MODELING THE TELE-IMMERSIVE SYSTEMS USING STOCHASTIC ACTIVITY NETWORK

Rahul Malik, Wanmin Wu, Klara Nahrstedt

Department of Computer Science
University of Illinois at Urbana-Champaign
{rmalik4, wwu23, klara}@illinois.edu

ABSTRACT

The next-generation social communication medium, tele-immersion, is receiving increasing attention from both the research and industrial communities. It enables interaction between geographically distributed sites through realistic reconstruction of physical scenes in immersive spaces. While the previous research focused on real-world measurements of implemented tele-immersive systems, we take a model-based approach to study the system more generally and thoroughly. With a model validated by actual measurements, we evaluate the performance of the system in a larger scale, and understand the quantitative effect of the algorithmic enhancements that were proposed in the literature. We also predict futuristic performance characteristic and show that high interactivity can be foreseen with hardware advances in the near future.

1. INTRODUCTION

Tele-immersion is emerging as the future of video-mediated interaction. It creates 3D immersive experience for geographically distributed users by implementing a pipeline of 3D capturing, transmission, and rendering in real time. The past research on tele-immersive systems involved the design and implementation of its complicated components [1], with some exploiting the performance issues on the deployed systems [2, 3]. While the real-world measurements provide a realistic view on system performance, they are very much limited by the existing software and hardware setup. Studying the system in a larger scale, for example, would require purchasing more end devices (e.g., cameras, displays) and hence is not cost-effective. Also, it is often effort-consuming (if not impossible) to vary certain system parameters in an actual system and evaluate the overall performance characteristics. Predicting future performance with enhanced hardware is even more difficult.

To understand tele-immersion of larger scale (in terms of devices, network, and sites, etc.), we take a model-based approach with the general stochastic activity network (SAN) framework [4]. The main advantage of using SANs is that they allow the automatic generation of large markov chains, so we can conceptually describe the richness and diversity of the whole system without worrying about low-level details. Our goals are threefold: (a) evaluate the scalability of the current architecture and identify the critical components, (b) understand the quantitative impact of some design enhancements that were proposed in previous work, i.e., the effect of algorithmic change on performance, and (c) predict futuristic performance with advances in the hardware architecture.

Our study methodology is as follows. We use the actual timing measurements that were taken in an implemented 3DTI system to build the model of the tele-immersion system in the Möbius tool [5], a SAN-based performance evaluation software package. The simulated results are matched with actual measurements to validate the model. We then study the scalability of the current system, i.e., the effect of future growth on performance. Further, we implement three advanced algorithms that were previously proposed (i.e., rate adaptation [1], packet spreading [1], and ViewCast [6]) and evaluate their impact on the performance. Last but not least, with the expected advances in hardware (e.g., multi-core, GPU, high-capacity networks, etc.), we study the quantitative effect on critical performance metrics such as end-to-end latency and frame rate.

The rest of the paper is organized as follows. Section 2 presents an overview of 3DTI systems and SAN. Section 3 illustrates our modeling process. Then the experimental results are shown in Section 4. Finally, Section 5 makes concluding remarks and discusses future work.

2. OVERVIEW

2.1. Tele-immersion

The tele-immersive system is modeled as a distributed multi-tier application that consists of several *participating environments*. [1] (Figure 1). Each environment has three tiers: *capturing*, *transmission*, *rendering*.

The *capturing tier* is used for 3D scene acquisition and

THE RESEARCH IS FUNDED BY NATIONAL SCIENCE FOUNDATION (NFS) UNDER GRANTS IIS 08-40323, CNS 05-20182, CNS 07-20702, CNS 08-34480.

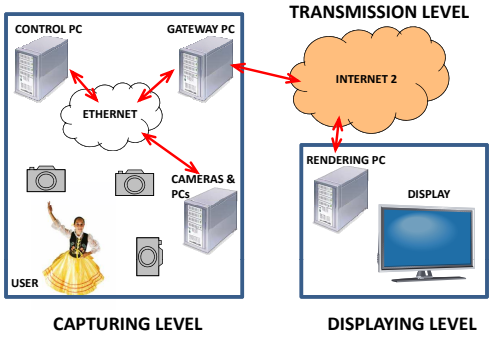


Fig. 1. Tele-immersion Application Model

reconstruction in real time. It consists of a set of 3D cameras organized in $180° - 360°$ within a room of the participating environment. Each 3D camera cluster consists of four 2D cameras, three of which are used for trinocular 3D reconstruction and one for color and texture acquisition. Each cluster is connected to a *camera host PC* that computes the depth map from the three 2D images (i.e., 3D reconstruction). The *control PC* (as shown in Figure 1) is responsible for synchronizing all the camera to take images from their viewpoint at the same instants of time.

