A Ns2 model for the Xbox System Link game Halo

Tanja Lang, Grenville Armitage
Centre for Advanced Internet Architectures. Technical Report 030613A
Swinburne University of Technology
Melbourne, Australia
tlang@swin.edu.au, garmitage@swin.edu.au

Abstract- **This paper presents traffic analyses for Xbox System Link traffic. The goal of this analyses is to create a traffic model for Xbox traffic that can be used for network evaluation purposes. Traffic characteristics observed were: packet length, packet inter-arrival times, packets per second (PPS) and data rates. We found that System Link traffic is very periodic. Servers send update packets every 40 ms to all clients. Clients transmit two types of packets to the server. One every 40 ms and the other one every 201 ms. Server to Client PPS was 25 and Client to Server PPS 30 for all experiments. Packet lengths depend on the number of players participating in a game. Based on these observations a simple ns2 simulation model for Server and Client Xboxes was developed.**

Keywords- Traffic analyses, network games, Xbox, Ethernet, LAN, Halo,System Link

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].

We chose to investigate Halo, a very popular Xbox System Link game. Even though the System Link feature has been designed to work only over LANs, several tunneling solutions are available to connect Xboxes over the Internet [4]-[6]. In addition, the newly released Xbox Live allows Xboxes to be directly connected to the Internet without the use of tunneling [7]. This variety of different solution and the reported success of Xbox Live [8], give an indication that ISPs might observe a substantial amount of Xbox traffic over their networks.

This paper presents the main traffic characteristics of Halo and a ns2 model for Xbox server and client based on these observed traffic patterns.

II. TRAFFIC CHARACTERISTICS OF HALO

A. Experiments and set-up

Several experiments were carried out to obtain a valid range of data points for the traffic pattern analyses. The System Link feature allows up to 4 Xboxes to be linked together via a hub over a LAN, with 1 to 4 players attached to each Xbox. Xbox System Link uses standard UDP/IP/Ethernet packets to transmit its data over the LAN. However, the IP address and UDP port of every Xbox is fixed to 0.0.0.1 and 3074 (making the packets un-routable over the Internet) and each Xbox uses MAC addresses to differentiate between the clients and server Xboxes.

We performed experiments with 2, 3 and 4 Xboxes with various combinations of players connected to each Xbox. The maximum number of players participating in an experiment was 14.

In addition to the Xboxes, a packet sniffer machine was attached to the hub to monitor the traffic generated on the LAN. An example experiment setup is shown in Figure 1

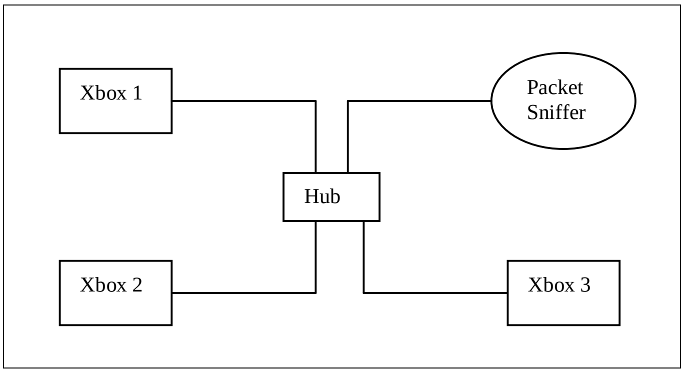

Figure 1: experimental set-up

The packet sniffer machine was a 600 MHz Celeron with 128 MB of RAM and was running the FreeBSD 4.6 operating system. The network card used was Intel Pro 10/100B/100+ Ethernet and was accessed through the 'fpx0' interface. The hub used was a 10Mbit/sec CentreCOM AT-MR820TR hub, therefore the LAN bandwidth was limited to 10 Mbps for all experiments.

