Network Analysis of Counter-strike and Starcraft

Mark Claypool, David LaPoint, Josh Winslow
{claypool}@cs.wpi.edu
Computer Science Department
Worcester Polytechnic Institute
100 Institute Road, Worcester, MA 01609, USA

Abstract

Network games are becoming increasingly popular, but their traffic patterns have received little attention from the academic research community. A better understanding of network game traffic can lead to more effective network architectures and more realistic network simulations. We gathered traffic traces from live Internet games of Counter-strike and Starcraft, representative games from two different gaming genres. In this paper, we analyzed the network traffic patterns and bandwidth consumption to better understand how such games impact the network. Both games were found to have very low bandwidth usage and packets significantly smaller than typical Internet applications. Starcraft games showed consistent traffic behavior across games while the traffic patterns for Counter-strike varied widely, especially for the server. In particular, Counter-strike clients and servers often send bursts of very small packets. Since current Internet routers are typically designed for large transfers with large packets, there may be opportunities to improve network architectures to better manage and support game traffic.

1 Introduction

The rapid growth in the connectivity of the Internet and the fall in computer hardware prices has fueled the growth of multi-player gaming over the Internet. In addition to computers, the latest generation of consoles systems from Nintendo, Sony, and Sega all include network support for multi-player game play on the Internet. The increase in Internet gaming is also a result of change in developer focus from single player games to multi-player games.

Over the years, there have been a number of studies measuring the performance of Internet backbones and end-hosts [13, 14], as well as detailed studies on the performance of Web clients [5, 8] and streaming media [7, 10, 15]. However, to the best of our knowledge there have not been wide-scale empirical measurement of multi-player network game traffic across the Internet. While the existing empirical studies have been valuable in helping understand Internet performance, they are not sufficient for characterizing network game traffic since network games have different application requirements, and hence different network traffic characteristics, than the majority of Internet traffic.

Unlike typical Internet traffic, network games typically do not use reliable Internet transport protocols. The majority of reliable Internet traffic runs on top of TCP, which guarantees sequential, in order arrival of all data by retransmitting lost packets. However, the additional time required for retransmissions and the window-based data rate imposed by TCP can be unacceptable to network games since interactive games often have strict delay constraints [2]. Instead of TCP, network games often use UDP since UDP packets are not guaranteed to reach their destinations, and thus do not require a mechanism for retransmissions. Unfortunately, UDP does not provide any built-in congestion control, presenting the risk of congestion collapse as the fraction of unresponsive UDP traffic increases [4].

Despite the very real burden that network games may impose on network bandwidth and the massive growth and large user base, issues related to the effects of these games on network congestion have been largely neglected in both academic and industry publications. Industry articles are more concerned with the management aspects of game development or focus on latency and maximum bandwidth issues only [3, 6, 11], whereas attention to detail like minimizing network load suffer due to the lack of economic pressure. Network gaming has not been a traditional research field in academia as network games are viewed primarily as a diversion for students, rather than a practical problem for computer scientists. Among other things, a better understanding of network game traffic can help design networks and architectures that more effectively accommodate their traffic footprints. Furthermore, careful empirical measurements of network games can provide the data required for accurate simulations, a typical tool for evaluating network research.

The goal of this research is to analyze the network traffic of two of the most popular games on the Internet. We chose two representative games from the most popular genres, First Person Shooters and Real Time Strategies (see Section 2.1), with the intent of representing a larger range of data than a single genre would likely provide. We used a network sniffer on both game servers and game clients participating in live Internet games in order to capture realistic network traffic. We then analyze the distribution of packet sizes, packet interarrival times, differences between client and server bandwidths and impact of number of players on network traffic.

The rest of this paper is laid out as follows: Section 2 details our approach in selecting network games and running our experiments; Section 3 provides results and analysis of the data traces collected during the experiments; Section 4 briefly describes how results from Section 3 could be used to simulate network games; and Section 5 summarizes our conclusions and presents possible future work.

2 Approach

We employed the following methodology: select network games for study (Section 2.1), run the selected games in an instrumented environment (Section 2.2), and analyze network traffic the games generate (Section 3).

