

Traffic Modelling for Fast Action Network Games

Johannes Färber

*University of Stuttgart, Inst. of Communication Networks and Computer Engineering
Pfaffenwaldring 47, 70569 Stuttgart, Germany*

E-mail: faerber@ind.uni-stuttgart.de

ABSTRACT

A significant share of today's Internet traffic is generated by network gaming. This kind of traffic is interesting in regard to its market potential as well as to its real time requirements on the network. For the consideration of game traffic in network dimensioning, traffic models are required that allow to generate a characteristic load for analytical or simulative performance evaluation of networks. In this paper the fast action multiplayer game „Counter Strike“ is evaluated based on one month of Internet traffic traces and traffic models for client and server are presented. The paper concludes with remarks on QoS metrics for an adequate assessment of performance evaluation results.

1. INTRODUCTION

Network game traffic generates a significant share of today's Internet traffic. In [2] it is reported that 3-4% of all packets in a backbone could be associated with only 6 popular games. A high market potential, increasing usage as well as sharp real time requirements make this kind of traffic interesting for Internet service providers and manufacturers. In order to profit from the high popularity of online gaming, networks [8] are enhanced for gamers by optimizing components and protocols for game traffic. In order to test the efficiency of such measures before their realization they are applied to a system model, i.e. an abstract description of the communication network. For a performance evaluation of this modified system under realistic load, either by mathematical analysis or by simulation, traffic models are needed for all traffic generating participants.

In 1999 Borella presented a traffic model for the first person shooter „Quake 2“ [5]. Since then many successful multiplayer games have been developed. Although there are other popular online game genres emerging with more focus on strategy or roleplaying, first person shooters are still the most popular multiplayer games found in the Internet and they impose the hardest real time requirements on the network.

We choose to characterize the traffic patterns of „Counter Strike“, a very popular first person shooter based on the Quake engine. In [7] we have presented first results of this evaluation which shows, that Borella's findings for „Quake 2“ are in general still valid for today's games.

In „Counter Strike“ players join one of two teams and attack or defend against the other team. It is a very fast paced game where a player's „life“ usually ends within few minutes. „Respawning“, i.e re-entering the match with a new „life“, is not allowed until the next turn which has a maximum duration, e.g. 6 minutes. The games communication model follows the client server approach and uses UDP packets for the exchange of small update information.

While the results in [7] are based on game traffic captured from a LAN party with 50 participants, a large set of traces of Internet game traffic was available for this work. The traces were taken with tcpdump at a dedicated game server hosting a single game for 1-32 players during the whole month of may 2001. Basically the traces hold for each UDP packet a timestamp, source and destination address and the payload size. With around 35000 different clients accessing the server and exchanging 20 million packets per day, the capture files are large. Table 1 shows an overview on the first 3 traces and reveals that much less than 35000 clients per day participated actively in a game, i.e. only 200-300 clients sent more than 2400 packets (corresponding to 1 minute of in-game traffic) or 12000 packets (5 minutes). Most clients only ping the server for lag and game information without joining a game.

In section 2 we describe a characterization of the patterns of client and server generated traffic with focus on packet size and packet interarrival times. In section 3 we present a traffic model framework and the corresponding parameter descriptions for client traffic and server traffic per client. In section 4 we conclude with some thoughts on quality of service metrics for network games.

Table 1: Trace overview

Trace	Description
01_05_2001	29003 clients in total, 15 mio packets 293 clients exchanged > 2400 p. 218 clients exchanged > 12000 p.
02_05_2001	32319 clients in total, 19 mio packets 321 clients exchanged > 2400 packets 237 clients exchanged > 12000 packets
03_05_2001 ...	38698 clients in total, 19 mio packets 343 clients exchanged > 2400 packets 268 clients exchanged > 12000 packets

2. TRAFFIC CHARACTERISTICS

2.1 General Traffic Characteristics

As most other first person shooters, Counter Strike is based on a client server communication model (see Figure 1). Before a match can start, someone has to setup a game server, either dedicated (the machine only runs the server software) or non-dedicated (i.e a player uses the server machine to participate). Then, players have to find the server by requesting server addresses from a central master server where public servers register on setup. They check for a low delay to the server and if enough other players are present and join the game. For the remainder of the paper we put focus on in-game traffic only, i.e. all this pre game communication is not subject to our analysis.