To test the packet capturing and time stamping capabilities of the sniffer machine, NetCom Systems SmartBits 2000 with a SX-7410B line card was used. The SmartBits 2000 was configured to send long (500 and 1000 packet) bursts of back-to-back packets to the sniffer computer, with intervals ranging from 12.5

usec to 50 msec. We found that the computer was capable of time stamping to shorter than ten microseconds accuracy, even when subjected to bursts of 1000 packets spaced tens of microseconds apart. This was considered sufficient accuracy for our Xbox analysis, since we are discussing intervals to the nearest millisecond.

Tcpdump [9] was run on the packet sniffer machine to obtain a raw packet trace of the entire LAN traffic during each experiment.

Pkthisto [10] was used to analyse these raw packet traces. Pkthisto creates packet length and packet inter-arrival histograms for each flow observed in the tcpdump file. Source and destination IP addresses and port numbers specify a flow. Because the IP addresses in System Link ethernet frames are meaningless pkthisto uses 4 bytes of the MAC address to derive an artificial IP address when isolating client-server and server-client flows. Pkthisto also logs packet per second (pps) and date rates (in bits per second (bps)) for each flow. For each histogram (or pps/rate value) 2000 consecutive packets were used. 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 can also create aggregate histograms based on all traffic to or from a specific node on the LAN.

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 Client

The packets sent from the server Xbox to all the client Xboxes are of constant length. The only parameter governing the packet length is the total number of players in the game (N). The packet length (in bytes) is related to the number of players by the following equation:

$$Pk\_length = 30*N + 100$$

Figure 2 shows the measured packet length compared to the packet length obtained by the above equation.

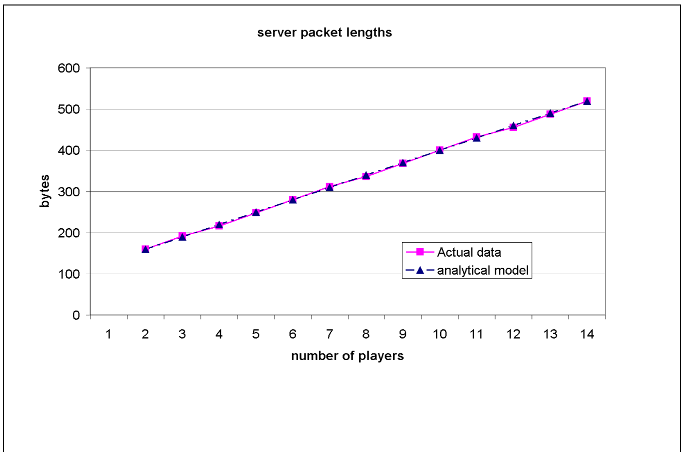

Figure 2 server to client packet lengths

Client to Server

Two types of packets are transmitted form each client Xbox to the server. Approximately 16 % of the client to server packets have a fixed length of 72 bytes. These packets have been observed in all experiments and are independent of how many Xboxes or players participated in the games.

The length of the remaining 84% of the client to server packets is dependent on the number of players connected to a particular client Xbox (C). The packet length can be approximated by:

$$Pk\_length = 30*C + 80$$

A comparison between the measured packet lengths and the lengths obtained by the above equation is shown in Figure 3.