2.1 Selecting Games

Broadly, the most popular categories of real-time network games are *First Person Shooters* (FPS) and *Massively Multi-player Online Role Playing* (MMORP), followed closely by *Real Time Strategy* (RTS).

FPS games, first made popular by Doom[1], have the player view the world through the perspective of a character (the first person) and have the player move around slaying monsters and other players, with an amalgamation of ranged weaponry (the shooter). On an average night, there are well over 10,000 servers Half-Life[2] games supporting over 40,000 players[3]. Other FPS games support slightly smaller user populations[4].

MMORP games, first made popular by Ultima Online's release in 1996[5], are similar to graphical Multi-User Dungeons, first pioneered in the 70s[6]. MMORP games provide a mechanism for character advancement, large lands to travel across, and other players with whom to interact. The three biggest MMORP games, Asheron's Call[7], Ultima Online[8], and Everquest[9], claim to have nearly 1 million subscribers combined.

RTS games, first made popular by Dune 2[10], are generally characterized by resource collection, unit construction, and battles that consist of large numbers of animated soldiers going through a repetitive, animated attack. All of these actions happen in real-time, unlike earlier strategy games such as Civilization[11], in which the player could take as much time as needed to plan the next turn. Since Dune 2, there have been several more RTS games released, one of the more popular being Starcraft[12]. Currently, the typical number RTS fans playing Starcraft on an average night numbers about 20,000 players.

We selected Starcraft, a RTS game, and Counter Strike, a FPS game, because they were familiar, best selling games in different genres. While we wanted to study Asheron's Call, a MMORP game, the time it took to setup and run our experiments precluded studying another genre. We leave that as future work.

2.1.1 The Starcraft Game Environment

Starcraft[13] is a RTS game that has players construct buildings and fighting units, and issue commands that cause the units to move, engage enemy units, and similar tasks. Games are played on one of many possible maps, either provided with the game or custom built by users. There are three races from which a player can choose, and each has a balanced set of advantages and disadvantages over the others. There are a number of ways in which players can be competitively grouped. In a free-for-all game, all players vie to have the last remaining army on the map. Players can also team up against each other and/or against artificially intelligent computer-controlled players in myriad ways.

For our experiments, all games were structured so that there were two teams of equivalent sizes: 1 vs.

[1] http://www.idsoftware.com/games/doom/doom-final/

[2] http://half-life.sierra.com/

[3] Average of 3 randomly selected nights between 10/2/01 and 12/15/01.

[4] Tribes 2, based on an average over 3 randomly selected nights between 10/16/01 and 12/15/01.

[5] http://www.uo.com

[6] http://www.legendmud.org/raph/gaming/book.html

[7] http://www.zone.com/asheronscall/, averages 12,000 players per night

[8] http://www.uo.com, over 300,000 total subscribers

[9] http://everquest.station.sony.com/, over 410,000 total subscribers

[10] http://www.dune2k.com/duniverse/dune2/

[11] http://www.civ3.com, the second sequel of Civilization

[12] http://www.blizzard.com/worlds-starcraft.shtml

[13] http://www.blizzard.com/worlds-starcraft.shtml

1, 2 vs. 2, 3 vs. 3, 4 vs. 4. The local player, analyzed in detail in Section 3, played as the same race in each game, and employed the same building strategy throughout. The games were played using Starcraft: Brood War, version 1.7. The local player logged on to `battle.net` using the USEAST gateway, and created the game sessions. Each game type was top players vs. bottom players. The same map, called Big Game Hunters found in the `maps/broodwar/webmaps` directory from where the game was installed, was used for each game.

2.1.2 The Counter-Strike Game Environment

Counter-strike[14] is a modification to the FPS Half-Life that is distributed free over the Internet for owners of Half-Life, or as a retail product in most game stores. Counter-strike puts the players in the role of either a terrorist attempting to hold hostages, blow up landmarks, or assassinate a VIP or a counter-terrorist agent trying to thwart the terrorists. To play, each player must connect to a server, usually located on an end-host on the Internet. When one or more other players join the same server, a game begins.

