3TP: An Application-Layer Protocol for Streaming 3-D Models

Ghassan AlRegib and Yucel Altunbasak, *Senior Member, IEEE*

***Abstract*—This paper addresses the problem of streaming progressively compressed three-dimensional (3-D) models over lossy networks. Out of all encoded packets that can be transmitted, we intelligently choose a subset of packets to be transmitted using transport control protocol in order to meet a distortion constraint, while transmitting the remaining packets using user datagram protocol to minimize the end-to-end delay. We call this new application-layer protocol 3-D models transport protocol. We show the effectiveness of this protocol both experimentally and theoretically. We compare the performance of the proposed protocol with systems that do not optimize transmission according to the content of the encoded bitstream. When the maximum distortion is 30, measured using the *Hausdorff* distance, we achieve savings in delay time ranging from 39% to 68% for packet-loss rates between 1% and 19%.**

I. Introduction

ACCESSING three-dimensional (3-D) animations over wired/wireless connections will impact many areas, including physics, engineering, education, entertainment, and commerce. For example, a real-time interactive technology will enable a physician to compare a 3-D data of the beating heart of a patient with selected animations in a central data-base of pathologies. Similarly, a technician with a wireless device will download an animation showing a device in operation. He will adjust the viewpoint and replay the animation to compare it with the behavior of the real device, identify the defective part, and download an animation that shows how to perform the maintenance task. Finally, large numbers of sports fans will be able to retrieve the motions of tennis or football players after a competition and replay them individually in fast mode to identify patterns or explore them in more detail using slow motion and new angles.

Such Internet applications utilize highly detailed 3-D models, giving rise to a large amount of data to be stored, transmitted, and rendered within a limited time frame. The latency in streaming these models prohibits smooth interaction with networked virtual environments. To alleviate such limitations, single-level compression methods [1], [2] can be employed to reduce the number of encoded bits and hence reduce the end-to-end transmission delay. A more efficient compression strategy aims to reduce the display latency by sending a coarse mesh of the 3-D model first, and transmitting refinement information later [3]–[10]. This refinement information is used to progressively transform the received crude mesh into a set of finer meshes until the full-resolution model is decoded on the client side. Even though progressive compression techniques reduce both the required bandwidth and the latency to display a 3-D model, they do not address the effect of packet losses on the decoded model quality. In a typical packet-switched network (e.g., the Internet), congestion and buffer overflow among other problems can cause packets to be lost.

In general, the client signs into a virtual world and requests to download a number of 3-D models with a minimum quality level or equivalently a maximum distortion level.[1] In order to deliver 3-D models with this distortion upper bound, several methods have been proposed in the literature to control the error introduced by losses. These error-control methods can be classified into three categories. The first class is *receiver-based* where lost parts of the bitstream are recovered on the client side from those parts that are correctly received [11]. The second category is *sender-based* [12]–[16]. Even though these methods are successful in protecting the streamed models, they do not account for the delay introduced by adding redundancy to the bitstream. The third class is *network-based* where the bitstream is considered as a stream of data without *a priori* knowledge of the content [17], [18].

In this paper, we propose a new application-layer protocol that uses both the transport control protocol (TCP) and the user datagram protocol (UDP) to stream the progressively compressed bitstream of the 3-D models in a virtual scene. The proposed 3-D models transmission protocol (3TP) ensures a minimum delay by carefully selecting the parts of the bitstream to be transmitted using TCP and the parts to be streamed using UDP. This choice depends on three factors: 1) the 3-D models; 2) the packet-loss rate; and 3) the maximum distortion level tolerated by the client. We demonstrate the effectiveness of 3TP both experimentally and theoretically. We show that 3TP reduces the delay time by a factor ranging from 39% to 68% compared to TCP at packet-loss rates between 1% and 19%. In contrast, Chen *et al.* in [19] randomly choose the part of the transmitted data to be sent using TCP regardless of the channel packet-loss rate.

This paper is organized as follows. Section II describes the compression method, while Section III describes streaming 3-D graphics using either TCP or UDP only. Sections IV and VI describe and demonstrate the efficacy of 3TP experimentally and

Manuscript received September 9, 2002; revised February 1, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Minoru Etoh.

G. AlRegib is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Savannah, GA 31407 USA (e-mail: gregib@ece.gatech.edu).

Y. Altunbasak is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: yucel@ece.gatech.edu).

Digital Object Identifier 10.1109/TMM.2005.858404

[1]Alternatively, the application specifies the time to deliver these meshes and the sender tries to maximize the quality of the received meshes. Even though the proposed method is suitable for both scenarios, in the rest of the paper, we will limit the discussion to the scenario where the user/application specifies the maximum distortion.

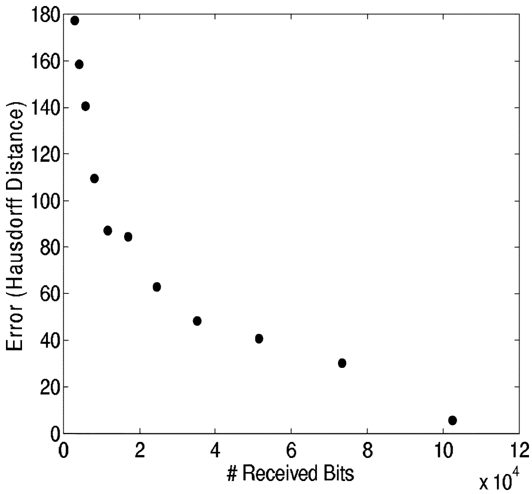


Fig. 1. R-D curve of the SMALL BUNNY model when progressively compressed into a base mesh and ten batches.

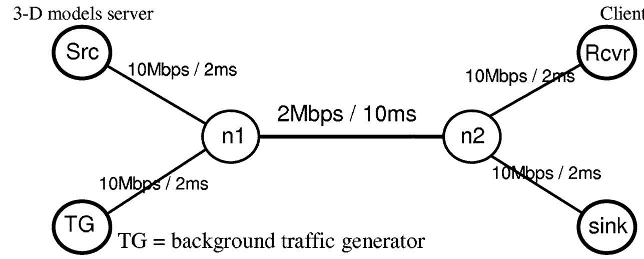


Fig. 2. Used network topology.

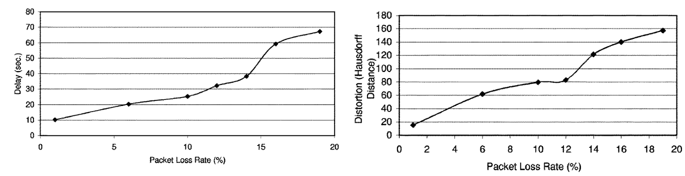


Fig. 3. (a) Measured delay when ten SMALL BUNNY models are streamed using TCP. (b) Average distortion (*Hausdorff* distance) when the base meshes are transmitted using TCP while all other batches are transmitted using UDP.

theoretically, respectively. Finally, conclusions are summarized in Section VII.