The *transmission tier* in each tele-immersive environment has a *service gateway* which is responsible for data dissemination across the participating sites. Locally, the gateway collects a compressed 3D video stream from each camera PC. Over Internet2, it exchanges the streams with the gateways in other participating sites, and performs tasks including traffic shaping, multi-stream coordination/synchronization, and skew control.

Finally, the *rendering tier* consists of a set of displays, each hosted by a *renderer PC* that receives the 3D video streams of all participating sites from its local gateway, decompresses them, and renders them into an immersive video containing all participants.

2.2. Stochastic Activity Network

Stochastic activity networks (SANs) are a stochastic extension of Petri nets. SANs are more powerful and flexible than most other stochastic extensions of Petri nets including stochastic Petri nets (SPNs) and generalized stochastic Petri nets (GSPNs). The main advantage of using SANs is that they allow the automatic generation of large markov chains. As a result, they have been extensively used for performance, dependability and performability evaluation.

SANs consist of five primitives: place, timed activity, instantaneous activity, input gate, and output gate. Timed activities represent the activities of the modeled system whose durations impact the system's ability to perform. Instataneous activities, on the other hand, represent system activities that, relative to the performance variable in question, are completed in negligible amount of time. Gates are introduced to permit greater flexibility in defining enabling and completion rules. For timed activities, definition of general probability distribution function is possible.

3. MODELING

We model the tele-immersive system using stochastic activity networks (SANs) [4], a variant of stochastic Petri nets. The underlying Markov processes are generated and solved using Möbius [5]. Table 1 gives the values of different system parameters that were used for the experiments. We formed the time taken for different system parameters as Normal random variables with mean values as shown in the table.

Table 1. Values of different parameters used

Parameter	Value
Number of cameras	12
Mean reconstruction time	0.076 s
Mean network delay	0.1051 s
Mean network spread time	0.015 s
Number of streams to send	12
Mean rendering time	0.0036 s
Bandwidth	1 Gbps
Network error rate	0.0

Instead of going into details of the model, we provide a high level overview of the model. We modeled the system into three main subcomponents, reconstruction subcomponent, transmission subcomponent and rendering subcomponent, each subcomponent consisting of the five primitives mentioned in Section 2.2. The rep/join model is shown in Figure 2. The replication node (denoted as Rep in red color)

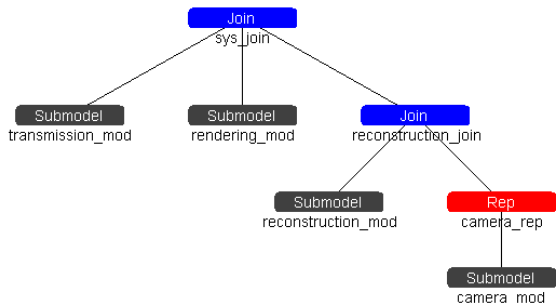
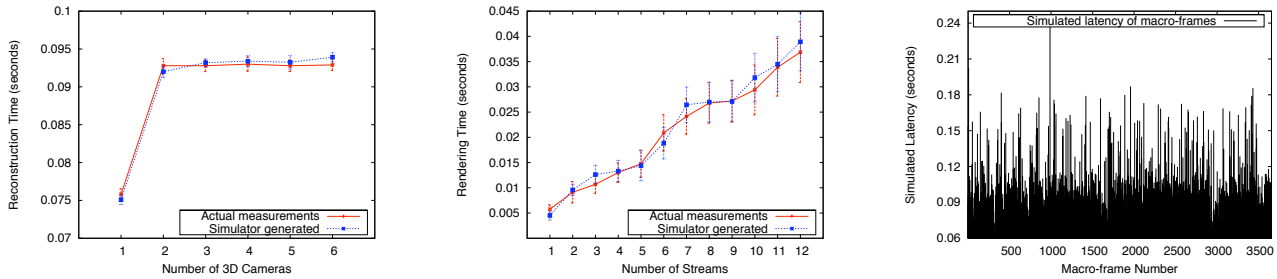


Fig. 2. Rep/Join model of system.



