3TP: 3-D Models Transport Protocol

Ghassan Al-Regib
Center for Signal and Image Processing
School of Electrical and Computer Engineering
Georgia Tech Regional Engineering Program
Georgia Institute of Technology
Savannah, GA 31407
E-mail: gregib@ece.gatech.edu

Yucel Altunbasak
Center for Signal and Image Processing
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
E-mail: yucel@ece.gatech.edu

Abstract

This paper addresses the problem of streaming progressively compressed 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 TCP in order to meet a given distortion constraint, while transmitting the remaining packets using UDP to minimize the end-to-end delay. We call this new application-layer protocol 3TP (3-D models transport protocol). In this paper, we *mathematically* model both the delay and the distortion for a given channel characteristics. Then, we minimize the expected delay for a given distortion upper bound. The proposed protocol results in savings of 39% and 62% in delay time at packet-loss rates of 1% and 14%, respectively, compared to systems that do not optimize transmission according to the encoded bitstream content.

CR Categories: E.4 [Coding and Information Theory]: Error control codes I.3.8 [Computer graphics]: Applications I.4.2 [Image Processing and Computer Vision]: Compression

Keywords: 3-D Graphics Streaming, Networked Graphics, Joint Source and Network Coding, Multimedia Streaming Protocols, 3-D Mesh Compression, Virtual Reality over IP Networks.

1 Introduction

An increasing number of 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. Moreover, limited-bandwidth links in the Internet cause latency that prohibits smooth interaction. To alleviate such limitations, compression methods [Taubin and Rossignac 1998; Rossignac 1999] have been proposed to reduce the number of bits representing 3-D models. Even though these compression methods achieve high compression ratio, the receiver needs to decode the whole bitstream before displaying the model on the client's screen. As a result, a new family of compression algorithms have been proposed where the model is progressively compressed into a number of levels [Pajarola and Rossignac 2000; Taubin et al. 1998; Hoppe 1998]. The most popular algorithms encode the model into a base mesh and a number of refinement levels. Applying each level produces a mesh that better approximates the original model. If one of these levels is lost, then all succeeding levels are not decodable but all coarser levels are still decodable. Even though these progressive methods are successful in reducing the end-to-end delay of displaying the model on the client's screen, they do not address the channel effect such as losses and delay. In this paper we use the *Compressed Progressive Mesh* (CPM) [Pajarola and Rossignac 2000] to produce M batches in addition to the base mesh. Each batch consists of connectivity and geometry parts. If the connectivity part of a batch is lost during transmission, decoding stops at that batch. Even though CPM is used in this paper, the proposed protocol can employ any progressive compression algorithm.

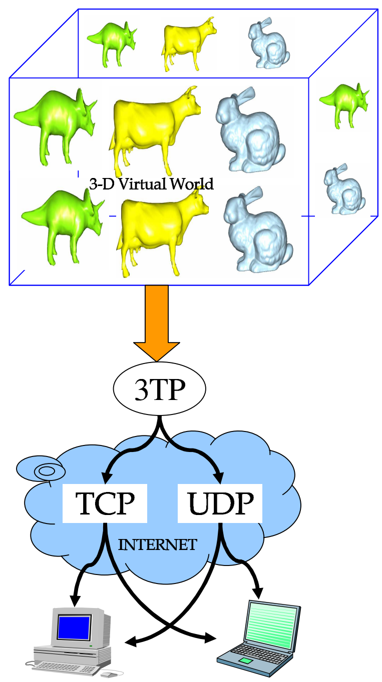


Figure 1: Illustration of the proposed 3-D models transport protocol (3TP).

In a typical 3-D application, the client signs into a virtual world and requests a number of 3-D models with a maximum distortion level to be downloaded. In order to deliver 3-D models with this distortion upper bound, several methods have been proposed in the literature. For example, sender-based methods estimate the losses in the channel and protects the transmitted bitstream by adding redundant bits to be able to recover lost data on the client side [Bajaj et al. 1998; Yan et al. 2001; Al-Regib and Altunbasak 2002; Al-Regib et al. 2002a; Al-Regib et al. 2002b]. These algorithms protect the transmitted bitstream using forward error correction (FEC) codes. A well-known network solution is to retransmit all lost packets until they are all received correctly on the client side. The transport control protocol (TCP) is an example of this approach. If the channel suffers from packet losses, then many TCP packets will be lost. TCP will retransmit all lost packets until they are all received error-free. This, unfortunately, increases the download time. Such technique is not applicable to the *time-sensitive* applications that are the focus of this paper. Finally, Martin in [Martin 2000a] and [Martin 2000b] proposed an adaptive system that segments the model into areas of importance. The choice of the part to stream first depends on the viewer's point of view. Nevertheless, the system is not adaptive to the channel packet-loss rate.