II. Three-Dimensional Compression

The *Compressed Progressive Mesh* (CPM) algorithm [3], [20], [21] is one of the most effective algorithms for progressive compression of 3-D models. Even thought CPM is used in this paper, the proposed protocol is applicable to other progressive compression methods with minor modifications. CPM applies a well-known simplification method that depends on two operations: *edge collapse* and *vertex split* that are applied at the encoder and the decoder, respectively [21], [22]. The encoding process is iterative. At the beginning of every iteration, a subset of edges is chosen to be collapsed. In addition to the resulting simplified mesh, a batch is produced that is used by the decoder to inverse the collapse operation. The base mesh can be compressed using any single-level mesh compression technique that exists in the literature [1], [2], [23]–[27]. The details of the algorithms can be found in [3]. Most importantly, if part of the connectivity data is lost, then the decoding process terminates.

Fig. 1 shows the rate-distortion (R-D) curve for the SMALL BUNNY model when progressively compressed using CPM into ten batches (levels) in addition to the base mesh. The *Hausdorff* distance is used as the distortion measure in this paper. In the following section, we show the effect of streaming 3-D models bitstreams using either TCP or UDP.

III. Three-Dimensional Streaming Using TCP/UDP

A. Streaming 3-D Models Using TCP

The congestion control algorithm of all variations of TCP employs an additive increase multiple decrease mechanism that controls the transmission rate. When no acknowledgment is received from the receiver, the sender assumes that congestion is taking place and it sharply reduces the transmission rate. The high delay associated with TCP, especially when the channel packet-loss rate is high, hinders the smooth online interactivity required by several 3-D applications. To illustrate the delay associated with TCP, we conducted several experiments to stream a number of progressively compressed 3-D models using TCP. In here we report the results for streaming ten SMALL BUNNY models where each model is progressively compressed into a base mesh and ten batches. We used a network simulator (ns-2) [28] to run these experiments with the topology shown in Fig. 2. The TCP packet size is chosen to be 256 bytes. The delay between the request to download the ten models and the time of receiving all packets correctly on the client side is shown in Fig. 3(a) for different packet-loss rates.

B. Streaming 3-D Models Using UDP

In general, for real-time applications, UDP is preferred in practice upon TCP because UDP does not require any feedback from the receiver. On the other hand, it does not guarantee the delivery of the transmitted packets to the client.[2] When the link between the sender and the receiver is error-free, UDP outperforms TCP in terms of end-to-end delay but as the packet-loss rate increases, the quality of the 3-D models on the client side degrades substantially. To illustrate the performance of UDP over different network conditions, we repeated the experiments in Section III-A using UDP as the transmission protocol. In order to keep a reasonable distortion level on the client side and because of the importance of the base mesh, we transmit all base meshes of all models in the scene using TCP, while UDP is used to transmit all other batches in the progressively compressed bitstream. The delay between the request to download the ten SMALL BUNNY models and the time of transmitting all packets is nearly the same at all packet-loss rates. In these experiments, the delay is approximately 8.14 s. The average distortion between the transmitted models and the received ones on the client side is depicted in Fig. 3(b). It is clear that the distortion constraint cannot be guaranteed, especially when the packet-loss rate is high.

By examining the results illustrated in Fig. 3, it is clear that there is a tradeoff between distortion and delay. The proposed

[2]One problem with using UDP is the fairness usage of Internet bandwidth. This problem is outside the scope of this paper.

protocol in this paper addresses this issue and is described in detail in the next section.

IV. 3TP: An Application-Layer 3-D Transport Protocol

The proposed 3TP protocol uses TCP and UDP intelligently to deliver the progressively compressed 3-D models to the client with an agreed-upon upper distortion bound within the minimum possible average time. In 3TP, the minimum delay is achieved by selecting certain parts of the encoded bitstream to be transmitted using TCP, while the remaining parts are transmitted using UDP. This choice is a function of the channel packet-loss rate and the maximum distortion.

The problem 3TP addresses can be stated as follows: "*Given (i) a virtual scene that contains M 3-D models and each model is progressively compressed into L levels, (ii) an upper bound on the distortion level ($\mathcal{D}_{\max}$), and (iii) the channel packet-loss rate (P_{LR}), determine the connectivity and the geometry parts of the encoded levels to be transmitted using TCP in order to minimize the delay*". In mathematical form, this problem can be re-stated as

$$\begin{aligned} &\min_{\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}} \arg \; \mathcal{T}\left(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}},P_{\mathrm{LR}}\right), \\ &\text{subject to}\; \mathcal{D}\left(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}},P_{\mathrm{LR}}\right) \le \mathcal{D}_{\max} \\ &\qquad 0 \le \chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}} \le L \end{aligned} \tag{1}$$

where $\mathcal{T}$ is the time delay between the request to download the M models and the time when all models are streamed out, $\mathcal{D}$ is the distortion on the client side, $\mathcal{D}_{\max}$ is the maximum acceptable distortion, χ^C_{TCP} is the number of connectivity levels to be transmitted using TCP, and χ^G_{TCP} is the number of geometry levels to be transmitted using TCP. In this paper, we solve this problem both experimentally and theoretically by modeling the delay ($\mathcal{T}$) and the distortion ($\mathcal{D}$) functions.

Because of the importance of the base mesh, we stream all base meshes at all conditions using TCP. Therefore, having $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}) = (0,0)$ corresponds to streaming the base mesh using TCP and all batches using UDP. On the other hand, when $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}) = (L,L)$, then all batches as well as the base mesh are transmitted using TCP.

A. Delay

In the proposed 3TP protocol, as more layers are transmitted using TCP, a higher delay is experienced. This is illustrated in Fig. 4, where we stream ten SMALL BUNNY models and each model is compressed into ten batches in addition to the base mesh. As shown, when only the base mesh is transmitted using TCP, i.e., $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}) = (0,0)$, then the delay, $\mathcal{T}$, is nearly the same at all packet-loss rates because base meshes constitute a small number of packets. The effect of the channel packet-loss rate on the delay is highest when all batches are transmitted using TCP, i.e., $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}) = (L,L)$.

B. Distortion

It is shown in Fig. 4 that the source of delay in 3TP is TCP. In contrast, the source of distortion is UDP when the packet-loss rate is higher than zero. This is illustrated in Fig.. 5(a) and (b). As shown in these plots, the distortion is minimum when $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}) = (L,L)$ since all levels are streamed using TCP.