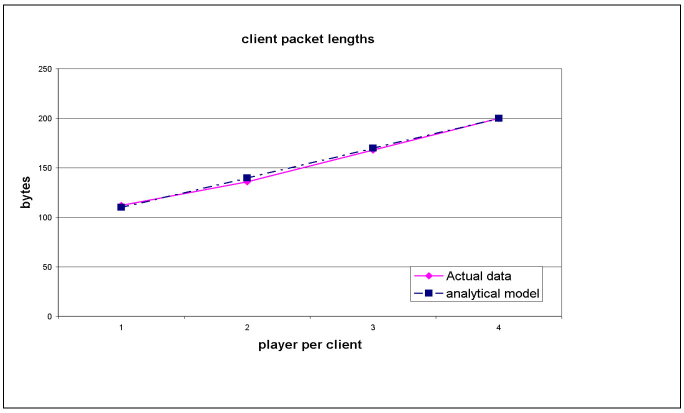

Figure 3 client to server packet lengths

C. Packet inter-arrival times

Server to Client

The server Xbox sends update packets to each of the clients every 40 ms. The packets to all client Xboxes are transmitted back-to-back. If L clients are connected to the server during a particular game, the server sends a burst of L data packets every 40 ms. The inter-arrival histograms therefore display a value of (100*(L-1)/L)% close to 0 ms and the rest of the inter-arrival times will be concentrated around 40 ms. Figure 4 presents an inter-arrival histogram for a 3 Xbox game (2 clients, 1 server) where 50 % of the packet inter-arrivals are close to 0 ms and the other 50 % close to 40 ms, while Figure 5 shows an inter-arrival histogram for a 4 Xbox game in which 66.6% of the packets are sent out back-to-back. The symbols 'IH nn' labeling one axes in each inter-arrival histogram plot signify the histogram number assigned to this particular histogram by pkthisto. Each histogram consists of 2000 observed packets.

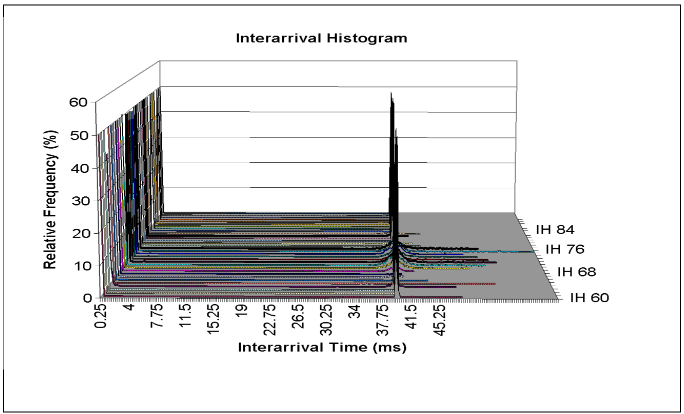

Figure 4: aggregate server to clients packet inter-arrival times (3 Xbox game)

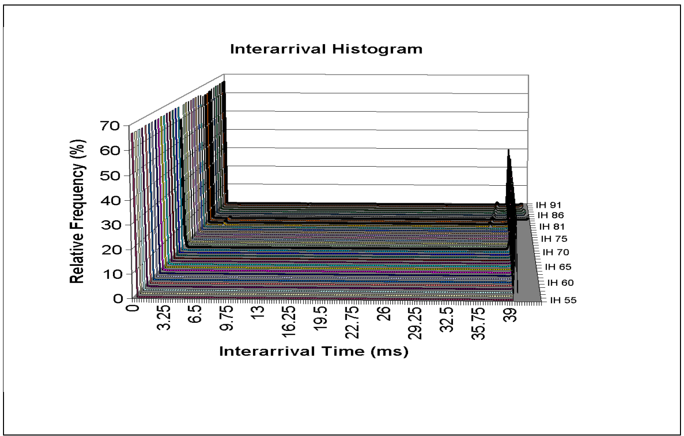

Figure 5: aggregate server to clients packet inter-arrival times (4 Xbox game)

Client to Server

The packet inter-arrival of an individual client to server data flow is independent of the number of players on the console and also independent of the number of other clients in the game. Therefore the individual client to server inter-arrival time histograms look the same for all the experiments performed and an example is shown in Figure 6.

