Voice Communication, Computer Games, Talkspurts, Silence Periods

Gabor Papp and Chris GauthierDickey *

CHARACTERIZING AND MODELING MULTIPARTY VOICE COMMUNICATION FOR MULTIPLAYER GAMES

Over the last few years, the number of game players using voice communication to talk to each other while playing games has increased dramatically. Unlike traditional voice-over-IP technology, where most conversations are between two people, voice communication in games often has 5 or more people talking together as they play. We present the first measurement study on the characteristics of multiparty voice communications and develop a model of the observed talking and silence periods that can be used for future research, simulation, network engineering, and game development. Over a 3 month period, we measured over 7,000 sessions on an active multi-party voice communication server to quantify the characteristics of communication generated by game players, including group sizes, packet distributions, user and session frequencies, and speaking (and silence) durations. Unlike prior results, our measurements and models demonstrate that the speaking and silence periods fit a Weibull distribution.

1. INTRODUCTION

Multiparty voice communication (MVC) is an important application that needs to be studied and researched. In multiplayer computer games, for example, thousands of players can interact and communicate with each other using built-in voice chat systems (e.g., World of Warcraft†). Game consoles have added multiplayer lobbies for player interaction as selling features of the hardware. Further, multiparty voice communication is widely used for conference calls and voice chat software and clearly will be part of future online collaborative systems. Thus, research in this area benefits the future design of MVC systems, is relevant and interesting to online game development and deployment, and is important to ISP network engineers supporting and hosting online game and voice systems.

Our research is the first work to look at characterizing multiparty voice communications over the Internet, and in particular when it is used with multiplayer games. Previous voice-over-IP (VoIP) measurement work has looked at quality of service parameters such as packet loss, packet reordering and its effects on sound quality, or at network characteristics and support of VoIP between two parties. Instead, our research attempts to characterize the traffic, packet arrival rates, group sizes, session frequencies and durations, and speaking and silence periods in order to develop mathematical models for multiparty communication that can be used for simulation and modeling.

*Department of Computer Science, University of Denver, {pgabor|chrisg}@cs.du.edu

†http://www.worldofwarcraft.com/

In the past, multiparty voice communication was limited to small groups because player bandwidth was limited to modem speeds. However, new computer games often include voice communication as an essential part of the playing experience. A typical multiplayer game might have up to 40 players coordinating with voice communication while spelunking. This differs significantly from a Skype[‡] session between two people.

To conduct our measurement study, we set up a TeamSpeak[§] server which allows clients to join, set up channels, and communicate with other clients on the same channel. We then recorded traffic over a three month period and analyzed the resulting data.

Our results show that multiparty voice communication differs from traditional two-party communication. We see that overall traffic follows a sinusoidal curve with a peak around 8PM and a period of 24 hours. Group sizes tend to average 5 people while groups with more than 20 people are the least common. Further, silence and talking periods followed a Weibull distribution, which differs from prior research on voice communications.

The main contribution of this work is a characterization of traffic patterns, group patterns, and voice patterns through measurements. In particular, we model the talking patterns mathematically, based on the measured multiparty voice communication sessions. Further, the characterization of voice patterns has typically been done on small sets of data; our study measures patterns from thousands of hours of voice data with thousands of unique sessions. Thus, future research in MVC systems will be able to use our models to drive experimental simulations, game developers can use these models to understand the impact of adding voice communications on network traffic already generated by their games, and ISPs can use this information for provisioning servers for hosting MVC systems.

2. BACKGROUND

Voice patterns consist of *on* and *off* periods (also called talkspurts and silence periods). Over the last 40 years, research has looked at these patterns and shown that the *on* periods follow an exponential distribution [3, 15] in traditional telephony. These results are important because they allow designers of hardware, codecs, and network administrators to predict the patterns of speech with mathematical models. Our research follows in this tradition, though we look at multiparty voice communication and study several orders of magnitude more sessions than previous research.

Markopoulou et al. measured the quality of voice communications over the Internet. They measured delay and loss over wide-area backbone networks and used these results with a voice quality model [10] to determine the efficacy of VoIP over the Internet for voice communication. The authors show that while many Internet backbones have sufficiently low delay, delay variability, and loss, several provide poor VoIP quality. Our work measures traffic directly at the server since we cannot provide a client that generates end-to-end measurements. However, we are more interested in the actual traffic patterns generated by the multi-party communications and less interested in whether or not Internet backbones support VoIP.