In this paper, we propose a new application-layer protocol that intelligently employs both the transport control protocol (TCP) and the user datagram protocol (UDP) to stream the progressively compressed bitstream of several 3-D models in a virtual scene (Figure 1). The proposed 3-D transport 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: (i) the 3-D models, (ii) the end-to-end channel packet-loss rate, and (iii) the maximum distortion level tolerated by the client. In [Al-Regib and Altunbasak 2003], we *experimentally* decide the partitioning of the bitstream into two parts; one part to be streamed using TCP and the other part is streamed using UDP. In this paper, we *mathematically* decide on this partitioning. We first mathematically model both the delay and the distortion for a given scene and an end-to-end channel. Then, we test all possible partitioning scenarios that satisfy the distortion upper bound. We choose the partitioning scheme that minimizes the expected delay. In contrast, Chen et al. in [Chen et al. 2003] randomly choose the amount of data to be sent using TCP regardless of the channel packet-loss rate.

The paper is organized as follows. We first show the performance of streaming the bitstream using either TCP or UDP in Section 2. Then, we present the proposed 3TP protocol in Section 3. We compare the performance of 3TP with other methods in Section 4 and we conclude the paper in Section 5.

2 Streaming 3-D Models

In this section, we show the high delay caused by streaming 3-D scenes using the transport control protocol (TCP), especially at high packet-loss rates. Similarly, we show the high distortion resulting from using the user datagram protocol (UDP) to stream these models.

2.1 TCP

Since TCP guarantees the delivery of all packets to the receiver error-free by retransmitting lost packets, it is attractive to applications that do not have time constraint. At high channel packet-loss rates, TCP suffers from high delay that 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) [University of California at Berkeley /LBNL/VINT] to run these experiments with the topology shown in Figure 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 Figure 3 for different packet-loss rates. As shown in the plot, the delay increases as the packet-loss rate increases. This is caused by the increase in the number of retransmissions, the TCP slow-start operation-area, and the sharp decrease of the transmission rate when congestion is detected [Floyd and Fall 1999].

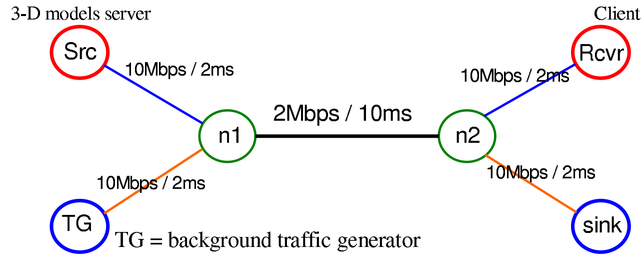


Figure 2: The used network topology.

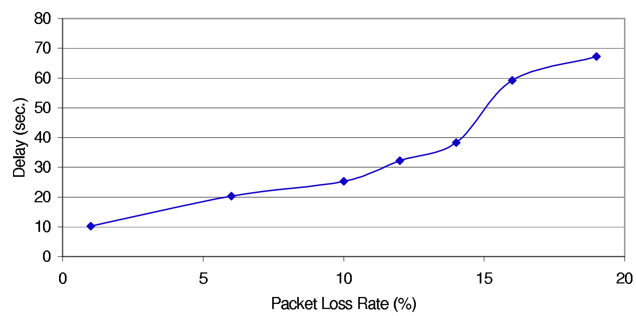


Figure 3: The measured delay when ten SMALL BUNNY models are streamed using TCP at different packet-loss rates.

2.2 UDP

Today, for real-time applications, UDP is preferred in practice over TCP because UDP does not require any feedback from the receiver. On the other hand, UDP does not guarantee the delivery of the transmitted packets to the client. 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 displayed 3-D models on the client side substantially degrades . To illustrate the performance of UDP over different network conditions, we repeated the experiments in Section 2.1 using UDP as the transmission protocol. In order to keep a reasonable quality 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 average distortion between the transmitted models and the received ones on the client side is depicted in Figure 4. In this case, all models are assumed to have the same importance and the reported distortion is the average distortion of the ten SMALL BUNNY models. It is clear from Figure 4 that the distortion constraint cannot be guaranteed when UDP is used to transmit all batches, especially when the packet-loss rate is high. This tradeoff between distortion and delay is addressed by the proposed protocol that is explained in the next section.

3 3TP: 3-D Models Transport Protocol

The proposed 3TP protocol intelligently uses TCP and UDP to deliver the progressively compressed 3-D models in a virtual scene to

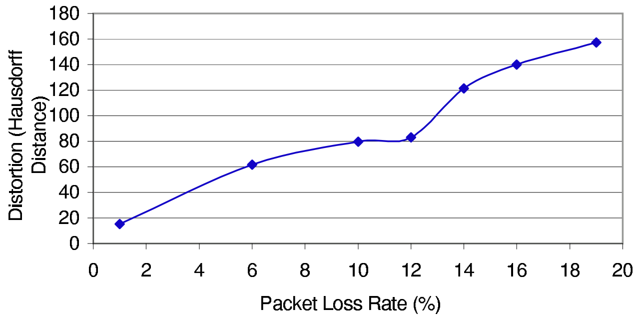


Figure 4: The measured distortion when the base meshes are transmitted using TCP, while all other batches of the ten SMALL BUNNY models are transmitted using UDP. The reported distortion is the average of the ten distortion measures.

the client with an agreed-upon upper distortion bound during the minimum possible time. In 3TP, the minimum delay is achieved by selecting certain parts of the encoded bitstream to be transmitted using TCP, while the remaining part is streamed 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 ($\mathscr{D}_{max}$), 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

$$\begin{aligned} &\min_{\chi^C_{TCP},\chi^G_{TCP}} arg \quad \mathscr{T}(\chi^C_{TCP},\chi^G_{TCP},P_{LR}),\\ &subject\ \ to\ \ \mathscr{D}(\chi^C_{TCP},\chi^G_{TCP},P_{LR}) \le \mathscr{D}_{max},\ \ and \\ &0 \le \chi^C_{TCP},\chi^G_{TCP} \le L, \end{aligned} \tag{1}$$

where $\mathscr{T}$ is the time delay between the request to download the M models and the time when all models are streamed out, $\mathscr{D}$ is the average distortion on the client side, 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 mathematically model the distortion ($\mathscr{D}$) and the delay ($\mathscr{T}$) functions and solve the optimization problem in Equation (1).

