Australian Telecommunications Networks & Applications Conference 2003 (ATNAC2003), Melbourne, Australia, December 2003

A Synthetic Traffic Model for Half-Life

Tanja Lang, Grenville Armitage, Phillip Branch, Hwan-Yi Choo
Centre for Advanced Internet Architectures
Swinburne University of Technology
Melbourne, Australia
tlang@swin.edu.au, garmitage@swin.edu.au

Abstract- **This paper presents our development of a synthetic traffic model for the interactive online computer game Half-Life. The goal is a traffic model that can be used by researchers and Internet Service Provider engineers to estimate the potential future impact of Half-Life traffic over IP networks. We developed our ns2 simulation model for Half-Life's IP traffic by running live network experiments and characterising the observed packet length, packet inter-arrival times, and data rates (in packets- and bits- per second). Our observations are documented in this paper, and we expect our traffic model will assist network planners who wish to better support real-time game traffic.**

I. Introduction

In recent years interactive network games have become more popular with Internet users and constitute an increasingly important component of the traffic seen on the Internet. Interactive game traffic has different characteristics to the WWW and e-mail traffic prevailing on the Internet today and therefore imposes different requirements on the underlying network.

Providing premium service to the increasing on-line gaming community could be a promising source of revenue for ISPs. To provide this service an ISP must have knowledge of the traffic load offered by game traffic to provision their networks accordingly. Some researchers have already looked at the traffic characteristics of different popular on-line games to provide a suitable traffic model to test existing or planed network for their capability to support game traffic [1] - [3].

In this paper we present our investigation of Half-Life an Internet first person shooter game. Due to the short reaction times necessary to play this kind of game, it is especially sensitive to network characteristics. An evaluation of the traffic load offered to the network by such a game can be used to evaluate the suitability of existing networks to support first person shooter games and to design new networks with special consideration of game traffic.

In this paper we first present the main traffic characteristics of Half-Life in terms of packet lengths, packet inter-arrival time, packet per second (PPS) and data rates and then a traffic model based on these observed characteristics. The traffic model is not aimed at describing the game in every detail but to capture the main characteristics of Half-Life game traffic. As in all modeling we were aiming for a balance of accurately describing real world data and reasonable execution speed of a simulation model including hundreds of network hosts.

II. Traffic characteristics of Half-Life

A. Experiments and set-up

We captured several months worth of Half-Life game traffic from games played over our university LAN. The general network setup is shown in Figure 1. The client machines were at most one hop away from the game server, resulting in round trip times less than 10 ms.

Concurrent with the game, the game server was also running pkthisto [4]. Pkthisto creates packet length and packet inter-arrival time histograms for each individual flow from client to server and server to client. Source and destination IP addresses and port numbers specify a flow. Pkthisto also logs packet per second and date rates (in bits per second) for each flow. We configured pkthisto to use 2000 consecutive packets for each histogram and pps/rate estimate. By default a flow is considered active if more than 200 packets are seen in less than 800 ms. In addition to per flow statistics, pkthisto also creates aggregate histograms based on all traffic to or from a specific node on the LAN.

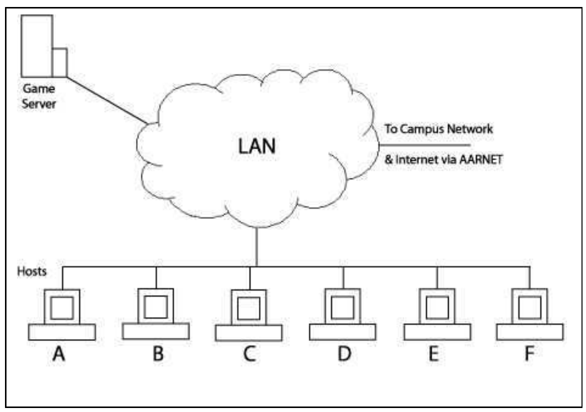

Figure 1: experimental set-up

We have observed 15 Half-Life games played over a 2 months period. The number of players varied from 3 to 7, both between and during games. For all of these games we found that the only parameters impacting the traffic pattern were the map played and the graphic rendering software of the client computer. The number of players in the game or the joining/leaving of a player was not visible in the investigated traffic traces.

