LATENCY-MINIMIZED DELIVERY FOR MULTI-RESOLUTION MESH GEOMETRY

Ghassan AlRegib and Dihong Tian

Center for Signal and Image Processing
Georgia Institute of Technology
{gregib,dhtian}@ece.gatech.edu

ABSTRACT

Three-dimensional (3D) meshes are used intensively in distributed graphics applications where model data is transmitted on demand to users' terminals and rendered for interactive manipulation. This paper presents a transmission system for such applications with an objective of minimizing the latency between the user input and the response. In particular, we represent 3D models by multiple resolutions to allow fast and scalable rendering, and provide them with unequal error protection and/or retransmission when sending the data over a lossy link. The transmission policies are determined adaptive to environment variables and in linear computation time. Simulation results show that the proposed transmission system achieves 20-30% reduction on delivering latency compared to the state-of-the-art approach in the literature.

1. INTRODUCTION

Internet-based multimedia applications are expanding from streaming video/audio to distributed 3D graphics, driven by growing demands of various applications such as electronic commerce, collaborative CAD, medical and scientific visualization, and virtual environments. Interaction is one of the key aspects of a distributed graphics application. Response time, which is the latency between the user input and the response (e.g., scenes displayed on the user's terminal) from the system, is one of the major considerations in designing high-performance distributed graphics systems. In contrast to specific 3D systems that assume all models are locally available and are essentially designed as stand-alone systems, distributed graphics applications often require on-demand exchange of 3D data in a networked environment, and impose requirements of real-time response and smooth performance over time on the interaction. In addition, 3D data differs from general media content for it requires rendering capability on the user terminal. Sending the same dimension of data to user terminals with disparate rendering capabilities may result in significant difference in response time between different clients. Transmission and rendering scalability is therefore desirable for 3D data. All the aforementioned requirements, not only have promoted the use of high-performance computing systems and distributed platforms, but also call for careful considerations on ways of reducing transmission latency, providing scalability, and maintaining high-resolution visualization when network delays and random data losses are involved.

Multi-resolution compression of 3D meshes (e.g., [1] is a partial solution to provide scalability for 3D data. Using multi-resolution encoders, the server can select the appropriate resolution for a particular client according to its quality requirement, or initially sends a coarse representation of the 3D model to the client for quick reconstruction and rendering, and then transmits refinement layers that allow the client to gradually increase model fidelity toward higher resolutions. Although, such methods are successful in exploring the space and time efficiency of 3D data, higher efficiency can be accomplished by addressing the effects of network behaviors. In particular, to display 3D scenes on the user's terminal with satisfactory quality and in real time, the impact of packet losses and transmission delays on the decoding process need to be explored. Typically, reliable or error-resilient transmission can be achieved by preprocessing techniques such as data partitioning [2], post-processing techniques such as error-concealment, and network-oriented techniques such as forward error correction [3] and retransmission techniques [4, 5]. All these techniques address efficient transmission of 3D data *separately*. Yet an appropriate combination of the techniques is desired to achieve better performance. Interaction and tradeoffs among the selected techniques need to be investigated, taking into account the properties of the 3D data and the network characteristics.

To provide a solution, a hybrid mechanism of unequal error protection and retransmission is proposed in this paper. Hierarchal data batches of the multi-resolution mesh are protected preferentially according to their distortion-rate performance, network parameters, and link statistics estimated by the transport layer. To minimize response time in interaction, the proposed mechanism is designed to have linear computational complexity. In addition, by integrating TCP-friendly congestion control [6] into the system, the proposed mechanism achieves smooth performance over time as well as bandwidth fairness for co-existing applications in the network. Simulation results show the efficacy of the proposed mechanism. For instance, compared with a recently presented 3D application protocol (3TP) [5], the proposed system achieves 20-30% reduction in transmission latency while delivering the same level of rendering quality.

The rest of the paper is organized as follows. Section 2 describes major aspects of the proposed mesh transmission system, and Section 3 presents a detailed study of the hybrid unequal-error-protection and selective-retransmission mechanism. Test results in simulated network environments are given in Section 4. Finally, Section 5 concludes the paper and summarizes future work.

