

Reliability Modeling and Analysis of Modern Distributed Interactive Multimedia Applications: A Case Study of a Distributed Opera Performance

Narasimha Veeraragavan, Roman Vitenberg, Hein Meling

► **To cite this version:**

Narasimha Veeraragavan, Roman Vitenberg, Hein Meling. Reliability Modeling and Analysis of Modern Distributed Interactive Multimedia Applications: A Case Study of a Distributed Opera Performance. Karl Michael Göschka; Seif Haridi. 12th International Conference on Distributed Applications and Interoperable Systems (DAIS), Jun 2012, Stockholm, Sweden. Springer, Lecture Notes in Computer Science, LNCS-7272, pp.185-193, 2012, Distributed Applications and Interoperable Systems. <10.1007/978-3-642-30823-9_16>. <hal-01527648>

HAL Id: hal-01527648

<https://hal.inria.fr/hal-01527648>

Submitted on 24 May 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Reliability Modeling and Analysis of Modern Distributed Interactive Multimedia Applications: A Case Study of a Distributed Opera Performance

Narasimha Raghavan¹, Roman Vitenberg¹, and Hein Meling²

¹ University of Oslo, P.O. Box 1080, Blindern, Oslo, Norway
{raghavan, romanvi}@ifi.uio.no

² University of Stavanger, N-4036 Stavanger, Norway
hein.meling@uis.no

Abstract. With the increase in processing power of computers, proliferation of specialized software and hardware, and the availability of greater communications bandwidth, it is possible to deploy more demanding modern Distributed Interactive Multimedia Applications such as World Opera. The World Opera application realizes the vision of a distributed real-time opera performance in which artists who are distributed around the globe are interacting with each other as though they were co-located. The application is driven by the artistic consortium that encompasses several major opera houses worldwide.

In this paper, we build a high-level reliability model to predict the system reliability of the deployed architecture for supporting World Opera. Furthermore, we conduct a sensitivity analysis to understand the reliability requirements that individual components need to meet in order to provide desired system reliability.

Keywords: Reliability Analysis, Reliability Block Diagrams, Stochastic Activity Networks, World Opera, Distributed Interactive Multimedia Applications

1 Introduction

Recently we have been witnessing a proliferation of Distributed Interactive Multimedia Applications (DIMAs), such as Massive Online Multiplayer Games (MOMG) [5] and video conferencing systems. These applications are characterized by interactive exchange of voice, video, control, and other data (such as sensor information) between a number of participants through an electronic medium in real-time. Because of its interactive and real-time nature, the paradigm imposes stringent requirements on latency and synchronization.

Classical DIMAs have traditionally been limited in terms of the complexity of interactions, types and numbers of streams, and the number of participants at each location, as we explain in Section 2. With the increase in processing power of the computers, advent of specialized software and hardware, and availability of greater communications bandwidth, it is possible to deploy more demanding modern DIMAs such as World Opera, an application envisioned by the World Opera artistic consortium [1]. In this application, artists, musicians and singers who are physically located at different opera

houses dispersed across the globe, participate in a single united performance and interact as if they were co-located.

One of the observations made during early experimentation with World Opera performances is that it is notoriously difficult to maintain a smooth technical operation for the entire duration of a performance, even with rigorous preparations. In this paper, we strive to give a rigorous explanation to this phenomenon by modeling the reliability of a World Opera performance. We further conduct sensitivity analysis in order to understand the reliability requirements of the components of the World Opera.

2 A World Opera Application

The World Opera (WO) consortium and its partners are engaged in conducting distributed, real-time, live opera performances across several world renowned opera houses. Each opera house represents a real-world stage with its own musicians, singers, dancers, and actors. Interaction between the artists is orchestrated by a single conductor present at a single selected stage. Participating artists from different real-world stages map to *virtual-world stages*, which are projected as video on display devices, and shown to the audience at the local opera house as well as audiences at geographically distributed (remote) opera houses. Additionally, virtual-world stages can display animated cartoon characters mimicking the behavior of the artists at remote stages. The virtual-world and real-world stages together form a mixed-reality stage. A collection of distributed mixed-reality stages together constitute a WO application.