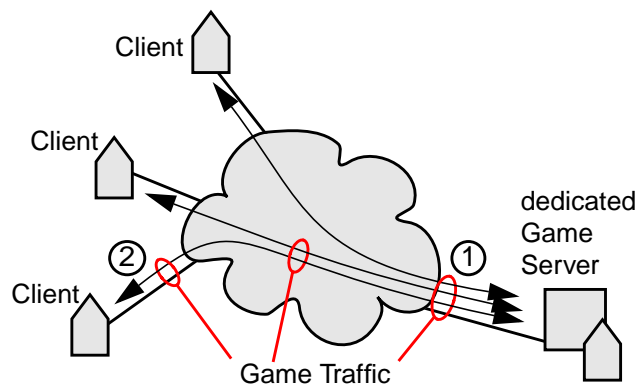


Figure 1: Client server architecture and different views on game traffic

The observed in-game traffic follows the transmit cycle already described in [5]: the server sends game state information to each client where packets are read and processed. Clients synchronize the server game state with their local game state, process player commands and return update packets with the players movement and status information. Since slower client machines require more processing time for rendering, their packet rate may be lower. Both, client and server packets are usually very small since they only contain movement and status information.

Figure 2 shows typical traffic rate plots of LAN traffic (from [7]). While client generated traffic shows almost constant behaviour for packet rate and data rate, traffic sent from server to clients is more variable but constantly high during a game turn. Between turns server rates may drop to zero for a short time.

The variation observed for the server significantly depends on the number of active players, as each active player receives an information packet, i.e the server sends a burst of packets during each transmit cycle. When players join, pause or leave a game, the burst length changes and thus the traffic rate. Note, that we look on the total traffic rate as seen at (1) in Figure 1. Traffic rates from server to one client, as observed at (2), do not vary that much. Thus, it makes sense to evaluate the server traffic to each client separately instead of its total traffic.

When many players are active in a match, we assume that additional information on the other players transferred to every client leads to increasing per client traffic rates. Although in [7] we have not found such a behaviour, now the large amount of data in fact reveals that server traffic changes. In order to find the rough scaling behaviour, we evaluate the packet rates for phases of different numbers of active players separately. For each trace we check in a 30 second sliding window for clients sending or receiving more than an average of 1 packet in 5 seconds. Those clients are considered to be active and their mean data rate and packet rate for the current window is determined.

Figure 3 and Figure 4 show the resulting mean traffic rates for client to server as well as server to client traffic over the number of active clients. We see, that with an increasing number of participants the average packet rates for clients and server stay approximately constant (the slight decrease in server packet rate is probably due to processor load). However the average data rate for the server to client traffic increases linearly while the client to server data rate remains constant. We observe an average server packet size of 50.4 Bytes + 6.15 Bytes per additional client (without UDP header).

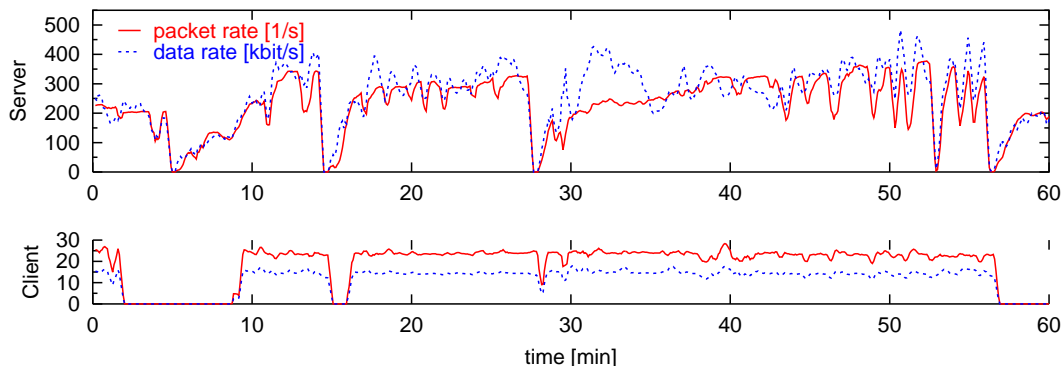


Figure 2: Example of server and typical client traffic of a 1h session (LAN traffic)

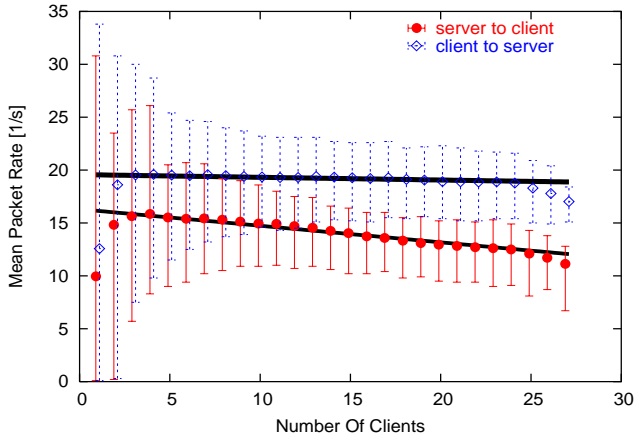


Figure 3: Change of mean packet rate with number of active clients (99% conf.)