2. SYSTEM OVERVIEW

In this section, we provide an overview of the 3D mesh transmission system that is under consideration. The proposed system has three major components: a 3D mesh codec, a hybrid unequal-error-protection and selective-retransmission mechanism at the application level (UEP/SR), and a transport protocol integrated with TCP-friendly rate control (TFRC). We briefly describe these components below.

The codec component: A compressed mesh stream is composed of a base mesh, M_0, and L enhancement batches, $\{B_i\}_{i=1,\ldots,L}$, each of which encodes a set of *vertex-split* [7] operations which transfer the triangulated mesh surface to a higher resolution. Sequentially, batch B_i refines the resolution of mesh M_{i-1} to higher resolution M_i until the full resolution is reached. Each resolution M_i, differs from the full resolution mesh by certain error (distortion), D_i, with a certain bit rate, R_i. A measure of such distortion that properly reflects the perceptual quality of different resolutions is important. In this paper, we measure such distortion by introducing a metric with a similar formulation to the peak signal-to-noise ratio (PSNR) which is commonly used in imaging. In particular, for a mesh with multiple resolutions $\{M_i\}_{i=0,\ldots,L}$ where M_0 is the base mesh and M_L is the full resolution, we define the quality of mesh M_i as

$$PSNR_m \triangleq -20\log_{10}\frac{\mathcal{E}_{rms}(M_i, M_L)}{\mathcal{E}_{max}(M_0, M_L)}\ (\text{dB}), \tag{1}$$

where $\mathcal{E}_{max}()$ and $\mathcal{E}_{rms}()$ are the measured maximum and root-mean-square surface distances between the corresponding pairs of meshes, respectively [8].

The transmission component: The L enhancement batches, $\{B_i\}_{i=1,\ldots,L}$, have different rate-distortion performance and are treated intelligently to achieve the best quality. In this paper, best quality is interpreted as minimized transmission latency τ under a distortion constraint D_{max}. Given network statistics reported by the transport layer, the sending application protects the batches with unequal-rate forward error correction (FEC) codes and/or retransmission according to their respective costs, and determines the optimal tradeoff that minimizes the transmission delay τ under the constraint of D_{max} or vice versa. The Reed-Solomon (RS) code is employed for FEC. We assume an (n, k) RS code with a block size of n packets (i.e., $n \times$ *packet-size* symbols) including k ($k < n$) information packets (code rate $\frac{k}{n}$). Considering a lossy channel, an RS code with $(n-k)$ parity-check packets will be able to recover the same number of packet losses. Hence the error probability of a batch using RS code (n, k) is

$$P(n, k) = Pr\{> (n-k) \text{ losses out of } n\}. \tag{2}$$

For an independent random error process on the link, it is obvious that we have

$$\begin{aligned} P(n,k) &= 1 - Pr\{\leq (n-k) \text{ losses out of } n\} \\ &= 1 - \sum_{i=0}^{n-k} \binom{n}{i} p^i (1-p)^{n-i} \end{aligned} \tag{3}$$

where p is the packet-loss rate. Similarly, the calculation of (2) can be performed for more sophisticated error models such as a Markov channel model. We consider the independent error process in this paper for simplicity.

The transport layer: UDP streams suffer from the lack of congestion control mechanism that prevents them from being reasonably fair[1] when competing for bandwidth with TCP-based traffic, as TCP throttles its transmission rate against the network congestion. A TCP-friendly system should regulate its data sending rate according to the network condition, typically expressed in terms of the packet size s, the round-trip time r, and the packet-loss rate p. Ideally, the TCP throughput equation is suitable in describing the steady-state sending rate of a TCP-friendly flow [9]:

$$T = \frac{s}{r\sqrt{\frac{2p}{3}} + 4r(3\sqrt{\frac{3p}{8}})p(1+32p^2)}, \tag{4}$$

where a recommended choice of the retransmission timeout, $t_{RTO} = 4r$, has already been integrated. To provide bandwidth fairness for parallel flows in the network as well as network stability, a TCP-friendly rate control protocol (TFRC) [6] based on (4) is integrated in our proposed transmission system.

[1]A flow is "reasonably fair" if its sending rate is generally within a factor of two of the sending rate of a TCP flow under the same condition [6].

