Quantifying the Effect of Content-based Transport Strategies for Online Role Playing Games

Chih-Ming Chen[1], Te-Yuan Huang[1], Kuan-Ta Chen[2], and Polly Huang[1]

[1]Department of Electrical Engineering, National Taiwan University
[2]Institute of Information Science, Academia Sinica
{cmchen,huangty}@gmail.com, ktchen@iis.sinica.edu.tw, phuang@cc.ee.ntu.edu.tw

ABSTRACT

The QoS requirements of game messages may differ because of the latter's intrinsic characteristics. In this paper, we propose three content-based strategies for quantifying the effects of different QoS levels. The strategies assign appropriate QoS requirements for game messages based on our analysis of Angel's Love action logs. We evaluate several transport protocols, including TCP, UDP, SCTP, DCCP, and our content-based transport protocol using the action logs of Angel's Love. Through simulations, we quantify the performance of our content-based strategies. The results show that the strategies incur much lower end-to-end delay and end-to-end jitter than existing transport protocols.

1. INTRODUCTION

Real-time transfer of game messages is essential for highly interactive online games. In other words, delays and jitters play important roles in players' gaming experiences [2]. *Thus, how to minimize end-to-end delay and jitter, i.e., maximize real-timeliness and interactivity, is generally considered the most critical networking issue in the design of game protocols.*

The network part of end-to-end delay and jitter perceived by game players is largely determined by the underlying transport protocol. Both TCP and UDP are widely used by popular MMORPGs. However, some studies [1, 3] have shown that TCP is not suitable for massive multi-players online games (MMOGs). Meanwhile, UDP is good for real-time transmission, but it cannot be applied to online game directly as some game data requires reliable transmission.

Our contribution in this work is two-fold:

1. We evaluate the performance of several existing transport protocols, namely TCP, UDP, SCTP, and DCCP, on MMORPGs with real-life traces.
2. We propose three content-based transport strategies for MMORPGs, and quantify their performance improvement over existing solutions.

NetGames'08, Worcester, MA, USA

Table 1: Angel's Love Action Traces per User

	Game Play Time	Number of Messages
Total	22h 7m 14s	7, 482, 951
Average	4h 37m 41s	4, 080
Maximum	22h 7m 14s	169, 116
Minimum	1h 1s	201

2. DATA DESCRIPTION

Angel's Love is a popular commercial MMORPG in Taiwan and thousands of people may be online at any one time. We asked the game company to record players' actions for us, shown in Table 1. The traces contain the time and type of each action a player performed. Collecting game-level action logs allows us to perform network simulations realistically and flexibly. We classify messages into three types: *move*, *attack*, and *talk*. Since only the latest location matters, the server will simply discard a move message if it is out-of-date. Attack messages detail a character's combat actions when it fights with creatures or other characters. Such messages cannot be missed as each combat action will have some impact on the target. However, if several successive attack messages describe exactly the same fight actions against the same target, then out-of-order arrivals of these commands can be tolerated because the final outcome will be the same. Finally, talk messages contain details of conversations between different players. Therefore, unlike the other two types of messages, talk messages must be transmitted reliably and orderly.

3. PROPOSED TRANSPORT STRATEGY

Our content-based transport strategies assign proper QoS levels to messages based on the QoS requirements of each message type.

1. Strategy **MRO**: MRO only employs *multi-streaming*. Different message types are put in different streams. For example, we classify messages into three types, namely move, attack, and talk, so we can place these three message types into three separate streams.
2. Strategy **MR**: MR combines *multi-streaming* and *optional ordering*. This strategy provides two kinds of streams: ordered streams and unordered streams. If a message type is tolerant of out-of-order processing, it can be transmitted via an unordered stream; otherwise it should be put in an ordered stream.
3. Strategy **M**: M combines all three options: *multi-streaming*, *optional ordering*, and *optional reliability*. If a message type cannot be lost and must be processed in order, it can be transmitted via an ordered

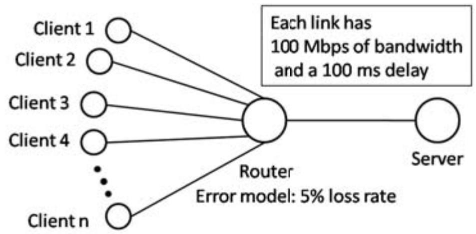

Figure 1: Experiment Topology

and reliable stream. If a message type can tolerate out-of-order processing and requires reliability, then it can be transmitted via an unordered and reliable stream. However, if a message type can be lost without affecting the correct execution of the game, it can be put in an unordered and unreliable stream.

In the next section, we evaluate the performance of the above three strategies for transporting game messages and compare them with existing transport protocols.

