Article

Interest Manager for Networked Driving Simulation Based on High-Level Architecture

Kareem Abdelgawad [1,*], Jürgen Gausemeier [1], Michael Grafe [2] and Jan Berssenbrügge [2]

[1] Heinz Nixdorf Institute, University of Paderborn, 33102 Paderborn, Germany; Juergen.Gausemeier@hni.upb.de

[2] Fraunhofer Institute for Mechatronic Systems Design IEM, 33102 Paderborn, Germany; Michael.Grafe@iem.fraunhofer.de (M.G.); Jan.Berssenbruegge@iem.fraunhofer.de (J.B.)

* Correspondence: Kareem.Abdelgawad@hni.upb.de; Tel.: +49-5251-606-228

Abstract: In networked driving simulation, two or more human drivers participate interactively within a shared virtual environment. Thereby, typical applications of driving simulation can be extended to consider multi-driver scenarios, where a much closer approximation of reality with its unpredictability is achieved. However, the utilized network is typically prone to a considerable amount of message traffic. In addition to restricted system scalability, the resulting degradation of network performance leads to invalid simulation outcomes or unacceptable system behavior. High-Level Architecture (HLA) is the IEEE standard 1516 that provides specific guidelines for networked simulation. Data distribution management (DDM) is one of the service groups provided by the HLA standard. The aim of the DDM service is to reduce network traffic and to save effort required to process unnecessary received data. However, existing approaches for current DDM implementations show major drawbacks in terms of utilization complexity, inefficiency, and yet added network overhead. This paper presents a new concept of an interest manager that takes over the DDM functionality and avoids these drawbacks. Simulation data is exchanged between the participating driving simulators only when it is necessary according to the driving situations. The interest manager was implemented and its efficient functionality was validated by analyzing the network load of two driving maneuvers with and without the interest manager.

Keywords: networked driving simulation; high-level architecture; network data traffic; data distribution management; interest manager

1. Introduction

Driving simulators are utilized in a variety of applications, such as the development and testing of vehicle systems, training and licensing purposes, and studying the behavior of drivers [1]. Various traffic scenarios that involve a human-driven vehicle and programmed traffic participants can be created. Environmental effects can be reproduced easily, such as, e.g., foggy or snowy roads, night-time driving conditions, etc. However, conventional driving simulation provides only a rough representation of the unpredictability level encountered, when multiple human drivers interact in the same real traffic environment. The ability to create a virtual driving environment simultaneously accessed by two or more human drivers allows a much closer approximation of real-world traffic interactions. With networked driving simulation, typical applications of driving simulation can be broadened to consider multi-driver scenarios [2]. Advanced automotive technologies based on communication between vehicles and road infrastructure can be developed and tested in an interactive, yet safe and cost-effective, manner [3]. Conventional driver training can be extended to multi-driver training, where a driving instructor handles several drivers at the same time in a life-like traffic environment [4]. Studying the performance and behavior of drivers can deliver more legitimate results, if drivers not only react to programmed traffic vehicles, but also interact with other human-driven vehicles [5]. A comprehensive discussion about the benefits of networked driving simulation is presented in Reference [6]. Figure 1 shows an example platform of a networked driving simulation developed and operated at the Heinz Nixdorf Institute, University of Paderborn, in Germany.

