An Area-of-Interest based Communication Architecture for Shared Haptic Virtual Environments

Clemens Schuwerk, Nakul Chaudhari and Eckehard Steinbach
Institute for Media Technology
Technische Universität München, Germany
Email: {clemens.schuwerk, eckehard.steinbach}@tum.de, nakul.chaudhari@in.tum.de

***Abstract*—Communication architectures conceived for Shared Haptic Virtual Environments (SHVEs) are based on either the client-server or the peer-to-peer paradigms. High-rate rendering and communication between collaborating users quickly leads to performance bottlenecks, if the virtual environment's size and complexity or the number of collaborating users increases. We propose a decentralized communication architecture for SHVEs, which exploits areas of interest (AoI) of a user to dynamically form smaller communication groups. High-rate information exchange following the client-server paradigm is used within these groups to support satisfactory haptic collaboration and consistency. A single group member is selected to simulate the object states and to relay this state to the other group members. Peer-to-peer inter-group communication is also used, based on spatial proximity, to further reduce the overall communication load. We implemented a prototype based on our proposed architecture and present some first evaluation results. They serve as a proof-of-concept and show the effectiveness of dynamic group maintenance in conjunction with additional traffic control schemes.**

I. Introduction

Integrating the haptic modality into computer applications is a relatively new, but quickly growing research area. The development of sophisticated force feedback devices pushes the research on computer haptics algorithms, used for rendering interaction forces with virtual objects [1]. The ability to touch and physically interact with virtual objects enables several interesting new applications, e.g., in the area of telepresence systems, education, training, assembly simulations, scientific data visualization or entertainment [2].

Several challenges in the area of computer haptics are inferred from the inherent exchange of energy between the human controlling the haptic device and the virtual environment (VE). The user commonly manipulates a virtual probe through the haptic device. Collisions between this probe and the objects in the VE are detected at runtime and interaction forces are rendered accordingly. Hooke's Law $F = k \cdot d$, where k is the object's stiffness and d the penetration depth of the probe into the virtual object, is commonly used for this purpose [3]. Thus, a control loop between the virtual environment and the human controlling the haptic device is closed. Stability of this loop and the high temporal resolution of the human haptic perception system require a high haptic update rate of about $1kHz$ to achieve high-fidelity force-feedback. Algorithms used for collision detection, force rendering and physics simulation are hence time-constraint, as only 1 ms is available for their computation. Although suitable algorithms have been proposed in the literature (an overview can be found in [3]) increased complexity of the virtual environment leads to computational bottlenecks in the rendering pipeline.

In Shared Haptic Virtual Environments (SHVEs), several users can collaboratively manipulate and interact with virtual objects. The incorporation of the haptic modality leads to an improved sense of togetherness between collaborating users [4]. Moreover, advances in communication infrastructure like the Internet enable haptic collaboration between globally distributed users. Current research in the area of SHVEs investigates mainly small virtual worlds in terms of spatial size, as well as number of participating users. SHVEs with hundreds of collaborating users, however, will become feasible, if affordable end-user haptic devices are available. Massively multiplayer online games (MMOGs) including haptic force feedback, for example, are an interesting future application. However, the afore-mentioned requirements for stable and high-fidelity haptic rendering in virtual environments impose severe challenges on the underlying communication architecture for SHVEs regarding scalability and consistency.

The decentralized architecture proposed in this paper is based on dynamically built communication groups. These groups are built according to the user's current area of interest (AoI) to reduce the overall communication load. The concept of AoI is adopted from research on MMOGs and is based on the idea that communication and consistency constraints between users can be alleviated according to their spatial distance in the virtual world. We consider two different AoIs (haptic and visual) to avoid the transmission of unnecessary update information. This concept is especially promising for SHVEs, as high-rate information exchange is needed only between haptically collaborating users. Note that haptic interaction with deformable objects is a challenging research area itself [5] and, thus, we focus on rigid objects in this work.

The remainder of this paper is structured as follows. In Section II we explain different communication architectures for SHVEs commonly used in the literature, followed by a short review of relevant traffic control schemes in Section III. In Section IV we describe the proposed communication architecture in detail. Our prototype as well as some first evaluation results are presented in Section V and Section VI. Finally, we summarize and conclude with planned directions for future work in Section VII.