Because of the importance of the base mesh, we stream all base meshes at all conditions using TCP. Therefore, having (χ^C_{TCP}, χ^G_{TCP}) $= (0,0)$ corresponds to streaming the base mesh using TCP and all batches using UDP. On the other hand, when (χ^C_{TCP}, χ^G_{TCP}) $= (L,L)$, then all batches as well as the base mesh are transmitted using TCP. For a given channel packet-loss rate higher than zero, it is anticipated that the delay is maximum and the distortion is minimum when (χ^C_{TCP}, χ^G_{TCP}) $= (L,L)$ because all packets are delivered to the client error-free. In contrast, it is anticipated that the delay is minimum and the distortion is maximum when (χ^C_{TCP}, χ^G_{TCP}) $= (0,0)$.

In [Al-Regib and Altunbasak 2003], the authors experimentally solve Equation 1 by running experiments to all possible scenarios and calculating the delay and the distortion at each experiment. This requires long processing time as discussed in [Al-Regib and Altunbasak 2003]. On the other hand, in this paper, we mathematically model the delay and the distortion. Then, we mathematically solve Equation 1.

3.1 Delay Model

To model the delay experienced by sending the bitstream using 3TP, we first need to model the delay for both TCP and UDP connections. We use the TCP throughput model proposed in [Floyd and Fall 1999] to compute the required time to deliver all TCP packets to the client. This latency is given by

$$\mathscr{T}_{TCP} = \frac{B_{TCP} \times RTT \times \sqrt{P_{LR}}}{k \times MSS}\ sec., \tag{2}$$

where $\mathscr{T}_{TCP}$ is the time required to deliver all TCP packets to the client, B_{TCP} is the total number of Bytes transmitted using TCP, RTT is the round-trip-time, P_{LR} is the average packet-loss rate, MSS is the maximum segment size, and k is a constant.

The other source of delay is the time of transmitting all UDP packets. This time depends on the transmission rate and is given by:

$$\mathscr{T}_{UDP} = \frac{B_{UDP}}{R_{UDP}}\ sec., \tag{3}$$

where $\mathscr{T}_{UDP}$ is the time required to transmit all UDP packets, B_{UDP} is the total number of Bytes transmitted using UDP, and R_{UDP} is the transmission rate of UDP stream.

In this paper, we assume that both RTT and P_{LR} do not vary during the streaming time for an end-to-end channel. Similarly, we assume that k, MSS, and R_{UDP} are constants. On the other hand, both B_{TCP} and B_{UDP} are functions of $(\chi^C_{TCP},\chi^G_{TCP})$. The total delay time ($\mathscr{T}$) is given by: $\mathscr{T} = \mathscr{T}_{TCP} + \mathscr{T}_{UDP}$.

Figure 5 depicts the total delay, computed using Equations (2) and (3), of streaming ten SMALL BUNNY models over different channels. In these calculations[1], $RTT = 28$ msec., $MSS = 256$ Bytes, and $k = 1.22$.

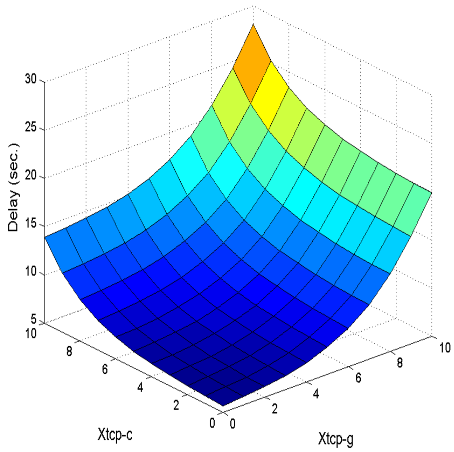
Next, we derive a mathematical model that estimates the average distortion introduced to the scene that is displayed on the client's screen for a given end-to-end channel.