doi:10.20944/preprints201704.0129.v1

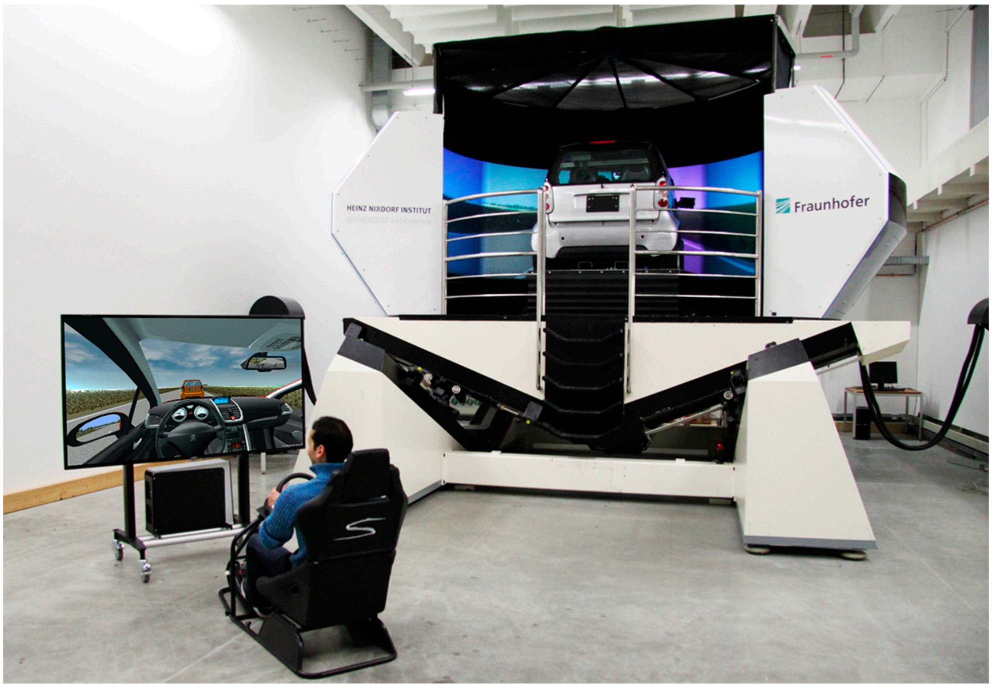

Figure 1. Platform for networked driving simulation at the University of Paderborn in Germany.

The shown platform incorporates two networked driving simulators of different fidelity levels. The smaller driving simulator variant has no motion platform. This fixed-based driving simulator is equipped with a commercial wheel-transmission-pedals set and a driving seat, providing low-cost, yet sufficient, physical feedback and control cues. The driving simulator utilizes a 60-inch high-definition screen. The simulation environment was developed with MATLAB/Simulink. The other driving simulator variant incorporates a complex motion platform. It consists of two dynamical motion parts providing five Degrees Of Freedom (DOF) to fully simulate vehicle lateral and longitudinal accelerations. The motion platform is equipped with a fixation system, which allows the utilization of different driving cabins, e.g., truck cabin or passenger vehicle cabin, so that drivers can experience various realistic control cues. This driving simulator has an eight-channel cylindrical projection system powered by eight LCD-projectors to cover a horizontal field of view of 240 degrees. The simulation environment was developed by dSPACE GmbH. The virtual driving scenes of both driving simulators were developed with Unity 3D. The following section presents a major problem that challenges mainly the scalability of networked driving simulation.

2. Problem Description

In networked driving simulation, a lot of data must be exchanged between the participating driving simulators. The position and orientation states of each main simulated vehicle represent the primary data that must be exchanged. Additional information can be exchanged to achieve more immersion and better visualization fidelity, such as, e.g., front and rear lamps state, turning indicator lamps state, front wheels orientation, wheels rotation speed. If a central traffic simulator is used, at least the position and orientation states of each simulated traffic participant must be exchanged with those of the participating driving simulators [7]. This massive amount of message traffic exchanged with each simulation step represents a lot of network overhead. The network overhead increases considerably by adding more driving simulators and simulated traffic participants. The resulting degradation of network performance is perceived mainly as data packets loss and cumulative latency [8]. Beside restricted system scalability, this leads to invalid simulation results or unacceptable system behavior. Similar networking issues have been recognized in the field of multi-player games. For playing games with slow-moving objects, networking issues do not present real problems. However, games involving fast interactions between players can be degraded severely by poor network performance [9]. The so-called dead reckoning and entity interpolation techniques are developed to overcome this problem [10]. The basic idea of dead reckoning/entity interpolation is to extrapolate/interpolate data of simulated entities in the multi-player game based on their predictable behavior. Thereby, data must not be sent very frequently in the network between the participating players. In multi-player games, entertainment remains principally the dominant factor [11]. As long as they do not perceive poor interaction behavior, game players usually accept some accuracy tolerance for the simulated entities. Dead reckoning and entity interpolation techniques proved good usability in the field of multi-player games. However, manipulating simulation data through extrapolation or interpolation techniques may not be satisfying for the scientific applications of networked driving simulation. Regarding development and test purposes, for instance, simulation results will not remain valid if the exchanged data is altered. Moreover, multi-driver training considers mainly road infrastructure and real traffic rules, such as, e.g., traffic lights and right-of-way rules. Altered and approximated position and orientation data cannot provide consequential understanding of training session results. This paper presents a concept for an interest manager that reduces network traffic load and avoids the drawbacks of existing approaches in this regard. The rest of this paper is organized as follows. Section 3 highlights the existing approaches of data distribution management in the high-level architecture standard and their drawbacks. Section 4 presents the concept of the developed interest manager used to reduce network traffic load. Concept validation using different driving maneuvers is provided in Section 5. Finally, Section 6 derives a conclusion.