Jiang et al. looked at the on-off patterns in VoIP by recording and digitizing conversations and then applying *gap* detectors to determine how long people talked and how long they were silent [8]. Their results show that the length that people talk for somewhat follows an exponential distribution while the gap they are silent for deviates significantly from the same distribution. Our measurements differ and show that the on-off patterns of VoIP in multiparty communications follow Weibull distri-

[‡] `http://www.skype.com/`

[§] `http://www.goteamspeak.com/`

butions more accurately, without significant deviation.

Skype [1], a VoIP application, was measured by Chen et al. in order to determine the level of user satisfaction [4]. By measuring network traffic characteristics, they correlated the amount of jitter and interactivity of a session with the length of the call. Our work measures similar traffic characteristics, but measures data between multiple parties. However, we do not examine or predict conversation quality based on traffic characteristics.

Borella analyzed game traffic from a popular online game server on a LAN and modeled the inter-packet arrival time and packet sizes with extreme distributions [2]. Their model was validated using the λ^2 test, which we also use to validate our models. The λ^2 test is important in our situation because we have large data sets with over 180k sample points and χ^2 tests perform poorly in these situations [2].

Henderson and Bhatti modeled network traffic of an online game over the Internet [7]. This work was measured over the Internet instead of over a LAN, providing a more realistic model. Their work shows a daily and weekly traffic pattern similar to what we measured with voice traffic: evenings have peak traffic while early mornings have the lowest traffic. Further, more traffic is seen overall on the weekends. Pittman et al. and Svoboda et al. had similar diurnal patterns in their measurement work on large-scale multiplayer games [14, 16]. In addition, other researchers modeled game traffic (packet sizes, arrival times, sessions) with similar results [5, 17].

Note that our early results were published in [11], but only included the initial CDFs of talkspurt and silence periods. However, in this paper, we present our full range of results and model multiparty voice communications.

3. TRACE COLLECTION

In this section, we describe the architecture of our network, the content of the server log file, the collection of the VoIP sessions and the procedure that we used to clean the collected data.

3.1. NETWORK SETUP

TeamSpeak is a group voice communication server that allows multiple people to connect using a TeamSpeak client, join *channels*, and talk simultaneously with other people in the same channel. In this client/server architecture, the clients encapsulate voice packets using one of many codecs, and send those packets to the server using UDP unicast. The server then unicasts the packets to the other $n-1$ clients connected on the same channel. Note that the server does *not* multiplex the voice packets, though we expect future architectures to do so.

We set up a TeamSpeak server and advertised it to game players as a free server beginning in November of 2006. We then began logging all traffic on port 8767 to the server using *tcpdump*. The server was set up to only allow 12.3kbps and 16.3kbps Speex encoding for voice.

Although TeamSpeak generates a server log file, the data contained in this file (even with maximum verbosity) is minimal and contains data only about logins, logouts, channel switching, and administrative operations. Thus, we used *tcpdump* to record all packet information generated with regards to the TeamSpeak server. We also discovered that when we compared data from the server log and the trace files, the server log was not always accurate. For example, the server log would show a player logging in multiple times without logging out. This was probably due to the fact that the player's connection died, but the server had not discovered it before the player re-logged in. However, from our packet traces we could determine a *session* by looking at the time that a player logged into the server to the last time they sent or received a voice packet from any player.

Using the data recorded by our tcpdump logs, we could identify voice packets and separate them from non-voice packets by their size and the codec ID. All voice packets were 155, 161, 205, and 211 bytes. Incoming packets were 155 and 205 bytes for 12.3 and 16.3 kbps Speex encoding. Outgoing packets were 161 and 211 bytes.

3.2. DATA CLEANING

As we began the analysis of our data, we discovered that some of the data points were extremely different from the rest. These included excessively long talk sessions as well as silent periods. For example, the extreme outliers of the talkspurts were data points where voice packets were delivered for close to an hour, which would be fairly difficult to accomplish when you consider that we can detect silence gaps as small as 100ms! We reckon that these are rare occasions resulting from something such as loud background music or human behavior such as forgetting to log off while leaving the computer for several hours. Therefore, we treated these data points as outliers and did not include them in our final results.