II. Communication Architectures for SHVEs

A client-server architecture, referred to as *Consistency Client-Server Architecture* throughout this paper, with local force rendering at the clients and a centralized physics engine running on the server to simulate the state of all objects in the VE is often used in related literature (e.g. in [6], [7], [8]). Herein, a local database is maintained at each client, storing the current state of the complete virtual environment. It is used to render the visual scene as well as interaction forces between the haptic device and the VE locally at the clients. The objects' state, calculated by the physics engine on the server, is transmitted to the clients to update these databases. The centralized physics engine has the advantage of being the only entity in the distributed application calculating the object state and, thus, ensures consistency between the users.

The centralized server constitutes a single point of failure and bottleneck, as it simulates the object state for all clients. Fully distributed applications, based on the peer-to-peer (P2P) communication paradigm, can also be conceived to build SHVEs (e.g. in [9], [10]). Each user transmits the device state to every other user and renders the VE's state completely independently by means of a locally deployed physics simulation. In presence of communication delay, however, every user will use outdated information about the other users' device to simulate the VE's state. As a result, the distributed object state databases will become inconsistent. Consistency maintenance schemes, as for example proposed in [10], need to be applied to resolve this issue. Although the communication with a centralized entity is avoided, a fully-meshed P2P network is necessary in a naive implementation as every user needs to exchange information with every other user. Hespanha et al. [11] introduced the concept of object ownership to share the computation and communication load between users. Every object is constantly owned by only one user, responsible for simulating this object's state. Then, based on spatial proximity of objects in the VE, communication groups with frequent information exchange are built. In our proposed architecture, object ownership can also roam between users to avoid the burden of simulating an object state a user is currently not interested in.

Several dedicated simulation servers, distributed across the communication network are used in [12]. Whenever a user wants to interact with an object, the server with the lowest communication delay is asked to simulate the object state. If the object is already simulated by another server, the user is redirected to this server. Thus, smaller client-server architectures are dynamically built and the object ownership can roam between the dedicated servers. These servers continually synchronize each other by means of direct server-to-server communication. Our proposed architecture also builds client-server architectures dynamically, however, without the use of dedicated servers. Although the purpose of the communication architecture in [12] is to serve only a small group of users, scalability can be achieved by increasing the number of dedicated servers.

Traditionally, MMOGs follow the same approach to achieve scalabilty [13]. Research, however, focuses more on P2P application-layer overlay networks and the exploitation of spatial proximity of users in the virtual world to reduce the communication load [14]. Compared to traditional MMOGs, SHVEs put more severe requirements on the communication delay and update rates between collaborating users due to the afore-mentioned physical interaction of the human with the VE. Only seven update packets per second per user need to be transmitted, for example, in the game investigated in [13]. In contrast, on the forward channel of a telepresence and teleaction system, which is comparable to the transmission of the users' device state in SHVEs, about 100 velocity/position packets are necessary, even with lossy perceptual data reduction applied [15].

III. Traffic Control

Due to the high update rate of commonly $1kHz$ in haptic applications and the resulting high packet rates, different traffic control schemes to reduce the communication load on the network are proposed. The update rate from the centralized consistency server to the clients is adjusted according to the current network conditions in [6] by simply dropping packets to avoid congestion in the network. In [8], haptic media units, meaning device position and object states, are assigned with priorities based on their predictability and loss effect. These priority values, together with the monitored network state, are then used to drop update packets eventually. Prediction of object states is used at the clients in both schemes to absorb missing state updates. Prediction of missing state updates is also one of the key concepts used in *dead reckoning* (DR), a method to reduce update packets in distributed simulations [16]. The entity issuing state updates, e.g. the afore-mentioned consistency server, implements the same prediction method as the receiving entities in the simulation. New updates are only triggered if the error between the real state and the predicted state exceeds a threshold. This threshold has to be defined carefully according to the VE's haptic rendering parameters, e.g. the object's stiffness, to achieve haptic transparency in SHVEs adopting the Consistency Client-Server architecture. If users are in contact with an object and the object's state is updated suddenly, they will feel a sudden change in force feedback. The dead reckoning approach proposed in [7] dynamically adapts the error threshold to keep these force changes below human perception limits.