3.2 Distortion Model

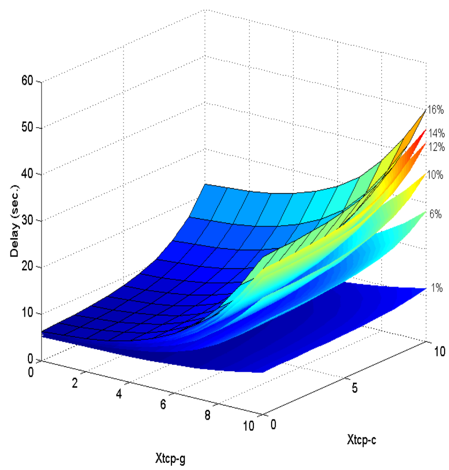
In order to estimate the distortion associated with the 3TP stream, the distortion for both TCP and UDP sub-streams is estimated. Because TCP delivers all packets to the client, the distortion associated with TCP is zero. Therefore, the main source of distortion is the dropped UDP packets. For example, when all levels of all models using TCP are transmitted (i.e., $(\chi^C_{TCP},\chi^G_{TCP}) = (L,L)$), the source of distortion is quantization, $\mathscr{D} = E_Q$, where E_Q is the quantization error. On the other hand, when $(\chi^C_{TCP},\chi^G_{TCP}) = (0,0)$, the expected distortion is a function of the packet-loss rate.

To derive the distortion model, we investigate three cases: $\chi^C_{TCP} = \chi^G_{TCP}$, $\chi^C_{TCP} < \chi^G_{TCP}$, and $\chi^C_{TCP} > \chi^G_{TCP}$. First, we illustrate the derivation of the distortion equation when $\chi^C_{TCP} = \chi^G_{TCP}$ using a specific example, then we give the general equation. Let the total number of batches to be 10 ($L = 10$) and let $(\chi^C_{TCP},\chi^G_{TCP}) = (8,8)$, then we will have two batches to transmit using UDP. Each batch consists of connectivity and geometry parts. Some of these parts might be lost during transmission and therefore we will have a total of seven scenarios as illustrated in Figure 6(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 Figure 6(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. Similarly, the expected distortion when scenario (5) in

[1] This choice of k is recommended in [Floyd and Fall 1999]



(a) P_{LR} = 6%



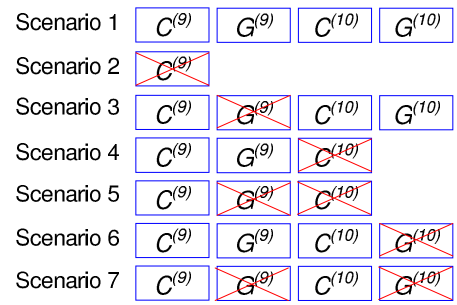
(b) P_{LR} = 1, 6, 10, 12, 14, 16, 19%

Figure 5: Theoretical delay ($\mathscr{T}$) of streaming ten SMALL BUNNY models on different channels with different packet-loss rates (P_{LR}).

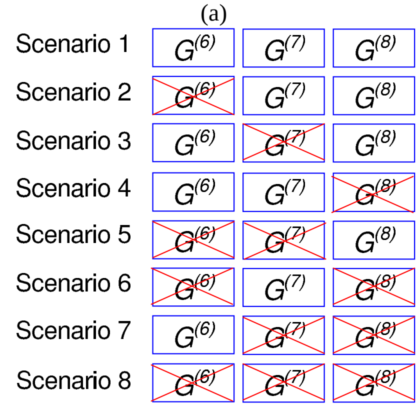
Figure 6(a) occurs is written as $E^{(9,10)}_{G,C}(1-P^{(9)}_C)P^{(9)}_G P^{(10)}_C$, where $E^{(9,10)}_{G,C}$ is the distortion resulting from losing the geometry part of batch (9) as well as the connectivity part of batch (10). Combining all these scenarios, the expected distortion becomes as follows (For simplicity, in the rest of the paper *TCP* is dropped from χ^C_{TCP} and χ^G_{TCP} to be χ^C and χ^G, respectively.)

$$\begin{array}{l}\mathscr{D}(\chi^C=\chi^G)= \\ E_Q(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2})(1-P_G^{\chi^G+1})(1-P_G^{\chi^G+2}) \\ +E_C^{\chi^C+1}P_C^{\chi^C+1}+E_C^{\chi^C+2}P_C^{\chi^C+2}(1-P_C^{\chi^C+1})(1-P_G^{\chi^G+1}) \\ +E_{C,G}^{(\chi^C+2,\chi^G+1)}P_C^{\chi^C+2}P_G^{\chi^G+1}(1-P_C^{\chi^C+1}) \\ +E_G^{\chi^G+1}P_G^{\chi^G+1}(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2})(1-P_G^{\chi^G+2}) \\ +E_G^{\chi^G+2}P_G^{\chi^G+2}(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2})(1-P_G^{\chi^G+1}) \\ +E_{G,G}^{\chi^G+1,\chi^G+2}P_G^{\chi^G+1}P_G^{\chi^G+2}(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2})\end{array}, \quad (4)$$

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



(a)



(b)

Figure 6: (a) In this illustration, the total number of batches is 10 and $(\chi^C_{TCP}, \chi^G_{TCP}) = (8,8)$, i.e., both the connectivity and the geometry parts of the first 8 levels are sent using TCP while the connectivity and the geometry parts of levels 9 and 10 are sent using UDP. (b) In this illustration, the total number of batches is 10 and $(\chi^C_{TCP}, \chi^G_{TCP}) = (8,5)$, i.e., the connectivity of the first 8 levels and the geometry of the first 5 levels are sent using TCP, while the connectivity of levels 9 and 10 and the geometry of levels 6-10 are sent using UDP. The crossed levels in these illustrations imply that part or the bitstream associated with this level is lost during transmission.

