Network Performance Measurement Framework for Real-Time Multiplayer Mobile Games

Chad Hansen, Nigel Jurgens
Dwight Makaroff and David Callele
Department of Computer Science
University of Saskatchewan
Saskatoon, SK, CANADA
{cmh631, nwj830}@mail.usask.ca,
{makaroff, callele}@cs.usask.ca

Philip Dueck
Experience First Design, Inc.
Saskatoon, SK, CANADA
pdueck@experiencefirstdesign.com

***Abstract*—Player satisfaction with real-time multiplayer mobile games is correlated with performance of the communications network. The network is the most dynamic component of such games; congestion and channel loss figure prominently in achieving bounds required for real-time response. If messages cannot be delivered on-time, game developers must use predictive techniques to maintain the game experience. Additional complexity in game play requiring more bandwidth and/or processing could be possible under favourable network conditions.**

In this paper, we provide a light weight, embedded measurement framework that obtains frame rate, one-way latency, and frame duration within a game session. The captured data can be used by game designers to tune game complexity and to manage predictive algorithm parameters. Game designers use these predictive algorithms to maintain an approximation of what occurs in real-time, despite delays from network transmission. Our initial test results show that we are able to obtain the necessary information efficiently and note that the game deployed in our case study experiences occasional large delays.

I. Introduction

Player satisfaction is directly correlated with network performance in real-time multiplayer games [1]. Players desire a shared experience, with seamless performance under different wireless networks – expecting only minor differences between networks with perhaps visually discernible, but tolerable, latency differences. We consider a single round of gameplay that concludes within 2 minutes. Players expect their experience to be consistent for the duration of the round; our work aims to support this consistency over different network configurations.

We have created a network measurement framework with the goal of improving the player experience. The simple, targeted framework does not require specialized hardware for network packet capture. The network measurement framework is targeted at three stakeholder groups: Game Developers, Network Service Providers (NSPs) and Gamers. Game developers can use the collected data to support design/implementation efforts, such as the tuning of predictive algorithms. With the help of the players it should be possible to acquire large datasets for tower-to-tower performance and device-to-device performance. Gamers can benefit from improved user experience and the mentioned incentives for collecting data on behalf of third parties (e.g. NSPs) with little conscious effort.

II. Related Work

Hourton *et al.* [2] provide a framework to assess cellular network performance for client devices in Chile. Their passive measurement technique uses crowd-sourced data to determine expected QoS parameters for users. It may be possible to use a similar crowd-sourced data acquisition technique to provide initial input to a game before any gameplay takes place. Gember *et al.* [3] performed extensive performance experiments on mobile device data transfer, determining that in-context measurements more accurately represented performance experienced by the end-user.

III. Context and Motivation

Video games are frame-based simulations; game state processing must take place at regular intervals. The frame processing consists of the reception of messages from other users from the previous frame, updating the game state, performing the required simulation computations and rendering the next video frame. Multi-frame network latency necessitates predictive techniques to maintain a coherent game state and manage transitions to new states.

Our network measurement framework provides the ability to obtain detailed application-level (for the game designer) and tower-to-tower level (for the network service provider) latency measurements, and the necessary context about the performance of the available communication options to provide users with the choice of game configuration.

IV. Implementation and Experiments

We implemented a network measurement framework that was integrated into an existing game engine (EDGE) built on the Unity3D[1] middleware platform. WiFi and cellular communications utilize the Photon[2] framework and the Bluetooth version is built on the Network framework. We measure bandwidth, frame durations and message latencies (in terms of milliseconds and number of frames). In our limited case study, we only analyze measurements that affected game experience. Total message latency consists of the outgoing and incoming message queue latencies, the 2-way network latency between

[1]www.unity3d.com. Last accessed 09-07-13.
[2]www.exitgames.com/Photon/Unity. Last accessed: 09-07-13

client and server, and server processing delay. In the Wifi and cellular environments, independent game clients send game and player state properties to the Photon forwarding server which then processes and broadcasts these messages to all other connected clients. Photon checks the incoming and outgoing message queues once per frame which can result in messages waiting for a maximum of 1 frame. Messages are batched such that a maximum of 5 ms delay is added to each message. The Bluetooth version has a game host (also a client) that acts as a forwarding server that also processes messages once per frame for a maximum additional delay of 1 frame.

We integrated the framework into an existing game built using EDGE. The game is a networked version of Pong that supports two players that was installed on and tested using a Samsung Galaxy S2 and Galaxy S3. We used SaskTel's LTE cellular network and the local Wifi network at the University of Saskatchewan. During each run, we collect 1000 message latencies plus a frame duration for each frame displayed.