4. EXPERIMENTAL METHODOLOGY

To evaluate our content-based strategies, we develop three content-based transport protocols, namely, P_{MRO}, P_{MR}, and P_M, based on the three proposed transport strategies discussed earlier.

We use the ns-2 simulator in our experiments. Our network topology, shown in Figure 1, contains $n + 2$ nodes. One node is configured as the server, one intermediate node serves as the network router and generates random packet loss events, and the remaining n nodes are game clients. For each link, we set the bandwidth at 100 Mbps, the propagation delay at 200 ms, and the loss rate at 5%. We use the module "error model" to generate packet loss events with an exponential distribution.

The game server and the game client modules are implemented to simulate communications between the server and clients of a MMORPG. The game client is *trace-driven*, which means we can feed a set of pre-recorded action logs into the module. Like a normal MMORPG server, the game server notifies other game clients whose characters are in the same zone.

In our experiment, we apply each transport protocol iteratively to transport game messages, and increase the number of clients from 20 to 100. We run each setting for 100 iterations, each of which is simulated with a different client arrival pattern. Specifically, clients arrive randomly between 0 and 1 second following a uniform distribution.

5. EVALUATION RESULTS

To quantify the performance of our content-based transport protocols, we compare the end-to-end delays of our transport protocols with those of TCP and UDP. Figure 2 shows the average end-to-end delays of each transport protocol as a function of the number of clients. The overall trends of our transport protocols are similar to those of existing protocols, but each of our protocols performs better than TCP.

We observe that, overall, the more flexible the strategies employed by our transport protocols, the better the performance they achieve. In all the scenarios, the performance of P_{MRO} is better than that of TCP. P_{MRO} transports data using three TCP connections so that the sequencing delays between different message types are eliminated. P_{MR} achieves lower end-to-end delays than P_{MRO}, as it supports multi-streaming and does not enforce ordering for move and attack messages. The performance of P_M is close to that of UDP because it does not enforce ordering and reliability for approximately 80% of move messages. Therefore, it reduces a large number of unnecessary retransmissions and sequencing delays for non-critical messages. As the end-to-end jitter results are similar to those for end-to-end delays, we do not repeat our arguments here.

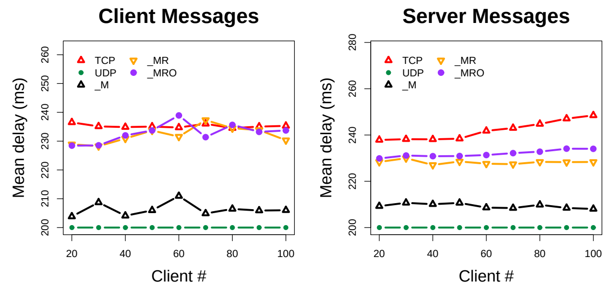

Figure 2: The average end-to-end delays of our transport protocol, TCP, and UDP

Table 2: Protocol Evaluation Summary

	Client to Server		Server to Client	
Protocol	Delay	Jitter	Delay	Jitter
TCP	ℵ	ℵ	ℵ	ℵ
SCTP	≈	≈	★	★
DCCP (TCP-like)	★★★★	★★★★	★★★	★★★
DCCP (TFRC)	☆	☆	☆	☆
UDP	★★★★	★★★★	★★★★	★★★★
MRO	≈	≈	★	★
MR	≈	≈	★★	★★
M	★★★	★★★	★★★	★★★

ℵ denotes incomparable, ≈ denotes similar, ☆ denotes worse, ★ denotes better, ★★ denotes much better, ★★★ denotes good, ★★★★ denotes very good, and ★★★★★ denotes excellent.

Table 2 summarizes the performance of each evaluated protocol. The evaluation of a protocol's performance is based on its relative performance compared to TCP, which is currently the most widely used protocol for MMORPGs. Although the table shows that DCCP (TCP-like) and UDP achieve the best performance, they cannot be applied to a MMORPG directly because they do not support reliable transmission. On the other hand, our proposed M strategy yields a slightly worse performance than UDP, but maintains the correct game semantics. Thus we conclude that our content-based transport strategies for MMORPGs are effective.

6. REFERENCES

[1] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei. An empirical evaluation of TCP performance in online games. In *Proceedings of ACM SIGCHI ACE'06*, Jun 2006.

[2] K.-T. Chen, P. Huang, and C.-L. Lei. How sensitive are online gamers to network quality? *Commun. ACM*, 49(11):34–38, 2006.

[3] S. Harcsik, A. Petlund, C. Griwodz, and P. Halvorsen. Latency evaluation of networking mechanisms for game traffic. In *NetGames '07: Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, pages 129–134, 2007.