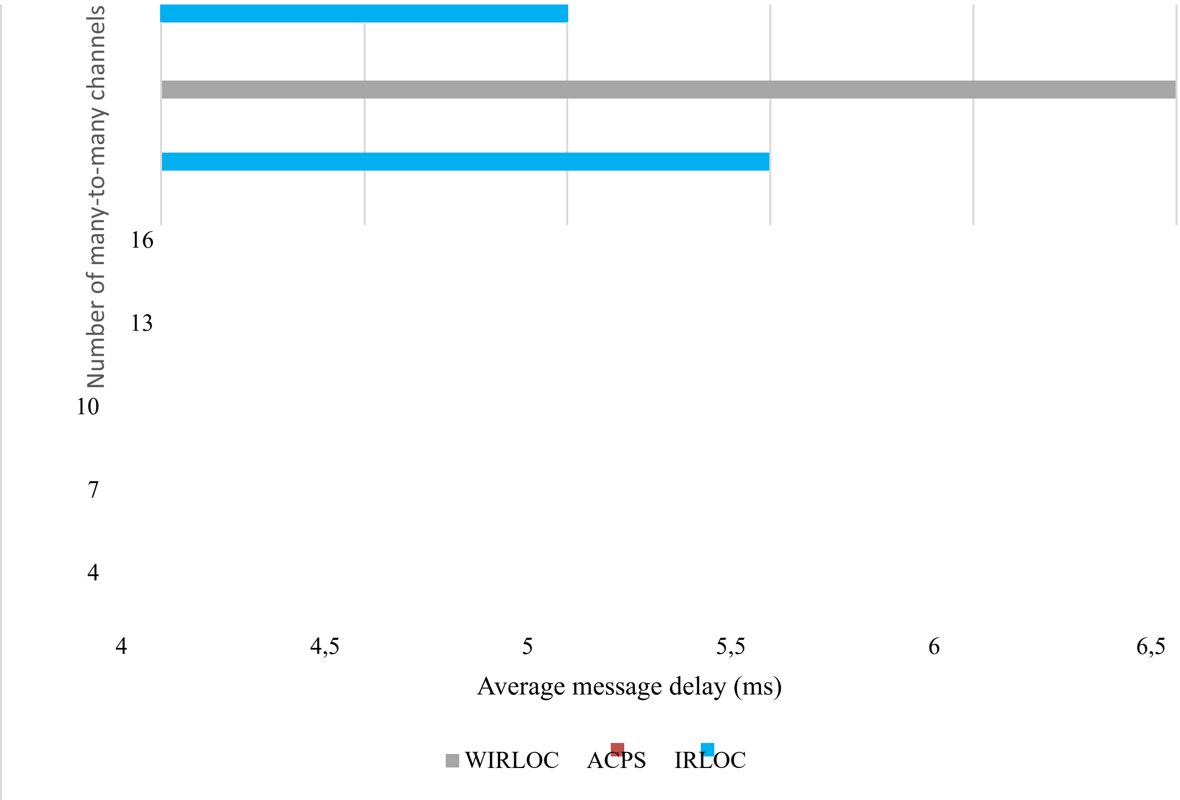


Fig. 3 Average message delay for average transmission time 1.5 ms.