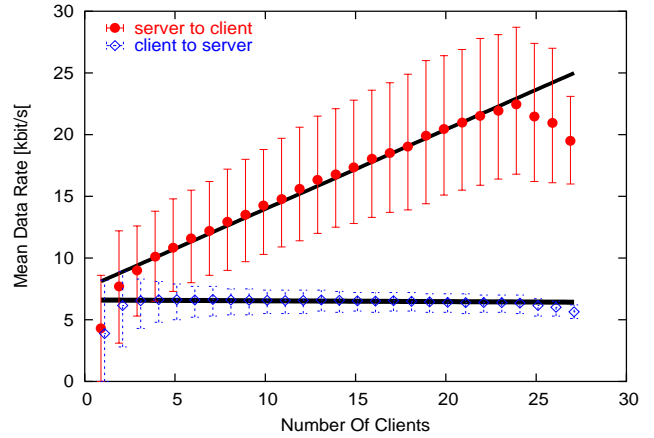


Figure 4: Change of mean data rate with number of active clients (99% conf.)

2.2 Server Traffic

The main characteristic of the server traffic is its bursty nature caused by information packets being sent to each client in every transmit cycle. We characterize this traffic by

- the interarrival time of bursts
- the length of the burst (i.e. number of clients) and
- the packet size (depending on number of clients).

Figure 5 shows the complementary cumulative distribution function (ccdf) of server packet interarrival times of the total traffic, i.e. the probability that the interarrival time exceeds the x-value. Around 70% of all packets are sent immediately one after the other, i.e. obviously the interarrival time within a burst. A significant share of around 25% of all packets shows an interarrival time of 20 ms and still 4% are sent after a gap of 40 ms. Less than 1% show 60 ms.

In order to explain this pattern, we have to look into per client traffic statistics. The traffic to the most active clients in the first trace was evaluated separately. Figure 6 shows the probability density function (pdf) of the packet interarrival time for 8 of those server to client communications. The statistics for the single clients are depicted mainly for illustra-

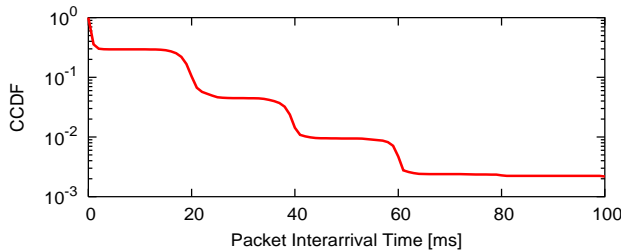


Figure 5: Complementary cumulative distribution function of server interarrival time (total traffic)

tion of the large differences between clients. The interarrival time pdf of all 44 evaluated clients (traces were concatenated for evaluation) shows a peak at 60 ms with small surrounding peaks every 10 or 20 ms. The Extreme value function depicted in the figure is fitted to that pdf. For each evaluated client we determine the mean of all interarrival times smaller than 1 second (to remove very large pauses for averaging). In average this mean is found at 64 ms with an average coefficient of variation of 0.2.

Note, that the interarrival time of client packets is almost equivalent to the interarrival time of the bursts the packets are sent in. With the results from Figure 5 we assume that the server is not always sending one single burst per transmit cycle but e.g. depending on the number of clients either one (3.3%), two (26.7%) or three (70%) bursts assigning clients to one of those sub cycles. As we like to keep the characterization simple, we abstain from a detailed verification and description of this behaviour.

Figure 7 shows the ccdf of the server packet interarrival time per client. The log-log plot reveals the tail behaviour of the data: there is a significant probability for interarrival times much larger than the mean. However, large interarrival times most certainly do not belong to regular in-game traffic and are not considered for further in-game traffic characterization.

The payload size of server generated packets shows a higher variability (see Figure 8 and Figure 9). Around 99% of all packets were smaller than around 300 Bytes and of course no packet was larger than the MTU of 1500 Bytes. A significant fraction of server packets reaching a size of about 1000 Bytes may be assigned to gameplay interruptions e.g. due to an end of turn or a change of scenario in which cases more information has to be transferred to the clients.