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

$$E^{(i,j)}_{C,G} \approx E^i_C \quad and \quad E^{(i,j,\cdots,k)}_{C,G,\cdots,G} \approx E^i_C, \qquad (5)$$

where $E^{(i,j,\cdots,k)}_{C,G,\cdots,G}$ is the distortion resulting from losing the connectivity part of batch (i) as well as the geometry parts of batches (j to k). The validity of this assumption is based on the fact that within a batch connectivity is more important than geometry [Al-Regib and Altunbasak 2003].

To further simplify the expression in Equation (4), 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.,

$$E_{G,G}^{(i,j)} \approx E_G^i + E_G^j \quad and \quad E_{G,G,\cdots,G}^{(i,j,\cdots,k)} \approx E_G^i + \cdots + E_G^k, \qquad (6)$$

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 Equations (5) and (6) into Equation (4), the expected distortion is simplified to

$$\begin{aligned}&\mathscr{D}(\chi^C = \chi^G) = \\ &E_Q(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2})(1-P_G^{\chi^G+1})(1-P_G^{\chi^G+2}) \\ &+E_C^{\chi^C+1}P_C^{\chi^C+1}+E_C^{\chi^C+2}P_C^{\chi^C+2}(1-P_C^{\chi^C+1}) \\ &+E_G^{\chi^G+1}P_G^{\chi^G+1}(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2}) \\ &+E_G^{\chi^G+2}P_G^{\chi^G+2}(1-P_C^{\chi^C+1})(1-P_C^{\chi^C+2}).\end{aligned} \qquad (7)$$

Equation (7) can be generalized for any number of batches, L, where $\chi^C (= \chi^G)$ levels are transmitted using TCP as

$$\begin{aligned}&\mathscr{D}(\chi^C = \chi^G) = \\ &E_Q \prod_{j=\chi^C+1}^{L}(1-P_C^j) \prod_{j=\chi^G+1}^{L}(1-P_G^j) \\ &+E_C^{\chi^C+1}P_C^{\chi^C+1} + \sum_{i=\chi^C+2}^{L} E_C^i P_C^i \prod_{j=\chi^C+1}^{i-1}(1-P_C^j) \\ &+ \sum_{i=\chi^G+1}^{L} E_G^i P_G^i \prod_{j=\chi^C+1}^{i}(1-P_C^j).\end{aligned} \qquad (8)$$

The second and the third terms are associated with the distortion resulting from losing connectivity levels. We will denote these two terms as

$$\mathscr{D}' = E_C^{\chi^C+1}P_C^{\chi^C+1} + \sum_{i=\chi^C+2}^{L} E_C^i P_C^i \prod_{j=\chi^C+1}^{i-1}(1-P_C^j) \qquad (9)$$

The distortion when $\chi^C < \chi^G$ can be derived similar to the above case when $\chi^C = \chi^G$. In the former case, more connectivity levels are transmitted using UDP and only $\mathscr{D}'$ is affected. The resulting distortion is the same as the one given in Equation (8). Therefore, we get

$$\begin{aligned}&\mathscr{D}(\chi^C \le \chi^G) = \\ &E_Q \prod_{j=\chi^C+1}^{L}(1-P_C^j) \prod_{j=\chi^G+1}^{L}(1-P_G^j) + \mathscr{D}' \\ &+ \sum_{i=\chi^G+1}^{L} E_G^i P_G^i \prod_{j=\chi^C+1}^{i}(1-P_C^j)\end{aligned} \qquad (10)$$

Now, we derive the distortion when $\chi^C > \chi^G$. For simplicity, we derive the equations for a specific example and then we give the general formula. Assume that $L = 10$, $\chi^C = 8$, and $\chi^G = 5$. When $5 \le \chi^G \le 7$, there will be eight possible scenarios as shown in Figure 6(b). Each scenario is combined with the seven scenarios shown in Figure 6(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 is the sum of the distortions resulting from losing these geometry levels individually as given by Equation (6). By incorporating this assumption and simplifying the resulting expression, we get

$$\begin{aligned}&\mathscr{D}(\chi^C > \chi^G) = \\ &E_Q \prod_{j=\chi^C+1}^{L}(1-P_C^j) \prod_{j=\chi^G+1}^{L}(1-P_G^j) \\ &+ \sum_{i=\chi^G+1}^{\chi^C} E_G^i P_G^i \sum_{i=\chi^C+1}^{L} P_C^i \prod_{i=\chi^C+1}^{L}(1-P_C^i) \\ &+E_C^{\chi^C+1}P_C^{\chi^C+1} + \sum_{i=\chi^C+2}^{L} E_C^i P_C^i \prod_{j=\chi^C+1}^{i-1}(1-P_C^j)\end{aligned} \quad . \qquad (11)$$

In summary, the expected average distortion on the client side is given by

$$\mathscr{D} = \begin{cases} E_Q' + D' + \sum_{i=\chi^G+1}^{L} E_G^{(i)} P_G^i \prod_{j=\chi^C+1}^{i}(1-P_C^j), & \chi^C \le \chi^G \\ E_Q' + \sum_{i=\chi^G+1}^{\chi^C} E_G^i P_G^{(i)} \sum_{i=\chi^C+1}^{L} P_C^i & \\ \qquad \times \prod_{i=\chi^C+1}^{L}(1-P_C^i) + D', & \chi^C > \chi^G \end{cases}, \qquad (12)$$

where $\mathscr{D}'$ is given in Equation (9) and

$$E_Q' = E_Q \prod_{j=\chi^C+1}^{L}(1-P_C^j) \prod_{j=\chi^G+1}^{L}(1-P_G^j). \qquad (13)$$