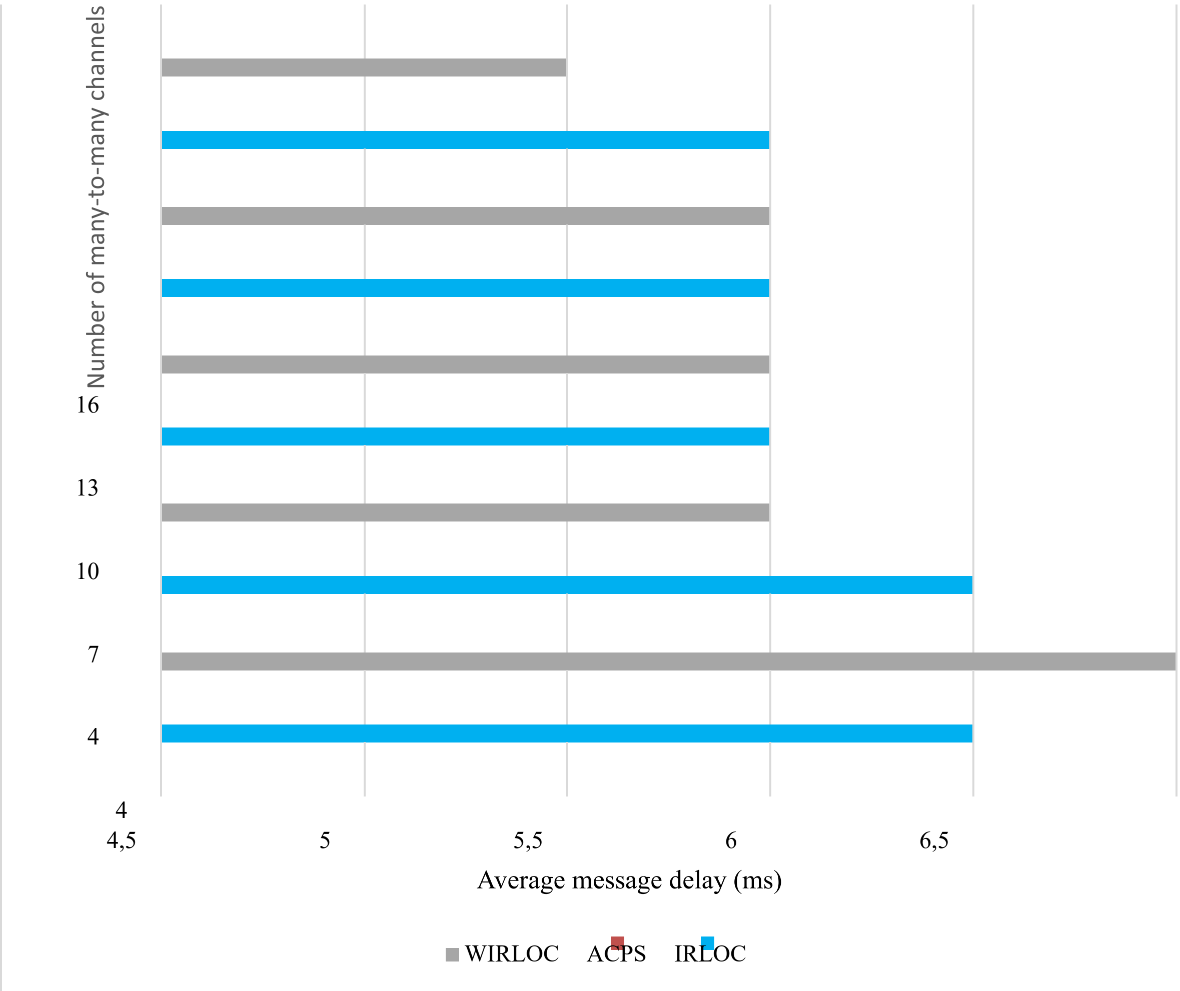


Fig. 4 Average message delay for average transmission time 2 ms.

Our results confirm observations of previous authors that there is no single method that is best for all the cases, and it heavily depends on parameters of simulation. Best example can be the method IRLOC, which in our experiment resulted in worst results in majority of the time. This can be contrasted with results found in [5] which showed that IRLOC was closer to ACPS in terms on giving best results. Our method gave the best solution the greatest number of times and never gave the worst result. We acknowledge that it may not be the case for all experiments and before selecting communication strategy one should compare all methods.

Conclusions

The paper describes a novel method of finding communication processes between simulators operating in the HLA architecture. The proposed method is generalization of IRLOC found in [4] and [5]. During method development we observed that solution gave by WIRLOC greatly depends on what federate weights are assigned. As there is no strict method for selecting federate weights, this may be an interesting area of research. Given that we observe great improvement in overall results by just modifying one weight, this problem may be good candidate to use genetic algorithms to find a solution. We compared three methods of selecting communication strategies, each having advantages in specific scenarios. As such it would be best if developers of HLA software would not limit their solutions to only one method but allowed them to compare and select one that best suites user needs.

Bibliography

- Topçu O., Oğuztüzün H. (2017). Guide to Distributed Simulation with HLA, Springer International Publishing, Simulation Foundations, Methods and Applications, doi.org/10.1007/978-3-319-61267-6.
- Akram A., Sarfraz M. S., Shoaib U. (2019). HLA Run Time Infrastructure: A Comparative Study. Mehran University Research Journal of Engineering and Technology, vol. 38, issue 4, pp. 961-972, DOI:10.22581/muet1982.1904.09.
- 1516.2-2010 - IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)- Object Model Template (OMT) Specification (2010).
- Morse K., Zyda M. (2002). Multicast grouping for data distribution management. Simulation Practice and Theory, vol. 9, issues 3-4, pp. 121–141, doi.org/10.1016/S0928-4869(01)00054-4.
- Wang J., Zheng T. (2014). A hybrid multicast–unicast assignment approach for data. Simulation Modelling Practice and Theory, vol. 40, pp. 39–63, doi.org/10.1016/j.simpat.2013.08.005.
- Nowicki T., Panufnik K., (2024). Mathematical model of communication of multiple simulators in an environment based on the HLA standard, 44th IBIMA Computer Science Conference, 27-28 November 2024, Granada Spain
- Chodowska-Wasilewska A., Nowicki T., Waszkowski, R., (2023). Simulation Exercises for the State
- Sanitary and Epidemiological Services Related to the Epidemic of Foodborne Diseases, in: Mrugalska B., Ahram T., Karwowski W., Human Factors in Engineering, Manufacturing Systems, Automation, and Interactions, CRC Press, Taylor & Francis Group, 2023
- Nowicki T., Saniuk A., Waszkowski R. (2017) Human-computer interaction in sanitary inspection simulation exercises, Advances in Ergonomics Modeling, Usability & Special Populations, Springer
- Series, Advances in Intelligent Systems and Computing, vol. 486, 2017, Soares M., Falcao Ch., Ahram
- T.Z. (Eds.), pp. 245-254, ISBN 978-3-319-41685-4