

Investigating the Influence of HTTP Traffic on the Accuracy of Packet Time Stamping

David A. Kennedy, Sebastian Zander

Centre for Advanced Internet Architectures. Technical Report 041124A
Swinburne University of Technology
Melbourne, Australia
{dkennedy, szander}@swin.edu.au

Abstract-For collecting information about game traffic characteristics and user behaviour, we are setting up a public game server that runs a few different game servers. We also want to run a web server on the same machine, to provide users with information about the server and the games. The web traffic then would interfere with the game traffic. In this paper we want to find out whether the web traffic would cause degradation in game performance, and whether it would influence the traffic characteristics of the game traffic. In particular we look at the influence of web traffic on the measured inter-arrival times of the UDP traffic.

Keywords- Game Traffic, HTTP, Inter-arrival Times

I. INTRODUCTION

In our studies of game traffic on the Internet, we decided to host a game server, so that we could capture and analyse the traffic patterns that are characteristic of online games [1] [2]. We set up a computer to host servers for different popular games: Counter-Strike, Wolfenstein: Enemy Territory, and Quake 3 Arena. We will put this computer on Grangenet's [3] high-speed network, and it will have a 100Mbps link to the Internet.

To help people find out about this game server, this computer will also host a web server with some pages containing information about the games and the server. The web server can also be used to allow users a faster download of game specific modifications. This presents a problem. While people are accessing these web pages, the generated web traffic interferes with the game traffic and may change the traffic characteristics measured. There are two main characteristics we are interested in: packet length, and inter-arrival times. Because the web traffic would have no influence on the packet length, we focus on investigating the influence on the packet inter-arrival times. The web traffic could either directly influence the game traffic (changing the inter-arrival times) or indirectly (changing the measurement results due to e.g. less accurate timestamps). In the extreme the web traffic would change the inter-arrival times dramatically creating a large jitter, which could also adversely affect the game play. (We assume that bandwidth is not a problem because we have a high-speed connection.)

Therefore, the web traffic must be limited in such a way that it does not affect the inter-arrival times and the accuracy of the time stamping. In this experiment we

aim to find out how web traffic affects the inter-arrival time characteristics of the game traffic.

The rest of the paper is organized as follows: section I describes the methodology of the tests; the results are presented in section II; section III concludes and outlines future work.

II. TEST METHODOLOGY

A. Programs used for the tests

The computer running the game servers (from now on referred to as "gs"), is running FreeBSD as its operating system. The program used to bandwidth limit the web traffic is dummynet [4], and the web server is Apache [5].

To control the UDP flow test variable, we used a package called Rude & Crude [6] (Realtime UDP Data Emitter & Collector for RUDE), which can generate a UDP stream with a constant packet rate (and hence constant inter-packet times). We use Rude & Crude instead of real game traffic because it can be much better controlled, produces a more precise traffic flow, and is easier to set up and use.

Rude generates more precise inter-packet times than real game clients would, allowing us to do perform a more accurate measurement. Furthermore, with real game clients it would be a very high effort to test against a wide range of different game traffic packet rates, as this would require a number of game clients, whereas with Rude we can easily change the packet rate. Rude can be easily remotely operated from the command line whereas a game client requires a monitor, X windows set up etc.

Because we are interested in the accuracy of inter-packet time measurements, it is essential that the inter-packet times of the sender are as precise as possible to properly determine any network or receiver effects on the distribution.

Rude was run on a separate computer and UDP packets were sent to gs, where Crude was running, capturing the packets for analysis.

We used a program called HTTPPerf [7] to generate requests and cause as much traffic as the server could sustain. The test bed was set up according to Figure 1.

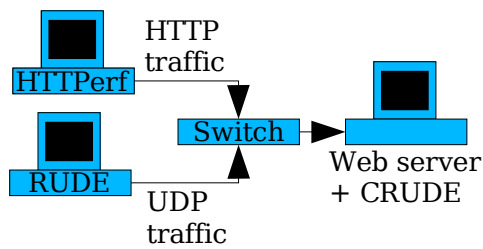


Figure 1 Test bed setup

B. Testing

We configured dummynet on gs to bandwidth limit the web traffic (the traffic on port 80) as follows:

```

ifpw add 1 pipe 1 ip from any to any 80
ifpw add 2 pipe 1 ip from any 80 to any
ifpw add 65535 allow ip from any to any
ifpw pipe 1 config bw <bandwidth>M
  
```

For each different test conducted, the bandwidth value was changed, and we measured the effect on the UDP traffic.