To find the distortion, $\mathscr{D}$, that is given in Equation (12), 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

$$P^{(j)} = \sum_{l=1}^{K^j} p(l,n) \qquad (14)$$

where K^j is the number of packets this part (connectivity/geoemtry) 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 shown in Figure 7 to model the end-to-end channel between the server and the client. More details on evaluating $p(l,n)$ can be found in [Horn et al. 1999]. 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.

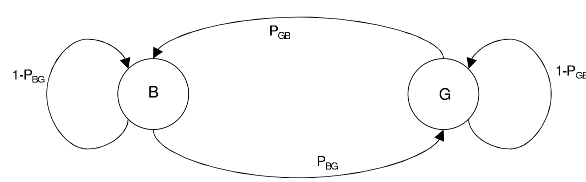
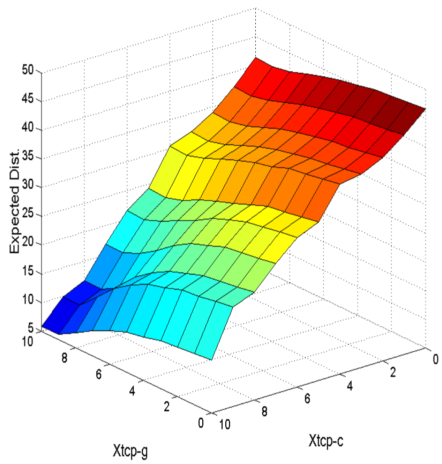
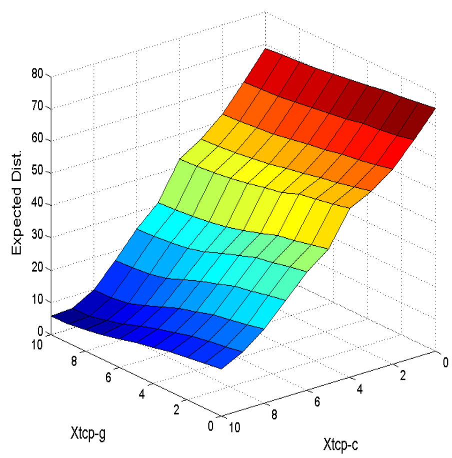


Figure 7: The Gilbert-Elliot end-to-end channel model.

We evaluated the average expected distortion ($\mathscr{D}$) for different packet-loss rates (P_{LR}) when ten SMALL BUNNY models are streamed with $(0,0) \leq (\chi^C_{TCP}, \chi^G_{TCP}) \leq (10,10)$. Figures 8(a) and (b) show the expected distortion ($\mathscr{D}$) when the packet-loss rate is 6% and 19%, respectively.



(a) P_{LR} = 6%



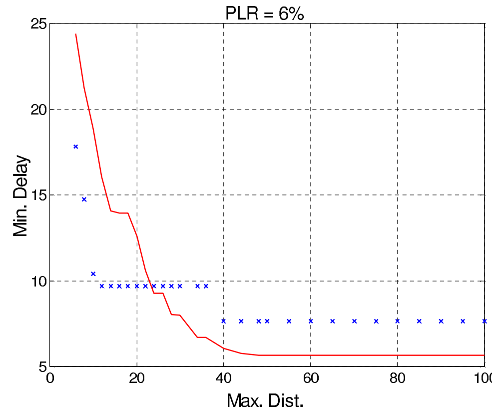
(b) P_{LR} = 14%

Figure 8: The theoretical average distortion ($\mathscr{D}$) of the ten SMALL BUNNY models displayed on the client's screen and streamed over different channels with packet-loss rates (P_{LR}) of (a) 6% and (b) 14%. This distortion is calculated using Equation (12).

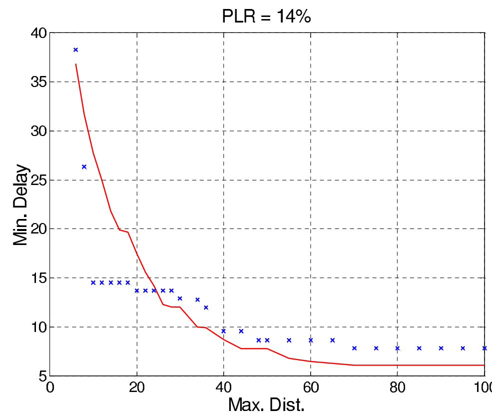
Figure 8 depicts the expected distortion when part of the bitstream is transmitted using UDP. These theoretical plots do not have the statistical problem the experimental plots are experiencing where the plot is not smooth [Al-Regib and Altunbasak 2003]. Finally, the plots in Figure 8 require less processing time than those obtained experimentally [Al-Regib and Altunbasak 2003]. In the next section, we show the performance of the theoretical solution.