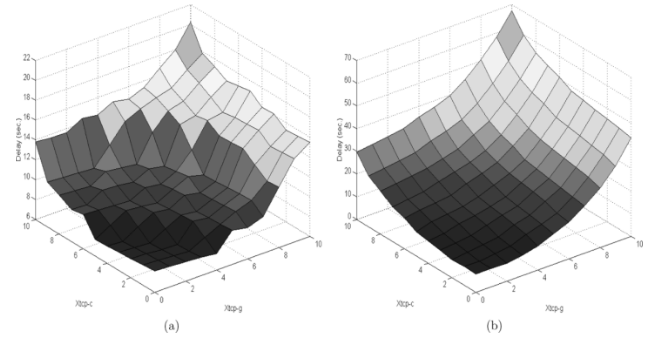


Fig. 4. Total delay ($\mathcal{T}$) of streaming ten SMALL BUNNY models. (a) $P_{\mathrm{LR}} = 6\%$. (b) $P_{\mathrm{LR}} = 19\%$.

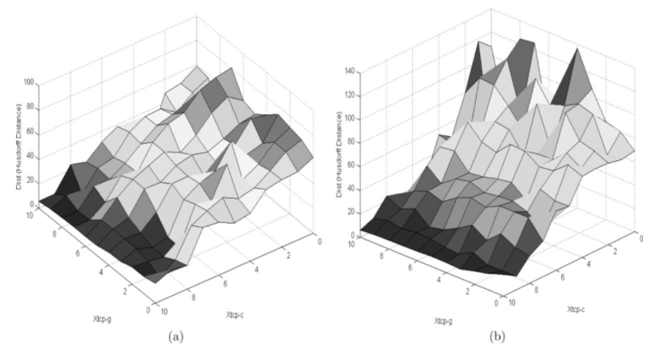


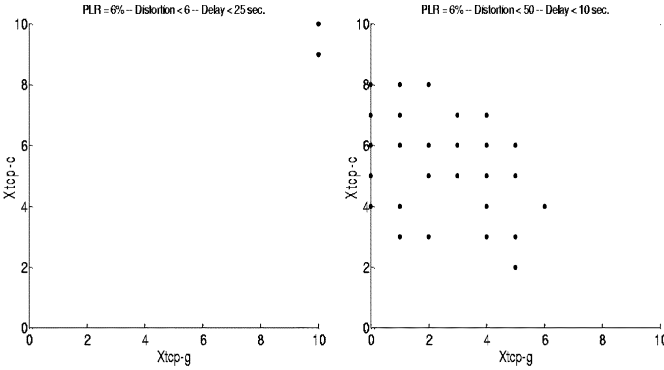
Fig. 5. Distortion ($\mathcal{D}$) of the displayed SMALL BUNNY models. (a) $P_{\mathrm{LR}} = 6\%$. (b) $P_{\mathrm{LR}} = 19\%$.

Even though it is expected that the distortion is maximum when $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}}) = (0,0)$ since only the base mesh is transmitted using TCP, the plots in Fig. 5(a) and (b) does not thoroughly reflect such expectation. The reason is that these experiments are performed statistically using ns-2 and in order to get more accurate curves, the experiments need to be repeated hundreds of times to compute the average, while we have only performed the experiment ten times. These experiments require a long processing time and repeating them many times is infeasible. This problem can be overcome by computing the delay and the distortion mathematically as will be shown in Section VI.

Notably, the plots in Fig. 5 reflect the fact that connectivity is more important than geometry. The choice of the number of connectivity/geometry levels to be transmitted using TCP depends on the 3-D models, the channel packet-loss rate, and the maximum distortion level. In the next section, we show how 3TP minimizes the delay for a given distortion and a given packet-loss rate by selecting the appropriate data to be transmitted using TCP, i.e., $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}})$.

V. Experimental Approach

In order to solve the problem in (1), we first follow an experimental approach where we determine all $(\chi^C_{\mathrm{TCP}},\chi^G_{\mathrm{TCP}})$ pairs that solve the problem experimentally.



(a) (b)

Fig. 6. Pairs of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ that satisfy: (a) $\mathcal{D} < 6$ and $\mathcal{T} < 25$ and (b) $\mathcal{D} < 50$ and $\mathcal{T} < 10$.

A. Delay Versus Distortion

To find the tradeoff between delay and distortion for a given channel and a set of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pairs, we follow the following simple procedure. This procedure finds the optimal solution of (1).

1) Obtain the packet-loss rate $(P_{\rm LR})$, the maximum distortion $(\mathcal{D}_{\max})$, the number of models (M), and the number of progressive levels each model is compressed into (L).
2) For each pair of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$, where $0 \leq \chi^C_{\rm TCP}, \chi^G_{\rm TCP} \leq L$, stream the M models over the channel using the topology shown in Fig. 2.
3) For each pair of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$, record the delay as well as the distortion of the displayed M models on the client's screen.
4) Choose the pair $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ that minimizes the delay, while the average distortion is equal or less than $\mathcal{D}_{\max}$.

Fig. 6(a) shows all pairs of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ that can stream ten SMALL BUNNY models with $\mathcal{D}_{\max} \leq 6$ and within a delay, $\mathcal{T}$, less than 25 s when the packet-loss rate, $P_{\rm LR}$, is 6%. As shown, two pairs of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ can meet these constraints when $P_{\rm LR} = 6\%$. In contrast, these constraints cannot be met when $P_{\rm LR} = 14\%$. At this packet-loss rate, more losses take place and hence more retransmissions are required. This results in delay higher than 25 s.

When the maximum acceptable distortion is increased to 50 and the delay is kept less than 10 s, several $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pairs satisfy these constraints, as shown in Fig. 6(b). When the packet-loss rate increases to 14%, only three $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pairs satisfy these constraints.

As shown in Fig. 6, for each maximum distortion, there are several choices of streaming out the bitstream. These choices introduce different delays. The choice that requires the minimum delay is the solution of (1). Fig. 7 depicts the minimum achievable delay for a given $\mathcal{D}_{\max}$. The corresponding $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pairs are listed in Table I for two packet loss rates. For example, when $\mathcal{D}_{\max} = 6$ and the packet-loss rate is 6%, the minimum achievable delay is 17.81 s when nine connectivity levels and ten geometry levels in addition to the base mesh are transmitted using TCP, while the remaining levels are transmitted using UDP. For a specific distortion level, as the packet-loss rate increases, more retransmissions are required. As a result, the minimum achievable delay increases as shown in Tables I and Fig. 7.