The transmission of the device state in SHVEs has gained less attention. Perception-based data reduction schemes for TPTA systems, e.g. the deadband approach presented in [15], or the afore-mentioned dead reckoning approach can be used to control the number of device state updates (position and velocity) in a similar way as used on force samples, respectively object state updates.

IV. Hybrid Communication Architecture

Our proposed communication architecture is a fully distributed P2P architecture, wherein spatial proximity in the virtual world is used to dynamically form groups and to reduce the overall communication load. We define the *Visual AoI* and the *Haptic AoI* in order to adapt the update rates between peers in three levels. If users are far away of each other in the VE and out of any AoI, updates are sent at the minimum rate. If they are close enough (the *Visual AoIs* are intersecting) the update rate is increased to provide a better visual reference at the other peers. As soon as they are close to each other (*Haptic AoIs* are intersecting), haptic interaction with shared objects is likely to occur and communication groups are formed. State information is exchanged frequently within these groups to support high-fidelity haptic interaction between

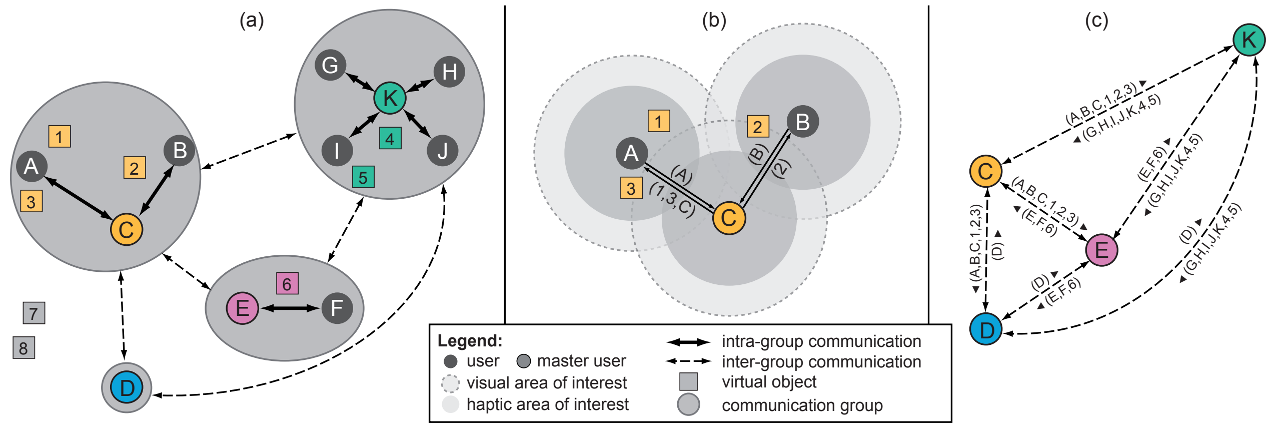

Fig. 1. Proposed communication architecture: (a) High-level design with dynamic groups, high-rate intra-group and low-rate inter-group communication. Selected master users are coloured. (b) Example for intra-group communication of the group with users A,B,C and objects 1,2,3. (c) Inter-group communication between master users.

users. Although our approach does not use a dedicated server, it is a hybrid architecture as every group acts like a small client-server architecture with only one peer simulating the state of all objects contained in the group. Besides using spatial proximity and AoIs to avoid unnecessary update packets, dead reckoning is used to further reduce the update rates between peers. The basic communication principles are explained in the following.

A. *High-level overview*

A high-level overview of the distributed haptic application is shown in Figure 1a. There are eleven users (A-K) and eight objects (1-8). Every user maintains a complete copy of the virtual environment. Several communication groups are formed dynamically, which might differ in the number of participants. While there are 5 users in the group at the top-right, there is also a single-user group at the bottom-left. A master user acts like the centralized server in the *Consistency Server Architecture* described in Section II. Every participating peer is equipped with a local physics engine, which is activated only for the master users to simulate the objects owned by the group. In Figure 1a, for example, user C calculates the state for objects 1, 2 and 3 depending on his own and the received device state from users A and B. This virtual object state is then sent back to A and B. A consistent VE state is guaranteed as only one physics engine is running within a group.