3. Related Existing Approaches in HLA and Major Drawbacks

High-level architecture (HLA) is an IEEE standard 1516 for networked simulation [12]. It provides rules and functions to connect simulators in a network. The Runtime Infrastructure (RTI) is a primary component of HLA [12]. It implements a set of services that can be invoked by applications (federates) to manage data and time within the networked simulation environment (federation). These services are divided into groups [12]: federation management, declaration management, object management, ownership management, time management, and data distribution management. The provided services enable the construction of modular, flexible, and well-defined environments of networked simulation. As basic knowledge of HLA concepts is assumed, a comprehensive discussion of its terminology, rules, and services is not provided in this work. However, the service of data distribution management (DDM) is considered in particular. In ideal cases, the DDM service provides relevance filtering between simulation components to improve communication efficiency. The aim is to reduce message traffic over the network and to reduce the effort required to process received data. Nonetheless, different approaches for current DDM implementations show major drawbacks. These are recognized mainly as complexity of utilization, inefficiency, and even some added network overhead [13]. Moreover, the efficient and proper use of the DDM service requires substantial knowledge of the DDM approach of the utilized RTI implementation [14]. The DDM service of current RTI implementations typically follows either a distributed region approach or a grid-based approach [15]. In the distributed region approach, each federate has publish and/or subscribe regions. The RTI must receive the region information of each federate to perform a matching process. Update and subscribe regions match only if they actually overlap, as shown in Figure 2a. A multicast group is assigned to each publisher [16]. Region matches are used to establish communication channels between the publishers and their subscribers.

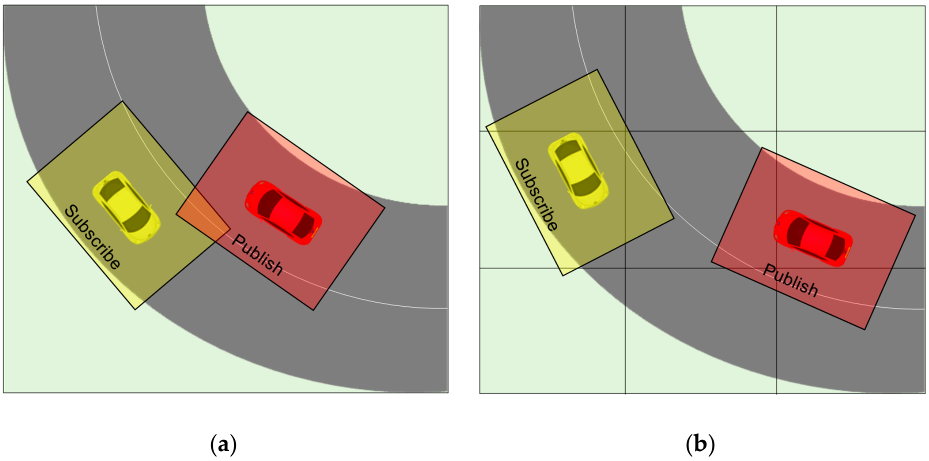

Figure 2. Data distribution management (DDM) approaches in High-level architecture (HLA): (**a**) Distributed region filtering; (**b**) grid-based filtering.

The distributed region approach guarantees that communication channels carry only relevant data to receiving federates. However, sending region information frequently to the RTI can overwhelm the network. Moreover, the RTI will be continuously busy receiving and processing a lot of notifications about changes of region information. This approach may be optimal for federations that use publish and subscribe regions with infrequent changes. For networked driving simulation in particular, frequent changes of publish and subscribe regions are typically expected as all federates are supposed to be moving continuously during the simulation.

In the grid-based approach, each federate also has publish and/or subscribe regions. The RTI partitions the entire routing space of the simulation along each dimension to form grid cells, as shown in Figure 2b. A multicast group is assigned to each grid cell. During federation execution, intersections of publish and subscribe regions with the grid cells are determined. Each federate sends data only to the multicast groups of the grid cells that intersect with its publish region. Similarly, each federate receives data only from the multicast groups of the grid cells that intersect with its subscribe region. The grid-based approach requires no exchange of region information. That is, the processes of associating regions and determining intersection with grid cells are localized. However, the intersections between regions and grid cells do not always mean intersections between publish and subscribe regions. Hence, a federate will most probably receive data that do not belong to its subscribe region. As depicted in Figure 2b, the subscriber simulator will receive data from the publisher simulator that shares the same grid cell, although the corresponding subscribe and publish regions do not actually intersect. Configuring the grid for cells of a smaller size will result in more computational efforts to update the subscriber and publisher lists of each single cell. Moreover, a large amount of smaller grid cells will be restricted by the limited number of available multicast groups.

Both distributed region and grid-based approaches of DDM use multicast routing as the underlying network mechanism for relevance filtering. However, the principle of using multicast filtering alone yields less than perfect results in practice [13]. Both approaches still require additional filtering schemes to ensure perfect filtering as required by the HLA standard [13]. Moreover, publish and subscribe regions have quadrilateral forms; usually they cannot represent the actual filtering criteria required by federates of some applications, such as, e.g., sight distance represented as a circle around a vehicle. Hence, federates still must filter received data locally. Furthermore, providing perfect filtering with DDM may require extensive changes to existing RTI implementations. Eventually, this can result in an RTI implementation that is not fully compliant with the HLA standard. Federation designers must weigh the benefits of using DDM against the added complexity and the underlying implementation factors that may impact the communication efficiency. Furthermore, federation designers that use one particular DDM approach may find that the DDM service adds larger network

doi:10.20944/preprints201704.0129.v1

overhead if another RTI implementation, i.e., a different DDM approach, is utilized. The following section presents a new concept for an interest manager that takes over the objective and functionality of the DDM service, while federations still remain fully compliant with the HLA IEEE standard 1516. In particular, the developed concept of the interest manager is applied to networked driving simulation in this work. However, it can be utilized in other applications for networked simulation in general.