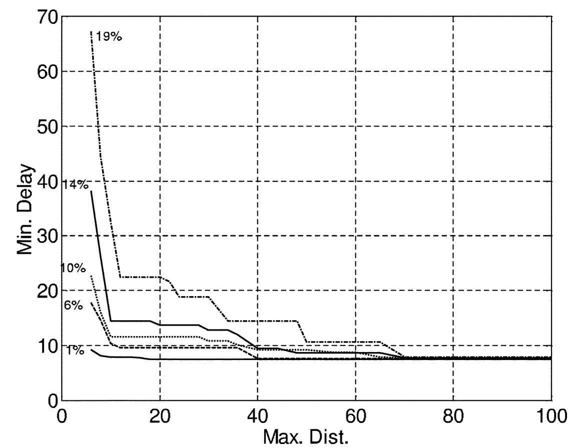
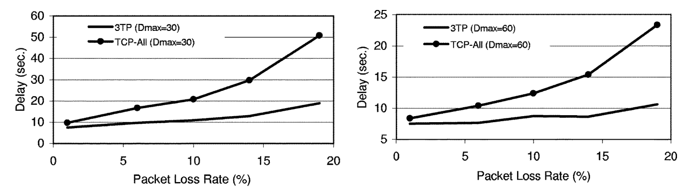


Fig. 7. Minimum delay versus maximum distortion specified by the client at different packet-loss rates.



(a) (b)

Fig. 8. Comparison between the performance of 3TP and the method of streaming all those levels, which satisfy the maximum distortion constraint, using TCP (TCP-All). (a) $\mathcal{D}_{\max} = 30$. (b) $\mathcal{D}_{\max} = 60$.

B. Performance Comparison

The area of streaming 3-D models over lossy channels is relatively new. The proposed methods in the literature minimize the end-to-end delay by reducing the number of bits required to be transmitted to the client. Existing progressive streaming algorithms start transmitting the encoded bitstream until the maximum distortion constraint is satisfied [3]. TCP is used as the transport protocol. For example, if $\mathcal{D}_{\max} = 30$, then from Fig. 1, the base mesh, the connectivity, and the geometry data of the first nine batches are transmitted using TCP only but if $\mathcal{D}_{\max} = 60$, then the base mesh and seven batches are transmitted using TCP only. In the rest of the paper, we will call this streaming strategy TCP-All.

Fig. 8 depicts a comparison between 3TP and TCP-All. In the 3TP case, we use Table I(a) and (b) to transmit $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ levels using TCP, while the remaining levels are transmitted using UDP. Note that 3TP ensures that all models are received at the client side with distortion smaller than $D_{\max}$.[3] When $P_{\rm LR} = 1\%$ and $\mathcal{D}_{\max} = 30$, 3TP saves 22% of the delay time compared to TCP-All but when $P_{\rm LR} = 14\%$, 3TP saves 56% of the delay time. Similarly, when the maximum distortion is increased to 60, 3TP saves up to 10% and 43% of the delay time when the packet-loss rates are 1% and 14%, respectively.

The proposed protocol finds the $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pair that minimizes the delay for a given $\mathcal{D}_{\max}$ at a given packet-loss rate. This process is performed off line after encoding the 3-D models. Then, a table similar to Table I is stored in the server together with the bitstream of the progressively compressed

[3]The average distortion of the 10 models and the distortion in each of these models is less than $D_{\max}$.

TABLE I
MINIMUM ACHIEVABLE DELAY FOR VARIOUS $\chi^C_{\rm TCP}$), $\chi^C_{\rm TCP}$, $\mathcal{D}_{\max}$, AND PACKET-LOSS RATES. (a) $P_{\rm LR} = 6\%$; (b) $P_{\rm LR} = 10\%$

$\mathcal{D}_{max}$, (χ^C_{TCP}, χ^G_{TCP})	6, (9,10)	8, (10,6)	20, (8,2)	40, (5,0)	60, (5,0)	80, (5,0)	90, (5,0)	100, (5,0)
Min delay (sec.)	17.8125	14.7594	9.7050	7.6360	7.6360	7.6360	7.6360	7.6360

(a)

$\mathcal{D}_{max}$,(χ^C_{TCP}, χ^G_{TCP})	6, (9,10)	8, (10,5)	20, (8,3)	40, (6,1)	60, (4,0)	80, (3,0)	90, (3,0)	100, (3,0)
Min delay (sec.)	22.8138	16.0655	11.6026	9.3191	8.7486	7.8313	7.8313	7.8313

(b)

3-D models. When a client requests the models, the server obtains the channel packet-loss rate from lower network-layer protocols such as Real-Time Control Protocol (RTCP). Then, the server chooses the $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pair that minimizes the end-to-end delay and starts streaming the bitstream accordingly.

To obtain more accurate distortion averages than the ones shown in Fig. 5, hundreds of experiments need to be repeated many times. Such processing is infeasible in practice. Alternatively, in the next section, we derive mathematical models of both the delay $(\mathcal{T})$ and the distortion $(\mathcal{D})$.

VI. MATHEMATICAL MODELLING

In Section V, we experimentally found the optimal $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$ pair that minimizes the delay for a fixed distortion level. Instead, in this section, we model mathematically the distortion $(\mathcal{D})$ and the delay $(\mathcal{T})$ functions and solve the optimization problem in (1).

A. Delay Model

We model the delay experienced by both TCP and UDP connections. We use the TCP throughput model proposed in [29] to compute the required time to deliver all TCP packets to the client. The total latency is the summation of the latency of the TCP and the UDP streams and is given by

$$\begin{aligned}\mathcal{T}\left(\chi^C_{\rm TCP}, \chi^G_{\rm TCP}\right) &= \mathcal{T}_{\rm TCP} + \mathcal{T}_{\rm UDP} \\ &= \frac{B_{\rm TCP} \times {\rm RTT} \times \sqrt{P_{\rm LR}}}{k \times {\rm MSS}} + \frac{B_{\rm UDP}}{R_{\rm UDP}} s \qquad (2)\end{aligned}$$

where $\mathcal{T}_{\rm TCP}$ is the time required to deliver all TCP packets to the client, $B_{\rm TCP}$ is the total number of bytes transmitted using TCP, RTT is the round-trip-time, LR is the average packet-loss rate, MSS is the maximum segment size, k is a constant, $\mathcal{T}_{\rm UDP}$ is the time required to transmit all UDP packets, $B_{\rm UDP}$ is the total number of bytes transmitted using UDP, and $R_{\rm UDP}$ is the transmission rate of the UDP stream.

In this paper, we assume that both RTT and $P_{\rm LR}$ do not vary during the streaming time for an end-to-end channel. Similarly, we assume that k, MSS, and $R_{\rm UDP}$ are constants. On the other hand, both $B_{\rm TCP}$ and $B_{\rm UDP}$ are functions of $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP})$.

Fig. 9(a) and (b) depicts the total delay, computed using (2), of streaming ten SMALL BUNNY models. In these calculations, ${\rm RTT} = 28$ ms, ${\rm MSS} = 256$ bytes, and $k = 1.22$.[4] By comparing these figures with the corresponding ones obtained experimentally in Fig. 4, it is clear that (2) approximates accurately the end-to-end delay. Moreover, the processing time required to obtain Fig. 9 is much less than the processing time required to obtain Fig. 4.