Special considerations need be taken into account regarding the master node selection. As it simulates the virtual world partially, it needs to be powerful enough to run the physics simulation within the time constraint of $1ms$ given by the haptic loop. Additionally, its communication link should be capable of serving high update rates. The most important criteria for realistic haptic interaction, however, is the communication delay to other group members and, thus, should be the prioritized metric. For simplicity, we select the peer with the highest ID as the master in our current prototype. Master selection algorithms are subject to future work and research related to general P2P networks can be adopted for this purpose [17]. Additionally, groups need to be split if a master becomes overloaded.

Object ownership can change at runtime. If, for example, a group moves away from an object, the master user dispenses the ownership as soon as the object comes to rest, by triggering a control message to the other master users including an event notification and the latest object state. Thus, also objects that are currently not owned by any master user are possible, e.g. the objects 7 and 8 in Figure 1a. The next group approaching such an object will take over the ownership, again by triggering a control message.

Users as well as objects can freely move within the virtual world and, thus, groups have to be maintained dynamically. Master users need to be aware of the positions of other users close to them. This is ensured by updates transmitted between group masters, however, at a lower rate compared to the intra-group communication. Users form a group, if their *Haptic AoIs* are intersecting. Special attention needs to be drawn to users moving along the group border, as several successive merge and split operations could be initiated. We apply some form of hysteresis as a countermeasure in our prototype and use slightly different distance thresholds for both actions. If two groups approach each other, they form a new bigger group by selecting a new master. The size of a single group is constrained by the capabilities of the selected master user. In the MMOG in [13] the group size is limited to 10. However, due to communication and computation demands of a distributed haptic application, group size limitations for SHVEs, depending on the type of application, need to be investigated in the future.

B. *Intra-group communication*

The object states (position, velocity, rotation, rotational velocity) are simulated only at the selected master user and relayed to the other group members to update the local object state databases. Interaction forces between the virtual probe, representing the haptic device, and virtual objects are rendered locally at each user. These forces are then displayed to the user through the haptic device. The current haptic device positions as well as the object states are transmitted at a high rate based on each user's area of interest. We use a simple sphere to represent the AoI in our current prototype. More sophisticated AoI descriptions for interest management in MMOGs are proposed in the literature [14], which might be used to further decrease the update rates in future work.

Figure 1b shows a detailed view of the high-rate intra-group communication in our prototype. The inner grey circles around the users A, B and C denote their *Haptic AoIs*. The selected master user C receives A's and B's haptic device position (the update packets are denoted as (A) and (B) in Figure 1b) and simulates the object states accordingly. The latest state is then sent back to the other group members according to their respective AoIs. The high-rate update packets sent to user A, for example, contain only the objects 1 and 3 as they are located within his *Haptic AoI*. Additionally, C's device state is piggybacked and used as a visual reference at A to support haptic collaboration between A and C. B's device state is not included in the high-rate updates sent from C to A, as A's and B's *Haptic AoIs* do not intersect. Their *Visual AoIs*, however, intersect and, thus, also B's position and object 2's state is piggybacked occasionally to provide a visual reference at user A. Thus, within a single group, the master user selectively updates the other users' object databases according to their respective AoIs.

We use three levels of update rates based on AoIs to reduce the packet rate. Thereby, a granularity between high-rate (intersecting *Haptic AoIs*), mid-rate (intersecting *Visual AoIs*) and low-rate (inter-group communication) information exchange can be achieved. In general, every traffic control mechanism introduced in Section III can be used to further reduce the packet rates in the dynamic client-server architectures. In our prototype, however, we implement standard dead reckoning. The performance of the proposed group management and traffic control scheme regarding packet and bit-rate reduction is evaluated in Section VI.

C. Inter-group communication

Low-rate state updates are transmitted between master users of different groups in a P2P fashion to relay the complete group's state to other groups as depicted in Figure 1c. These updates include the state of all simulated objects, as well as the positions of all group members. Besides this, control messages about object ownership changes are transmitted immediately to other groups. Inter-group communication is generally handled at a maximum rate of 1 update/s in our current implementation. Only if two groups are close to each other and their *Visual AoIs* intersect, the rate is increased to a maximum of 10 updates/s to provide a better visual reference. Obviously, the fully-meshed communication between all current master users leads to another bottleneck in the architecture. This burden is alleviated compared to the high-rate communication necessary between collaborating users due to the proposed group management, as inter-group communication not necessarily needs to be kept at the highest rate. Overlay P2P architectures proposed for MMOGs [14] might be adopted in the future to improve organization and scalability of the inter-group P2P communication.