The game presented in this paper was a 6 player game with 4 custom maps: xflight, chilDM, rats3, and odyssey. Each of these maps was played for 10 minutes, which results in 4 to 5 histograms per map. The observations made from this example game are representative for all the games investigated.

Australian Telecommunications Networks & Applications Conference 2003 (ATNAC2003), Melbourne, Australia, December 2003

B. Packet lengths

The packet lengths discussed in this section are the lengths of the IP datagrams. The UDP payload is 28 bytes shorter and the Ethernet frame 14 bytes longer than the IP datagram length.

Server to Individual Clients

The packet length from the server to the different clients is dependent on the map played. Figure 2 shows a server to individual client packet length plot for the complete game including all 4 different maps played. The zero data lines between the maps have been included to improve the visibility of the different maps.

In general the packet lengths for Half-Life have quite a wide spread and range from 60 to 300 bytes. Xflight has the shortest length distribution with packets ranging from 64 to 188 bytes with a strong peak at 67 bytes. ChilDM is the map with the largest spread of packet lengths starting from 68 bytes and going up to oever 400 bytes. The length of Rats3 packet ranges from 60 to 300 bytes, with packet lengths concentrated around 70 bytes. The packets transmited by the Odyssey map start at 74 bytes and again go up to 300. Most packets however are between 140 and 180 bytes long.

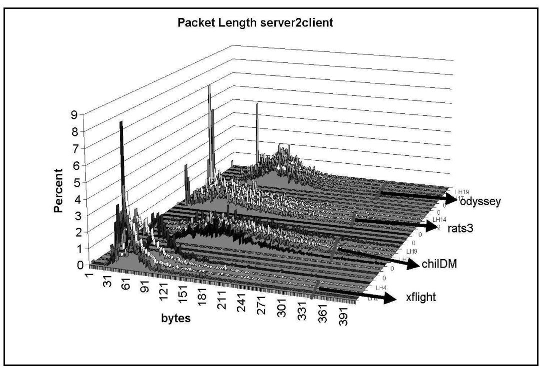

Figure 2 server to client packet lengths

Client to Server

The packet lengths from the individual clients to the server is independent from all parameters. Figure 3 shows an aggregate packet length plot, representing the packets that the server has received from all clients during the entire game. 93% of these packets fall inside the outlined corridor, 7% of the packets are responsible for the peaks. These peaks occur at random time intervals and do not correspond to map changes. They may represent a short idle period of different players. Cleint to server packets have a more limited range than server to client packets.The smaller packets are about 60 bytes long and the largest ones are around 90 bytes.

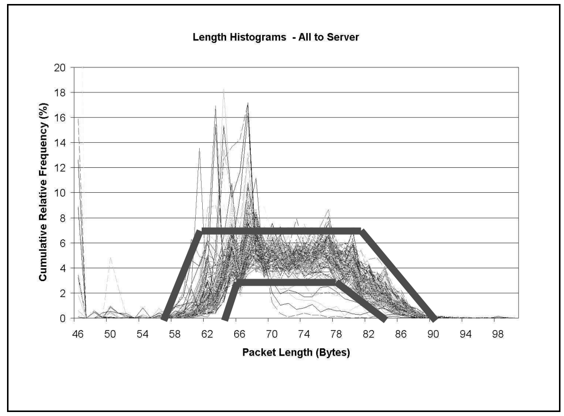

Figure 3 all clients to server packet lengths

C. Packet inter-arrival times

Server to Client

The server to client packet inter-arrival times are very regular (Figure 4). For most maps, with the exception of rats3, the server sends one update packets to the client every 60 ms. For the rats3 map about half of the packets are sent around 50 ms and the other half around 70 ms.