All Counter-strike games are played on a pre-loaded map, with each map having its own set of victory objectives. Most objectives have the counter-terrorists attempting to rescue a set of hostages from close to where the terrorists start, or the terrorists attempting to plant a bomb close to where the counter-terrorists start. Typically, each map is played several times, with each time being called a round, lasting several minutes. Each round ends either when the victory objectives are met, time runs out, or when one team has been totally eliminated. At the start of each round, both sides are allowed to buy weapons and ammunition with the money they earned from the previous rounds. The better a team did in the previous round, the more money they have to spend. Once each team is equipped, they attempt to completely wipe out the other team with their weaponry or complete the objective, although the former outcome ends far more rounds than the latter.

Controlling the environment in a real Counter-strike game is difficult because players can join and leave as they please. We chose to analyze several different maps to see the effect the campaign has on network traffic. The number of players was not controlled, as players joined and left the server in a typical Counter-Strike gaming fashion. We used Counter-strike version 1.3 on Half-life version 1.1.0.8. The maps used were `de_dust`, `de_aztec`, and `cs_assault`, all from the standard Counter-strike install. The Counter-strike server we connected to was located on the WPI network.[15]

[14] http://www.counter-strike.net

2.2 Running Games

We used Commview (version 2.6, build 103)[16] to capture all network traffic related to the games ran. Commview is a robust network packet sniffer with the ability to filter packets, compute statistics, and generate reports periodically.

All Starcraft data was collected on the machine:

```
Intel Pentium III 800mhz processor with 100mhz FSB
512 megabytes PC-100 SDRAM
nVidia geForce2 3d w/64 Mbytes of DDR SDRAM
UltraWide SCSI hard drive interface
10baseT nic connected to 608/108 kbps DSL modem
Windows 98B Operating System
```

All Counter-strike data was collected on the machine:

```
AMD Athlon 800mhz processor with 200mhz FSB
256 megabytes PC-100 SDRAM
nVidia geForce 3d w/32 Mbytes of DDR SDRAM
ATA-66 hard drive interface
10baseT nic connected to WPI LAN
Windows 98 v4.10.98 Operating System
```

3 Results and Analysis

This section presents results and analysis for bandwidth (Section 3.1) and packets (Section 3.2), for both Starcraft and Counter-strike.

3.1 Bandwidth Analysis

Figure 1 depicts the bandwidth sent by each player for 2 player, 4 player, 6 player and 8 player Starcraft games. The amount of bandwidth sent is linear with the number of game players with a 2-player game sending around 650 bytes/second and each pair of additional players adding about 1500 bytes/second of data. Also, as the number of players increases, so does the variance in bandwidth sent (standard error of 32% for a 2-player game and 41% for an 8-player game).

Figure 2 depicts the bandwidth sent by a Counter-strike game client using the `cs_assault` map for the first 20 minutes and then switching to the `de_aztec` map. The average bandwidth sent is around 2800 bytes/second. There is considerable variation in the bandwidth per second (standard error of 110%), with a noticeable drop in bandwidth at 1100 seconds when there was a map change.