4. Development Methodology

In general, interest management schemes in networked games can be classified into two categories: space-based and class-based [17]. The space-based scheme considers the relative position of objects in a virtual environment. The class-based scheme depends on the type of entities and the logical relations between them. In this work, an interest manager module is developed to avoid the drawbacks of current DDM approaches in the HLA standard. It can adopt the space-based and the class-based schemes. Principally, the interest manager module is a federate that follows the general publish-subscribe model of interest management, which is inherently adopted in the HLA standard [18]. Publishers are entities that send data, whereas subscribers are entities that receive data. An entity can be assigned arbitrarily to act as publisher and/or subscriber. Each entity can send/receive objects and interactions. The primary difference between objects and interactions is persistence. Objects can be exchanged during the whole simulation time, whereas interactions can be exchanged eventually and do not persist throughout the whole simulation time. The basic idea of the developed interest manager is to analyze the whole driving situation with respect to a given filter criterion and a threshold value. Accordingly, the interest manager notifies the participating simulators through the RTI to change their interest of data exchange. This is achieved by altering their publish—Subscribe model. In other words, the publish—Subscribe model of each participating simulator is adaptive throughout the simulation instead of using a fixed model. In general, data network communication can be performed using different packet delivery methods: unicast, broadcast, and multicast [16]. The multicast approach provides group-based packet delivery. That is, data packets are addressed to a group of destinations simultaneously. Copies of these data packets are automatically created in network routers and sent only to network segments that contain members of the group. The interest manager developed in this work assumes that the utilized RTI implementation and network support multicast routing. Figure 3 shows a layout for the interplay between the developed interest manager, RTI software, and two participating driving simulators. The following subsections discuss a typical communication process between the aforementioned system components.

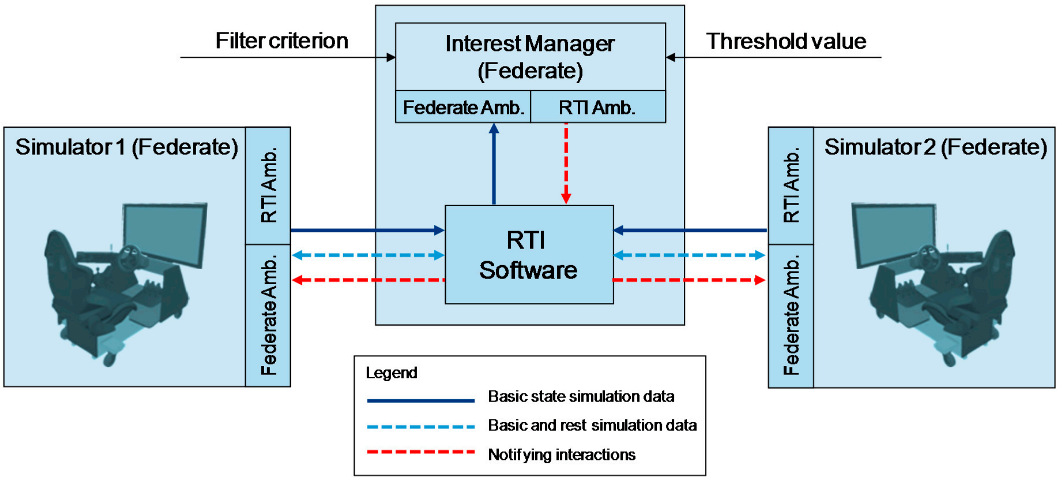

Figure 3. Layout for the interplay between system federates and the developed interest manager.

4.1. Sending Basic State Simulation Data

During the simulation, each participating driving simulator sends its x and y positions to the RTI software over the network. This information represents the basic state simulation data. The rest of simulation data, discussed in the section of problem descriptions in this work, must not be sent to the RTI software over the network at this instance. The participating driving simulators do not receive information about each other through the RTI software and the network is loaded minimally. The basic state simulation data are forwarded to the interest manger only, so that it can monitor the driving situation during the simulation. As the interest manager and the RTI software run on the same machine, their data exchange does not present any network load.

