

# 3TP: AN APPLICATION-LAYER PROTOCOL FOR STREAMING 3-D GRAPHICS

Ghassan Al-Regib and Yucel Altunbasak

Center for Signal and Image Processing  
Georgia Institute of Technology  
Atlanta, GA 30332-0250  
E-mail: {gregib,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 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). Experimental results show savings of 42% and 68% in delay time at packet-loss rates of 6% and 19%, respectively, compared to systems that do not optimize transmission according to the encoded bitstream content.

## 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. Distributed gaming, which is the most well known example of these applications, has transmission requirements that are significantly different from traditional data applications. Most importantly, data has to be correctly delivered within certain time frame in order to preserve real-time interactivity among participants. Nevertheless, the Internet imposes two major limitations on smooth interactivity within a distributed gaming environment. First, the limited-bandwidth links cause latency that prohibits smooth interactivity among participants. Second, the losses caused by congestion introduces distortion that significantly affects the game quality.

In this paper, we consider the scenario when a client signs into a virtual world and requests to download a number of 3-D models with a maximum distortion level<sup>1</sup>. We envision the proposed protocol to be a basis for a more general transport protocol that addresses more distributed gaming scenarios where scene states are exchanged between participants [1].

The effect of both limited-bandwidth links and packet losses elevates considerably when dynamically created content or complex 3-D models are used. To alleviate such limitations, single-level compression methods [2] can be employed to reduce the end-to-end transmission delay. The disadvantage of these methods is that nothing can be displayed on the client's screen before the entire bitstream is downloaded. A more efficient compression strategy aims to reduce the display latency by sending a coarse mesh of the 3-D model first, and then transmitting refinement information later. For example, the *Compressed Progressive Mesh* (CPM) [3] algorithm produces  $M$  batches in addition to the base mesh. Application of each batch yields a different level of detail that further

approximates the original 3-D model. Each batch consists of a connectivity part and a geometry part. If the connectivity part of a batch is lost during transmission, decoding stops at that batch. We use CPM in this paper to generate a progressive bitstream of the 3-D models. Nevertheless, the proposed protocol can employ any other progressive compression algorithm. 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. Such losses have catastrophic effect on the game quality.

In order to deliver 3-D models with a distortion upper bound, several methods have been proposed in the literature. For example, sender-based methods estimate the losses in the channel and protect the transmitted bitstream by adding redundant bits to be able to recover lost data on the client side. Such error-resilient transmission methods have been proposed by Bajaj et al. in [4], Yan et al. in [5], and Al-Regib et al. in [6] and [7]. 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. Other class of network-based methods considers the bitstream as a stream of data without a priori knowledge of the content. 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), which is the most commonly used transport protocol on the Internet, is an example of this approach. Because of high delay caused by retransmission, such techniques are not applicable for the *time-sensitive* distributed gaming applications that are the focus of this paper.

The proposed 3-D models transport protocol (3TP) consists of three major parts. The first part addresses the problem of downloading 3-D models when a client signs into a game. In this scenario, the user specifies the upper bound on the distortion and the 3TP protocol minimizes the end-to-end delay required to download these models. The second part addresses the problem of streaming actions and states of all participants. A streaming framework for this part has been proposed in [6]. The third part of 3TP is a multicast protocol that addresses efficient streaming techniques for large number of participants. In this paper, we focus on the first part where we propose a streaming protocol for static 3-D models.

The proposed protocol 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. Since connectivity and geometry data affect the quality of the decoded model differently, the proposed 3-D models 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 and the sizes of connectivity

<sup>1</sup>This work was supported in part by NSF award CCR-0133221.

<sup>1</sup>Alternatively, the server might calculate the maximum distortion level using algorithms that are beyond the scope of this paper.

and geometry bitstreams, (ii) the end-to-end channel packet-loss rate, and (iii) the maximum distortion level tolerated by the client.

## 2. STREAMING 3-D MODELS USING TCP

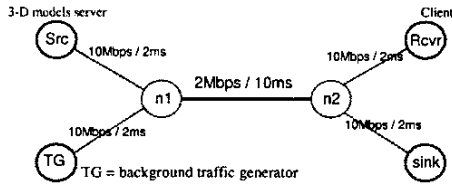


Fig. 1. The used network topology.