(a) Modeling reconstruction time

(b) Modeling rendering time

(c) Simulated latency of macroframes

Fig. 3. Validating the Model

makes a number a replicas of a subcomponent below it and join node (denoted in blue color) joins two or more subcomponents. The reconstruction subcomponent consisted of a number of cameras that were replicated and joined with reconstruction. Finally, all three of them were combined together using a join of the entire system.

4. EXPERIMENTAL RESULTS

4.1. Validating the Model

We first validate our model by matching the simulated values with the real-world measurements that were taken in an implemented and deployed system.

Figure 3(a) shows the actual reconstruction time (which was measured with six real cameras [3]) together with the simulated values. Figure 3(b) shows the rendering time (which was taken with twelve recorded 3D video streams [3]) with the modeled results. Figure 3(c) presents the modeled latency for the arrival of macro-frames (due to space limit the results with actual measurements were not shown and can be found in the previous paper [3]). As shown in the figures, the model reasonably captures the characteristics of the system and produces accurate estimation results.

4.2. Modeling Advanced Algorithms

In this section, we study the impact of three advanced algorithms in the tele-immersive system: rate adaption, packet spreading, and ViewCast [1, 6]. As a baseline, Figure 4(a) shows the simulated measurements of end-to-end delay and its breakdown without any enhancements. With twelve cameras, we observe that the scalability bottleneck of the system lies in the rendering component.

Yang *et al.* [1] introduced a rate adaptation algorithm for data dissemination in 3DTI systems. The idea was to drop some less important frames in times of network congestion. We study its effect on latency, with the results shown in Figure 4(b). We find that the end-to-end latency is reduced, with the rendering time having the most significant drop.

Since the tele-immersion traffic comes in burst, it is important to avoid network congestion which may result in packet loss and long delay. A packet spreading algorithm was proposed to serve this purpose [1]. We study the impact of this algorithm on the end-to-end delay, with the results shown in Figure 4(c). The reconstruction or rendering delay was not affected, and thus not shown. We note that spreading of packets does not affect the latency much while relieving congestion of network.

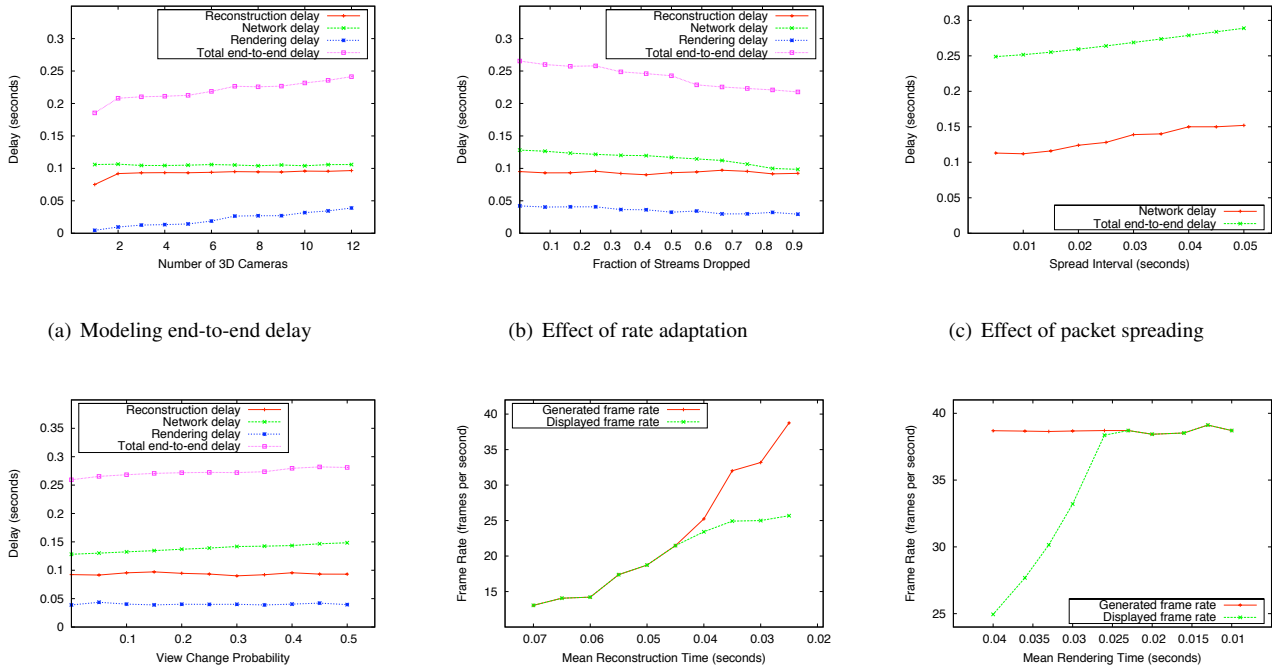
While the rate adaptation protocol works on the intra-stream level, another protocol, ViewCast, was presented [6] to adapt to network dynamics on the inter-stream level by dropping less important streams in times of sparse resources. Its impact on the delay of the system was exploited and illustrated in Figure 4(d). Even with the view change probability of 0.5, for example, the end-to-end delay only increases from 0.25 to 0.32 seconds, which suggests the robustness of the ViewCast protocol.