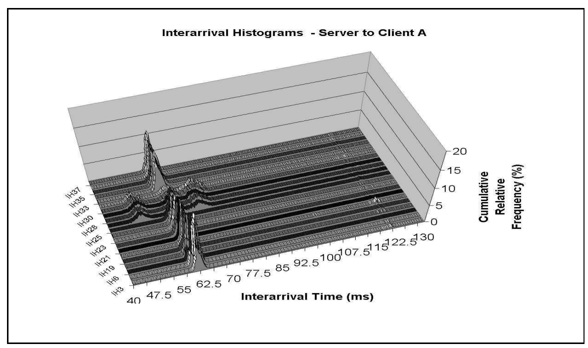

Figure 4: server to individual client packet inter-arrival times

Client to Server

The packet transmissions from the different client computers to the server are very dependent on the graphic rendering software used. Three different graphic rendering methods were observed during our traffic measurements, OpenGL [6], Direct3D [7] and Software.

We only present the graph for OpenGL (Figure 5), which is the most commonly used graphic rendering method. OpenGL clients transmit packets very periodically in 33 ms and 50 ms intervals, while the transmission intervals of clients using Software rendering is one packet every 41 ms most of the time.

Only one of the computers participating in the games was using Direct3D. The packet inter-arrival times from this machine are very random There were no clearly visible recurring peaks and the packet transmission pattern varies from game to game. More measurements with Direct3D clients wuold need to be performed if a model is to be created. However the usage of direct3D as rendering method is not very common.

Australian Telecommunications Networks & Applications Conference 2003 (ATNAC2003), Melbourne, Australia, December 2003

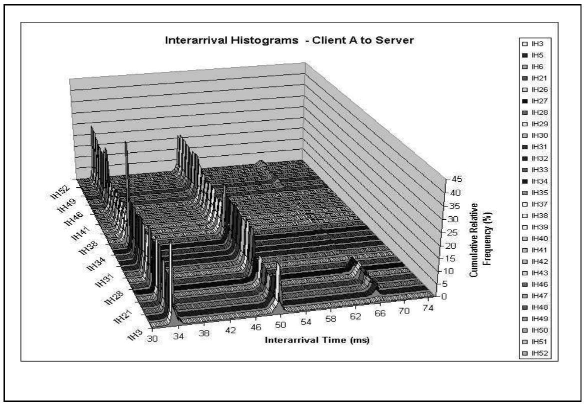

Figure 5: client to server packet inter-arrival times for OpenGL

D. Packet per second (PPS) and data rates

Server to Client

The PPS rate from server to client is only slightly varying between 16 and 16.5 packets per second with a mean of 16.3

The data rate (presented in Kbits per second) is dependent on the PPS rate and the packet length. We have discussed in section II.B that the packet lengths of server to client traffic are dependent on the map played, therefore the data rate is equaly dependent on the map. Xflight has the shortest packets and the lowest rate, while chilDM has the longest packets and correspondingly highest rate.

The aggregate data rate from the server to all the clients shows the same dependency on the map as individual data rates. In addition, dependencies on the number of active players can be seen (Figure 6). In the initial phase of the game (indicated by the square) the players are joining. Afterwards 3 maps (xflight, chilDM, rats3) are played with 6 players. One player leaves at the end of the rats3 map, therefore Odyssey is only played by 5 players. This results in equal data rates for the Odyssey (dashed circle) and rats3 map in the aggregate flow graph, while, when observing individual data rates, Odyssey has a higher data rate than rats3.

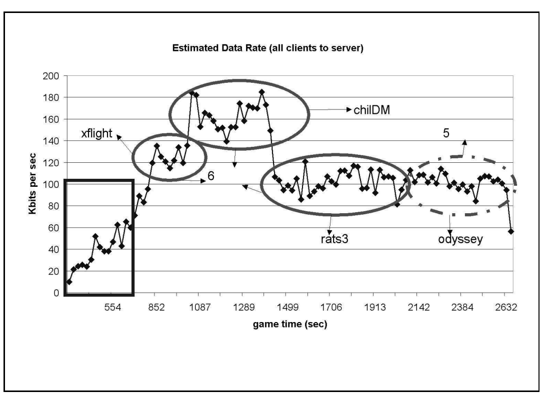

Figure 6: aggregted data rate server to all clients

Client to Server

The PPS rate from the client to the server varies between 22 and 24 packets per second with the mean approximately 23. The exception is the computer using Direct3D, which has a very constant PPS rate of 20.