4 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 needed to be transmitted to the client. Therefore, we will compare the performance of the proposed 3TP protocol with a method that we call TCP-All where



(a)



(b)

Figure 9: The minimum achievable delay using 3TP for different maximum distortion levels at different packet-loss rates. The straight lines are the theoretical results while the starred points are obtained experimentally.

both connectivity and geometry parts of all levels that just guarantee the distortion constraint are streamed using TCP. For example, if $\mathscr{D}_{max} = 30$, then from the R-D curve (Figure10) of the SMALL BUNNY model, the base mesh, the connectivity, and the geometry data of the first nine batches are transmitted using TCP only but if $\mathscr{D}_{max} = 60$, then the base mesh and six batches are transmitted using TCP only[2]. In the 3TP case, we use the $(\chi^C_{TCP}, \chi^G_{TCP})$ pairs that have been mathematically found in Section 3.

Figures 9(a) and (b) compare the experimental and the theoretical solutions for two packet-loss rates[3]. It is noticed that at low $\mathscr{D}_{max}$, the experimental solution gives lower delay than the corresponding theoretical solution. This is caused by the insufficient number of iterations when we solved the problem experimentally in [Al-Regib and Altunbasak 2003]. Nevertheless, for $\mathscr{D}_{max} > 30$, both methods give similar delay. More specifically, the delay computed theoretically is smaller than the one found experimentally. As we repeat the experiments thousands of time, we anticipate that the two curves will behave identically.

Figure 11 depicts the theoretical comparison between 3TP and TCP-All. When $P_{LR} = 1\%$ and $\mathscr{D}_{max} = 30$, 3TP saves 39% of the

[2] This distortion is measured using the *Hausdorff* distance.

[3] Such information is obtained using network techniques that are beyond the scope of this paper. Real-Time Control Protocol (RTCP)[Wang et al. 2002], for example, provides sender and receiver reports that contain long- and short-term packet-loss rates.

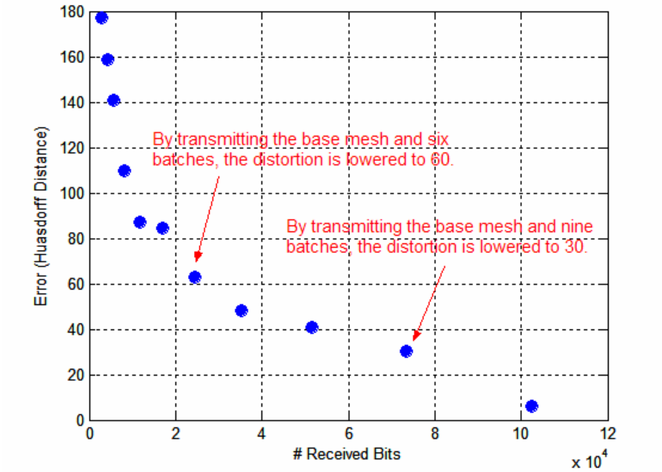


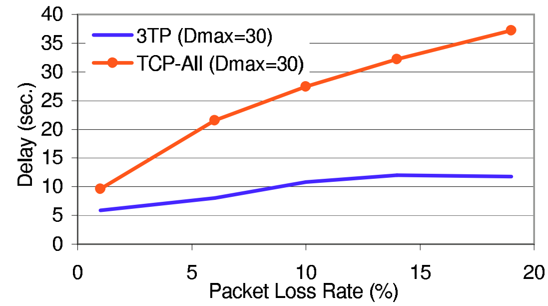
Figure 10: The rate-distortion (R-D) curve of the SMALL BUNNY model when progressively compressed into a base mesh and ten batches.

delay time compared to TCP-All but when $P_{LR} = 14\%$, 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.

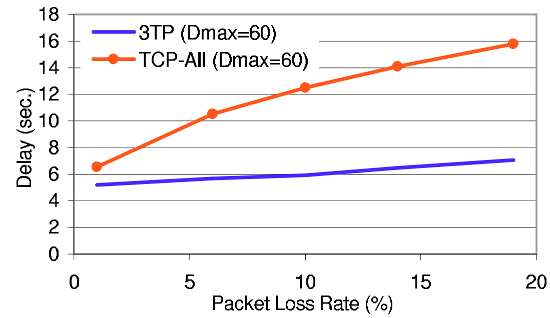
5 Conclusion

In this paper, we proposed an application-layer protocol for streaming 3-D models over lossy channels. The proposed protocol combines source and channel characteristics to minimize the end-to-end delay of streaming the bits that guarantee a distortion constraint. For every channel and a distortion constraint, we mathematically choose the number of connectivity and geometry levels $(\chi^C_{TCP}, \chi^G_{TCP})$ to be transmitted using TCP, while the remaining levels are transmitted using UDP. We first mathematically model the expected delay and the expected distortion for a given 3-D scene and a given end-to-end channel. Then, we solve the optimization problem mathematically for all partitioning scenarios. The proposed protocol outperforms the alternative method of using TCP for all connectivity and geometry levels that satisfy the distortion constraint. The process used to obtain the solution of Equation (1) is performed off line. Then, a table that lists the $(\chi^C_{TCP}, \chi^G_{TCP})$ pairs for every $\mathscr{D}_{max}$ and P_{LR} combinations is stored, on the server, together with the bitstream of the progressively compressed 3-D models. When a client requests the models, the server chooses the $(\chi^C_{TCP}, \chi^G_{TCP})$ pair, from the stored table, that minimizes the end-to-end delay for the given channel packet loss rate and starts streaming the bitstream accordingly.