V. Prototype

We implemented a distributed haptic application based on Chai3D, an open source library for haptic rendering [18]. Within the environment, a grid of movable and touchable virtual cubes is distributed on the ground. Each user's viewport is limited to a small fraction of the complete environment. Users are able to roam in the VE with the help of the keyboard, while they can use their haptic device (PHANToM Omni, Sensable Technologies Corp) to manipulate and touch the virtual cubes. Every user maintains a full copy of the virtual environment, containing a database of all objects states and other users' position and velocity.

The communication architecture proposed in Section IV is deployed on top of the basic haptic application to update the distributed databases. UDP is used as transport layer protocol to exchange information between users. Although UDP is generally preferred in haptic applications due to the smaller overhead and better real-time capabilities [19], control messages for group maintenance should be sent reliable, by using either TCP or using reliability control on top of UDP. We use only UDP in our first prototype, as the evaluation targets towards a proof-of-concept of the proposed underlying communication architecture. It is subject of future work to investigate our distributed application with a sophisticated network simulation, including delay, jitter and packet-loss, to improve the system's resilience against these network impairments. In the current test bed, two machines, denoted as A and B in the following, run the application and are connected through a 1 Gbit/s switch.

High, mid and low update rates are set to 1000, 10 and 1 updates/s and are chosen according to the users' AoIs. Dead reckoning is used additionally to reduce the packet rates. Thus, a new update packet is triggered only if the state error exceeds a pre-defined threshold. Position and rotational thresholds were chosen in pre-tests to allow satisfactory haptic interaction with virtual objects. First order linear prediction is used between state updates to extrapolate the object and device movement.

VI. Evaluation

We logged the packets transmitted during a test session with an interaction between the users A and B to evaluate the effect of the proposed architecture. The measured results are shown in Fig. 2. Both users are separated in the beginning of the session and do not form a group (phase 1, $0s \leq t \leq 10s$). Thus, they have ownership of objects around them and simulate the object state locally. Afterwards, B moves towards A and the inter-group communication is increased as soon as their visual AoIs intersect (phase 2, $11s \leq t \leq 21s$). At $t = 22s$ they are close enough to form a group and A is selected to be the master user. After some collaborative interaction with a shared object (phase 3, $23s \leq t \leq 62s$), B moves again away from A. The group is split (phase 4, $63s \leq t \leq 72s$) and inter-group communication is further reduced (phase 5, $73s \leq t \leq 80s$).

Fig. 2a and Fig. 2b plot the update rates for the haptic interface point (HIP), the haptic device representation in the virtual world, sent to B, respectively A. The object state update rates are plotted in Fig. 2c and Fig. 2d. The reduced update rates due to the exploitation of the users' AoIs are denoted by the green bars in Fig. 2a-d. The yellow bars denote the update rates if dead reckoning is applied on top. The resulting packet rates and bit rates transmitted over the communication network are shown in Fig. 2e and Fig. 2f.

Both users are separated during phase 1 and 5 and, thus, inter-group communication is kept at the lowest rate for both the HIP and the object state, resulting in only 1 packet/s