[15] http://www.wpi.edu/Admin/Netops/MRTG/

[16] http://www.tamos.com/products/commview/

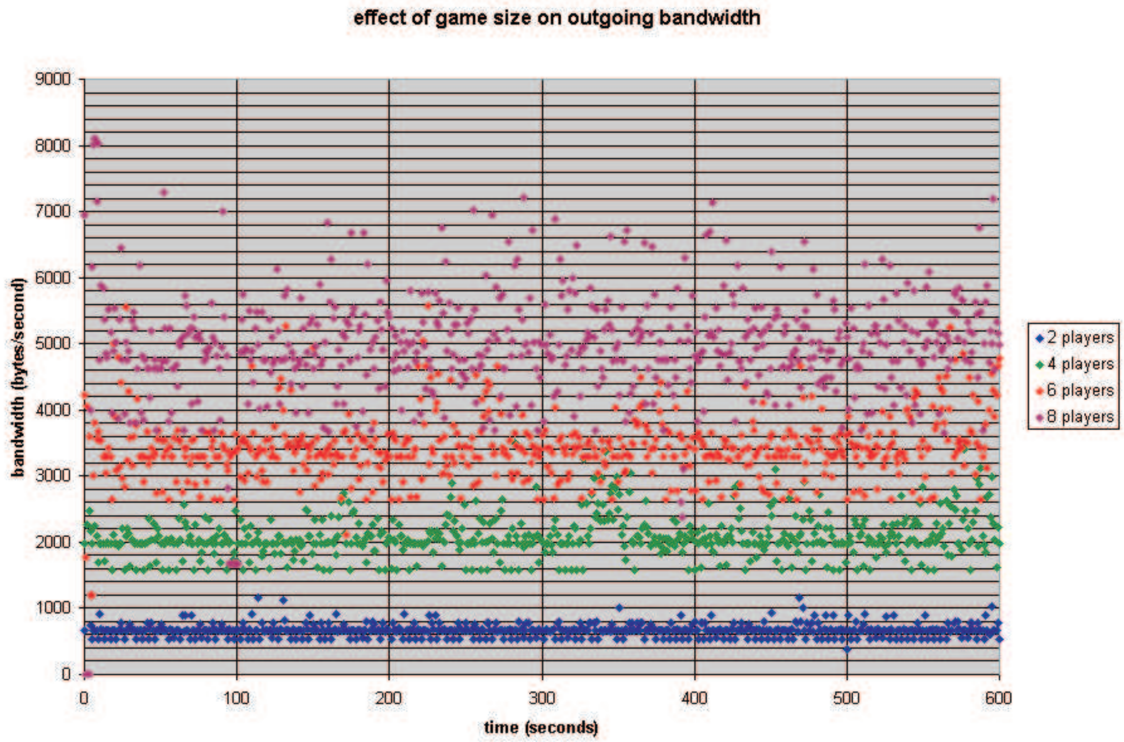

Figure 1: Starcraft Bandwidth Sent for Different Game Sizes

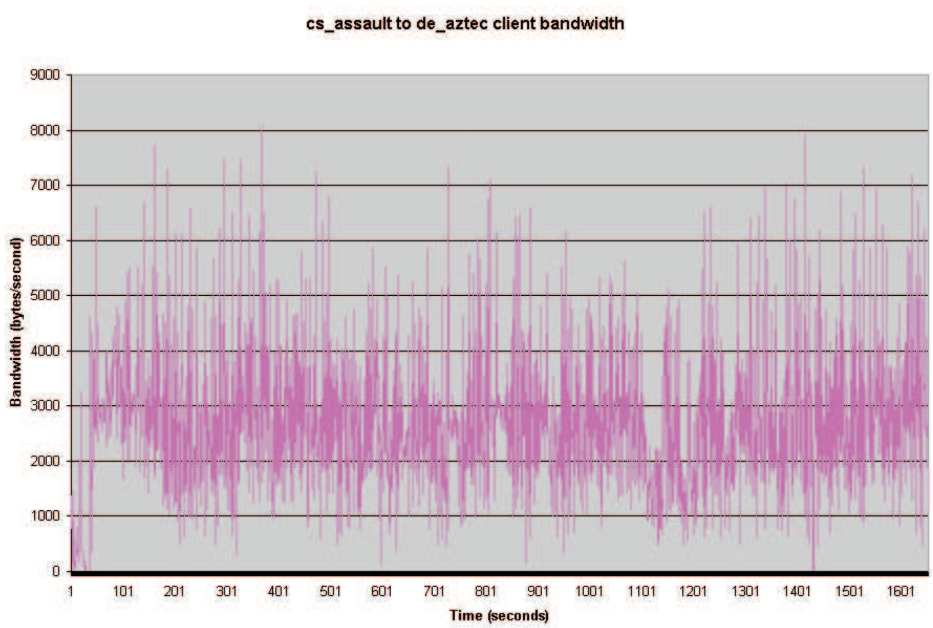

Figure 2: Counter-strike Bandwidth Sent by Client

Figure 3 depicts the bandwidth sent by the server for the same Counter-strike 3-person game. The average bandwidth is around 6470 bytes/second, with considerable variation in the bandwidth per second (standard error of 135%). The map change occurs at about 1110 seconds, depicted by the drop in bandwidth.