The congestion control algorithm of all variations of TCP employs an additive increase multiple decrease mechanism that controls the transmission rate [8]. When no acknowledgment is received from the receiver, the sender assumes that congestion is taking place and it sharply reduces the transmission rate. Since TCP guarantees the delivery of all packets to the receiver error-free, it is attractive for applications that do not have time constraint. On the other hand, 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 such as distributed gaming. 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) [9] to run these experiments with the topology shown in Figure 1. 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 2 for different packet-loss rates. As shown, the delay is a function of the packet-loss rate. The increase in delay, as the packet loss increases, is caused by the increase in the number of retransmissions, the slow-start operation-area in TCP, and the sharp decrease of the transmission rate when congestion is detected [8]. An alternative transmission method is to use UDP where no congestion control is employed as illustrated in the following section.

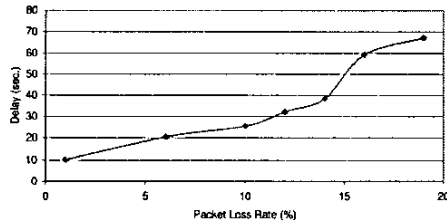


Fig. 2. The measured delay when ten SMALL BUNNY models are streamed using TCP at different packet-loss rates.

## 3. STREAMING 3-D MODELS USING UDP

Today, for real-time applications, UDP is preferred in practice upon TCP because the former 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. 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 2 using UDP as the transmission protocol instead of TCP for the refinements information. 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 3. 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 3 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 in the next section.

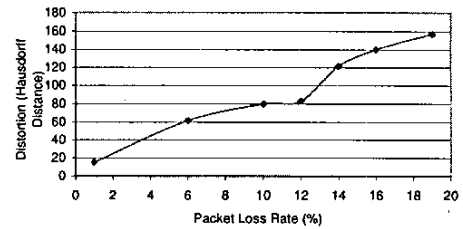


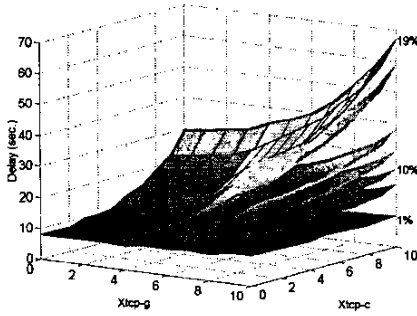
Fig. 3. The average distortion (Hausdorff distance) when the base meshes are transmitted using TCP, while all other batches of the ten SMALL BUNNY models are transmitted using UDP.

## 4. 3TP: AN APPLICATION-LAYER 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 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 ( $\mathcal{D}_{max}$ ), and (iii) the channel packet-loss rate ( $PLR$ ); determine the connectivity ( $\chi_{TCP}^C$ ) and the geometry ( $\chi_{TCP}^G$ ) parts of the encoded levels to be transmitted using TCP in order to minimize the delay ( $\mathcal{T}$ ), while the remaining parts will be transmitted using UDP". The delay ( $\mathcal{T}$ ) is the time difference between the request to download the  $M$  models and the time when all models are streamed out. In this paper, we experimentally evaluate both the delay ( $\mathcal{T}$ ) and the distortion ( $\mathcal{D}$ ).

Because of the importance of the base mesh, we stream all base meshes at all conditions using TCP. Therefore, having  $(\chi_{TCP}^C, \chi_{TCP}^G) = (0, 0)$  corresponds to streaming the base mesh using TCP and all batches using UDP. On the other hand, when  $(\chi_{TCP}^C, \chi_{TCP}^G) = (L, L)$ , 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 dis-



$P_{LR} = 1, 6, 10, 12, 14, 16, 19\%$

**Fig. 4.** Total delay ( $T$ ) of streaming ten SMALL BUNNY models on different channels with different packet-loss rates ( $P_{LR}$ ).

distortion is minimum when  $(\chi_{TCP}^C, \chi_{TCP}^G) = (L, L)$  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  $(\chi_{TCP}^C, \chi_{TCP}^G) = (0, 0)$ . Next, we experimentally demonstrate the effect of  $(\chi_{TCP}^C, \chi_{TCP}^G)$  on both the delay and the distortion. In these experiments, we used the topology shown in Figure 1. We first transmit the TCP packets followed by the UDP packets. In order to have both streams experiencing the same packet-loss rate, we adjust the UDP transmission rate in ns-2 [9].