The proposed processing is performed off-line and the results can be stored in a look-up table along with the progressively-encoded bitstream. This table would have an entry for a list of channel packet-loss rates and delays. For every entry, the table stores the corresponding batches pair that minimizes the delay and maximizes the quality. When a client requests a 3D model, we can measure the channel end-to-end delay and packet-loss rates, then fetch the corresponding parameters from the look-up table, and stream out the bitstream using TCP/UDP according to the table.



(a)$\mathscr{D}_{max} = 30$



(b)$\mathscr{D}_{max} = 60$

Figure 11: Comparison between the performance of 3TP and the method of streaming all those levels, which satisfy the maximum distortion constraint, using TCP (TCP-All). Two different maximum distortion levels, $\mathscr{D}_{max}$ are shown: (a) $\mathscr{D}_{max} = 30$, and (b) $\mathscr{D}_{max} = 60$.

References

AL-REGIB, G., AND ALTUNBASAK, Y. 2002. An unequal error protection method for packet loss resilient 3-D mesh transmission. In *Proceedings of INFOCOM*, vol. 2, 743–752.

AL-REGIB, G., AND ALTUNBASAK, Y. 2003. 3TP: An application-layer protocol for streaming 3-D graphics. In *The Proceedings of the IEEE Int. Conf. on Multimedia and Expo*, vol. 1, 421–424.

AL-REGIB, G., ALTUNBASAK, Y., AND ROSSIGNAC, J. 2002. A joint source and channel coding approach for progressively compressed 3-D mesh transmission. In *The Proceedings of the IEEE Int. Conf. in Image Processing*, vol. 2, 161–164.

AL-REGIB, G., ALTUNBASAK, Y., ROSSIGNAC, J., AND MERSEREAU, R. 2002. Protocol for streaming compressed 3-D animations over lossy channels. In *The Proceedings of the IEEE Int. Conf. on Multimedia and Expo*, vol. 1, 353–356.

BAJAJ, C. L., CUTCHIN, S., PASCUCCI, V., AND ZHUANG, G. 1998. Error resilient transmission of compressed VRML. In *Technical report, TICAM, The University of Texas at Austin, 1998.*

BAJAJ, C. L., PASCUCCI, V., AND ZHUANG, G. 1999. Progressive compression and transmission of arbitrary triangular meshes. In *Proceedings of the IEEE Conference on Visualization '99*, 307–316.

CHEN, Z., BODENHEIMER, B., AND BARNES, J. F. 2003. Robust transmission of 3D geometry over lossy networks. In *Proceed-*

ings of the 8th International Conference on 3D Web Technology, ACM, 161–208.

FLOYD, S., AND FALL, K. 1999. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Transactions on Networking 7*, 4 (August), 458–472.

HOPPE, H. 1998. Efficient implementation of progressive meshes. *Microsoft Research Technical Report (MSR-TR-98-2)* (January).

HORN, U., STUHLMULLER, K., LINK, M., AND GIROD, B. 1999. Robust internet video transmission based on scalable coding and unequal error protection. *Signal Processing: Image Communications 15*, 77–94.

MARTIN, I. M. 2000. Adaptive rendering of 3D models over networks using multiple modalities. *Technical Report, IBM T. J. Watson Research Center RC 21722*, 97821 (April).

MARTIN, I. M. 2000. Arte - adaptive rendering and transmission environment. In *Proceedings of ACM Multimedia*.

PAJAROLA, R., AND ROSSIGNAC, J. 2000. Compressed progressive meshes. *IEEE Transactions on Visualization and Computer Graphics 6*, 1 (January-March), 79–93.

ROSSIGNAC, J. 1999. Edgebreaker: connectivity compression for triangle meshes. *IEEE transactions on visualization and computer graphics 5*, 1 (January), 47–61.

TAUBIN, G., AND ROSSIGNAC, J. 1998. Geometric compression through topological surgery. *ACM Transactions on Graphics 17*, 2 (April), 84–115.

TAUBIN, G., GUEZIEC, A., HORN, W., AND LAZARUS, F. 1998. Progressive forest split compression. In *Proceedings of SIGGRAPH'98*, 123–132.

UNIVERSITY OF CALIFORNIA AT BERKELEY /LBNL/VINT. *network simulator ns (version 2)*. Available at http://www.isi.edu/nsnam/ns/.

WANG, Y., OSTERMANN, J., AND ZHANG, Y.-Q. 2002. *Video processing and communications*. Prentice Hall, NJ.

YAN, Z., KUMAR, S., LI, J., AND KUO", C.-C. 1999. Error resilient coding of 3D graphics based on morphing and volume splitting. In *SPIE Conference on Multimedia Systems and Applications II*, vol. 3845, 222–23.

YAN, Z., KUMAR, S., AND KUO, C.-C. J. 2001. Error resilient coding of 3-D graphic models via adaptive mesh segmentation. *IEEE transactions on circuits and systems for video technology 11*, 7 (July).