Overall, Starcraft shows considerably more bandwidth consistency than either the Counter-strike client or the Counter-strike server. The largest Starcraft game possible, an 8-player game, is still sufficiently optimized to play at modem speeds. Counter-strike clients can similarly play at modem speeds (overall, our clients averaged about 1200 bytes/second), but since Counter-strike servers can support a much higher number of players than Starcraft (games with 20 players are not uncommon), server bandwidth may become constrained over a modem.

3.2 Packet Analysis

Figure 4 depicts a cumulative density function (CDF) of packet sizes (both inbound and outbound) for Starcraft and Counter-strike. Both games produce small packets, with the median packet size for Starcraft about 120 bytes and the median packet for Counter-strike clients about 160 bytes. Both Starcraft and the Counter-strike client exhibit a narrow range of small packets. The Counter-strike server, however, has a much heavier tail, with 10% of all packets being over 500 bytes.

While a Starcraft game has very little deviation in terms of packet size or bandwidth used throughout the run of one game, Counter-strike, especially from the viewpoint of the server, has a non-uniform, but distinct network pattern. Figure 5 depicts the packet sizes sent over time for a 3-player Counter-strike game using the `cs_assault` map for the first 23 minutes and then switching to the `de_aztec` map. Rounds can be distinguished by the slow decline, caused by players dying, in the packet sizes sent from the server. For example, one round goes from approximately 500s to 650s, featured above in the first small box. The mean packet size is 465 bytes. It seems likely that the circled large packets of nearly 3000 bytes are round initialization or round termination packets, as they strongly correlate with the end of the decline in packet sizes. The ovals correlate with large firefights within the game. The break in the graph around 1400 seconds, indicated by the second large box, is where the map was changed to `de_aztec`.

Figure 6 depicts depicts a CDF of packet interarrivals for Starcraft and Counter-strike. The Starcraft packets arrive fairly uniformly over a range of 10 to 300 milliseconds, as evidenced by the gradual slope. Counter-strike, on the other hand, has about 30% of its packets arrive in large bursts.

Overall, both Counter-strike and Starcraft send considerably smaller packets than the typical Internet traffic packet size of over 400 bytes [9]. The number of players does not have a noticeable effect on the packet sizes for either Starcraft or Counter-strike. Starcraft packet sizes are consistent throughout the game, but Counter-strike packet sizes, on the other hand, are greatly influenced by the action in the game. Counter-strike sends periodic bursts of small packets while Starcraft send packets at a much steadier rate. Since current Internet routers are designed for large transfers with large packets, there may be opportunities to improve network architectures to better manage and support game traffic.

4 Simulation of Network Game Traffic

Empirical experiments with network games are often difficult because of variable network conditions and the costs involved with deploying large numbers of network game clients under controlled conditions. However, by using simulators, such as NS-2 [12], the previous section results may be useful for network game protocol designs, new network router queue management disciplines that react to network game traffic, and understanding interactions between network game traffic and traditional traffic. We briefly sketch out the design of traffic generators that could be used in such simulations.

First, separate traffic generators should be chosen for Starcraft and Counter-strike, with Counter-strike traffic generators separated into client and server components. One can model an overall network game client by selecting an RTT based on average conditions seen in [2]. Packet interarrivals can be probabilistically selected based on data in Figure 6 and the packet sizes can be selected based on data in Figure 4. Once developed, bandwidth consumption can be validated using the graphs in Figures 1, 2 and 3.