V. RESULTS

The most important measurements obtained were the one-way message latencies between two clients. Figure 1 shows latency for a single test in the cellular network. The median

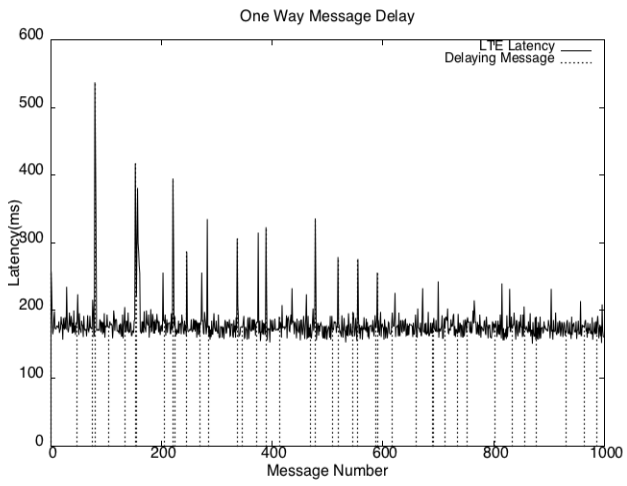

Fig. 1. Cellular message latency over time

latency is 175 ms, with recurring spikes that are between 1.5 and 3 times the median. In a sample, but representative, test on the WiFi network, there is a median latency of 212 ms (20% higher), but we experienced substantial variation in the first 500 messages and very low variation thereafter. The median Bluetooth latency is much smaller (50 ms) but the variation is much higher; the highest latency is 577 ms with 2 periods of instability in the sample test. Further investigation into the game logic showed that some high latency messages occurred when points were scored in the game and extra processing was necessary in the game state update engine.

The distribution of frame delays is shown in Figure 2. Most messages incur a 2 or 3 frame delay on the cellular network. The lower 70% of delay distribution is similar under WiFi, though more messages incur a shorter delay. The Bluetooth network provides longer frame delays because the game state processing is more intensive.

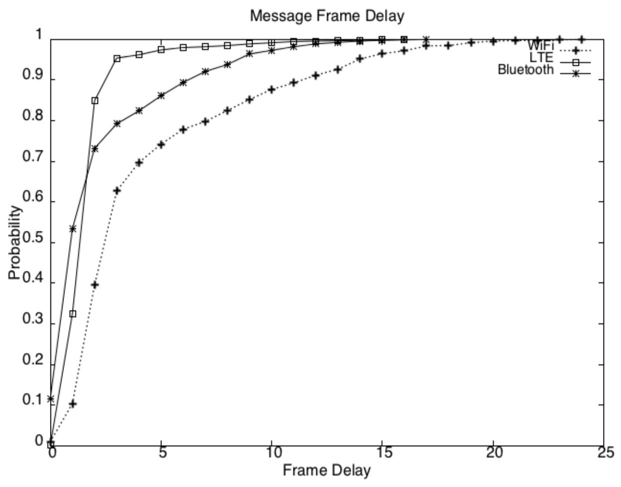

Fig. 2. CDF of Message Frame Delay

The user's principal performance indicator is the frame rate, computed once per second. Our target rate is 60 frames per second (fps). If processing on the client device plus the message generation and reception overhead at the network and transport layer exceeds the frame duration, the display of the next frame cannot take place and the realism of the simulation degrades. Delayed messages prevent calculations based on the real state from occurring and new game state cannot be reliably calculated. For the cellular network, the 75th percentile frame rate was 59 fps and fewer than 25% of the measurements were below 55 fps. The drops in frame rate coincide with spikes in message latency. These values were nearly identical for Wifi and slightly lower for Bluetooth, with a 25% of the measurements below 47 fps, for reasons mentioned earlier.

VI. CONCLUSIONS AND FUTURE WORK

Real-time multiplayer games rely on prediction to maintain the illusion of real-time synchronization across all clients. The ability to tune this prediction with in-context measurements allows game developers to tune their prediction model for network conditions rather than designing around a single assumed level of network quality, thereby delivering even more convincing real-time experiences. Our case study shows that the Wifi and cellular networks tested provided similar statistical results while the Bluetooth network experienced significantly more variation. Our next steps are to provide more comprehensive testing on more cellular networks to determine the performance range and performance variability to guide game designers when providing experiences for the resource-rich and resource-poor network environments in which the games are to be deployed.

REFERENCES

[1] M. Dick, O. Wellnitz, and L. Wolf, "Analysis of factors affecting players' performance and perception in multiplayer games," in *ACM NetGames*, Hawthorne, NY, Oct. 2005, pp. 1–7.

[2] G. Hourton, G. D. Canto, J. Bustos, and F. Lalanne, "Crowd-measuring: Assessing the quality of mobile internet from end-terminals," in *NetG-CooP*, Nov. 2012, pp. 145–148.

[3] A. Gember, A. Akella, J. Pang, A. Varshavsky, and R. Caceres, "Obtaining in-context measurements of cellular network performance," in *ACM IMC*, Boston, MA, Nov. 2012, pp. 287–300.