3. THE PROPOSED MECHANISM

The objective of this work is to minimize the response time for on-demand graphics transmission toward high-quality display. After encoding the 3D object into multi-resolutions to enable fast and scalable rendering, we aim to minimize transmission latency for a guaranteed level of quality, or equivalently, a constrained receiving distortion, D_{max}. Because for a multi-resolution mesh stream, further decoding a higher LOD requires successful decoding of the preceding LODs, satisfying D_{max} is equivalent to selecting the least number of batches that need to be transmitted reliably. Denoted by χ, this least number of batches is expressed as

$$\chi = \min\{x | D_d(x) \leq D_{max}\}, \tag{5}$$

where $D_d(x)$ denotes the decoding distortion of the first x batches.

Once the number of selected batches is determined, the challenge is to find the optimal tradeoff between forward error protection and retransmission such that the user-requested distortion level can be satisfied with minimum transmission latency. To do so, an optimal distribution of parity-check packets for the selected batches is computed, taking into account their unequal rate-distortion performance and potential retransmission costs. The selected data is then protected with the corresponding parity-check packets, and is transmitted (with possible retransmissions) until the batches are correctly decoded. A simple illustration of this transmission mechanism is presented in Figure 1, where χ batches are selected to be reliably transmitted with a calculated distribution of the parity-check packets (noted with 'RS'). Packets marked with '$*$' indicate the losses occurred during transmission. As depicted in Figure 1, Batch 2 is not decodable due to packet losses and is retransmitted in order to satisfy the distortion constraint. After the χ batches are correctly received, the remaining data is transmitted (with or without error resilience, provided that the requested distortion level has been met).

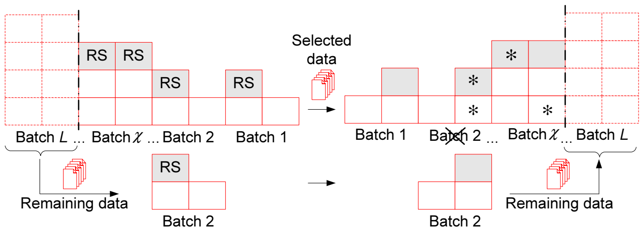

Fig. 1. A simple demonstration on the proposed hybrid UEP/SR method, where the packets marked with 'RS' are parity-check packets and '*' indicates packet losses during transmission.

Using $\mathbf{s}_\chi$ and $\mathbf{c}_\chi$ to represent the vectors (with length χ) of source and parity-check packets, respectively, for the selected χ batches, the expectation of the transmission latency τ is given as

$$\begin{aligned} E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi) &= E(\tau_{\mathbf{s}_\chi} + \tau_{\mathbf{c}_\chi} + \tau_R|\mathbf{s}_\chi, \mathbf{c}_\chi) \\ &= E(\tau_{\mathbf{s}_\chi} + \tau_{\mathbf{c}_\chi}|\mathbf{s}_\chi, \mathbf{c}_\chi) + E(\tau_R|\mathbf{s}_\chi, \mathbf{c}_\chi), \quad (6) \end{aligned}$$

where $E()$ represents the probabilistic expectation, and τ_R denotes the total latency incurred by retransmission; $\tau_{\mathbf{s}_\chi}$, $\tau_{\mathbf{c}_\chi}$ denote the transmission costs for all the source and parity-check packets, respectively, in the selected batches, i.e.,

$$\tau_{\mathbf{s}_\chi} = |\boldsymbol{\tau}_{\mathbf{s}_\chi}| = \sum_{i=1}^{\chi} \boldsymbol{\tau}_{\mathbf{s}_\chi}(i), \text{ and } \tau_{\mathbf{c}_\chi} = |\boldsymbol{\tau}_{\mathbf{c}_\chi}| = \sum_{i=1}^{\chi} \boldsymbol{\tau}_{\mathbf{c}_\chi}(i),$$