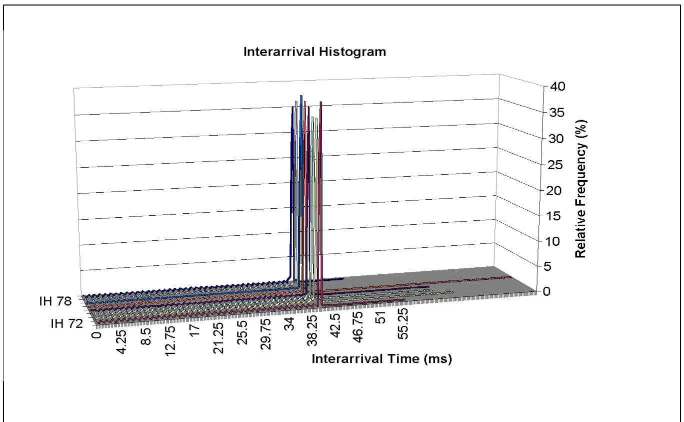

Figure 6: individual client to server packet inter-arrival time

Figure 7 presents the cumulative inter-arrival times for the same data flow (CIH nn signifies the nth cumulative inter-arrival histogram created by pkthisto). In this figure it can be seen that about 33% of the packets are sent closer spaced than 40 ms. The inter-arrival time distribution for these packets is uniformly random for inter-arrival times between 0 and 40 ms (Figure 6). The remaining 67% of data packets are sent at 40 ms intervals. This uniform distribution between 0 and 40 ms is caused by the concurrent transmission of 72 byte packets every 201 ms and the packets dependent on the number of client players every 40 ms. The inter-arrival times between these 2 kinds of packets can be anywhere between 0 and 40 ms.

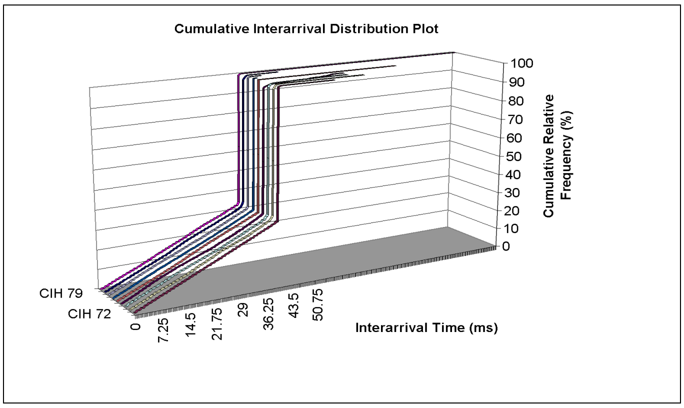

Figure 7: individual client to server cummulative packet inter-arrival times

The aggregate inter-arrival time pattern is dependent on the client Xbox hardware. We present the graphs for the experiments with 3 Xboxes. The aggregate inter-arrival times for 4 Xbox games follow the same pattern but it is much harder to recognize it in the graphs.

Of the 3 Xboxes used in the presented experiments, two (x1 and x2) were transmitting packets at the same rate of one packet every 40 ms, while the 3rd Xbox (x3) was transmitting one packet every 40.001 ms. Figure 8 shows the packet inter-arrival time at the server of a game during which x1 was the server and x2 and x3 were the clients. The inter-arrival times in this figure are not pkthisto results (one histogram for 200 packets) but the inter-arrival times for all data packets captured with tcpdump. The red line represents the inter-arrival between packets from x2 and x3 while the blue line represents the gap between packets from x3 and x2. For example x2 starts transmitting at time 0, the next packet is transmitted at 40 ms, the next at 80 ms and so on; while x3 starts transmitting at 10 ms, then 50 ms, 90 ms… The gap x2-x3 is 10 ms while the gab x3-x2 is 30 ms. However, since one Xbox transmits one packet every 40 ms, while the other every 40.001 ms, the gaps in the inter-arrival times shift during the game causing the diamond shaped pattern.