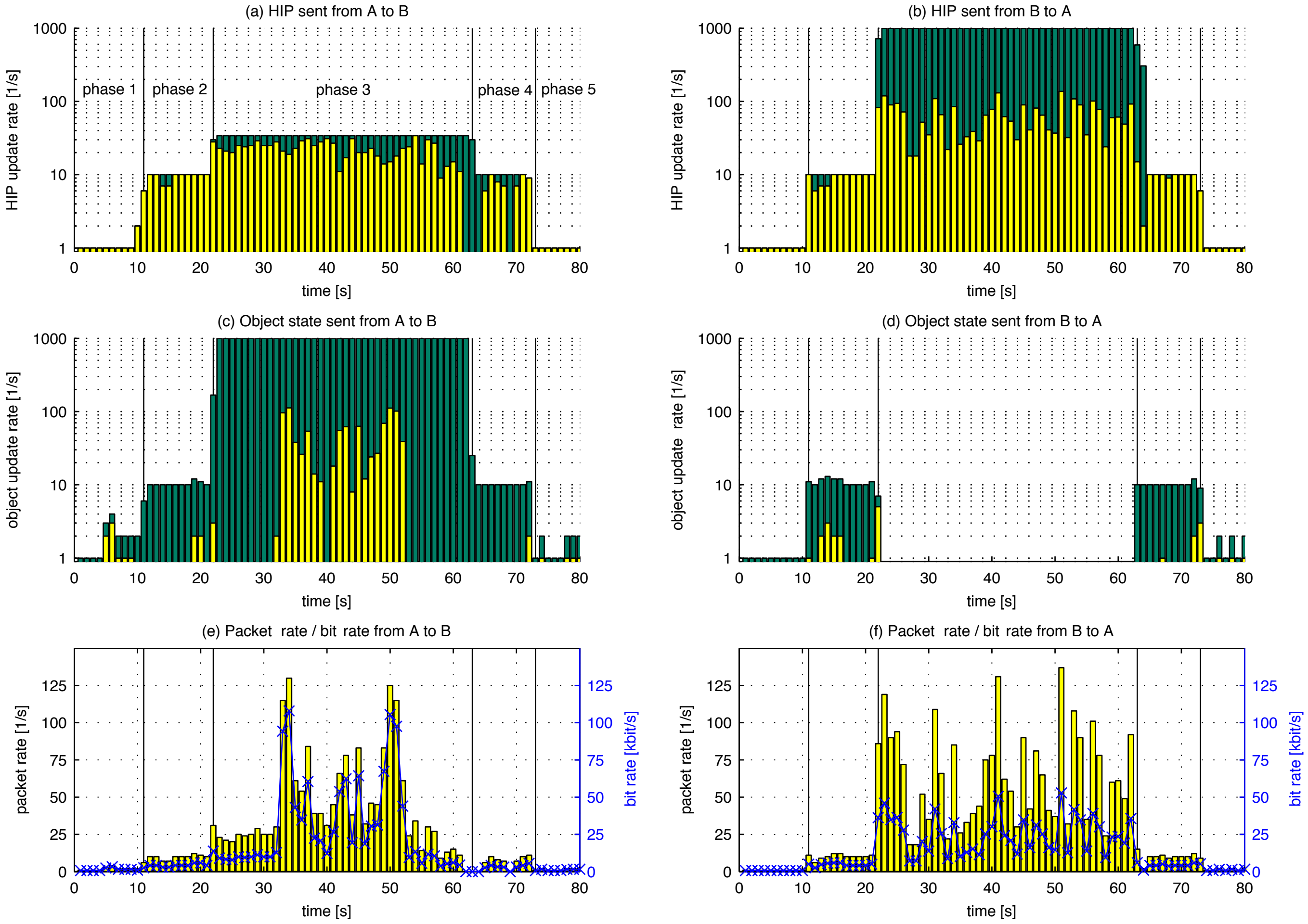

Fig. 2. Evaluation results. (a), (b): HIP (haptic interface point) updates from A to B, respectively B to A. Reduced update rates due to AoI are shown in green, update rates with additional dead reckoning in yellow. (c), (d): reduced object state updates due to AoI (green) and with additional dead reckoning (yellow). (e), (f): resulting packet rates (yellow) and bit rates (blue).

transmitted in both directions. Occasionally, the packet rate is slightly increased as user A and B trigger additional control messages about object ownership changes which also include the state of the affected objects (observable e.g. at time $t = 5s$ and $t = 6s$ in Fig. 2c/e). Please note that both users are transmitting object states until phase 3 as they form a single-user group and simulate their owned objects' state locally.

As soon as both users are closer to each other and their visual AoIs intersect, the inter-group update rate increases to 10 updates/s for both HIP and object state. Again, a slight increase in the number of object updates can be observed due to the afore-mentioned control messages. Although dead reckoning avoids the transmission of any updates if all objects are stationary (e.g. in Fig. 2c, phase 2), the user's movement with the haptic device leads to update packets triggered to user B, which include only A's device position (compare Fig. 2e, phase 2).