While removing extreme outliers can be controversial, we justify our actions by noting that our method removed few or no data points and that the methods used for curve fitting often pick the first and last end-points of the data to begin and end the curve, and then adjust values to force the rest of the graph to fit. Thus, the extreme outliers can cause a curve to not fit the data well, whereas by removing the outliers and fitting the curve allows one to obtain a better fit, according to various metrics. In Section 4.5, we detail the effects of our data cleaning.

To remove the extreme outliers, we first analyzed the linearity of the data. Prior researches show that talkspurt and silence periods often follow an exponential distribution [3, 8, 15]. We also plotted preliminary graphs to get an idea of the general trend of the data. Note that linearization for the purpose of cleaning does not need to be perfect (e.g., we used an exponential form, though our data turned out to be Weibull). The purpose of this process is to remove extreme outliers and, given the large number of data points, removing only a small percentage of data points is acceptable.

Once the data was linearized, we identified the first and third quartiles. To keep as much data as possible, we deleted only the extreme outliers. The data points, *e*, that are beyond the outer fences are defined as:

$$e < Q_1 - 3IQR \quad \text{or} \quad e > Q_3 + 3IQR \tag{1}$$

Here, Q_1 is the first quartile, Q_3 is the third quartile and IQR means the inter-quartile range $(Q_3 - Q_1)$. While methods that remove all the outliers and not just the extreme ones use $1.5IQR$ we decided to use $3IQR$ and only remove a very small amount of data, which we felt was sufficient enough for our purposes.

4. MEASUREMENTS

Our measurements cover a 3 month period from December 2006 to February 2007. During this time, we measured over 7000 sessions from over 800 IP addresses dispersed geographically for an average of 1.46 GB/day in traffic.

Table 1: Heaviest user distributions with rankings.

(a) Distribution by country.

Country	**Player Distribution %**
United States	60.53
Canada	26.63
Singapore	9.68
Australia	1.57

(b) Distribution by state.

State	**Player Distribution %**
Pennsylvania	17.80
New York	8.40
California	7.20
Colorado	4.60
Florida	4.20
Texas	3.80
Washington	3.20

4.1. GEOGRAPHICAL DISTRIBUTION OF USERS

In order to ensure that our data was not biased due to the geographical location of clients connecting to the server, we took advantage of the fact that all the client IP addresses were obtainable from our log files. Thus, we could estimate the locations of the clients and ensure that they were not all from the same place. Using the free MaxMind tool, GeoLite Country[¶], we determined the latitude, longitude, country and state where applicable of each IP address. This free version of the software claims to have over 98% accuracy. After processing our data we found that more than 87% of our users were from North America, more than 60% of our users were from the United States, each of the 7 most popular states was responsible for more than 3% of the U.S. users and combined they were responsible for almost 50% of the U.S. users, and none of the remaining states contributed more than 3% of the population individually.

Table 1a shows that the majority of our users are from the United States and Canada while Table 1b shows the breakdown by states in the United States. Note that only two countries contributed more than 10% of the population, the U.S. and Canada. *We conclude that the primary result of our server location being in the MST time-zone is simply that most users are from the U.S. and Canada.* Generally, server location affects the user locations due to latency issues, but given the wide-spread locations of users within the continental U.S. and Canada, our data is not biased towards a particular area within these two countries, except to follow natural populations.

4.2. OVERALL SERVER TRAFFIC

The first set of measurements we present are the overall traffic seen by the server during an average day. Figure 1 shows the averages, averaged per hour on the x-axis and the number of packets sent and received on the y-axis. Thus, this figure is an indication of the volume of traffic seen by the TeamSpeak server. Incoming voice packets are always 155 or 205 bytes while outgoing voice packets are always 161 or 211 bytes respectively.

This result shows that server input doubled and server output increased by an approximate factor of 4 during the evenings (approximately 7pm-9pm MST[‖]). This indicates that more users are online using multiparty voice communication during the evenings.

[¶] `http://www.maxmind.com/app/geolitecountry`

[‖] Throughout this paper, times are listed as MST, but this is only as a convenience indicating the time-zone the server is located in and has no bearing on the measurements or results.

We also observed that during the peak period (7pm-9pm) the traffic rate is also almost constant. In other words, the number of sessions started is the same as the number of sessions finished during this period and thus resembles a balanced birth-death process.