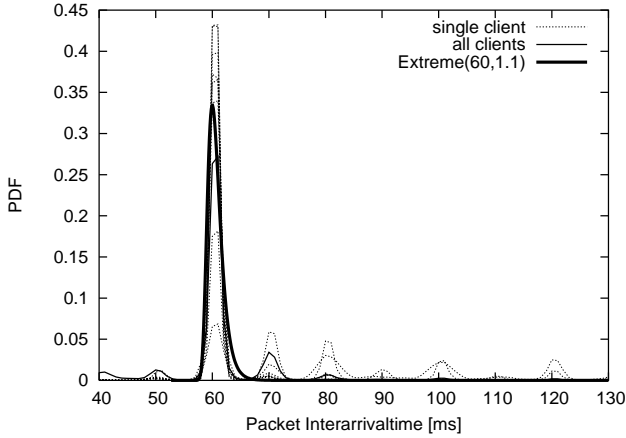


Figure 6: Probability density function of server packet interarrival time for 10 clients

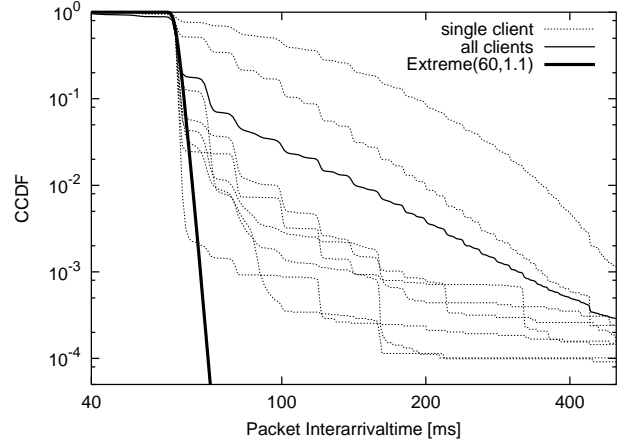


Figure 7: Complementary cumulative distribution function of server packet interarrival time for 10 clients

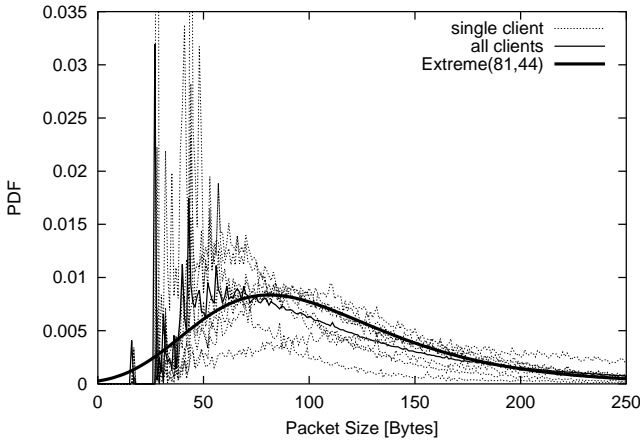


Figure 8: Probability density function of server packet size for 10 clients

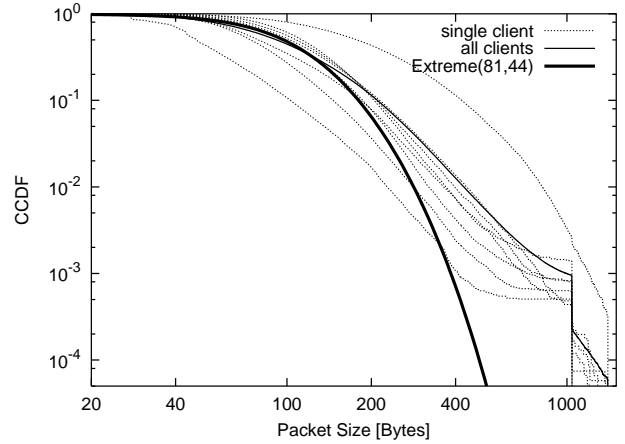


Figure 9: Complementary cumulative distribution function of server packet size for 10 clients

In average, the mean packet size of all packets is 119 Bytes with a coefficient of variation of 0.67. According to the results of section 2.1 this is equivalent to an average of 10 active players. Note that Figure 6 through Figure 9 reflect

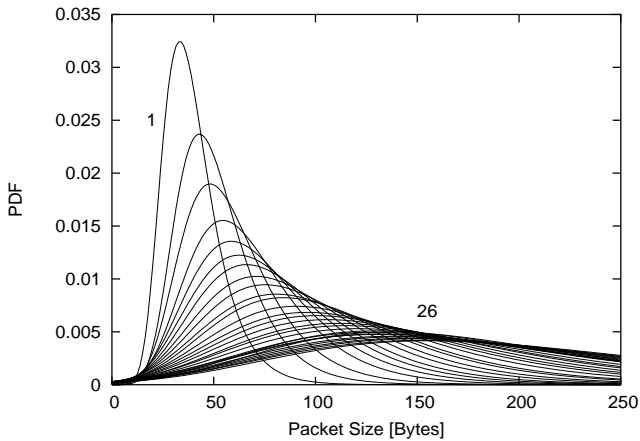


Figure 10: Fitted probability density functions of server packet size for 1 to 26 active clients

the total traffic without regarding the number of active players. If the mean of the packet size changes with the number of active players as seen above, it is likely that also the shape of its distribution changes with the number of active players. Thus, Figure 8 only shows half of the truth, i.e. the statistics for all possible player numbers together. In fact, a separate evaluation for each number of active clients leads to a different view. Using a sliding 30 second window for each trace we determine the number of active clients as described in section 2.1. The packets in the window are then accounted to the statistic associated with the number of active clients. Figure 10 shows approximations to the resulting packet size pdfs for each number of active players. Clearly we see that the shape changes from a very strong peak for a single player to a wide distribution for more players up to the maximum of 26 participants found in the traces.