Since neither PPS nor packet lengths show a dependency on the map for client to server traffic, the estimated data rate varies randomly and is centered around 13 Kbits per second.

The aggregate client to server data rate is not surprisingly dependent on the number of players active in the game. For each player that joins/leaves the aggregate data rate increases/decreases by approximately 13 Kbits per second.

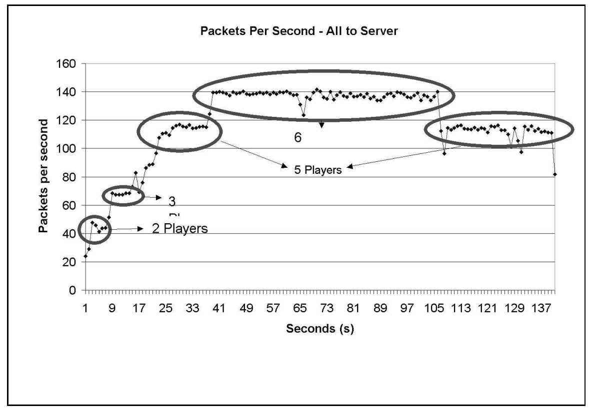

Figure 7: PPS all clients to server

III. SIMULATION MODEL

Based on the observed traffic characteristics, in terms of packet inter-arrival times and packet lengths, we develop ns2 models for Half-Life servers and Half-Life clients.

Server to Client

The server to client packet inter-arrival time is easy to model. Depending on the map the server either sends packets every 60 ms or 50% at 50 ms and 50% at 70ms.

The packet length distributions per map were more of a challenge. We used the SPSS statistics package to analyze the distribution of the data packet lengths and to perform goodness-of-fit tests with hypothetical distributions. For reasons of execution speed and ease of modeling we concentrated on simple distributions that were available as part of simulation software and investigated their fit to the observed data.

The Lognormal distribution with different parameters proved to be the best fitting distribution for all maps in the game. The authors of [1] observed that the lognormal distribution also results in acceptable fits for Quake3 traffic.

In [2] it was already observed that the traditional goodness of fit methods do not lead to acceptable results for packet length or inter-arrival time distributions of game traffic. Q-Q graphs were used in this paper to visually display the quality of a fit. We had the same experience when applying goodness-of-fit test to the Half-Life data and also chose to represent the fit in from of Q-Q plots. Figure 8 shows the Q-Q plot of the chilDM map packet lengths versus the lognormal distribution. For packet lengths below 400 bytes the 2 distributions have an

Australian Telecommunications Networks & Applications Conference 2003 (ATNAC2003), Melbourne, Australia, December 2003

excellent match, they only start to deviate for higher packet lengths. Even though packets with higher lengths are present in the real data set, their probability is as low as 0.1%, therefore it is safe to exclude them from the model. The same behavior of a good fit for the main body of the distribution and a divergence for the tail was observed for the three other maps as well.

Figure 9 represents a comparison between the actual and the simulated data, again for the chilDM map, while Table 1 to Table 4 list the mean and standard deviation of the observed data as well as the simulated one along with the parameters used for the lognormal distribution, for all 4 maps. Like the model presented in [1] for Quake3 data, our model results in a lower variance than the original data. This is caused by the fact that parameter estimation methods ignore the small probabilities of the packets in the tail of the distribution.