4.3. Modeling Futuristic Performances

The multi-core architecture is becoming prevalent nowadays, with which the performance of many components can potentially be significantly boosted.

As shown in Figure 3(a), the current reconstruction algorithm takes about 0.07 seconds without considering the synchronization overhead [3]. Figure 4(e) shows the simulated frame rate (at the rendering side) by varying the reconstruction time and fixing the other parameters. We observe that when the reconstruction time is more than 0.045 seconds, the rendering frame rate is not affected because in the producer-consumer pipeline the consumer (i.e., renderer) is still the bottleneck. However, when the reconstruction time further decreases below 0.045, the generated frame rate greatly increases.

On the other end, as the GPU architecture becomes more powerful, the rendering time is expected to decrease as well. Figure 4(f) presents the simulated frame rate by varying the



(a) Modeling end-to-end delay

(b) Effect of rate adaptation

(c) Effect of packet spreading

(d) Effect of view change

(e) Effect of reconstruction time

(f) Effect of rendering time

Fig. 4. Modeling Advanced Algorithms and Futuristic Performances

rendering time and fixing the other parameters. We note that with the rendering time dropping from 0.04 to 0.027 seconds, the total frame rate improves significantly. When the rendering time is below 0.027 seconds, the other components in the system become the bottleneck as it grows with the number of cameras.

5. CONCLUSION AND FUTURE WORK

We show how the stochastic activity network can be used to model a practical 3D tele-immersive system. Through modeling, we gain a better understanding of the different system components and their impact on the overall performance. We find that the advanced algorithms proposed to the system would incur acceptable overhead. We exploit the effect of performance improvement in different parts of the system, and show that the high interactivity in 3D tele-immersive remote collaboration is possible in the near future. For the future work, we plan to look at ways of modeling N participants in a distributed tele-immersive system and study the tradeoffs there. We will also model the various network topologies.

6. REFERENCES

[1] Zhenyu Yang, Yi Cui, Bin Yu, Jin Liang, Klara Nahrstedt, Sang-Hack Jung, and Ruzena Bajscy, "TEEVE: The next generation architecture for tele-immersive environments," in *Proc. of ISM*, 2005.

[2] H. Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, M. Goss, W. Culbertson, and T. Malzbender, "Understanding performance in coliseum: an immersive videoconferencing system," *ACM TOMCCAP*, 2005.

[3] Dongyun Jin Klara Nahrstedt Wanmin Wu, Zhenyu Yang, "Implementing a distributed 3d tele-immersive system," in *Proc. of ISM'08*, 2008.

[4] J. F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: Structure, behavior, and, application," in *Proceedings of the International Workshop on Timed Petri nets*, 1985, pp. 106–115.

[5] D. Daly, D. D. Deavours, J. M. Doyle, P. G. Webster, and W. H. Sanders, "Mobius: An extensible tool for performance and dependability modeling," in *Proc. of TOOLS*, 2000.

[6] Klara Nahrstedt Zhenyu Yang, Wanmin Wu, "Viewcast: View dissemination and management for multi-party 3d tele-immersive environments," *Proc. of ACM Multimedia*, 2007.