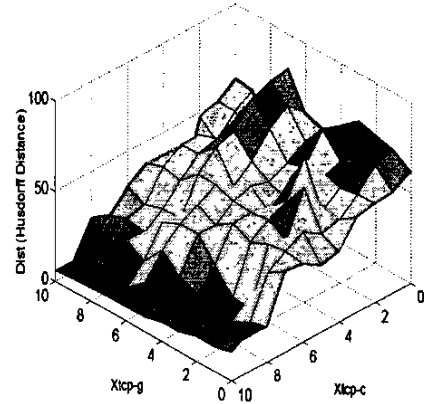
#### 4.1. Delay

In the proposed 3TP protocol, the main source of delay is the flow/congestion control algorithm of TCP. As more layers are transmitted using TCP, higher delay is experienced. In addition, when the channel is congested, more retransmissions are required by TCP and as a result delay increases. This is illustrated in Figure 4 where we stream ten SMALL BUNNY models and each model is compressed into ten batches in addition to the base mesh. The effect of the channel packet-loss rate on the delay is highest when all batches are transmitted using TCP, i.e.,  $(\chi_{TCP}^C, \chi_{TCP}^G) = (L, L)$ . Each point in these curves has a certain distortion associated with it as will be shown in the following section.

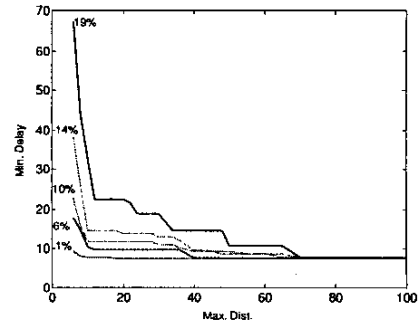
#### 4.2. Distortion

It is shown in Figure 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 Figure 5 for packet-loss rate ( $P_{LR}$ ) of 6%. As shown in this plot, the distortion is minimum when  $(\chi_{TCP}^C, \chi_{TCP}^G) = (L, L)$  since all levels are streamed using TCP and therefore they are all correctly received at the client.

This and similar plots reflect the fact that connectivity is more important than geometry because  $\mathcal{D}(\chi_{TCP}^C = 10, \chi_{TCP}^G = 0) < \mathcal{D}(\chi_{TCP}^C = 0, \chi_{TCP}^G = 10)$ . More specifically, the improvement in the quality that results by adding one more connectivity level to be transmitted using TCP is larger than the improvement in the quality when, instead, one more geometry level is transmitted using TCP. This can be seen from Figure 5 by looking at the cases:  $(\chi_{TCP}^C = 10, \chi_{TCP}^G = 0)$ ,  $(\chi_{TCP}^C = 0, \chi_{TCP}^G = 10)$ , and  $(\chi_{TCP}^C = 10, \chi_{TCP}^G = 10)$ . 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.,  $(\chi_{TCP}^C, \chi_{TCP}^G)$ , while the remaining data is transmitted using UDP.



**Fig. 5.** The average distortion ( $\mathcal{D}$ ) of ten SMALL BUNNY models displayed on the client's screen and streamed over a channel with packet-loss rate ( $P_{LR}$ ) of 6%.



**Fig. 6.** Minimum delay (found using 3TP) vs. maximum distortion at different packet-loss rates.  $P_{LR} = 1, 6, 10, 14,$  and  $19\%$ .

#### 4.3. Delay vs. Distortion

To find the tradeoff between delay and distortion for a given channel and a set of  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pairs, we follow the following simple procedure: (i) obtain the packet-loss rate ( $P_{LR}$ ), the maximum distortion ( $\mathcal{D}_{max}$ ), the number of models ( $M$ ), and the number of progressive levels each model is compressed into ( $L$ ), (ii) for each  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pair, where  $0 \leq \chi_{TCP}^C, \chi_{TCP}^G \leq L$ , stream the  $M$  models over the channel using the topology shown in Figure 1, (iii) for each  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pair, record the delay as well as the average distortion of the displayed  $M$  models on the client's screen, and (iv) choose the  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pair that minimizes the delay, while the average distortion is equal or less than  $\mathcal{D}_{max}$ .

By following the above experimental procedure, we can find all  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pairs that guarantee the delay  $T$  and the distortion  $\mathcal{D}$  constraints. For example, no  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pair can deliver the ten SMALL BUNNY models for  $\mathcal{D}_{max} \leq 6$  and  $T < 10$  seconds at  $P_{LR}$  of 6% (Figure 5). When the delay upper bound is increased to 10, two  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pairs can meet these constraints when  $P_{LR} = 6\%$ . In contrast, these constraints can still not be met when  $P_{LR} = 14\%$  because more losses take place and more layers are transmitted using TCP to satisfy the distortion constraint. This results in more lost TCP packets and con-

sequently more retransmissions, which leads to higher delay. On the other hand, when fewer levels are transmitted using TCP, then more packets will be lost and the distortion constraint cannot be satisfied. Therefore, such delay and distortion constraints cannot be met when  $P_{LR} = 14\%$ . As the upper bounds on distortion and delay are increased, more  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pairs can meet these constraints.