[4]This choice of k is recommended by [29].

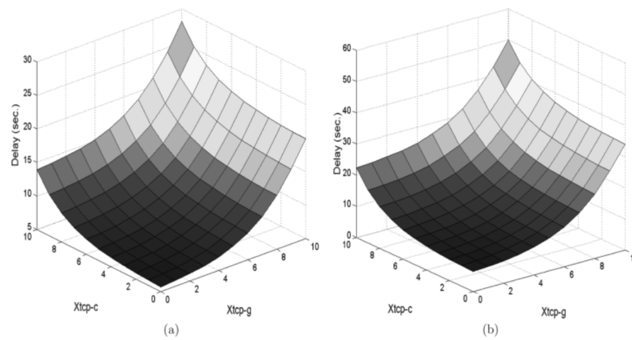


Fig. 9. Theoretical delay $(\mathcal{T})$ of streaming ten SMALL BUNNY models. (a) $P_{\rm LR} = 6\%$. (b) $P_{\rm LR} = 19\%$.

B. Distortion Model

There is no distortion associated with the TCP stream because TCP delivers all packets to the client. Therefore, the main source of distortion is the dropped UDP packets and it is a function of the packet-loss rate. To derive the distortion model, we investigate three cases: $\chi^C_{\rm TCP} = \chi^G_{\rm TCP}$, $\chi^C_{\rm TCP} < \chi^G_{\rm TCP}$, and $\chi^C_{\rm TCP} > \chi^G_{\rm TCP}$. First, we illustrate the derivation of the distortion equation when $\chi^C_{\rm TCP} = \chi^G_{\rm TCP}$ using a specific example, then we give the general equation. Let the total number of batches to be 10 $(L = 10)$ and let $(\chi^C_{\rm TCP}, \chi^G_{\rm TCP}) = (8, 8)$, then we will have two batches to transmit using UDP. Each batch consists of a connectivity and a geometry part. Some of these parts might be lost during transmission and therefore we will have a total of seven scenarios as illustrated in Fig. 10(a). Note that when the connectivity part of a batch is lost, then decoding stops, which is not the case when the geometry part is lost. For example, the expected distortion resulting from losing the connectivity part of batch (9) [i.e., scenario (2) in Fig. 10(a)] is written as $E^{(9)}_C P^{(9)}_C$, where $E^{(9)}_C$ is the distortion resulting from losing the connectivity part of batch (9), and $P^{(9)}$ is the probability of losing this connectivity part. $E^{(9)}_C$ is measured by decoding the bitstream prior to level-9. Then, we measure the error between the resulting mesh and the original model; i.e., $E^{(9)}_C$ is the error between the original model and the mesh produced by decoding level-8. Similarly, $E^{(9)}_G$ is measured as the error between the original model and the mesh produced by decoding all connectivity levels and all geometry levels except the geometry data associated with level-9.

Similarly, the expected distortion when scenario (5) in Fig. 10(a) occurs is written as $E^{(9,10)}_{G,C}(1 - P^{(9)}_C)P^{(9)}_G P^{(10)}_C$,

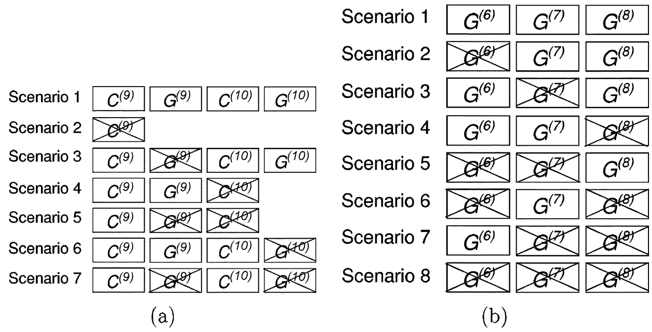


Fig. 10. (a) Illustration of the possible scenarios when sending the connectivity (C) and the geometry (G) bits of batches 9 and 10 using UDP. In this example, the total number of batches is 10 and $(\chi^C_{\text{TCP}}, \chi^G_{\text{TCP}}) = (8, 8)$. (b) Illustration of the possible scenarios when sending the geometry (G) bits of batches 6, 7, and 8 using UDP. In this example, the total number of batches is 10 and $(\chi^C_{\text{TCP}}, \chi^G_{\text{TCP}}) = (8, 5)$. The crossed levels imply that part or all the bitstream associated wit this level is lost during transmission.

where $E^{(9,10)}_{G,C}$ is the distortion resulting from losing the geometry part of batch (9) as well as the connectivity part of batch (10). Combining all these scenarios, we get the expected distortion as follows:

$$\begin{aligned}
&\mathcal{D}\left(\chi^C_{\text{TCP}} = \chi^G_{\text{TCP}} = 8\right) \\
&\quad = E_Q\left(1-P_C^{(9)}\right)\left(1-P_C^{(10)}\right)\left(1-P_G^{(9)}\right)\left(1-P_G^{(10)}\right) \\
&\qquad + E_C^{(9)}P_C^{(9)} + E_C^{(10)}P_C^{(10)}\left(1-P_C^{(9)}\right)\left(1-P_G^{(9)}\right) \\
&\qquad + E_{C,G}^{(10,9)}P_C^{(10)}P_G^{(9)}\left(1-P_C^{(9)}\right) + E_G^{(9)}P_G^{(9)}\left(1-P_C^{(9)}\right) \\
&\qquad \times\left(1-P_C^{(10)}\right)\left(1-P_G^{(10)}\right) + E_G^{(10)}P_G^{(10)}\left(1-P_C^{(9)}\right) \\
&\qquad \times\left(1-P_C^{(10)}\right)\left(1-P_G^{(9)}\right) + E_{G,G}^{(9,10)}P_G^{(9)}P_G^{(10)} \\
&\qquad \times\left(1-P_C^{(9)}\right)\left(1-P_C^{(10)}\right),
\end{aligned} \tag{3}$$

where E_Q is the quantization error, $P_C^{(j)}$ is the probability of losing the connectivity part of batch (j), $P_G^{(j)}$ is the probability of losing the geometry part of batch (j), $E_C^{(j)}$ is the distortion introduced by losing the connectivity part of batch (j), $E_G^{(j)}$ is the distortion introduced by losing the geometry part of batch (j), and $E_{C,G}^{(j,i)}$ is the distortion introduced by losing the connectivity part of batch (j) as well as the geometry part of batch (i). Calculations of $P_C^{(j)}$, $P_G^{(j)}$, and $E^{(j)}$ are detailed later in this section.