HTTPerf is a command line based program that generates HTTP requests. It is designed to test web servers by generating HTTP requests at very high rates. The command used is as follows:

```

httpperf -server <hostname> --port 80 --uri /quake3.html --rate 10000 --num-conn 6000000
  
```

This command caused HTTPerf to connect to the web server on the host with IP name <hostname>, running at port 80. The web page requested was quake3.html. Quake3.html is approximately 20 kilobytes and is a HTML page with server-side includes. The rate at which the requests were issued was 10,000 per second, the test included a total of 6,000,000 TCP connections (10 minutes) and there was one HTTP request sent for each connection. While HTTPerf was running, Rude ran for 5 minutes for each test.

III.RESULTS

When we first started testing, we found a lot of large values for the inter-arrival times of the packets at the receiver. Upon checking the inter-arrival times of the packets at the source, we could see that these strange, large values were caused by Rude [4]. After freeing up all of the CPU time on the machine with Rude running, shutting down all unnecessary processes, the tests were conducted again, and although greatly improved, the source distribution was still very spread-out. Each time the 5 minute test was conducted, the source distribution was different, so we concluded that we could not measure the distribution at the receiver alone, but the difference between the distribution at the receiver and the distribution at the source.

For each test, we calculated the standard deviation of the inter-arrival time distribution of the UDP packets, as seen by both the source and the receiver. We then took the difference between these two standard deviations as the value that we used to compare the tests.

As Table 1 shows, the more bandwidth the HTTP traffic was allowed to use, the larger the standard deviation difference is. Although this is the case, the absolute values are very small. We can see that even under a full web traffic load of up to 10 Mbps, the UDP inter-packet times were only affected by a standard deviation in the order of a few microseconds. In the case of game traffic, which has inter-packet times in the order of milliseconds, this change would be negligible.

<i>HTTP bandwidth (Mbit)</i>	<i>St.Dev. @ sender (us)</i>	<i>St.Dev. @ receiver (us)</i>	<i>St.Dev. difference (us)</i>
0	0.7808	0.7989	0.0181
1	0.7822	0.8140	0.0318
2	0.7899	0.8342	0.0443
5	0.7782	0.8904	0.1122
10	0.8763	2.1922	1.3159

Table 1 Difference between standard deviation at sender and receiver, at different HTTP bandwidth limits.

IV.CONCLUSION

The tests that were performed were done with time constraints, as the server was due to be shipped to Grangenet. More tests with different UDP packet rates should be performed. More extensive testing should also be done (repeating the tests and using the average standard deviations) to produce more reliable results. The tests could be done with real game traffic and both the inter-arrival time distributions, and the actual perceived game performance could be measured and observed.

The tests could be refined to be more indicative of how the server would actually perform. For example if gs was actually running the game servers, and there were multiple game clients connected to it, the performance hit from added web traffic could be worse than it has shown to be in this simple test environment. Also the CPU load associated with running the game servers and the web server should be measured because it could have more of an impact on the time stamping than the web traffic has on the inter-packet times.

For our purposes, there are only four HTML pages of around 20 kilobytes each being served, and it is not expected that the server will be dealing with many concurrent sessions. For these reasons we decided that a bandwidth limit of 1 Mbps for the web traffic is sufficient to maintain a decent speed for web transfer.

REFERENCES

- [1] T.Lang, P.Branch, G.Armitage, "A Synthetic Traffic Model for Quake 3", ACM SIGCHI ACE2004 conference, Singapore, June 2004, <http://atnac2003.atcrc.com/ORALS/Lang2.pdf>
- [2] T.Lang, G.Armitage, P.Branch, H-Y.Choo, "A Synthetic Traffic Model for Half Life", Proceedings of the Australian Telecommunications Networks & Applications Conference 2003 (ATNAC 2003), Melbourne, Australia, December 2003, <http://atnac2003.atcrc.com/ORALS/Lang.pdf>

- [3] GrangeNet - GRid And Next GEneration Network website, <http://www.grangenet.net> (viewed October 2004)
- [4] Dummynet, http://info.iet.unipi.it/~luigi/ip_dummynet (viewed October 2004)
- [5] Apache, <http://www.apache.org> (viewed October 2004)
- [6] RUDE & CRUDE, <http://rude.sourceforge.net> (viewed October 2004)
- [7] HTTPerf research paper, and download link, http://www.hpl.hp.com/personal/David_Mosberger/httpperf.html (viewed October 2004)