4.2. Analyzing Driving Situation

The interest manager obtains a filter criterion from the user, such as, e.g., sight distance or rendering distance, together with a threshold value. The sight distance defines the area available for the driver around the simulated vehicle. The render distance is the maximum distance, at which 3D objects are drawn by the utilized rendering engine. The filter criterion and its threshold value define the interest region of the simulated vehicles. The interest manager analyzes the driving situation according to the (x, y) position of each simulated vehicle. That is, it determines the relevance between each pair of simulated vehicles with respect to the given filter criterion and its threshold value.

4.3. Generating Notifying Interactions

According to the analyzed driving situation, the interest manager eventually notifies the driving simulators to change the publish—Subscribe models. The notifications are sent individually to each driving simulator through the RTI software in the form of interactions. To that end, each driving simulator must subscribe to the interactions of the interest manager. However, the interest manager does not send notifications, i.e., interactions, to driving simulators of an unchanged neighborhood. The neighborhood is defined in this context as a set of driving simulators, which have to exchange simulation data according to the given filter criterion and its threshold value. A large number of notifying interactions may be sent by the interest manager if simulated vehicles frequently enter and exit the interest region of each other. One possible solution to avoid this problem is to define the filter criterion with hysteresis threshold values. That is, simulated vehicles enter the interest regions of each other below a certain threshold value and exit it above another value.

4.4. Sending and Receiving the Rest of Simulation Data

When a driving simulator receives a notifying interaction from the interest manager, it alters the publish—Subscribe model. Then, it starts eventually to send the rest of simulation data to the RTI software, which forwards them to the relevant driving simulators. Similarly, it starts eventually to receive the basic and rest of simulation data from the relevant driving simulators through the RTI software. This means, the RTI software uses the changing publish-subscribe models to throttle the data placed on the network adaptively. Receiving interactions from the interest manager through the RTI software may present some minimal network load. However, the rest of simulation data is generated and sent to the RTI software, and hence, to the other driving simulators over the network, only when it is necessary according to the driving situations.

Using this approach, the RTI software must not be notified about the frequently changing interest region information. This adds considerable benefits for reducing network load in comparison to the distributed region approach of DDM. Moreover, the RTI software is disbanded from the computational effort and the assignment of multicast groups is reduced in comparison to the grid-based approach of DDM. The following section presents the results of example validation maneuvers for the introduced concept of the interest manager.

doi:10.20944/preprints201704.0129.v1

5. Results and Discussion

The concept of the interest manager presented in this work was implemented as a C++ federate. For networked simulation, there are various commercial and non-commercial tools supporting the HLA IEEE standard. The Portico software is utilized during the validation process in this work. It is an open source and cross-platform RTI implementation fully compliant with the HLA IEEE standard [19]. The Portico software includes the basic services necessary to coordinate the operation of the federates and to support the data exchange during the runtime execution. The Portico software follows a decentralized RTI approach, where a component of the RTI implementation resides at each system participant side.

Two examples with different numbers of driving simulators are presented to validate the new concept of the interest manager and its implementation as a C++ federate. In these particular examples, the (x, y) positions of the simulated vehicles represent the basic state simulation data, whereas the z position and the three orientation angles represent the rest of simulation data. The sight distance is chosen as a filter criterion with an exemplary threshold value of four meters for validation convenience. The Wireshark tool is used during the validation process. It is an open source and cross-platform analyzer used to observe network traffic [20]. With a 0.1 s tick interval, this software gives the number of exchanged data packets with and without the utilization of the developed interest manager. Accordingly, the percentage of reduced network load is calculated. The network traffic in this context is represented by the rest of simulation data only. The network load of the basic simulation data is constant and that of the eventual notifying interactions is minimal. Hence, they are excluded from the calculation of the percentage of reduced network load.

5.1. Example Maneuver 1: Two Driving Simulators

The first validation example uses two simple vehicle models that drive toward each other with a constant speed of 22 km/h, as shown in Figure 4. Each vehicle model represents a separate federate. Together with the interest manager, the whole federation is composed of three federates.

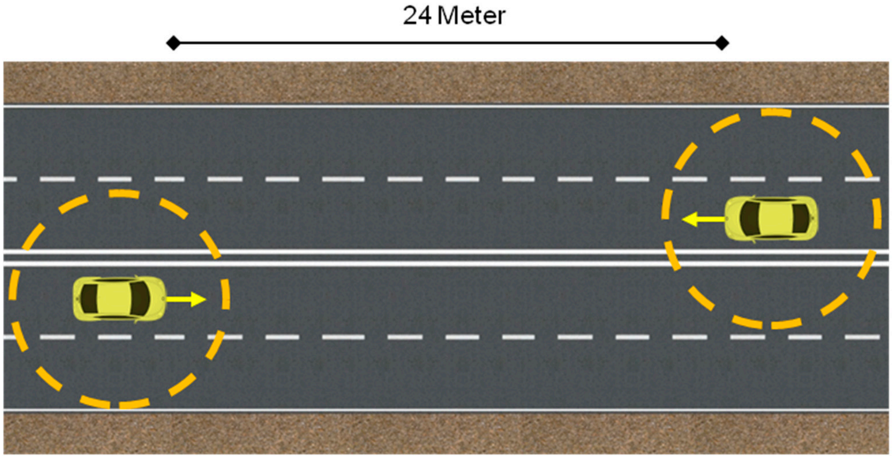

Figure 4. Validation example maneuver with two networked driving simulators.