The decoding process stops when the connectivity information of a certain batch is lost and therefore the error introduced by losing the connectivity part of batch (j) is higher than the error introduced by losing the geometry part of batch (i). Therefore, we will assume the following:

$$E_{C,G}^{(i,j)} \approx E_C^{(i)} \quad and \quad E_{C,G,\cdots,G}^{(i,j,\cdots,k)} \approx E_C^{(i)} \tag{4}$$

where $E_{C,G,\cdots,G}^{(i,j,\cdots,k)}$ is the distortion resulting from losing the connectivity part of batch (i), as well as the geometry parts of batches (j to k).

To further simplify the expression in (3), we assume that the distortion resulting from losing two or more geometry levels is the sum of the distortions resulting from losing these parts individually, i.e.,

$$E_{G,G}^{(i,j)} \approx E_G^{(i)} + E_G^{(j)} \quad and \quad E_{G,G,\cdots,G}^{(i,j,\cdots,k)} \approx E_G^{(i)} + \cdots + E_G^{(k)} \tag{5}$$

where $E_{G,G,\cdots,G}^{(i,j,\cdots,k)}$ is the distortion resulting from losing the geometry parts of batches (i to k).

After incorporating the assumptions in (4) and (5) into (3), the expected distortion is simplified to

$$\begin{aligned}
&\mathcal{D}\left(\chi^C_{\text{TCP}} = \chi^G_{\text{TCP}} = 8\right) \\
&\quad = E_Q\left(1-P_C^{(9)}\right)\left(1-P_C^{(10)}\right)\left(1-P_G^{(9)}\right)\left(1-P_G^{(10)}\right) \\
&\qquad + E_C^{(9)}P_C^{(9)} + E_C^{(10)}P_C^{(10)}\left(1-P_C^{(9)}\right) \\
&\qquad + E_G^{(9)}P_G^{(9)}\left(1-P_C^{(9)}\right)\left(1-P_C^{(10)}\right) \\
&\qquad + E_G^{(10)}P_G^{(10)}\left(1-P_C^{(9)}\right)\left(1-P_C^{(10)}\right).
\end{aligned} \tag{6}$$

Equation (6) can be generalized for any number of batches, L, where $\chi^C_{\text{TCP}}(=\chi^G_{\text{TCP}})$ levels are transmitted using TCP as

$$\begin{aligned}
&\mathcal{D}\left(\chi^C_{\text{TCP}} = \chi^G_{\text{TCP}}\right) \\
&\quad = E_Q \prod_{j=\chi^C_{\text{TCP}}+1}^{L}\left(1-P_C^{(j)}\right)\prod_{j=\chi^G_{\text{TCP}}+1}^{L}\left(1-P_G^{(j)}\right) \\
&\qquad + E_C^{\left(\chi^C_{\text{TCP}}+1\right)}P_C^{\left(\chi^C_{\text{TCP}}+1\right)} + \sum_{i=\chi^C_{\text{TCP}}+2}^{L} E_C^{(i)}P_C^{(i)} \\
&\qquad \times \prod_{j=\chi^C_{\text{TCP}}+1}^{i-1}\left(1-P_C^{(j)}\right) + \sum_{i=\chi^G_{\text{TCP}}+1}^{L} E_G^{(i)}P_G^{(i)} \\
&\qquad \times \prod_{j=\chi^C_{\text{TCP}}+1}^{i}\left(1-P_C^{(j)}\right).
\end{aligned} \tag{7}$$

The second and the third terms are associated with the distortion resulting from losing connectivity levels. We will denote the summation of these two terms as two terms as $\mathcal{D}'$.

The distortion when $\chi^C_{\text{TCP}} < \chi^G_{\text{TCP}}$ can be derived similar to the above case when $\chi^C_{\text{TCP}} = \chi^G_{\text{TCP}}$. In the former case, more connectivity levels are transmitted using UDP and only $\mathcal{D}'$ is affected. The resulting distortion is the same as the one given in (7)

$$\begin{aligned}
&\mathcal{D}\left(\chi^C_{\text{TCP}} \le \chi^G_{\text{TCP}}\right) \\
&\quad = E_Q \prod_{i=\chi^C_{\text{TCP}}+1}^{L}\left(1-P_C^{(i)}\right)\prod_{j=\chi^G_{\text{TCP}}+1}^{L}\left(1-P_G^{(j)}\right) + \mathcal{D}' \\
&\qquad + \sum_{i=\chi^G_{\text{TCP}}+1}^{L} E_G^{(i)}P_G^{(i)} \prod_{j=\chi^C_{\text{TCP}}+1}^{i}\left(1-P_C^{(j)}\right).
\end{aligned} \tag{8}$$

Now, we derive the distortion when $\chi^C_{\text{TCP}} > \chi^G_{\text{TCP}}$. For simplicity, we derive the equations for a specific example and then we give the general formula. Assume that the total number of batches is $L = 10$, $\chi^C_{\text{TCP}} = 8$, and $\chi^G_{\text{TCP}} = 5$. When $5 \le \chi^G_{\text{TCP}} \le 7$, there will be eight possible scenarios as shown in Fig. 10(b). Each scenario is combined with the seven scenarios shown in Fig. 10(a). As a result, there will be 56 possible scenarios. To further simplify the expression, we assume that the distortion resulting from losing two ore more geometry levels as the sum of the distortions resulting from losing these geometry levels individually as given by (5). By incorporating this assumption and simplifying the resulting expression, we get

$$\begin{aligned}\mathcal{D}\left(\chi_{\mathrm{TCP}}^{C} > \chi_{\mathrm{TCP}}^{G}\right) &= E_Q \prod_{i=\chi_{\mathrm{TCP}}^{C}+1}^{L}\left(1-P_C^{(i)}\right)\prod_{j=\chi_{\mathrm{TCP}}^{G}+1}^{L}\left(1-P_G^{(j)}\right)\\ &+ \sum_{i=\chi_{\mathrm{TCP}}^{G}+1}^{\chi_{\mathrm{TCP}}^{C}} E_G^{(i)}P_G^{(i)}\sum_{j=\chi_{\mathrm{TCP}}^{C}+1}^{L} P_C^{(j)}\\ &\times \prod_{k=\chi_{\mathrm{TCP}}^{C}+1}^{L}\left(1-P_C^{(k)}\right)+E_C^{(\chi_{\mathrm{TCP}}^{C}+1)}P_C^{(\chi_{\mathrm{TCP}}^{C}+1)}\\ &+ \sum_{i=\chi_{\mathrm{TCP}}^{C}+2}^{L} E_C^{(i)}P_C^{(i)}\prod_{j=\chi_{\mathrm{TCP}}^{C}+1}^{i-1}\left(1-P_C^{(j)}\right). \end{aligned} \tag{9}$$

To find the distortion $\mathcal{D}$ that is given in (8) and (9), we need to evaluate the distortion resulting from losing the connectivity part of level (j) $(E_C^{(j)})$, the distortion resulting from losing the geometry part of level (j) $(E_G^{(j)})$, and the probability of losing the connectivity or the geometry parts of level (j) ($P_C^{(j)}$ or $P_G^{(j)}$). In this paper, we evaluate the first two terms using the *Hausdorff* distance. We achieve this by decoding the encoded bitstream for every $0 \le j \le L$ and measure the *Hausdorff* distance between the decoded and the original models.