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 the problem we address in this paper. Figure 6 depicts the minimum achievable delay for a given upper bound on the distortion ( $D_{max}$ ) for different packet-loss rates. Each of these points has a corresponding  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pair. For example, when  $D_{max} = 6$ , and  $P_{LR} = 1\%$ , the minimum achievable delay is 9.22 seconds when 8 connectivity levels and 10 geometry levels in addition to the base mesh are transmitted using TCP, while the remaining levels are transmitted using UDP. When the maximum acceptable distortion is increased to 20, the minimum delay is 7.52 seconds with  $(\chi_{TCP}^C, \chi_{TCP}^G) = (5, 1)$ . In this case, more distortion is tolerable and, therefore, fewer levels are transmitted using TCP.

#### 4.4. 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 both connectivity and geometry parts of all levels that just guarantee the distortion constraint are streamed using TCP. For example, if  $D_{max} = 30^2$ , then from the R-D curve 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  $D_{max} = 60$ , then the base mesh and seven batches are transmitted using TCP only. In the 3TP case, we use the  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pairs that have been found in Section 4.3.

Figure 7 depicts a comparison between 3TP and TCP-All. When  $P_{LR} = 1\%$  and  $D_{max} = 30$ , 3TP saves 22% of the delay time compared to TCP-All but when the packet-loss rate is increased to 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.

#### 5. CONCLUSIONS

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 given channel and distortion constraints, we choose the number of connectivity and geometry levels  $(\chi_{TCP}^C, \chi_{TCP}^G)$  to be transmitted using TCP, while the remaining levels are transmitted using UDP. 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 the addressed problem is performed off line. Then, a table that lists the  $(\chi_{TCP}^C, \chi_{TCP}^G)$  pairs for every  $D_{max}$  and  $P_{LR}$  possible combination is stored, on the server, together with the bitstream of the progressively compressed 3-D models. When a client

<sup>2</sup>This distortion is measured using the Hausdorff distance.

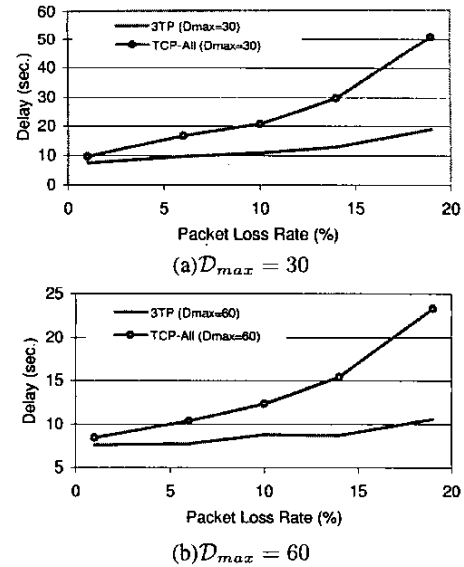


Fig. 7. Comparison between the performance of 3TP and the method of streaming all those levels, which satisfy the maximum distortion constraint, using TCP (TCP-All). Two different maximum distortion levels ( $D_{max}$ ) are shown.

requests the models, the server chooses the  $(\chi_{TCP}^C, \chi_{TCP}^G)$  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.

#### 6. REFERENCES

- [1] L. Gautier, C. Diot, and J. Kurose, "End-to-end transmission control mechanisms for multiparty interactive applications in the Internet," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, New York, NY, March 1999, vol. 3, pp. 1470-1479.
- [2] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Transactions on Graphics*, vol. 17, no. 2, pp. 84-115, April 1998.
- [3] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, no. 1, pp. 79-93, January-March 2000.
- [4] C. L. Bajaj, S. Cutchin, V. Pascucci, and G. Zhuang, "Error resilient transmission of compressed VRML," in *Technical report, TICAM, The University of Texas at Austin, 1998*, Austin, TX, 1998.
- [5] Z. Yan, S. Kumar, and C.-C. J. 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, July 2001.
- [6] Ghassan Al-Regib, Yucel Altunbasak, Jarek Rossignac, and Russell Mersereau, "Protocol for streaming compressed 3-D animations over lossy channels," in *The Proceedings of the IEEE Int. Conf. on Multimedia and Expo*, Lausanne, Switzerland, August 2002, vol. 1, pp. 353-356.
- [7] Ghassan Al-Regib and Yucel Altunbasak, "An unequal error protection method for packet loss resilient 3-D mesh transmission," in *Proceedings of INFOCOM*, New York, NY, June 2002, vol. 2, pp. 743-752.
- [8] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, NJ, 1996.
- [9] UCB/LBNL/VINT, *network simulator ns (version 2)*, Available on URL: <http://www.isi.edu/nsnam/ns/>.