A similar evaluation for the interarrival time of the server traffic reveals, that with more players the very sharp peak at 60 ms is turning into a ternary distribution with peaks at 50,

60 and 70 ms where the 70 ms peak becomes dominant for more than 17 players. The server obviously slightly decreases the packet rate for large multiplayer matches (see also Figure 3).

2.3 Client Traffic

Client traffic is characterized by an almost constant packet and data rate as shown in Figure 2 above. As the view on the total client traffic does not reveal much, we again evaluate the captured data per client (for a subset of the most active clients). We characterize client traffic by

- packet interarrival time and
- packet size

In Figure 11 and Figure 12 the probability density and distribution functions of the interarrival time for the 8 example clients are shown. We see that most packet interarrival times are found between 30 and 60 ms. Some pdfs have

peaks while others show a wider distribution. The differences are caused by different client hardware performance and settings which lead to different mean interarrival times. We have seen similar differences between client traffic in [7], however, in the LAN game traffic, all client pdfs showed clear peaks. As our traces were captured on the server side, we assume that we see the effect of network jitter for some distant clients, while closer clients did not suffer from large jitter.

The long tailed behaviour of the ccdf in Figure 12 is caused by some large interarrival times at around 600 to 800 ms as well as long pauses between matches or turns.

In average the mean packet interarrival time for the most active clients is found at 51 ms with an average coefficient of variation of 0.4 (again we only consider interarrival times smaller than 1 second for calculating the mean for in-game packet traffic).

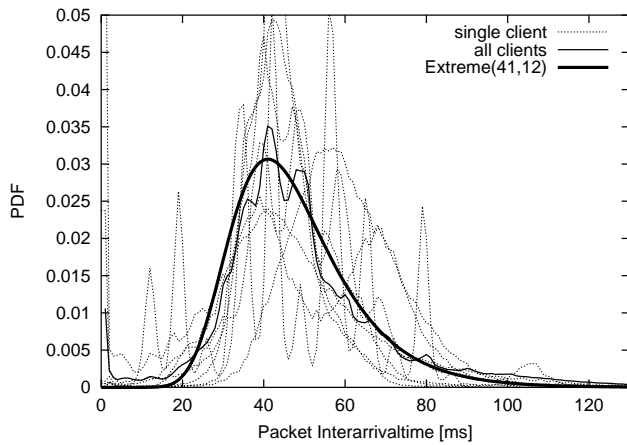


Figure 11: Probability density function of client packet interarrival time for 10 clients

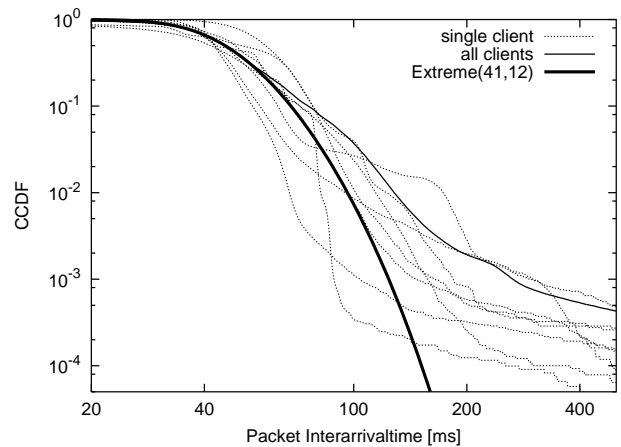


Figure 12: Complementary cumulative distribution function of client packet interarrival time for 10 clients