At a high level, a WO performance is organized into several sessions, each representing a time interval during which a specified set of components is in use. A session starts when directors³ at all stages agree on the set of components to be used for the duration of the session.

The WO application exhibit complex interactions between the real-world and virtual-world. These interactions are established on a per-session basis. They require synchronization across several dimensions: spatial, temporal, and precedence.

Spatial synchronization means that real-world locations must be mapped to appropriate locations in the virtual-world. Consider two cameras at a remote stage each capturing video of the upper and lower part of a human body, respectively. When these two video streams are projected to the virtual stages, their spatial location must be correlated so as to produce a single image of the human body.

Temporal synchronization means that artists at a given real-world stage should act in unison and remain synchronized with artists at the remote stages. For example, a singer at a local stage should be synchronized with musicians located at remote stages. Similarly, actors at the local stage should have a correct lip synchronization with respect to the dialogues of actors at remote stages.

Precedence synchronization means that some streams are given a higher priority for processing than the other streams. For example, a stream from the conductor is given a higher priority than a stream from a violinist.

³ A director is a technician with artistic knowledge responsible for controlling the technical components based on the artistic requirements. One director is present at each stage during the performance.

These complex interactions depend upon a) the software components such as middleware services and b) hardware components such as cameras, transmitters, audio and video receiving devices, musical instruments used by the artists, and network links.

Component failures taking place during these complex interactions, could severely affect the synchronization between the real-world and virtual-world. This in turn would affect the entire performance.

3 World Opera Architecture

We provide a brief selective description about details of the deployed World Opera architecture that are relevant to the reliability model in Section 4 and evaluation in Section 5. There are three participating physical stages in the current preliminary deployment scenario. The stages are symmetric: Each stage includes the same number of components of the same type. Specific component types in the deployed architecture as well as the number of components of each type at a stage are shown in Table 1.

Table 1: Components used in each stage in WO architecture

Component Types	# components
Camera	3
Camera workstations	3
Display workstations	3
Audio workstation1	1
Audio workstation2	1
Projector	3
Speaker	3
Wired microphone	4
Wireless microphone	4
Mixer	1
Gateway	1

4 Reliability Model for World Opera

It is imperative for WO application to operate smoothly for the entire duration of a performance. The probability for continuous smooth operation is represented by the reliability metric [16], which is the main focus of our work. The reliability of WO is defined as the probability that the functionality provided by all the components of WO is operational for the duration of the performance, or the mission-time. While in principle it is possible to define an acceptable degraded quality of experience based on reduced functionality (the audience can hear the singer without seeing her), we leave it for future work to consider such metrics. Furthermore, the script for the opera performance is written considering all the participating stages. Hence, the failure of one stage results in the failure of the script, which in turn leads to the failure of the opera performance.

In this section we provide reliability models for the WO application that are used for reliability analysis in Section 5. We employ the Reliability Block Diagram (RBD) [16] modeling method that captures the operational relations between the components of

WO. This method also allows us to predict the reliability requirements of individual components.

For a system consisting of a number of hardware components, RBD represents functional relationships between the components using a series-parallel structure. When connected in a series structure, all components need to be working for a system to be operational. On the other hand, a parallel structure requires that at least one of the components must be working for the system to be operational. A variety of applications have used RBDs for modeling system reliability, e.g. in [10], [13], [4], and [8].

It should be noted that the models only focus on the new set of components installed in opera houses exclusively for WO performances. They do not consider permanent components and functionality inherent to opera houses such as power supplies or physical stage movement because it is assumed that experienced stage engineers of reputed opera houses are able to deliver such functionality with a very high reliability. We now present our modelling assumptions.

Independent failures: In line with an assumption commonly made in the dependability modelling literature [9], we consider component failures to be mutually independent.

Crash failures: Only crash failures of the components are considered because other failure types are less common. In particular, failures in the network connecting the stages are deemed to have a negligible probability because the network consists of dedicated lines leased from a commercial provider. Similarly, byzantine failures are significantly less typical in the tightly-controlled production environment of WO.