5 Summary

The network traffic generated by Starcraft and Counter-strike look significantly different. There is a significant degree of variation in the packet sizes and bandwidth used in Counter-strike games, in contrast to Starcraft in which the differences between sizes of packets sent are barely distinguishable. For larger games, Starcraft does have more packets transmitted when the bandwidth requirements increase. The Counter-strike client also follows this model, although the packet sizes are more variable than those in Star-

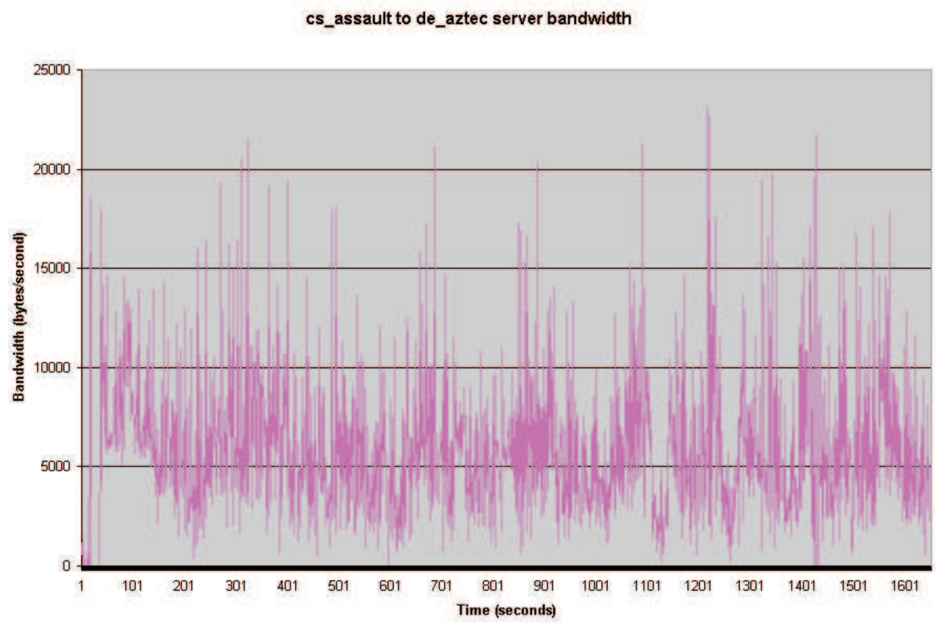

Figure 3: Counter-strike Bandwidth Sent by Server

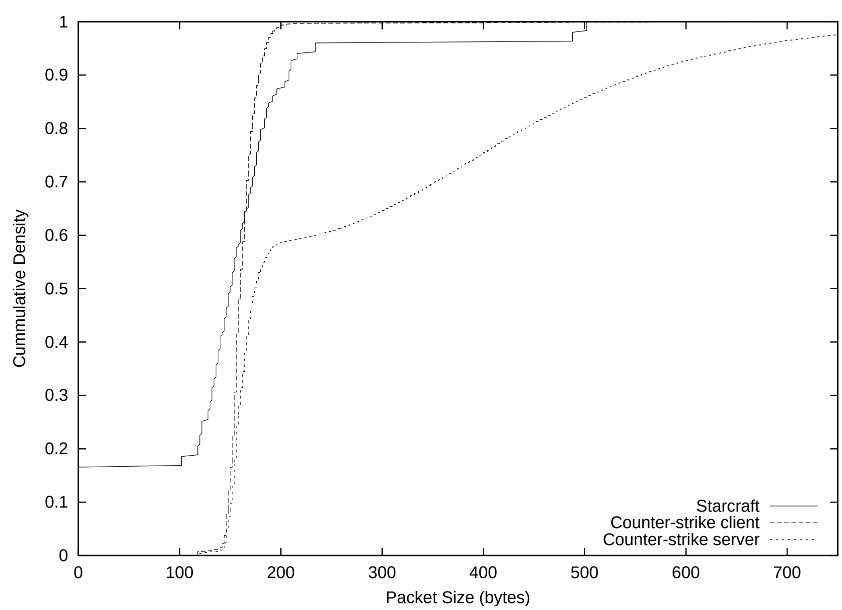

Figure 4: Packet Sizes

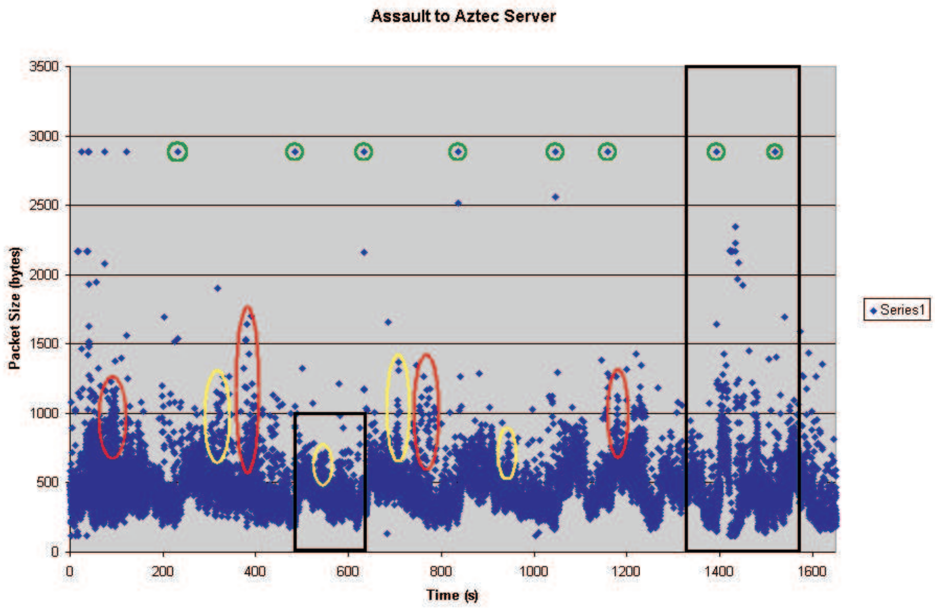

Figure 5: Sizes of Counter-strike Packets over Time Sent by Server

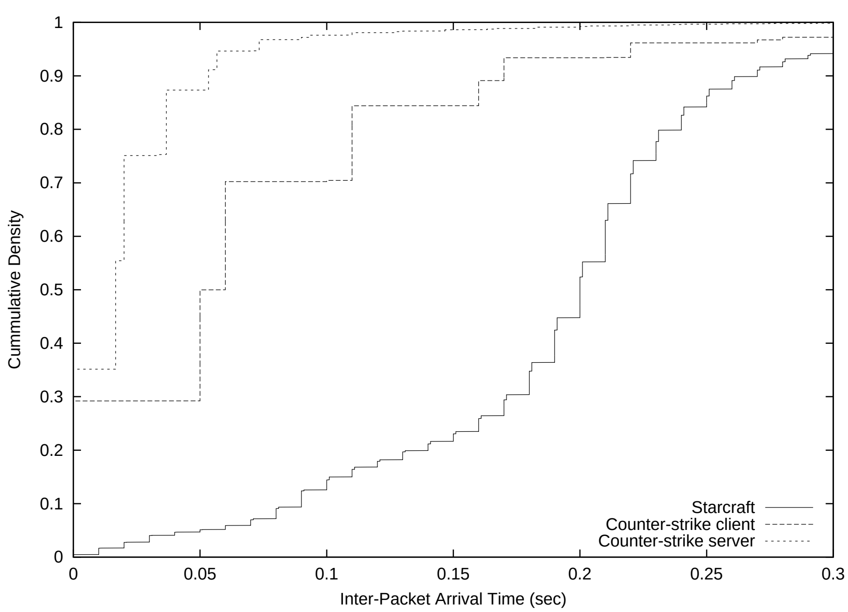

Figure 6: Packet Interarrival Times

craft. The Counter-strike server, on the other hand, increases the size of the packets when it needs to send more data to the client. These differences are not visible when viewing a bandwidth graph but are important to note due to their possible effects on network congestion.

Starcraft and Counter-strike are also different in the bandwidth they consume over time. Starcraft's bandwidth consumption varies little over the course of a game regardless of events occurring within the game. Counter-strike, however, has a distinct cyclic pattern in bandwidth, which vary over time as players get killed, and have a marked correlation to game events. Overall, the amount of bandwidth consumed by a Starcraft player is comparable to bandwidth consumed by a Counter-strike client. A 6-player game of Starcraft has each client send between 3000 and 3500 bytes/second, and a Counter-strike client connected to a mostly-full server typically sends a little over 3200 bytes/second.