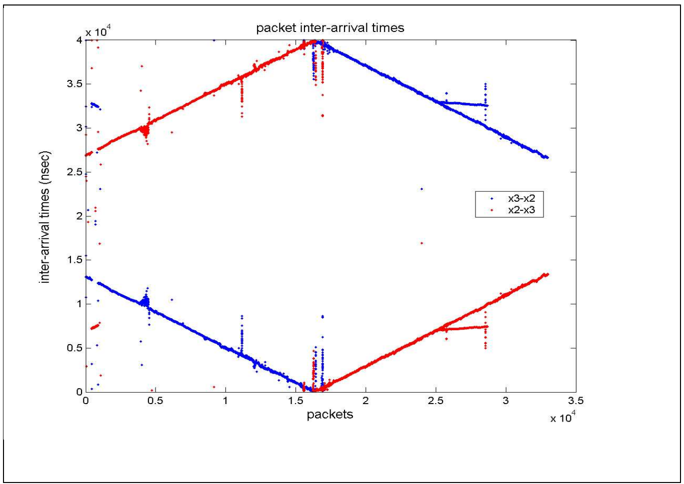

Figure 8: aggregate client to server packet inter-arrival times (tcpdump)

Figure 9 shows the same packet inter-arrival time plot, but this time after the processing with pkhisto. The same diamond shaped pattern of packet inter-arrival times can be observed.

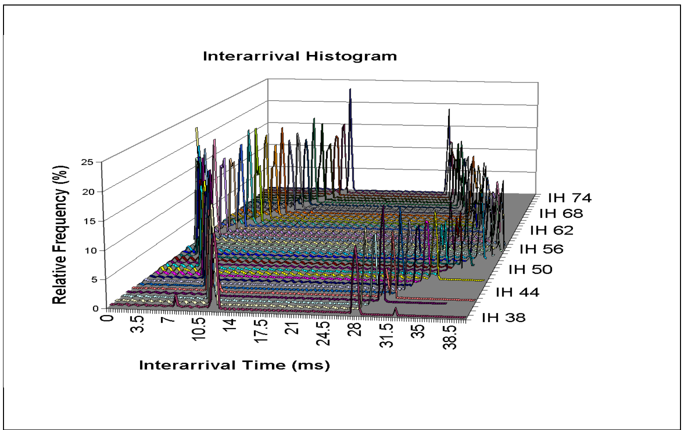

Figure 9: aggregate client to server packet inter-arrival times (pkhisto)

Figure 10 on the other hands presents a game in which x1 and x2 were the clients while x3 was the server. Since x1 and x2 both transmit packets every 40 ms, the gaps in the inter-arrival times remain the same from start to finish, resulting in the two lines at approximately 10 ms and 30 ms presented in the plot.

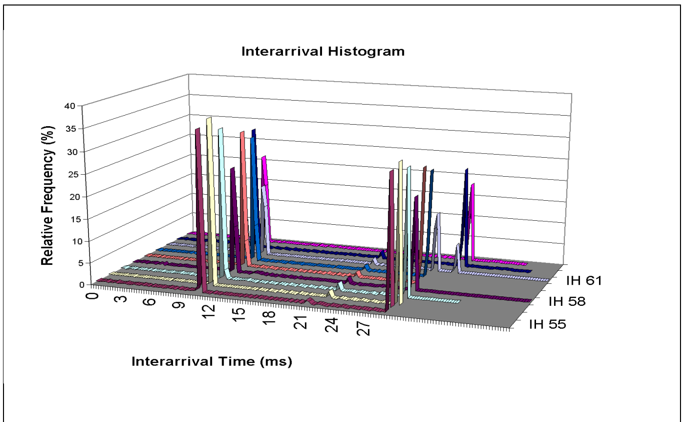

Figure 10: aggregate client to server packet inter-arrival time (z1 and x2 as clients)

D. Packet per second (PPS) and data rates

The number of packets sent per second remaind constant throughout all experiments conducted. PPS form the server to individual clients was 25 and aggregate server to client PPS was L*25 where L is the number of clients in the game. Individual client to server PPS was 30 and the aggregate client to server PPS was always equal to L*30.