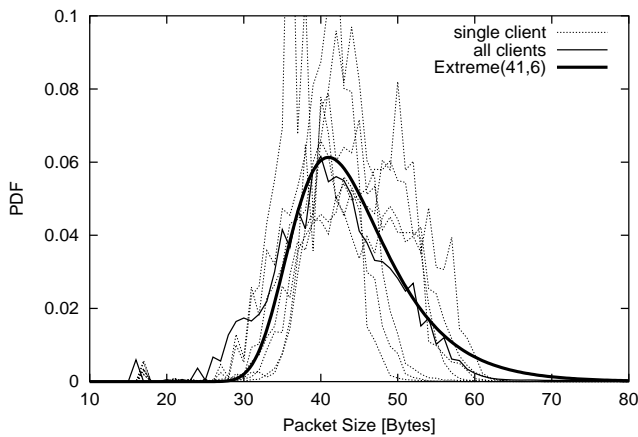


Figure 13: Probability density function of client packet size for 10 clients

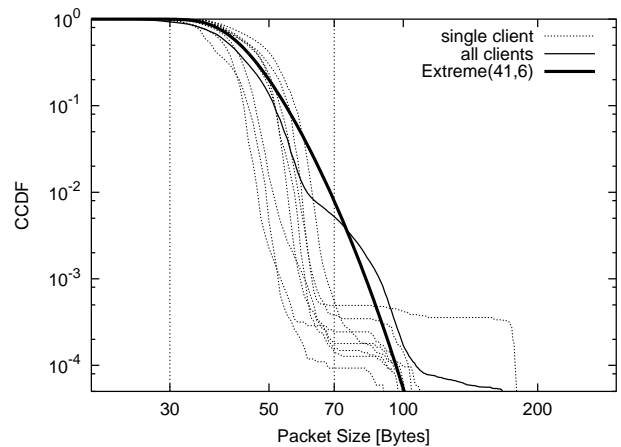


Figure 14: Complementary cumulative distribution function of client packet size for 10 clients

The pdfs of packet sizes of the active clients shown in Figure 13 vary around a mean of 42 Bytes (without UDP header) with a coefficient of variation of only 0.15. This is a little smaller than the mean of 52 Bytes found in [7].

The ccdf in Figure 14 shows, that 99.9% of all packets have a size between 30 and 70 Bytes. Only very few show larger size of 100 Bytes or more.

3. GAME TRAFFIC MODEL

Our intention is to provide a simple traffic model for fast action multiplayer games. We distinguish between in-game traffic (i.e. traffic of players actively participating in a running match) and communication during match preparation or pauses. During game pauses client and server traffic may stop for a short time after which larger update packets are transferred to synchronize all clients. Note, that this traffic is not time critical. Those dynamics are out of the scope of this work and have to be modelled on a higher level if desired. We put our focus on in-game traffic and do not consider any pre and post match communication for modelling which would also most certainly be too specific to Counter Strike.

Although multiplayer games have inherent correlations between the clients due to a shared game state we have shown in section 2 that the variances for packet sizes and interarrival times are small, i.e. any such dependencies only lead to slight traffic changes. Thus, the game traffic can be modelled by independent traffic streams from each client to the server and a burst traffic stream from the server to the clients. In our approach we assume that clients behave independent of each other and that client traffic is independent of the corresponding server traffic.

Our game traffic model consists of only two independent modules, the client traffic model and the server traffic model with a burst size equal to the number of clients participating in the simulated traffic.

For a mathematical description of the distribution functions for interarrival time or packet size we need to find a function of similar shape and fit it to the empirical data. As Borella has identified the extreme value distribution (see Table 2) to fit best for Quake traffic, we also choose this function for better comparison. Similar functions like shifted Lognormal or shifted Weibull lead to acceptable fits as well. As the extreme function allows negative values, it has to be bounded and scaled appropriately in order to use it in simulation or analysis.

We use least square fitting of the extreme value function to the pdf to find the parameters a (location) and b (scale). As least square fitting neglects small probabilities especially in the tail of the distribution functions, the resulting parameters will lead to a smaller variability than actually observed.

Table 2: Extreme value function

pdf	ccdf
$\frac{1}{b} e^{-\frac{x-a}{b}}$	$e^{-e^{-\frac{x-a}{b}}}$, $b > 0$

Note, that the empirical data still contains packet sizes or interarrival times captured during game interruptions. As we do not want to consider this in our description of the very regular in-game behaviour of server and clients modelling a long tailed distribution is not desired here.

3.1 Server Model

For the burst interarrival time of the server, we assume a simplified single burst per transmit cycle which holds packets to all clients. Within a burst a packet is sent to every client as soon as possible. The peak in Figure 6 suggests to use a deterministic burst interarrival time of 60 ms.

As we have shown in section 2.1 average packet sizes depend on the number of clients. The statistical evaluation of all 29 traces presented in section 2.2 reveals that also the distribution changes significantly with an increasing number of players n . Figure 10 shows the approximated extreme value distributions for each number of active players. The parameters a and b of these 26 distributions follow a strict linear behaviour:

$$a(n) = 34.5 + 4.2 n \quad (1)$$

$$b(n) = 9 + 3n \quad (2)$$

For each destination (client) within a burst a new packet size should be generated. We suggest to bound the extreme value function at a minimum of 20 Bytes and a maximum of 1000 Bytes.