Fail-stop failures: All the components are non-repairable during the performance. During WO performance, a repair is considered too time consuming to be practical, and it would severely impact on the performance.

4.1 Reliability Block Diagram Models for World Opera

In this section, we present three-level hierarchical RBD models reflecting the deployed architecture of WO. In hierarchical RBD models, lower-level RBD models are used as building blocks in larger higher-level RBD models so that the top-level RBD model corresponds to the overall architecture.

Each block in the top-level RBD (see Figure 1a)) represents a participating stage in WO so that the number of blocks is 3, following the actual number of stages. The blocks are connected in a series structure as the failure of any participating stage will lead to the failure of WO. The model reflects the current settings for WO, in which the participating stages have identical architecture with respect to the number of components and component types. Each block at the middle level (see Figure 1b)) of the hierarchy represents a component type used within a stage. Similarly to the top-level RBD, the blocks are connected in a series structure because the failure of any single component type will lead to the failure of a stage which will in turn lead to the failure of the WO performance.

At the bottom of the hierarchy, each block represents one component of a particular type. Since a failure of any single component will lead to a failure of a component type (we do not consider degraded quality of experience in this paper), the blocks are

connected in a series structure. To illustrate these points, consider a scenario where 3 cameras capture a stage from three different viewpoints and all these viewpoints are required for the performance. In this case, each block represents a camera with a specified failure rate as shown in Figure 1c). Since we need all the cameras, we connect the blocks representing the cameras in a series structure. Likewise, Figures 1d), 1e), 1f), 1g), and 1h) show the RBD models for the following component types: camera workstations, display workstations, projectors, speakers, wired microphones, and wireless microphones, respectively.

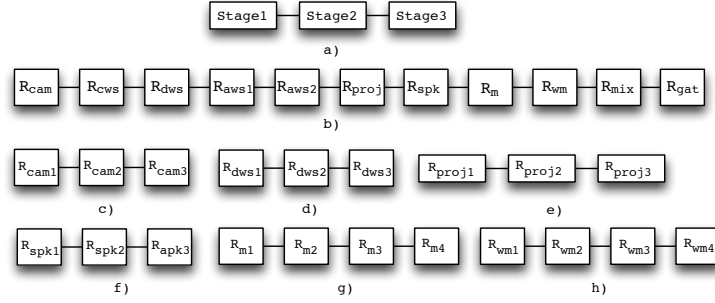


Fig. 1: Hierarchical RBD models for World Opera with three stages

5 Evaluation Results

We use SHARPE [14] to construct and evaluate the RBD models shown in Figure 1. We fix the mission-time to 2 hours, representative of most Opera performances. The RBD models are evaluated with the failure rates shown in Table 2. The rates are based on conservative estimates and partially known product failure rates information made available by the manufacturers. While our model is independent of the failure rate distribution, we assumed for our experiments that failure rates are exponentially distributed following conventional practice.

Table 2: Parameter notation and values

Component Types	Failure rate	Reliability
Camera	λ_{cam} 0.002	R_{cam}
Camera workstations	λ_{cws} 0.00001	R_{cws}
Display workstations	λ_{dws} 0.00001	R_{dws}
Audio workstation1	λ_{aws1} 0.00001	R_{aws1}
Audio workstation2	λ_{aws2} 0.00001	R_{aws2}
Projector	λ_{proj} 0.006	R_{proj}
Speaker	λ_{spk} 0.004	R_{spk}
Wired microphone	λ_m 0.002	R_m
Wireless microphone	λ_{wm} 0.002	R_{wm}
Mixer	λ_{mix} 0.0001	R_{mix}
Gateway	λ_{gat} 0.00001	R_{gat}

We have two objectives for our experiments. First, we evaluate the reliability of a WO performance (i.e., the probability that all the participating stages remain operational

for the duration of the performance) under the settings shown in Table 2. Table 3 shows our results wrt reliability and unreliability values for different time points in the course of a WO performance. The mean time to performance failure is 6.38691959 hours.