At time $t = 22s$ A and B form a group and high-rate information exchange is applied. Without dead reckoning, user B would now transmit his device position at the full rate of 1000 packets/s (Fig. 2b, phase 3) as A needs his exact device position for the physics simulation. The rendered object state would be sent back to B at the full rate, too (Fig. 2c, phase 3). A's device position is used at B just as a visual reference and, thus, needs to be piggybacked only occasionally (Fig. 2a, phase 3). The effect of dead reckoning is clearly evident during phase 3 where the number of state updates/s to B (Fig. 2c) and HIP updates/s to A (Fig. 2b) is reduced from 1000 to a maximum of around 100. Please note that user B transmits only HIP updates during phase 3 as he behaves like a client in the *Client-Server Architecture* (Fig. 2d, phase 3).

HIP and object updates, if currently pending, are encapsulated in single UDP packets, resulting in the packet rates shown in Fig. 2e and Fig. 2f. The bit rate depends, of course, on the payload, which might vary from packet to packet. As explained in Section IV, only states of objects lying in the *Haptic AoI* are included in the high-rate update packets. Dead reckoning additionally leads to the exemption of stationary or predictable object states. During intra-group communication in phase 3 user A demands a higher bit rate as B as he transmits object states (represented by 112 bytes[1] per object), while B includes only his device state (48 bytes[2]) in his updates. In phase 1 and 2, respectively 5 and 4, only about 1 kbit/s or 4

[1] $3 \cdot 8$ bytes each for object position and velocity vector, $4 \cdot 8$ bytes each for rotation and rotational velocity quaternion

[2] $3 \cdot 8$ bytes each for device position and velocity vector

kbit/s are needed for the inter-group communication. During collaborative interaction the bit rate increases to about 110 kbit/s for the master user A and 50 kbit/s for the user B.

VII. Conclusion

We propose a fully-distributed hybrid architecture, wherein peers dynamically form communication groups based on their proximity in the virtual world. Two different AoIs for haptic and visual interaction are introduced to adaptively adjust the update rates between peers.

A first prototype based on the proposed architecture is implemented and evaluated in a two-user testbed. From the measured resulting packet rates we conclude that the proposed approach is promising to reduce the overall communication load in a large SHVE. Without the exploitation of AoIs and dead reckoning, 416 kbit/s would be needed per user to transmit his HIP position (24 bytes) and 672 kbit/s to transmit one object position/rotation (56 bytes) at the full rate of $1kHz$, including 28 bytes for IP and UDP headers. Peak rates of only 100 kbit/s for the master user respectively 50 kbit/s for the other user are needed in our prototype during direct haptic interaction. Otherwise, less than 5 kbit/s are needed. More fine-grained update levels might be adopted in the future to further reduce packet and bit rates.

The evaluation serves as a first proof-of-concept of our work in progress as we used only two collaborating users. A complete network simulation, emulating different network impairments, together with a more realistic application needs to be deployed to investigate group maintenance in more detail and to further improve details of the proposed architecture. For example, provisions to avoid conflicting object ownership hand overs need to be considered. Automated dummy users need to be developed for large scale evaluation of the proposed architecture, as experiments with many real users are difficult to realize.

The architecture proposed in this paper uses application-layer information to adopt the communication between users, without taking the underlying communication network into account. User experience can be improved if there is a bidirectional information exchange between the application and the network about their current state. The network itself can try to improve QoS conditions according to interaction within the VE and, on the other hand, the distributed application can try to adapt to the characteristics the network currently provides.

Note that recent advances in cloud computing infrastructure might allow the usage of a centralized architecture for building a large SHVE and to avoid the server to quickly become a computational bottleneck [20]. The proposed decentralized approach, however, is able to consider communication delay between users in the process of selecting a master user, which is a key factor for satisfactory haptic interaction between users. Furthermore, even if the complete SHVE is simulated centrally, dynamic group building and exploitation of different AoIs is still necessary to reduce the overall communication load.

Acknowledgment

This work has been supported by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no. 258941.

References

[1] M. Srinivasan and C. Basdogan, “Haptics in virtual environments: Taxonomy, research status, and challenges,” *Computers & Graphics*, vol. 21, no. 4, pp. 393 – 404, April 1997.

[2] E. Saddik, “The potential of haptics technologies,” *IEEE Instrumentation Measurement Magazine*, vol. 10, no. 1, pp. 10–17, April 2007.

[3] M. Lin and M. Otaduy, *Haptic Rendering: Foundations, Algorithms, and Applications*. A K Peters/CRC Press, 2008.