The probability of losing a certain part (either connectivity or geometry) of batch (j) (i.e., $P^{(j)}$) depends on the number of packets in this part. In general, $P^{(j)}$ is given by $P^{(j)} = \sum_{l=1}^{K^{(j)}} p(l,n)$, where $K^{(j)}$ is the number of packets this part (connectivity/geometry) of level (j) and $p(l,n)$ is the probability of losing l packets out of n packets. To evaluate $p(l,n)$, we use the two-state Markovian model to model the end-to-end channel between the server and the client. More details on evaluating $p(l,n)$ can be found in [30]. The choice of n depends on both $k^{(j)}$ and the number of models that are transmitted. For example, if the connectivity part of a certain level constitutes five packets and there are ten models, then $n = 5\times 10 = 50$ packets.

We evaluated the average expected distortion $(\mathcal{D})$ for different packet-loss rates (P_{LR}) when ten SMALL BUNNY models are streamed with $(0,0) \le (\chi_{\mathrm{TCP}}^{C}, \chi_{\mathrm{TCP}}^{G}) \le (10,10)$. Fig. 11(a) and (b) shows the distortion when the packet-loss rate is 6% and 19%, respectively.

By comparing the distortion plots in Fig. 11 with the corresponding plots that were obtained experimentally in Fig. 5, one can conclude that the mathematical model achieve good approximation of the distortion when part of the encoded bitstream is streamed using UDP. Moreover, the theoretical plots do not have the statistical problem the experimental plots are experiencing where the plot is not smooth (Fig. 5). Finally, the plots in Fig. 11 require less time than those obtained experimentally in Fig. 5.

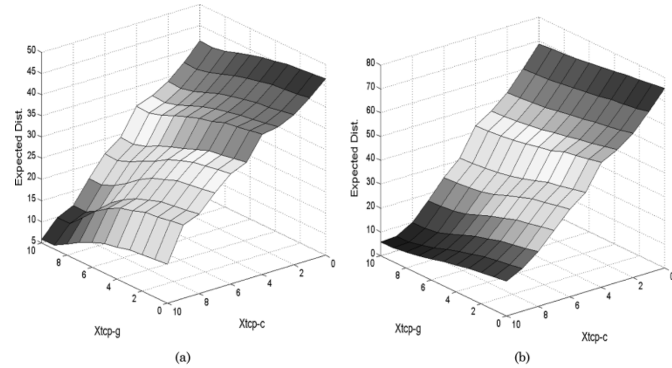


Fig. 11. Theoretical average distortion. (a) $P_{\mathrm{LR}} = 6\%$. (b) $P_{\mathrm{LR}} = 19\%$.

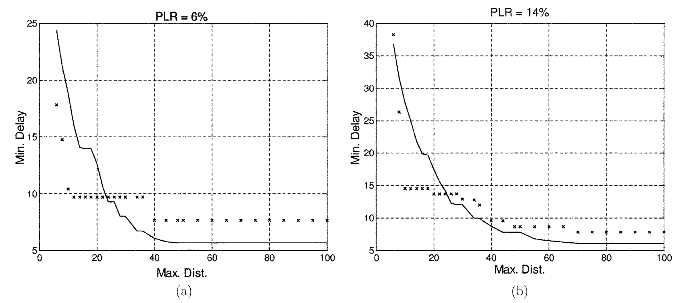


Fig. 12. Minimum achievable delay using 3TP. The straight lines are the theoretical results while the starred points are obtained experimentally.

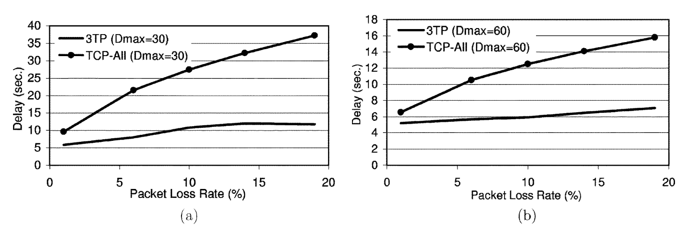


Fig. 13. Comparison between the performance of 3TP and TCP-All. (a) $\mathcal{D}_{\max} = 30$. (b) $\mathcal{D}_{\max} = 60$.

C. Performance Comparison

We repeated the comparison experiments that we performed in Section V-B but in here the delay $(\mathcal{T})$ and the distortion $(\mathcal{D})$ curves are obtained mathematically using (2) and (8) and (9), respectively. Fig. 12(a) and (b) compares the experimental and the theoretical solutions for two packet-loss rates. It is noticed that at low $\mathcal{D}_{\max}$, the experimental solution gives lower delay than the corresponding theoretical solution. This is caused by the insufficient number of iterations of the experiments as we discussed at the end of Section IV. Nevertheless, for $\mathcal{D}_{\max} > 30$, both methods give similar delay.

Fig. 13 depicts the theoretical comparison between 3TP and TCP-All. When $P_{\mathrm{LR}} = 1\%$ and $\mathcal{D}_{\max} = 30$, 3TP saves 39% of the delay time compared to TCP-All but when $P_{\mathrm{LR}} = 14\%$, 3TP saves 62% of the delay time. Similarly, when the maximum distortion is increased to 60, 3TP saves up to 20% and 54% of the delay time when the packet-loss rate is 1% and 14%, respectively. Similar results are obtained when the HORSE model is transmitted.

VII. Conclusions

In this paper, we proposed an application layer protocol for streaming 3-D models over lossy channels that combines source and channel characteristics to minimize the end-to-end delay while satisfying the distortion constraint. For every channel and distortion constraint, we choose the number of connectivity and geometry levels that are to be transmitted using TCP while the remaining levels are transmitted using UDP. We choose these pairs both experimentally and theoretically and in both cases, the proposed protocol outperforms the alternative method of using TCP for all connectivity and geometry levels that satisfy the distortion constraint.

References

[1] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Trans. Graph.*, vol. 17, no. 2, pp. 84–115, Apr. 1998.

[2] J. Rossignac, "Edgebreaker: connectivity compression for triangle meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 1, pp. 47–61, Jan. 1999.

[3] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 6, no. 1, pp. 79–93, Jan.–Mar. 2000.

[4] G. Taubin, A. Gueziec, W. Horn, and F. Lazarus, "Progressive forest split compression," in *Proc. SIGGRAPH'98*, Jul. 1998, pp. 123–132.

[5] D. Cohen-Or, D. Levin, and O. Remez, "Progressive compression of arbitrary triangular meshes," in *Proc. IEEE Conf. Visualization '99*, San Francisco, CA, Oct. 1999, pp. 67–72.