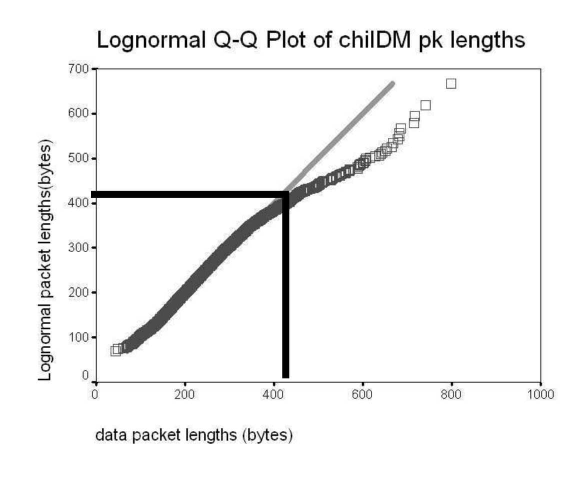

Figure 8: Q-Q plot of chilDM packet length and lognormal distribution

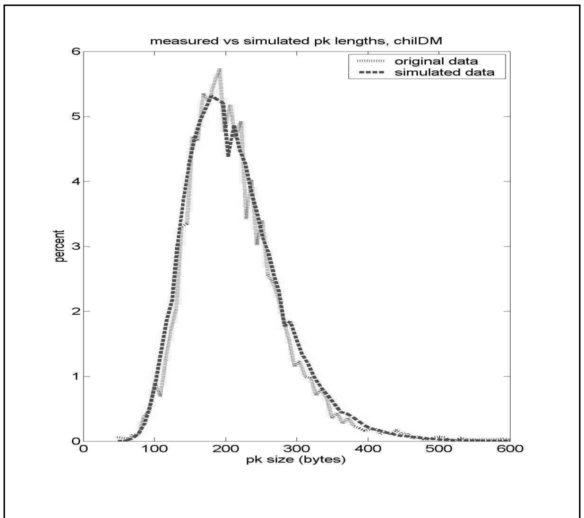

Figure 9: comparison between observed and simulated chilDM packet lemgths

chilDM	observed	simulated	lognormal
mean	213.0773	213.5428	202.91875
stdev	70.45093	67.8754	0.3102072

Table 1

Odyssey	observeed	simulated	Lognormal
Mean	159.278	159.4463	154.15274
Stdev	44.1898	4.727	0.25086516

Table 2

Rats3	observed	Simulated	lognormal
Mean	139.443	139.4307	129.63025
Stdev	58.7277	53.6403	0.3723556

Table 3

Xfligth	Observed	Imulated	Lognormal
Mean	113.282	113.2366	109.76946
Stdev	32.5058	27.8248	0.2415373

Table 4

Client to sServer

For client to server packet inter-arrival times we only chose to implement OpenGL and software clients, since OpenGL is the most commonly used graphic rendering method and we only had one Direct3D client participating in our game.

Both OpenGL and Software packet inter-arrival times are easy to model. For OpenGL 50% of packets are sent in 33 ms intervals, while the outher 50% are transmitted every 50 ms. For Software clients our model transmits all packets in 41 ms intervals.

Client packet length are indpendent of computer hardware, number of players or maps, as discussed in section II.B. Therefore only one model for client packet lengths is necessary. We excluded the random peaks discussed in section II.B and only modeled the distribution indicated in the corridor in Figure 3.

As for server packet lengths we used SPSS to obtain Q-Q plots of hypothetical distributions against the measured data. For client packet length the normal and the lognormal distribution resulted in equally good fits. For ease of implementation we therefore chose the normal distribution to model client packet lengths.

Figure 10 shows the Q-Q plot of the measured data agains a normal distribution, while Figure 11 presents a comparison between the measured and the simulated packet lengths. The mean and standard deviation of the measured and simulated packet lengths, as well as the parameters used for the normal distribution are shown in Table 5.