The two vehicles are separated by a distance of 24 m and they start moving at the same time. The simulation has been stopped after the two vehicles finished a travel distance of 24 m. The network loads represented as the number of packets of the rest of simulation data sent over the network with and without the interest manager have been captured, as depicted in Figure 5.

doi:10.20944/preprints201704.0129.v1

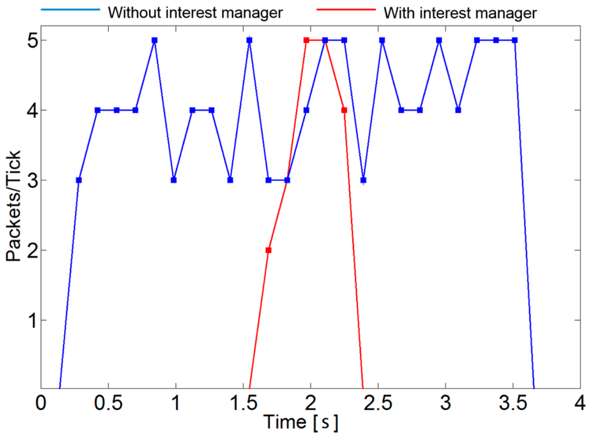

Figure 5. Network load with and without interest manager—example maneuver 1.

With using the interest manager, the rest of simulation data is exchanged only when the interest regions of both simulated vehicles overlap. Thereby, the network is minimally loaded with the basic state simulation data during the time when the interest regions do not overlap. For this particular driving situation and with respect to the rest of simulation data, about 80% of network load has been reduced in comparison to the case where no interest manager and no DDM approach were utilized. This considerable network load reduction ensures that eventually other exchanged simulation data packets have sufficient network bandwidth. Hence, there are less chances for network latency accumulation. This benefit increases the system reliability and enhances the simulation performance. Moreover, it increases the scalability of the simulation system. That is, there is a sufficient network bandwidth to integrate more driving simulators within the same environment as proven by the following validation example.

5.2. Example Maneuver 2: Four Driving Simulators

The second validation example uses four simple vehicle models to prove the efficiency of the developed interest manager if more than two driving simulators participated within the same environment. Each pair drives toward each other with a constant speed of 22 km/h as shown in Figure 6. Each vehicle model represents a separate federate. Together with the interest manager, the entire federation consists of five federates.

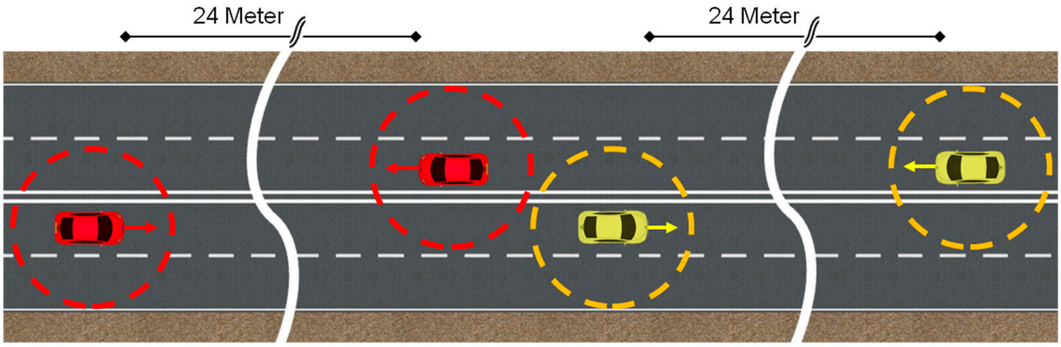

Figure 6. Validation example maneuver with four networked driving simulators.

In the first variant of this validation example, the four vehicle models start moving at the same time. Similar to the first validation example, the simulation has been stopped after the four vehicles finished a travel distance of 24 m. The network loads with and without the interest manager have been captured, as depicted in Figure 7.