Data rates are of course equal to PPS * pk-length and are therefore dependent on the number of total players for server to client flows and dependent on the number of players connected to the client Xbox for client to server flows. When discussing the packet lengths , we stated that pkhisto measures the length of the IP datagrams. The data rates presented here are therefore the rates of the IP traffic not the Ethernet traffic. A 14 byte header needs to be added to obtain the Ethernet packet lengths and the corresponding data rates.

Figure 11 shows the data rate estimated by pkthisto for the aggregate server to clients flow for a 4 Xbox game with 4 players. This estimated data rate is mostly equal to 129 kbps. For a 4 player game the server sends 216 byte packets and since 3 client Xboxes are present the PPS is 75, which results in a data rate of 129.6 kbps

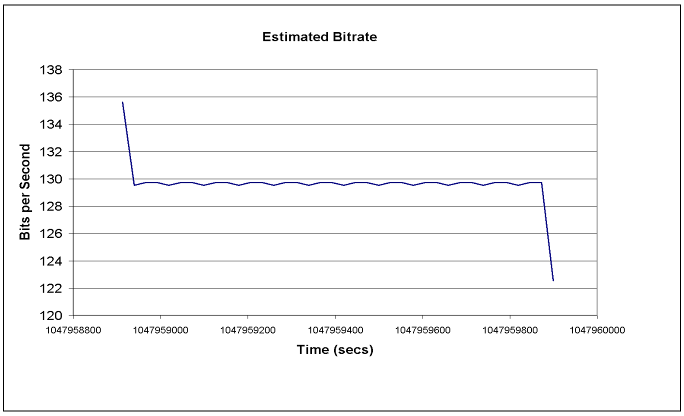

Figure 11: aggregate server to client data rate (estimated by pkthisto)

Figure 12 and Figure 13 show an individual and the aggregate client to server data rates. The same connection between packet sizes and PPS values is true as for the server to client flows. The individual client to server data rate is 25 kbps most of the time. The client to server PPS is 30; 25 packets per second are 112 bytes long (since one player is connected to the client) and 5 packets are the 72 byte long packets that are present in every game. These packet lengths and packet per second rates give 25.28 kbps. The aggregate client to server data rate is around 75 kbps, which is the sum of the individual client to server flows.

The knowledge of the packet per second rate and the packet length of the UDP payload, which is 28 bytes less than the IP datagram size, can be used to compute the expected data rate of Xbox traffic over any kind of underlying network. This is especially useful for users wanting to use an IP tunnel from their home to participate in a network game. IP tunneling enables System Link games via the Internet by transparently transmitting the Xbox traffic between the source and destination LAN.