[4] C. Basdogan, C.-H. Ho, M. Slater, and M. Srinivasan, “The role of haptic communication in shared virtual environments,” in *Proceedings of the Third PHANToM Users Group Workshop*, October 1998, pp. 85–89.

[5] K. Huang and D. Lee, “Consensus-Based Peer-to-Peer Control Architecture for Multiuser Haptic Interaction Over the Internet,” *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 417–431, April 2013.

[6] K. Hikichi, H. Morino, I. Fukuda, S. Matsumoto, Y. Yasuda, I. Arimoto, M. Iijima, and K. Sezaki, “Architecture of haptics communication system for adaptation to network environments,” in *IEEE International Conference on Multimedia and Expo, ICME*, August 2001, pp. 563–566.

[7] C. Schuwerk, R. Chaudhari, and E. Steinbach, “Perceptually robust traffic control in distributed haptic virtual environments,” in *Haptics: Perception, Devices, Mobility, and Communication*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, June 2012, vol. 7282, pp. 469–480.

[8] S. Lee and J. Kim, “Priority-based haptic event filtering for transmission and error control in networked virtual environments,” *Multimedia Systems*, vol. 15, no. 6, pp. 355–367, December 2009.

[9] J. Kim, H. Kim, B. K. Tay, M. Muniyandi, M. a. Srinivasan, J. Jordan, J. Mortensen, M. Oliveira, and M. Slater, “Transatlantic Touch: A Study of Haptic Collaboration over Long Distance,” *Presence: Teleoperators and Virtual Environments*, vol. 13, no. 3, pp. 328–337, June 2004.

[10] R. Iglesias, S. Casado, T. Gutiérrez, A. García-Alonso, W. Yu, and A. Marshall, “Simultaneous remote haptic collaboration for assembling tasks,” *Multimedia Systems*, vol. 13, no. 4, pp. 263–274, October 2007.

[11] J. Hespanha, M. McLaughlin, and G. S. Sukhatme, “Haptic collaboration over the internet,” in *Proceedings of the Fifth PHANToM Users Group Workshop*, October 2000, pp. 1–5.

[12] J. Marsh, M. Glencross, S. Pettifer, and R. Hubbold, “A Network Architecture Supporting Consistent Rich Behavior in Collaborative Interactive Applications,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 3, pp. 405–416, March 2006.

[13] B. Knutsson and H. Lu, “Peer-to-peer support for massively multiplayer games,” in *INFOCOM. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, March 2004, pp. 107 – 119.

[14] C. Carter, A. El Rhalibi, and M. Merabti, “A Survey of AoIM, Distribution and Communication in Peer-To-Peer Online Games,” in *21st International Conference on Computer Communications and Networks (ICCCN)*, July 2012, pp. 1–5.

[15] P. Hinterseer, S. Hirche, S. Chaudhuri, E. Steinbach, and M. Buss, “Perception-based data reduction and transmission of haptic data in telepresence and teleaction systems,” *IEEE Trans. on Signal Processing*, vol. 56, no. 2, pp. 588–597, February 2008.

[16] IEEE Computer Society, “IEEE Standard for Distributed Interactive Simulation–Application Protocols,” *IEEE Std 1278.1-2012 (Revision of IEEE Std 1278.1-1995)*, pp. 1 – 747, December 2012.

[17] H. Yoshinaga, T. Tsuchiya, and K. Koyanagi, “Coordinator election using the object model in p2p networks,” in *Agents and Peer-to-Peer Computing*, ser. Lecture Notes in Computer Science, vol. 3601. Springer Berlin Heidelberg, July 2005, pp. 161–172.

[18] F. Conti, F. Barbagli, D. Morris, and C. Sewell. (2005) Chai 3d: An open-source library for the rapid development of haptic scenes. [Online]. Available: http://www.chai3d.org

[19] E. Steinbach, S. Hirche, M. Ernst, F. Brandi, R. Chaudhari, J. Kammerl, and I. Vittorias, “Haptic communications,” *Proceedings of the IEEE*, vol. 100, no. 4, pp. 937–956, April 2012.

[20] M. Marzolla, S. Ferretti, and G. D’Angelo, “Dynamic resource provisioning for cloud-based gaming infrastructures,” *Computers in Entertainment*, vol. 10, no. 3, pp. 4:1–4:20, December 2012.