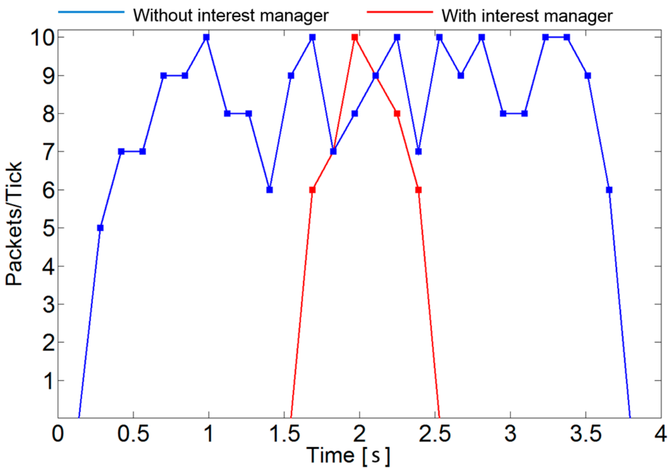

Figure 7. Network load with and without interest manager—first variant of example maneuver 2.

The interest manager allows the exchange of the rest of simulation data only when the interest regions of each pair of simulator vehicles overlap. For this particular driving situation, about 78% of network load has been saved in comparison to the case where no interest manager and no DDM approach were utilized. In the second variant of this validation example, two vehicle models start moving toward each other at the same time. After reaching the end point, the other two vehicle models start moving toward each other at the same time. The network loads with and without the interest manager have been captured, as depicted in Figure 8.

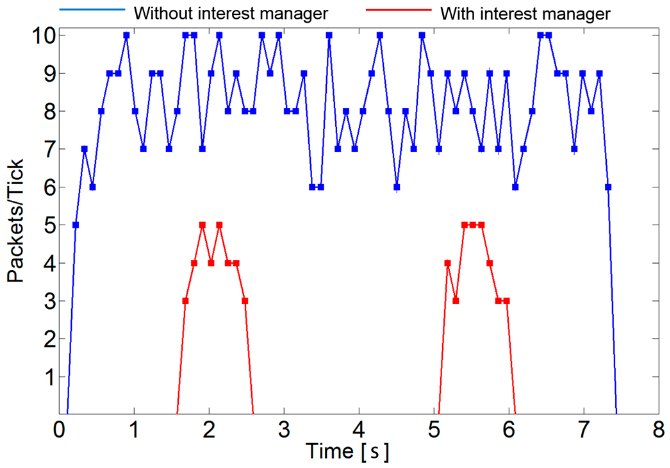

Figure 8. Network load with and without interest manager—second variant of example maneuver 2.

Similar to the previous cases, the interest manager allows the exchange of the rest of simulation data only when the interest regions of each pair of simulated vehicles overlap. For this particular driving situation, about 88% of network load has been saved in comparison to the case where no interest manager and no DDM approach were utilized.

In general, the amount of reduced network load depends on the driving situation and the given filter criterion along with its threshold value. In a driving situation where all participating simulator vehicles belong to the same neighborhood, there will be no reduction for the network load. However, this drawback applies to the typical DDM approaches of HLA as well. The introduced concept of the interest manager, as well as the typical DDM approaches, represents an endeavor to reduce network load when applicable. Thereby, there is eventually less chance for data packets loss and cumulative network latency. Nonetheless, thorough network design remains a crucial requirement and challenge for a functional and useful environment of networked simulation. The following section derives a conclusion for the presented work.

6. Conclusions

In networked driving simulation, different human-driven vehicles can participate and interact in a common virtual traffic environment. Typical applications of driving simulation can consider new promising scenarios. However, the large amount of data exchange between the driving simulators may affect network performance, and hence, the usability of the simulation platform. High-level architecture (HLA) is an IEEE standard 1516 for networked simulation. Its data distribution management service (DDM) aims to reduce network traffic. However, the utilization of the DDM service is accompanied by some drawbacks, where users usually have to find a compromise. This work presented a concept for an interest manager to take over the functionality of the DDM service while avoiding its common drawbacks. It can be applied to any number of participating driving simulators, as shown during the validation process. The interest manager will reduce network traffic load when applicable. However, careful network design remains a primary aspect for a functional networked simulation environment. This is represented by selecting the right network technology, transmission protocol, and eventually networked simulation architecture. The presented concept can be applied to other distributed simulation areas. This can be realized by simply using different filtering criteria and threshold values for interest matching. The concept has been validated using simple vehicle models in different driving maneuvers. The validation process showed very good results for reducing network traffic load, and hence, enhancing simulation performance and increasing system scalability.

Acknowledgments: The presented work was carried out by the authors during the collaborative research and development activities at the Heinz Nixdorf Institute and the Fraunhofer Institute in Paderborn, Germany. No necessary special or external funding resources were acquired to produce or publish the research and development outcomes of this work.

Author Contributions: Kareem Abdelgawad developed the concept, carried out the implementation, and wrote the paper. Jürgen Gausemeier was consulted regarding the call for action and directed the research topic. Michael Grafe contributed to the validation and evaluation processes. Jan Berssenbrügge developed the visualization environment for the driving simulators. The latter three authors examined the work and reviewed the paper.