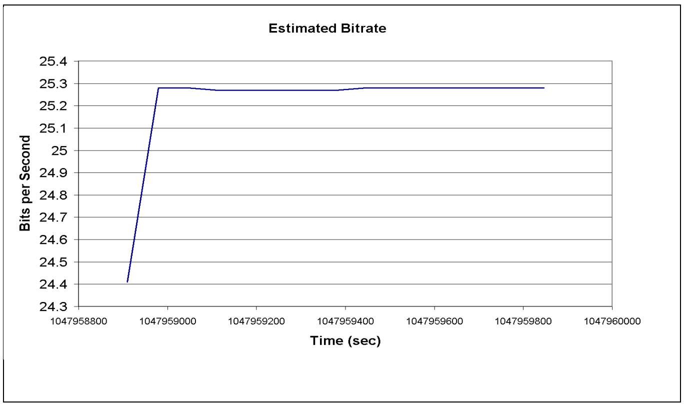

Figure 12: individual client to server data rate

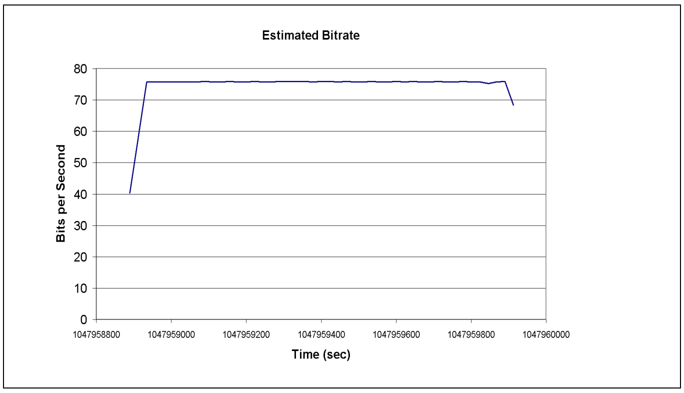

Figure 13: aggregate client to server data rate

III. Simulation Model Based on the Traffic Characteristics

The goal of our study was to developed traffic models that accurately describe System Link games. These models can be used to estimate the expected performance of an Xbox game over the Internet and to help ISPs when provisioning their network services. To achieve this, the traffic generated by the simulation model should be almost indistinguishable form traffic genarated by a real Xbox. We needed to process ns2 packet traces with pkthisto to be able to make this comparison. This was achieved by implementing a small programm converting ns2 trace files into tcpdumps. The resulting dumps were fed into pkthisto to create the graphs in Figure 14 to Figure 16.

A copy of the ns2 Xbox client and server models as well as the program to convert ns2 packet traces into tcpdumps can be obtained from the GENIUS web-site [11]

Packet lengths are modeled according to the formulae presented section II.B. The accuracy of these formulae compared to the actual data is already discussed in these sections.

The server model transmits a burst of ‘number of clients’ packets every 40 ms. The aggregate server to client inter-arrival times created by the server model in an 3 Xbox game are shown in Figure 14.

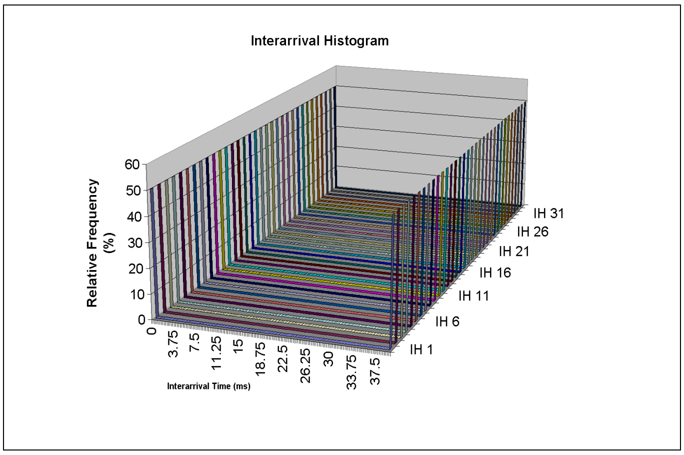

Figure 14: aggregate server to client packet inter-arrival times (ns2 mdoel)

Figure 15 and Figure 16 show the aggregate client to server packet inter-arrival times for two different 3 Xbox games. For the game in Figure 15 the two clients transmit the data packet at a slightly different rate, resulting in the diamond shaped inter-arrival time pattern also observed in Figure 9. Figure 16 on the other had shows the inter-arrival time pattern of a game where both clients transmit their data packets at exactly the same rate, equivalent to the game presented in Figure 10.