Table 3 shows traffic characteristics of the observed data as well as the suggested distribution. The corresponding plots for the extreme value distribution can be found in Figure 8 and Figure 9. The neglect of the tail behaviour is clearly visible in Figure 9. Extreme peaks shown in Figure 8 are client specific and are not modelled here.

3.2 Client Model

Although the statistical evaluation in section 2.3 shows wide distributions for the packet interarrival times, we like to suggest a deterministic value of 50 ms for a client source traffic model (see Table 3) as we have observed a very constant packet rate in LAN game traffic and see network jitter as the main reason for the wide distributions. If, however, incoming client traffic arriving at the server should be simulated, the fitted extreme distribution with parameters shown in Figure 11 and Figure 12 should be used instead.

Table 3: Counter Strike traffic characteristics and suggested approximation

n active clients	Server (per client)		Client	
	characteristic	approximation	characteristic	approximation
(burst) interarrival time	peak = 60 ms mean = 64 ms coeff. of variation = 0.21	Deterministic(60)	peak = 46 ms mean = 51 ms coeff. of variation = 0.4	Deterministic (50)
packet size (without UDP header)	mean = $50.4+6.15n$ Bytes coeff. of variation = 0.67	Extreme($a(n),b(n)$) [20:1000]	mean = 42 Bytes coeff. of variation = 0.123	Extreme (41,6) [20:100]

The client packet size has very little variation around the mean. The probability function depicted in Figure 13 shows that an extreme function describes the shape well. We did not differentiate client packet sizes per number of active clients as we did for server traffic. Fitting an extreme function to the pdf of packet sizes of all active clients leads to the suggested function for the client source traffic model (see Table 3).

4. QOS METRICS FOR GAMES

The suggested game traffic model is intended to be used for performance analysis for networks in regard to their gaming capability. In order to assess the impact of packet delay or packet loss experienced in such an analysis, it is necessary to define Quality of Service (QoS) metrics for gaming applications. We identify packet loss, delay and jitter as the main criteria for the assessment of game traffic quality.

Today's games can cope with an enormous lag (ping, round trip time) and loss. These applications are thought to be used over the Internet with a typical round trip time of 50 to 150 ms. If analog modems are used, each use introduces an additional latency of 30 to 40 ms, i.e. an additional 120 to 160 ms to the round trip time for a dial-up player [13]. Ping times frequently show 300 ms and more. Consideration of loss and lag are an essential part of the game design. Game designers try to optimize for 200 to 250 ms round trip time and provide robustness for larger lag. This is achieved by client-side prediction of the game state, i.e. movement of objects and other players [3,6]. By combining movement with inertia or reducing maximum velocity of objects prediction is even more effective [13].

Such considerations result in very robust games, technically tolerating lag up to one second and loss up to 40%. However, these values are not suited as criteria for a good or bad service, as QoS should be seen from the players perspective. Acceptable gameplay requires far better performance. Whether a gamers connection has ping times of 50 ms or 150 ms makes a huge difference [5]. In [1] an evaluation of player effectiveness in a first person shooter over that players ping time shows that players with lower ping times score significantly higher kill rates than others.

Probably the only really QoS criteria from the players perspective is fun which is hard to measure. However, players like to talk about the quality of their connection in terms of lag and correlate it with their experienced fun. The impact of packet loss or jitter is rarely discussed. Qualitatively it can be said that low packet loss and small jitter help compensate for high lag. Opposed to that, a high jitter or packet loss totally ruins gameplay despite of a very small lag.

In an Internet discussion board [11] players commented on impact of ping times on gameplay. While many players only accept minimum ping times, some players report that they adapt their playing strategies to high ping times and may even enjoy a game with 200 ms lag. The impact of lag also depends on the game. As „Quake III Arena“ is very fast and responsive the ping time almost automatically decides on winning or losing. „Quake World“ or „Unreal Tournament“ are reported to behave much better in this regard, i.e. ping times are not as decisive for successful playing. Based on [12] and [11] we find that a ping below 50 ms is associated with excellent game play. A ping below 100 ms is good and above that, playability decreases noticeably. Ping times above 200 ms are often reported to be intolerable but many players claim to have no problems with ping times around 200 ms (see also Table 4).

Please note, that these are no results from a large survey but only a summary of player comments and opinions. It is doubtful if players really notice the difference between 50 ms and 100 ms lag if the game wouldn't report it. Also, many players today simply have no fast Internet connection and use analog modems. An evaluation on „Half Life“ reported in [10] shows that players who experience high ping times of over 225 ms do not quit and look for a faster

Table 4: QoS metric „lag“ for first person shooters

Lag	Users Opinion
< 50 ms	excellent gameplay
50 - 100 ms	good gameplay
100 - 150 ms	noticeably decreased gameplay
150 - 200 ms	significantly affected gameplay
> 200 ms	„intolerable“ Gameplay

server but stay and continue to play with this high lag. We assume that those players use 56k modems and do not expect to get a better connection elsewhere. The study reveals that many gamers (40%) play with a high lag of over 225 ms despite of the decreased playability.