We hypothesized that traffic was actually higher on weekends, and therefore we divided our averages into individual days so that we averaged all Mondays separately, all Tuesdays separately, etc. Figure 2 shows the inbound server traffic from users. From this figure, we see that most days are very similar with a small amount of variance, though in terms of total input, Fridays and Sundays have the highest amount of inbound traffic.

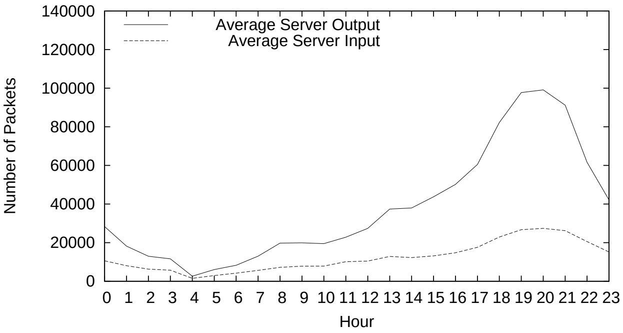

Figure 1: *Server Traffic:* Average voice server traffic over a 24 hour period (times shown are MST). Server input doubled while server output quadrupled during evening hours. The peak is around 7pm-9pm whereas the most quite period is around 4am.

In Figure 3, we can more clearly see that the server output has more traffic on weekends than on weekdays. In addition, Sunday traffic increases earlier than on any other day, starting at 1pm MST while the peak of traffic is highest late on Friday evenings at approximately 130k packets/hour. Interestingly, Saturdays have a lower peak traffic than Sundays or Fridays, but have a higher average traffic during the early hours of the day. This difference is most likely due to people who are on late Friday continuing to use TeamSpeak into the early hours of Saturday morning (and then probably sleep late that day).

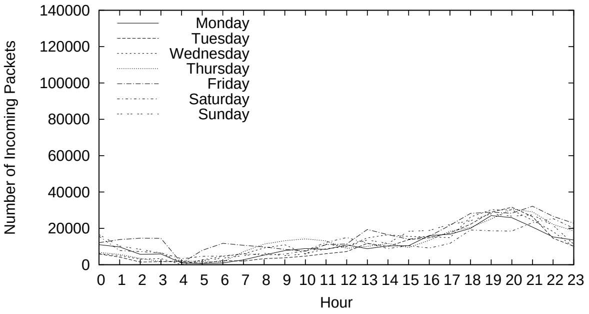

Figure 2: *Server Input:* Average voice server input traffic over a 24 hour period (times shown are MST). Server input is similar on all days, with a peak during evening hours.

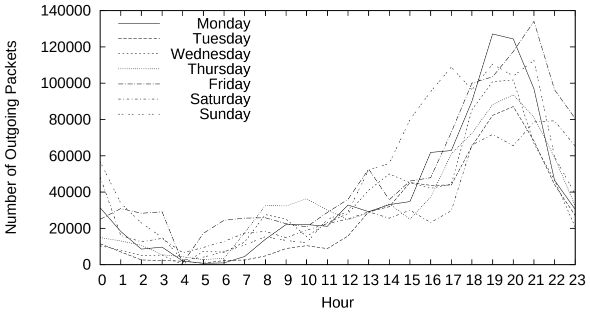

Figure 3: *Server Output:* Average voice server output traffic over a 24 hour period (times shown are MST). Server output is higher and extends over more hours during the weekends, and in particular on Sundays.

A final interesting trend is that Monday also seems to have a high outbound traffic peak that is similar to Friday. The primary difference appears to be that the traffic is shifted about two hours earlier, probably because game players start earlier so that they can go to bed earlier.

Given the TeamSpeak architecture, which unicasts packets to other players in the same channel, these results provide an insight into the size of the group that is talking to each other in the same channel. First, on Fridays, the output is approximately 4 times the size of the input. This implies that for each voice packet that is input, TeamSpeak is replicating it 4 times, for a group size of 5. On a day such as Tuesday, the traffic is 2 to 3 times that of the input, indicating group sizes on average of 3 to 4. We conjecture that on Fridays and Sundays, game players are more likely to use multiparty communication to converse with a larger group of other people than on other days. Most likely this is because players have more free time on those days and are able to coordinate getting together online with other players more readily.