[6] C. L. Bajaj, V. Pascucci, and G. Zhuang, "Progressive compressive and transmission of arbitrary triangular meshes," in *Proc. IEEE Conf. Visualization '99*, San Francisco, CA, 1999, pp. 307–316.

[7] P. Alliez and M. Desbrun, "Valence-driven connectivity encoding for 3D meshes," in *Proc. Eurographics*, Manchester, U.K., 2001.

[8] A. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin, "MAPS: multiresolution adaptive parameterization of surfaces," in *Proc. SIGGRAPH'98*, 1998, pp. 95–104.

[9] J. Li and C. C. Kuo, "Progressive coding of 3D graphic models," *Proc. IEEE*, vol. 86, no. 6, pp. 1052–1063, Jun. 1998.

[10] H. Hoppe, "Efficient Implementation of Progressive Meshes,", Microsoft Research Tech. Rep.(MSR-TR-98-2), Jan. 1998.

[11] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communications: a review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[12] C. L. Bajaj, S. Cutchin, V. Pascucci, and G. Zhuang, "Error Resilient Transmission of Compressed VRML," Univ. Texas at Austin, 1998.

[13] Z. Yan, S. Kumar, and C.-C. J. Kuo, "Error resilient coding of 3-D graphic models via adaptive mesh segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 7, pp. 138–144, Jul. 2001.

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

[15] G. Al-Regib, Y. Altunbasak, and J. Rossignac, "An unequal error protection method for progressively compressed 3-D models," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing*, vol. 2, Orlando, FL, May 2002, pp. 2041–2044.

[16] G. Al-Regib, Y. Altunbasak, J. Rossignac, and R. Mersereau, "Protocol for streaming compressed 3-D animations over lossy channels," in *Proc. IEEE Int. Conf. Multimedia and Expo*, vol. 1, Lausanne, Switzerland, Aug. 2002, pp. 353–356.

[17] S. Bischoff and L. Kobbelt, "Toward robust broadcasting of geometry data," Comput. Graph., vol. 2, no. 5, pp. 665–675, 2002, to be published.

[18] I. M. Martin, "Arte—adaptive rendering and transmission environment," in *Proc. ACM Multimedia*, Los Angeles, CA, 2000.

[19] Z. Chen, B. Bodenheimer, and J. F. Barnes, "Robust transmission of 3D geometry over lossy networks," in *Proc. 8th Int. Conf. 3D Web Technology*, Saint Malo, France, Mar. 9–12, 2003, pp. 161–208.

[20] J. Popovic and H. Hoppe, "Progressive simplicial complexes," in *Proc. SIGGRAPH'97*, 1997, pp. 217–224.

[21] H. Hoppe, "Progressive meshes," in *Proc. ACM SIGGRAPH'96*, 1996, pp. 99–108.

[22] R. Ronford and J. Rossignac, "Full-range approximations of triangulated polyhedra," in *Proc. Eurographics'96*, vol. 15-3, Aug. 1996, p. C-67.

[23] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. Graphics Interface*, Vancouver, BC, Canada, Jun. 1998.

[24] M. Deering, "Geometry compression," in *Proc. SIGGRAPH'95*, Aug. 1995, pp. 13–20.

[25] S. Gumhold and W. Strasser, "Real time compression of triangle mesh connectivity," in *Proc. SIGGRAPH'98*, Jul. 1998, pp. 133–140.

[26] M. Chow, "Optimized geometry compression for real-time rendering," in *Proc. IEEE Visualization'97*, Phoenix, AZ, Oct. 19–24, 1997, pp. 347–354.

[27] G. Taubin, W. Horn, F. Lazarus, and J. Rossignac, "Geometry coding and VRML," *Proc. IEEE*, vol. 86, no. 6, pp. 1228–1243, Jun. 1998.

[28] UCB/LBNL/VINT. Network Simulator ns (Version 2). [Online] Available: http://www.isi.edu/nsnam/ns/

[29] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. Netw.*, vol. 7, no. 4, pp. 458–472, Aug. 1999.

[30] U. Horn, K. Stuhlmuller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Process.: Image Commun.*, vol. 15, pp. 77–94, 1999.

Ghassan AlRegib received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology (Georgia Tech), Atlanta, in June 2003. His Ph.D. dissertation focused on developing error-resilient techniques to stream 3-D graphics over lossy channels.

He joined the faculty at Georgia Tech in August 2003, where he is an Assistant Professor at the School of Electrical and Computer Engineering and is currently working on projects related to multimedia networking, video and 3-D compression and streaming, shared reality, and multimodal processing of 3-D objects in shared spaces.

Dr. AlRegib received the Outstanding Graduate Teaching Award from the School of Electrical and Computer Engineering in 2000–2001, the Center for Signal and Image Processing Research Award in spring 2003, and the Center for Signal and Image Processing Service Award in spring 2003, all from Georgia Tech. He serves as the Special Sessions Program Chair for ICIP-2006.

Yucel Altunbasak (M'97–SM'02) received the Ph.D. degree from the University of Rochester, Rochester, NY, in 1996.

He joined the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, in 1999, where he is currently an Associate Professor. He was previously with Hewlett-Packard Research Laboratories in 1996 and taught at Stanford and San Jose State Universities as a Consulting Assistant Professor. He is currently working on industrial- and government-sponsored projects related to multimedia networking, wireless video, video coding, genomics signal processing, and such inverse imaging problems as super-resolution and demosaicking. His research efforts to date have resulted in over 110 peer-reviewed publications and 15 patents/patent applications. Some of his inventions have been licensed by Office of Technology Licensing at Georgia Tech.

Dr. Altunbasak is an Associate Editor for the IEEE Transactions on Image Processing, the IEEE Transactions on Signal Processing, *Signal Processing: Image Communications*, and for the *Journal of Circuits, Systems and Signal Processing*. He served as the lead Guest Editor on the *Image Communications* Special Issue on Wireless Video. He is the Vice President for the IEEE Communications Society Multimedia Communications Technical Committee and has been elected to the IEEE Signal Processing Society IMDSP Technical Committee. He has served as a Co-Chair for "Advanced Signal Processing for Communications" Symposia at ICC'03, as a Track Chair at ICME'03 and ICME'04, as a Panel Sessions Chair at ITRE'03, as a Session Chair at various international conferences, and as a Panel Reviewer for government funding agencies. He will serve as the Technical Program Chair for ICIP-2006. He is a co-author for a conference paper that received the Best Student Paper Award at ICIP'03 and co-authored a conference paper that has been selected as design finalist at EMBS'2004. He received the National Science Foundation (NSF) CAREER Award and is a recipient of the "2003 Outstanding Junior Faculty Award" from the School of Electrical and Computer Engineering.