5. CONCLUSIONS

We have presented a traffic characterization of the popular multiplayer game „Counter Strike“. For simple source modelling of this traffic we treat clients and server independently and focus on in-game phases with very regular traffic patterns. Although the model neglects effects of correlations between clients and server it allows the performance evaluation of network systems in regard to popular game traffic.

The simplicity of the presented model allows to use it either to simulate traffic on a link to and from a subset of clients as well as traffic to and from the server communicating with all active clients. The number of active clients as well as session durations have to be set for the duration of the simulation or must be described on a higher model level, e.g. using the results of [9]. The game traffic model is not suited to provide background traffic for evaluations of other traffic flows. Its use is clearly in the evaluation of Quality of Service (QoS) aspects of networks in respect to games. The discussion on QoS metrics for first person shooters is far from complete but gives insight into network requirements of games and allows a rough assessment of simulation results.

Compared to Borellas observations for „Quake 2“ over LAN we see larger packets and lower packet rates. The main characteristics, however, are very similar. Also, a brief evaluation we have done for „Unreal Tournament“ (another very popular first person shooter) reveals a similar general characteristic (small packets, high and very regular packet rate, bursts for server). As a large part of today's first person shooters are either based on the Quake engine or the Unreal engine, we feel that the class of first person shooters or fast action multiplayer games can be described with a general traffic model.

Other game genres require different traffic models. For Age of Empires, a popular strategy game, a totally different networking architecture is used (fully meshed, max. 8 players) and QoS requirements differ as well (lag of below 250 ms is good, lag up to 500 ms payable and beyond that the lag is noticeable) [4].

In the near future, the quality of game graphics will continue to increase enormously but game traffic is not so likely to change much from what it looks like now except for the number of players and additional voice communication which will be incorporated in most online games soon. Even today many gamers use headsets and voice communication software to talk to their team members.

Acknowledgements

The author likes to express his thanks to Tristan Henderson who made his large archive of Counter Strike data available for evaluation. Also, thanks to all reviewers for their helpful comments.

References

- [1] Armitage, Grenville: Sensitivity of Quake3 Players To Network Latency, IMW2001 workshop poster session, Nov 2001, http://www.geocities.com/gj_armitage/q3/quake-results.html
- [2] McCreary S., claffy k.: Trends in wide area IP traffic patterns - A view from Ames Internet Exchange, *ITC Spec. Seminar*, 2000
- [3] Bernier Y.W: Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization, *Game Developers Conference 2001*, http://www.gdconf.com/archives/proceedings/2001/prog_papers.html
- [4] Bettner P., Terrano M.: 1500 Archers on a 28.8: Network Programming in Age of Empires and Beyond, *Game Developers Conference 2001*, http://www.gdconf.com/archives/proceedings/2001/prog_papers.html
- [5] Borella M.S.: Source Models of Network Game Traffic, *Network + Interop '99 Engineer's Conference*, May 1999
- [6] Bonham S., Grossman D., Portnoy W., Tam K.: Quake: An Example Multi-User Network Application - Problems and Solutions in Distributed Interactive Simulations, *CSE 561 Term Project Report*, University of Washington, July 2000, <http://www.cs.washington.edu/homes/grossman/projects/561projects/quake/>
- [7] Färber, J.: Network Game Traffic Modelling, *Netgames 2002*, Braunschweig, pp. 53-57, April 2002, <http://www.ibr.cs.tu-bs.de/netgames2002/>
- [8] Game2k1, <http://www.game2k1.com/>
- [9] Henderson T., Bhatti S.: Modelling user behaviour in networked games, *Multimedia 2001*, June 2001, <http://www.acm.org/sigs/sigmm/MM2001/ep/henderson/>
- [10] Henderson T., Latency and user behaviour on a multiplayer games server. *NGC 2001*, London, UK, pp1-13, Nov 2001, <http://www.cs.ucl.ac.uk/staff/T.Henderson/docs.html>
- [11] Slashdot, How Fast Too Slow? A Study Of Quake Pings, discussion forum, May 2001, <http://slashdot.org/article.pl?sid=01/05/24/2044233>
- [12] Trier M.: Highspeed-Internet, *GameStar (Gaming Magazine)*, March 2002, pp.164-165
- [13] Ng Yu-Sheng: Designing Fast-Action Games for the Internet, *Gamasutra (Online Game Developer Magazine)*, May 1997, http://www.gamasutra.com/features/19970905/ng_01.htm