Table 3: Evaluation Results of the RBD model

Time (hrs)	Three stages	
	Reliability	Unreliability
0.5	0.924700849	0.0752991508
1.0	0.855071661	0.144928339
1.5	0.790685491	0.209314509
2.0	0.731147545	0.268852455

Second, we perform sensitivity analysis in order to derive minimum reliability of the individual components (i.e., the probability that the component is operational for the entire duration of the mission time) that would result in the desirable level of reliability for WO. We consider the target reliability of 0.99 for a three-stage WO, representing that one out of 100 performances fail. This level of reliability would allow a business model in which the audience of the failed performance get free tickets for a new performance.

We used a dependability package [6] for Mathematica [12] to perform the sensitivity analysis on the RBD model shown in Figure 1. As a part of the sensitivity analysis, we vary the combination of the reliability values for different components. Table 4 shows one possible solution out of many solutions of the sensitivity analysis. The table presents the minimum reliability of various component types, that is the probability that all individual components of a specific component type remain operational during the mission time. It also provides the minimum reliability of a single stage, that is the probability that all of the components at a stage remain operational during the mission time.

As the table shows, the “most relaxed” reliability requirement of a component is 0.9997 and most components must support a minimum reliability of 0.99999995 in order to achieve the target reliability for WO. This observation holds for all other possible solutions of the sensitivity analysis. In other words, the component failure rate is required to be very low, e.g., 10^{-6} per hour, during the two hour mission time.

Table 4: Sensitivity analysis results derived from the RBD model

R_{stage1} R_{stage2} R_{stage3}														
0.996656														
R_{cam}	R_{mix}	R_{cws}	R_{dws}	R_{aws1}	R_{aws2}	R_m	R_{wm}	R_{gat}	R_{proj}			R_{spk}		
$1 - 10^{-4}$		$1 - 10^{-6}$				$1 - 10^{-3}$		$1 - 10^{-5}$	$1 - 2.16 \times 10^{-4}$			$1 - 9 \times 10^{-4}$		
R_{cam1}	R_{cam2}	R_{cam3}	R_{cws1}	R_{cws2}	R_{cws3}	R_{dws1}	R_{dws2}	R_{dws3}	R_{proj1}	R_{proj2}	R_{proj3}	R_{spk1}	R_{spk2}	R_{spk3}
0.999967			0.9999995						0.999928			0.9997		
R_{m1} R_{m2} R_{m3} R_{m4} R_{wm1} R_{wm2} R_{wm3} R_{wm4}														
0.99975														

6 Related Work

While both client-server and P2P teleconferencing applications are wide-spread nowadays, the complexity of most architectures supporting these application stems from the protocols rather than the variety, scale, and complexity of utilized components. Accordingly, the research into fault-tolerance aspects of teleconferencing applications has traditionally been focusing on overcoming network packet losses by decoders [2], redundant routing [15], and network coding techniques [11]. P2P systems additionally devote a lot of attention to handling node churn [3]. Since these applications are primarily oriented towards desktop users, the architectures designed for these applications can only support limited hardware (microphone and web camera) and software components (client application) per location, limited number of participants at each location, and limited number of simultaneous multi-point communications. This reduces the need for reliability analysis for such architectures. In [7], the authors provide a service dependency model for a teleconferencing scenario, mainly focusing on analyzing faults at the level of network connectivity. Failure of the client-side hardware devices (such as webcamera and microphone) is not considered.

In contrast, recently emerging tele-immersive applications are designed to provide more sophisticated features such as extensive configurability, high-resolution audio and video. Such systems typically include a multitude of specialized hardware and software components. To meet the bandwidth requirements of the high resolution audio and video, these systems are designed over the advanced networking infrastructure such as Internet2. Accordingly, network packet losses have a smaller impact on the dependability of tele-immersive applications compared to malfunctioning of individual components.

[17], [18], [19] have shown that these applications benefit the artists community to a great extent. [17] developed a distributed 3D-tele-immersive system, where the motion of a human body from different geographical locations are captured and shared in a virtual space among the participants. The architecture is mainly designed to deal with video streams and not audio or sensor streams. [19] described a distributed immersive performance application and presented the requirements and challenges it posed for managing the data generated in this environment. [18] presented the experimental results of performing collaborative dancing in the context of a 3D-tele-immersive system. The results conclude that the collaborative dance performance over network is feasible.