References

1. Fisher, D.L.; Caird, J.; Rizzo, M.; Lee, J. Application in engineering. In *Handbook of Driving Simulation for Engineering, Medicine, and Psychology*, 2nd ed.; CRC Press Taylor & Francis Group: Boca Raton, FL, USA, 2011; pp. 33–43. ISBN: 978-1-4200-6100-0.
2. Muehlbacher, D. Multi-driver simulation—The link between driving simulation and traffic simulation. In Proceedings of the International Scientific Conference on Mobility and Transport Mobil.TUM 2015, Munich, Germany, 30 June–1 July 2015.
3. Lu, N.; Cheng, N.; Zhang, N.; Shen, X.; Mark, J. Connected vehicles: Solutions and challenges. *IEEE Internet. Things. J.* **2014**, *1*, 289–299. [CrossRef]
4. Kandhai, K.; Smith, M.; Kanneh, A. Immersive driving simulation for driver education and analysis. In Proceedings of the IEEE 16th International Conference on Computer Games (CGAMES), Louisville, KY, USA, 27–30 July 2011; pp. 288–292.
5. Maag, C.; Muhlbacher, D.; Mark, C.; Kruger, H.P. Studying effects of advanced driver assistance systems (ADAS) on individual and group level using multi-driver simulation. *IEEE Intell. Transp. Syst. Mag.* **2012**, *4*, 45–54. [CrossRef]
6. Oeltze, K.; Schießl, C. Benefits and challenges of multi-driver simulator studies. *J. Intell. Transp. Syst. (IET)* **2015**, *9*, 618–625. [CrossRef]
7. Abdelgawad, K.; Henning, S.; Biemelt, P.; Gausemeier, S.; Trächtler, A. Advanced traffic simulation framework for networked driving simulators. In Proceedings of the 8th IFAC Conference on Advances in Automotive Control (ACC), Kolmården, Sweden, 20–23 June 2016; Volume 49, pp. 101–108.

8. Wu, Q.; Gu, Y.; Zhu, M.; Rao, N. Optimizing network performance of computing pipelines in distributed environments. In Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS), Dublin, Ireland, 14–18 April 2008.
9. Debroy, S.; Ahmad, M.; Iyengar, M.; Chatterjee, M. Critical sections in networked games. In Proceedings of the 13th IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013.
10. McLoone, S.; Walsh, P.; Ward, T. An enhanced dead reckoning model for physics-aware multiplayer computer games. In Proceedings of the 16th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Dublin, Ireland, 25–27 October 2012.
11. Quandt, T.; Kröger, S. Introduction: Multiplayer gaming as social media entertainment. In *Multiplayer: The Social Aspects of Digital Gaming*, 1st ed.; Routledge: Abingdon, UK, 2013; ISBN: 978-0415828857.
12. Awais, M.; Mueller, W.; Elsheikh, A. Using the HLA for distributed continuous simulations. In Proceedings of the 8th Congress on Modeling and Simulation (EUROSIM), Wales, UK, 10–13 September 2013.
13. Tietje, H. *Erfahrungen mit dem Einsatz von DDM. White Paper from the Competence Center Informatik GmbH (CCI)*; Springer-Verlag: Meppen, Germany, 2001; ISBN: 978-3-540-62415-8.
14. Pan, K.; Turner, S.; Cai, W.; Li, Z. Implementation of data distribution management services in a service oriented HLA RTI. In Proceedings of the Winter Simulation Conference (WSC), Austin, TX, USA, 13–16 December 2009.
15. Sarbazi-Azad, H.; Zomaya, A. *Large Scale Network-Centric Distributed Systems*; Wiley-IEEE Computer Society Press: London, UK, 2013; ISBN: 978-0-470-93688-7.
16. Karaoglu, B.; Heinzelman, W. Multicasting vs. broadcasting: What are the Trade-Offs? In Proceedings of the IEEE Conference on Global Telecommunications (GLOBECOM 2010), Miami, FL, USA, 6–10 December 2010.
17. Wang, L.; Turner, S.; Wang, F. Interest management in agent-based distributed simulations. In Proceedings of the 7th IEEE International Symposium on Distributed Simulation and Real-Time Applications, Delft, The Netherlands, 23–25 October 2003.
18. Ravindran, K.; Mukhopadhyay, S.; Sidhanta, S.; Sabbir, A. Managing shared contexts in distributed multi-player game systems. In Proceedings of the 6th IEEE International Conference on Communication Systems and Networks (COMSNETS), Bangalore, India, 6–10 January 2014.
19. Nouman, A.; Anagnostou, A.; Taylor, S. Developing a distributed agent-based and DES simulation using portico and repast. In Proceedings of the IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Delft, The Netherlands, 30 October–1 November 2013.
20. Wang, S.; Xu, D.; Yan, S. Analysis and application of wireshark in TCP/IP protocol teaching. In Proceedings of the International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT), Shenzhen, China, 17–18 April 2010.