where $\boldsymbol{\tau}_{\mathbf{s}_\chi}(i)$ and $\boldsymbol{\tau}_{\mathbf{c}_\chi}(i)$ correspond to the transmission cost for the i-th batch. Given the RS codes $(\mathbf{s}_\chi, \mathbf{c}_\chi)$, the packet-loss rate p and the round-trip time r, the steady-state transmission throughput T is described by (4), and $\boldsymbol{\tau}_{\mathbf{s}_\chi}$, $\boldsymbol{\tau}_{\mathbf{c}_\chi}$ can be considered as constant-value vectors. With the notations $\boldsymbol{\tau}_0 = (\boldsymbol{\tau}_{\mathbf{s}_\chi} + \boldsymbol{\tau}_{\mathbf{c}_\chi})$ and $\tau_0 = \tau_{\mathbf{s}_\chi} + \tau_{\mathbf{c}_\chi} = |\boldsymbol{\tau}_0| = \mathbf{1}\cdot\boldsymbol{\tau}_0$, we have $E(\tau_{\mathbf{s}_\chi} + \tau_{\mathbf{c}_\chi}|\mathbf{s}_\chi, \mathbf{c}_\chi) = E(\tau_0)$, and (6) is further expressed as

$$\begin{aligned} E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi) &= E(\tau_0) + E(\tau_R|\mathbf{s}_\chi, \mathbf{c}_\chi) \\ &= |\boldsymbol{\tau}_0| + \mathbf{P}(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi)\cdot[\boldsymbol{\tau}_0 + E(\boldsymbol{\tau}_R|\mathbf{s}_\chi, \mathbf{c}_\chi)] \\ &\approx \sum_{n=0}^{\infty} \mathbf{P}^n(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi)\cdot\boldsymbol{\tau}_0 \\ &= \frac{1}{1 - \mathbf{P}(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi)}\cdot\boldsymbol{\tau}_0. \quad (7) \end{aligned}$$

Note that in (7), $E(\tau_R|\mathbf{s}_\chi, \mathbf{c}_\chi)$ was expanded as a summation of recursive retransmission costs multiplied by the corresponding probabilities, with an assumption that all processing cost upon data loss is ignorable. $\mathbf{P}(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi)$ is the vector of batch error probabilities with each element computed according to (2), and $\mathbf{P}^n(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi)$ is defined as

$$\mathbf{P}^n(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi) = \mathbf{P}^{(n-1)}(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi)\cdot\mathbf{P}(\mathbf{s}_\chi + \mathbf{c}_\chi, \mathbf{c}_\chi),\ n > 1.$$

The optimal distribution of parity-check packets, $\mathbf{c}_{\chi opt}$, is then given by

$$\mathbf{c}_{\chi opt} = \arg\min_{\mathbf{c}_\chi} E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi). \quad (8)$$

Equations (7-8) with (5) provide the theoretical solution to the problem. Yet an operational solution should also take into account the computation complexity as exhaustive search will not be feasible (the choice of $\mathbf{c}_\chi$ in (8) is arbitrary and therefore the solution space is not a close space). To accommodate real-time applications, a generalized *steepest decent* algorithm is used in this work to find the optimal (or a possibly suboptimal) solution. This fast algorithm starts with $\mathbf{c}_\chi = \emptyset$, i.e., no parity-check packets are added at the initial point. At each iteration, the algorithm adds one more parity-check packet to either one of the χ batches, which results in χ possibilities. It then finds amongst the one that decreases the expected delay $E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi)$ the most, or stops the computation if $E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi)$ increases for all cases, where $E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi)$ is calculated using (7). Figure 2 presents a simple illustration on this fast algorithm. The double-circled nodes indicate the searching path of the steepest decent algorithm, which have smaller expected transmission delays than their parents and siblings; the solid one represents the optimal operating point that has the minimum expected transmission delay, meaning that all its descendants (skipped in the plot) have larger expected delays. To simplify notation, the source vector $\mathbf{s}_\chi$ is ignored in Figure 2 and only the vector of parity-check packets

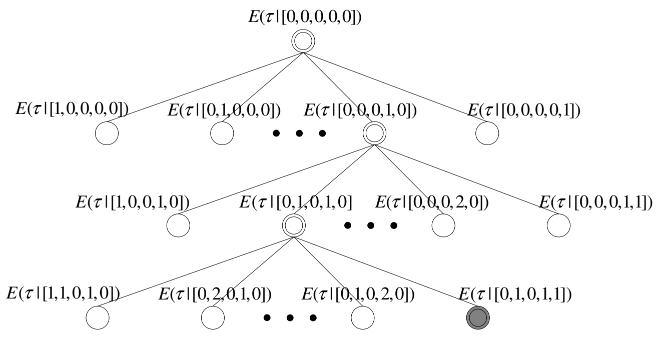

Fig. 2. Illustration on the steepest decent algorithm for finding $\mathbf{c}_{\chi opt}$.