Counter-strike traffic exhibits a bursty pattern, with 30% of the packets arriving in single bursts. Starcraft traffic, on the other hand, is much more uniform in its arrival.

Overall, the traffic patterns for Counter-strike and Starcraft are optimized for clients with the limiting bandwidth of a modem connection. Typical Web browsing and streaming audio can also be accommodated by modem connections. However, both Counter-strike and Starcraft exhibit different packet sizes than typical audio or Web packet sizes, with audio traffic having a median packet size of about 300 bytes [10] and Web traffic having a median packet size of around 1500 bytes [1]. Moreover, the inter-arrival times for Counter-strike and Starcraft are significantly different than streaming audio, with audio being bursty with a median departure time of about 1 second [10]. These differences present the opportunity to design networks and simulations of networks that more effectively support network game traffic above and beyond the current support for streaming media and Web traffic.

Notes

The authors would like to thank the anonymous game players who helped obtain the packet traces. All traces used in providing the analysis for this paper and a Java-based tool to help parse Commview output can be downloaded from http://perform.wpi.edu/downloads/#net-game. The authors would also like to thank the anonymous reviewers for their suggestions on how to improve an earlier draft of this paper.

References

[1] Mark Allman. A web server's view of the transport layer. *ACM Computer Communication Review*, 30(4), October 2000.

[2] Grenville Armitage. Lag Over 150 Milliseconds is Unacceptable, May 2001. [Online] http://gja.space4me.com/things/quake3-latency-051701.html.

[3] Paul Bettner and Mark Terrano. 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond. *Gamasutra*, March 2001. [Online] http://www.gamasutra.com/features/-20010322/terrano_02.htm.

[4] Sally Floyd and Kevin Fall. Promoting the Use of End-to-End Congestion Control in the Internet. *IEEE/ACM Transactions on Networking*, February 1999.

[5] Balachander Krishnamurthy and Craig E. Wills. Analyzing Factors that Influence End-to-End Web Performance. *WWW9 / Computer Networks*, 33(1-6):17–32, 2000.

[6] Peter Lincroft. The Internet Sucks: Or, What I Learned Coding X-Wing vs. Tie Fighter. *Gamasutra*, September 1999. [Online] http://www.gamasutra.com/features/19990903/-lincroft_01.htm.

[7] Dmitri Loguinov and Hayder Radha. Measurement Study of Low-bitrate Internet Video Streaming. In *In Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, November 2001.

[8] Bruce A. Mah. An Empirical Model of HTTP Network Traffic. In *Proceedings of IEEE INFOCOM*, pages 592–600, April 1997.

[9] S. McCreary and k. claffy. Trends in Wide Area IP Traffic Patterns: A View from Ames Internet Exchange. In *Proceedings of ITC Specialist Seminar on Measurment and Modeling of IP Traffic*, pages 1 – 11, September 2000.

[10] Art Mena and John Heidemann. An Empirical Study of Real Audio Traffic. In *In Proceedings of the IEEE Infocom*, pages 101 – 110, March 2000.

[11] Yu-Shen Ng. Designing Fast-Action Games for the Internet. *Gamasutra*, September 1997. [Online] http://www.gamasutra.com/features/-19970905/ng_01.htm.

[12] Universiy of California Berkeley. The Network Simulator - ns-2. Interent site `http://www.isi.edu/nsnam/ns/`.

[13] Vern Paxson. End-to-End Internet Packet Dynamics. *IEEE/ACM Transactions on Networking*, Fall 1999.

[14] K. Thompson, G. Miller, and R. Wilder. Wide-Area Internet Traffic Patterns and Characteristics. *IEEE Network*, pages 10–23, November 1997.

[15] Yubing Wang, Mark Claypool, and Zheng Zuo. An Empirical Study of RealVideo Performance Across the Internet. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, November 2001.