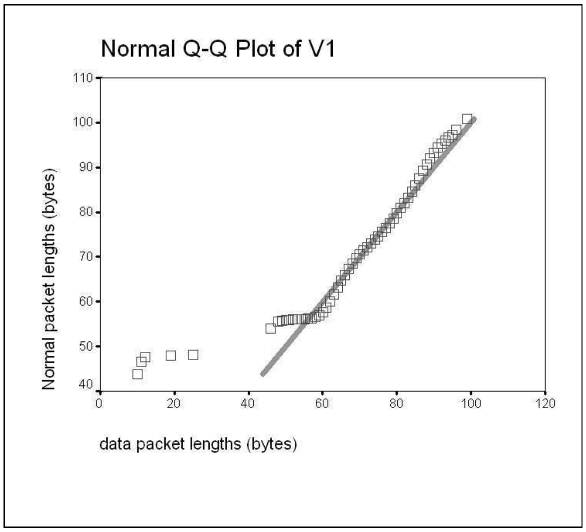

Figure 10: Q-Q plot for packet lengths from all client to server

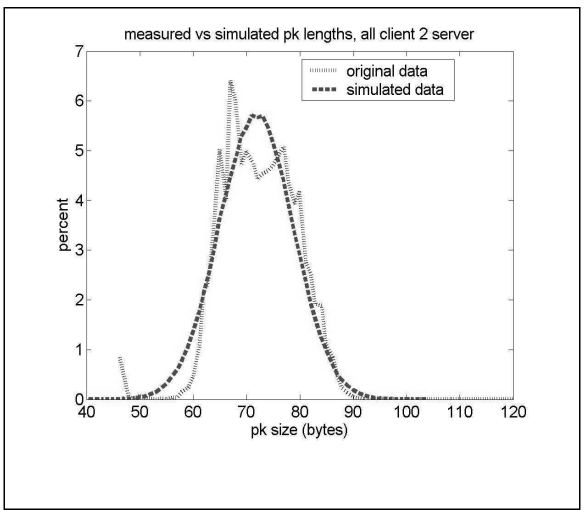

Figure 11: comparison between observed and simulated packet lengths fromm all clients to the server

Client pk length	Observed	Simulated	normal
Mean	72.3106	72.3150	72.291433
stdev	7.0025	7.4279	6.9742849

Table 5

Simulation Model

Since ns2 implements the normal as well as the lognormal distribution it was very easy to implement the server and the client model in this simulation tool. Below we present the main sections of server and client code. The server code is for the chilDM map. The map is passed as a parameter into the server model and case statement exists for every map.

```
Case map == ‘chilDM’:
   interval = 0.06;
   timer.resched(interval)
```

Whenever the timer expires a packet is transmitted

```
Case map == ‘chilDM’:
    Pk_size = Random::lognormal
          (202.91875, 0.3102072);
    timer.resched(interval)
```

The client model is only parameterized by the rendering type (OpenGL or Software)

```
Case rendering == ‘OpenGL’:
   Interval1 = 0.05;
   Interval2 = 0.033;
   Timer1.resched(interval1)
   Timer2.resched(interval2)
```

Whenever one of these 2 timers expires a packet is sent and the appropriate timer rescheduled

```
Pk_size = Random::normal
    (72.291433, 6.9742849)
```

IV. Conclusion

Inter-active network games are very popular and can be a promising source of revenues for ISPs. To be able to offer premium service to the customers, it is important to understand the traffic characteristics of games to provision the network accordingly.

In this paper we analyzed Half-Life a popular Internet first person shooter game. We characterized Half-Life traffic in terms of packet inter-arrival times, packet lengths and data rates. Based on the observed characteristics we developed simulation models for Half-Life servers and Half-Life clients that can be used to evaluate existing networks for their suitability to support game traffic, as well as help in the design of new QoS networks.

V. References

[1] J. Faerber, “Network game traffic modelling”, Proceedings of the 1st ACM workshop on Network and System Support for games, April 2002

[2] M.S. Borella, "Source models of network game traffic", Proceedings of networld+interop '99, Las Vegas, NV, May 1999

[3] W. Feng, F. Chang, W. Feng, J. Walpole, "Provisioning On-line Games: A Traffic Analysis of a Busy Counter-Strike Server", SIGCOMM Internet Measurement Workshop, November 2002

[4] pkthisto, http://caia.swin.edu.au/genius/genius-tools.html (as of 3rd of April 2003)

[5] Game Environments Internet Utilisation Study (GENIUS) http://caia.swin.edu.au/genius/

[6] OpenGL, http://www.opengl.org/ (as of 26th of July 2003)

[7] Direct3D, http://www.gamedev.net/reference/programming/features/d3do/ (as of 26th of July 2003)