is indicated. From Figure 2, it is apparent that the steepest decent algorithm has a linear computational complexity $O(K\cdot\chi)$, or more generally, $O(K\cdot L)$, where K is the total number of parity-check packets that are added (the depth of the tree) and L is the number of generated batches (the maximum width of the tree).

4. SIMULATION RESULTS

Simulation in this paper is performed using ns-2 [10]. Figure 3 shows the simulated topology and the test models, each of which is encoded to generate 10 enhancement batches. The bottleneck link is shared by f parallel flows. All the background traffic is FTP traffic transmitted over TCP.

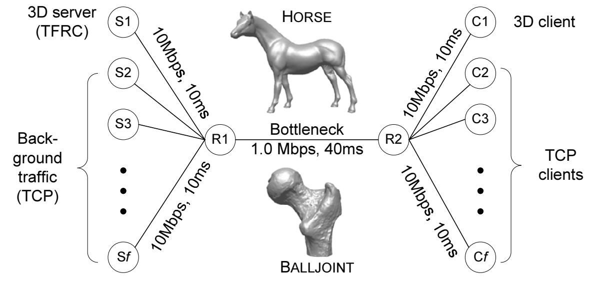

Fig. 3. Simulation topology in ns-2.

We compare the proposed mechanism (REP) with the 3TP protocol presented in [5] which deploys hybrid TCP/UDP to reduce the transmission latency. For the two mechanisms, Figures 4(a-b) plot the transmission latency with respect to different numbers of batches that are selected to be reliably transmitted, determined upon the distortion constraint. The results have been averaged over all received meshes during the simulation period. The number of parallel flows in the network is set to be $f = 12$, which results in a moderate network congestion situation as reflected by the measured packet-loss rate ($p \approx$ 6-7%). The packet sizes are $s = 1000$ bytes for Figure 4(a) and $s = 500$ bytes for Figure 4(b), respectively. As can be seen from these plots, REP considerably reduces the transmission latency compared to 3TP. Specifically, when the full resolution model is required by the receiving application, all the enhancement batches need to be reliably transmitted ($\chi = 10$). Simply, 3TP returns to be sole TCP in this case. In contrast, REP still achieves substantial delay reduction by using hybrid UEP/SR. For example, in Figure 4(a) where the packet size is $s = 1000$ bytes, 27% reduction in the average delay is observed for the HORSE model.

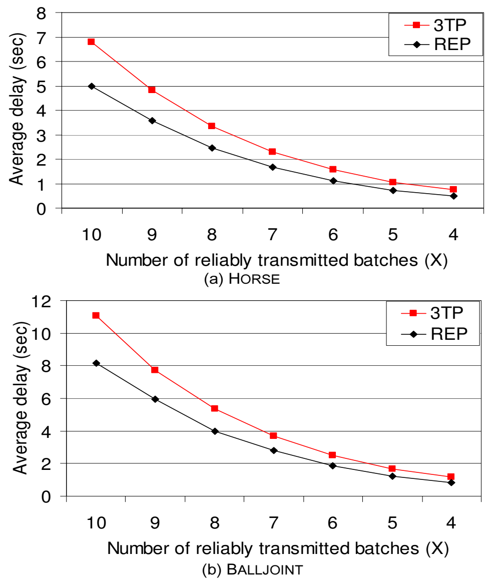

Fig. 4. Transmission latency for various portions of data.