To the best of our knowledge, none of the existing tele-immersive applications considered failure scenarios in the design of their architecture. For this reason, the applications developed on top of these architectures do not provide guaranteed quality of experience to the end users in presence of failures. The models that we propose in Section 4 are directly applicable to the reliability analysis of these applications.

7 Conclusion

We have provided a reliability model for the WO application and a corresponding sensitivity analysis. We show that the currently deployed architecture requires components with very high reliability in order to satisfy the requirement of only one failed perfor-

mance out of 100. In the future we would like to extend and generalize the analysis to address the large class of modern tele-immersive applications.

References

1. The World Opera. <http://theworldopera.org>, March 2012.
2. J. G. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. In *VCIP*, pages 392–409, 2001.
3. M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in cooperative environments. In *ACM Symposium on Operating Systems Principles (SOSP)*, New York, NY, USA, 2003. ACM.
4. D. Chen, S. Dharmaraja, D. Chen, L. Li, K. Trivedi, R. Some, and A. Nikora. Reliability and availability analysis for the jpl remote exploration and experimentation system. In *Proceedings of International Conference on Dependable Systems and Networks*, pages 337 – 342, 2002.
5. T. Hampel, T. Bopp, and R. Hinn. A peer-to-peer architecture for massive multiplayer online games. In *Proc. NetGames '06*. ACM, 2006.
6. B. E. Helvik. Dependability analysis with reliability block diagrams. <http://library.wolfram.com/infocenter/MathSource/7371/>, 2009.
7. S. Katker and K. Geihs. A generic model for fault isolation in integrated management systems. *J. Netw. Syst. Manage.*, 5:109–130, June 1997.
8. G. Kervarrec and D. Marquet. A realistic reliability and availability prediction methodology for power supply systems. In *Proceedings of 24th Annual International Conference in Telecommunications Energy*, pages 279 – 286, 2002.
9. C. D. Lai, M. Xie, K.-L. Poh, Y.-S. Dai, and P. Yang. A model for availability analysis of distributed software/hardware systems. *Information and Software Technology*, 44(6):343–350, 2002.
10. M. Nanda and S. Rao. A formal method approach to analyze the design of aircraft flight control systems. In *3rd Annual IEEE Systems Conference*, pages 64 –69, March 2009.
11. A. T. Nguyen, B. Li, and F. Eliassen. Chameleon: Adaptive peer-to-peer streaming with network coding. In *INFOCOM*, pages 2088–2096, 2010.
12. W. Research. Mathematica edition: Version 8.0, 2010.
13. B. Roczen, R. Arno, and P. Hale. Reliability block diagram methodology applied to gold book standard network. In *IEEE Industrial and Commercial Power Systems Technical Conference*, pages 116 – 126, May 2004.
14. R. A. Sahner, K. S. Trivedi, and A. Puliafito. *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Kluwer Academic Publishers, Norwell, MA, USA, 1996.
15. Y. Shan, I. V. Bajic, S. Kalyanaraman, and J. W. Woods. Overlay multi-hop FEC scheme for video streaming over peer to peer networks. *IEEE ICIP*, 2004.
16. K. S. Trivedi. *Probability and Statistics with Reliability, Queueing, and Computer Science Applications, 2nd Edition*. Wiley-Interscience, Oct. 2001.
17. W. Wu, Z. Yang, D. Jin, and K. Nahrstedt. Implementing a distributed tele-immersive system. In *Proc. of the 10th IEEE ISM'08*, pages 477–484.
18. Z. Yang, B. Yu, W. Wu, R. Diankov, and R. Bajscy. Collaborative dancing in tele-immersive environment. In *ACM Multimedia*, pages 723–726, Santa Barbara, CA, USA, Oct. 2006.
19. R. Zimmermann, E. Chew, S. A. Ay, and M. Pawar. Distributed musical performances: Architecture and stream management. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4:14:1–14:23, May 2008.