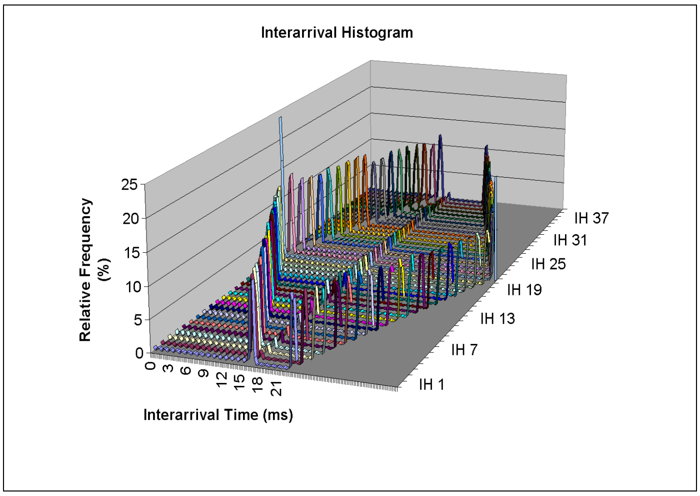

Figure 15: aggregate client to server packet inter-arrival times, 2 different transmission rate (ns2 mdoel)

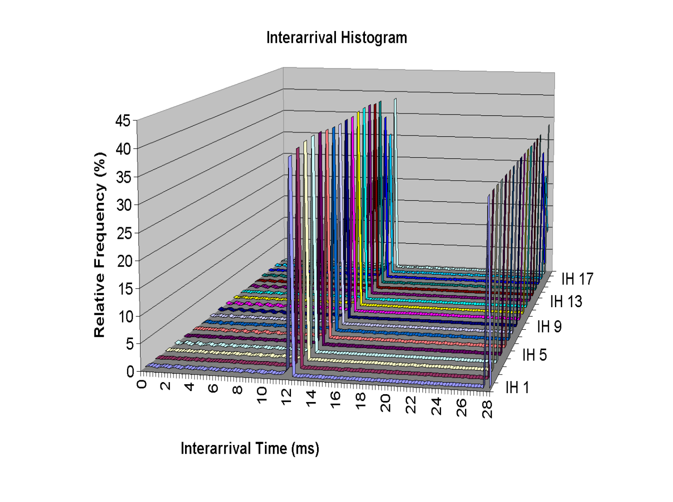

Figure 16: aggregate client to server packet inter-arrival times, 2 different transmission rate (ns2 mdoel)

IV. Conclusion

In this paper we presented our study on Xbox System Link traffic while playing Halo and a simulation model based on the observed traffic characteristics. Although Xbox System Link only supports inter-connection via LANs, programs such as XBConnect allow System Link games to be bridged over the Internet. The study and modeling of Xbox traffic characteristics can give an insight on the performance that players can expect when their Xboxes are connected via the Internet.

The analyses of Xbox traffic showed that it has very simple traffic pattern and is highly periodic. The important parameters are: total number of players in the game, number of clients and number of players connected to a client. The server packet lengths depend on the total number of players in the game and the server sends a burst of number of client packets every 40 ms. The client packet length depends on the number of players connected to the client and again one of these packets is sent every 40 ms. In addition to these packets each client transmits a 72 byte packet every 201 ms.

When observing aggregate packet inter-arrival times form the clients to the server, we found that they are dependent on the Xbox hardware. Some Xboxes used in our experiments transmitted packets at exactly 40 ms intervals while others sent one packet every 40.001 ms. Depending on which Xboxes were used as clients and server the aggregate inter-arrival times plots either display a diamond shaped pattern or two straight lines.

Based on these traffic patterns we developed ns2 models for Xbox server and Xbox clients. These models offer the same traffic load to the network as an actual Halo System Link game would and can be used to evaluate access networks for their suitability to support multi-user System Link games.

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] XBConnect, http://www.xbconnect.com (as of 6th June 2003)

[5] GameSpy Tunnel, http://www.gamespyarcade.com/support/tunnel_xbox.shtml (as of 6th June 2003)

[6] Xbox Gateway, http://www.xboxgw.com (as of 6th June 2003)

[7] Xbox Live, http://www.xbox.com/LIVE/default.htm (as of 6th June 2003)

[8] Xbox Live Community, http://www.xboxlivecommunity.com/index.php as of 6th June 2003)

[9] Tcpdump, www.tcpdump.org (as of 3rd of April 2003)

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

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