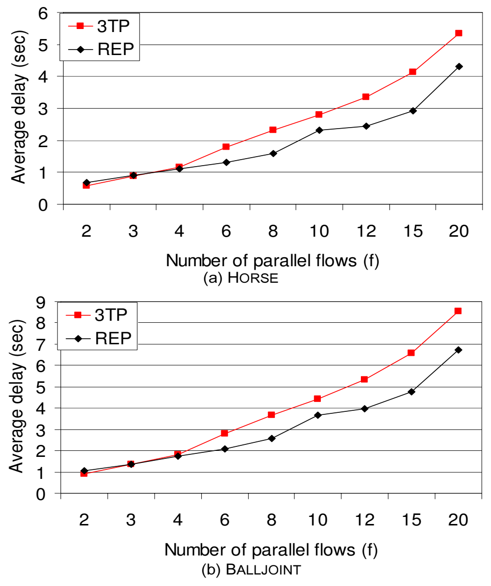

Fig. 5. Transmission latency under a quality requirement 36 dB.

In Figures 4(a-b), it is observed that as the portion of reliably transmitted data becomes smaller (i.e., smaller χ), which corresponds to a lower constraint on decoding distortion, the performance of REP gradually merges to 3TP. This behavior is anticipated because as the distortion constraint is lowered, both mechanisms transmit most of the data using TFRC without FEC or retransmission. Yet, considering a distributed online presentation of 3D scenes, a small reduction in the average delay provided by REP may still result in significant improvement on overall performance when a number of 3D meshes are transmitted on demand.

In Figure 5, variation of the average delay for different network congestion situations (characterized by the number of competing flows) is investigated. Suppose an upper bound of the decoding distortion, $D_{max} \geq 36$ dB, is requested for the models to be rendered. To satisfy D_{max}, $\chi = 8$ enhancement batches are selected to be reliably transmitted[2]. Figure 5 presents the delay results of REP and 3TP for the selected batches for the test models. It is shown that REP outperforms 3TP under most of the network conditions, and the gain is especially significant when the network encounters moderate or heavy congestion ($f \geq 6$ in the figures). For the network with light traffic load ($f \leq 4$), REP and 3TP perform similarly. When only two flows exist in the network and the packet size is $s = 1000$ bytes (Figure 5(b)), one may notice that 3TP provides slightly better results than REP. This is resulted from TCP's advantage of quick adaptation to the changes in available bandwidth compared with TFRC [6].

5. CONCLUSIONS

We have proposed a hybrid unequal-error-protection and retransmission mechanism for streaming multi-resolution meshes over lossy networks. To minimize the transmission latency under a distortion constraint, a portion of multi-resolution mesh data is selected to be reliably transmitted. Then, the best tradeoff between forward packet-loss resilience and retransmission is calculated in linear time using a steepest decent algorithm. The algorithm requires light-weight computation, making it attractive for distributed graphics applications.

[2]This number in general depends on the rate-distortion performance of the model.

6. REFERENCES

[1] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Trans. Visualization and Computer Graphics*, vol. 6, no. 1, pp. 79–93, 2000.

[2] Z. Yan, S. Kumar, and C.-C. Kuo, "Error-resilient coding of 3-D graphic models via adaptive mesh segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 7, pp. 860–873, 2001.

[3] G. Al-Regib and Y. Altunbasak, "An unequal error protection method for packet loss resilient 3-D mesh transmission," in *Proceedings of IEEE INFOCOM*, 2002, vol. 2, pp. 743–752.

[4] Z. Chen, B. Bodenheimer, and F. J. Barnes, "Robust transmission of 3D geometry over lossy networks," in *Proceedings of Web3D*, 2003, pp. 161–172.

[5] G. Al-Regib and Y. Altunbasak, "3TP: An application-layer protocol for streaming 3-D graphics," *IEEE Transactions on Multimedia*, 2005, to appear.

[6] M. Handley, S. Floyd, J. Padhye, and J. Widmer, *TCP friendly rate control (TFRC): protocol specification*, Request for Comments (RFC) 3448, The Internet Society, January 2003.

[7] H. Hoppe, "Progressive mesh," in *Proceedings of ACM SIGGRAPH*, 1996, pp. 99–108.

[8] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," in *Proceedings of Eurographics*, 1998, vol. 17(2), pp. 167–174.

[9] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-based congestion control for unicast applications," in *Proceedings of ACM SIGCOMM*, 2000, pp. 43–56.

[10] UCB/LBNL/VINT, *network simulator ns (version 2)*, available on http://www.isi.edu/nsnam/ns/.