When we look at this data in conjunction with the general server traffic, we see an interesting trend. Even though group sizes may increase, the amount of incoming traffic does not increase at the same rate as the outgoing traffic. Given these traffic patterns we believe that while many people may be able to talk at the same time in a large group, human protocols prevent this from occurring. Typically, only one person can talk at a time and they take turns during the sessions. In essence, if more than one person begins talking, the speakers stop to allow only one person to talk so that the conversation can be understood.

We expect our results to be similar to traffic patterns seen on game servers. Indeed, similar diurnal patterns and weekly patterns have been observed in related game traffic measurement work [17, 7, 14, 16]. There is clearly a peak, a local minimum each day, a strong correlation between days and a higher load on the weekends.

4.3. GROUP SIZES

We next examine group sizes to gain an insight into the size of a group that is typical in multiparty voice communication when used with games. As we noted previously, the ratio between the inbound and outbound traffic is an indicator of the average group size.

To perform this measurement, we looked at the trace logs and determined the sender ID for all the outgoing packets. The number of outgoing packets with the same ID and the same sequence number is one less than the actual group size. We binned all data according to how many people it was duplicated to, allowing us to examine the data based on the size of the group. Thus, we can determine the effect of the groups with different sizes on both the incoming and outgoing traffic on the server. Note that, neither the server log file nor the TeamSpeak packet format provides information about the used channel and thus it is impossible to identify the actual groups based on the packet alone. The results are seen in Figure 4. The solid line indicates the incoming traffic and the dashed line indicates the outgoing traffic that is simply the incoming traffic multiplied by the number of players that the particular packet is replicated.

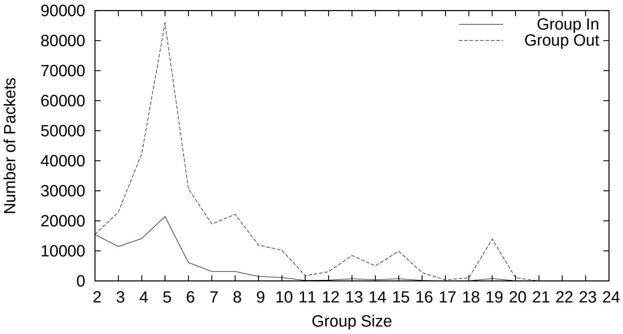

Figure 4: *Group Sizes:* The number of groups of a given size, categorized by days over the measurement period. We see that, on average, the most groups are between 2 to 8 people talking, with a maximum of 24 people in a group.

The results on group sizes show that the most active groups are the ones that are formed by 5 people. It can also be concluded that these groups generate the most outgoing traffic among all the groups. It is also worth mentioning that the amount of incoming packets from groups formed by pairs is almost as high as it is by these groups, but because of the replication the outgoing traffic is much less affected. Similarly, the counter effect can be seen in case of groups that contain 19 people;

although the amount of incoming traffic is low, the amount of outgoing traffic is high due to the large amount of replication necessary. The largest group we observed was 24 people.

We believe that a correlation between using TeamSpeak and the game being played exists. Currently, one of the most popular online games being played is World of Warcraft. In this game, players are often limited to 5 people in special areas, biasing the data towards a small group of people talking and playing the game together. On the other hand, a large class of multiplayer games, called *first-person shooters*, tend to group players into two groups, each between 8 and 16 players. Multi-party voice communication has also become very important for this class of games. If our TeamSpeak server was used by players of these kinds of games, we would expect the group sizes to correlate. Thus, we concluded that our server was mostly used by players on games which promoted small groups. However, because determining the game being played is impossible from our logs, and because the server was advertised to a wide variety of sources, we believe our results are general enough to at least apply to MVC for games in general.

4.4. SESSION CHARACTERISTICS

We define a session as the period from when a user logs into the server until they log out of the server. These can be determined finding the *login* and *logout* entries in the TeamSpeak log file. Note that entries where the user was seen logging in more than once without logging out were not considered in our measurements.

We recorded 7,749 sessions, including the packets that were sent to and from the server and how long users were logged into TeamSpeak. *On average, we observed 86.1 logins per day from 826 individual users.* To understand this data further, we calculated the session times and generated a CDF as shown in Figure 5.

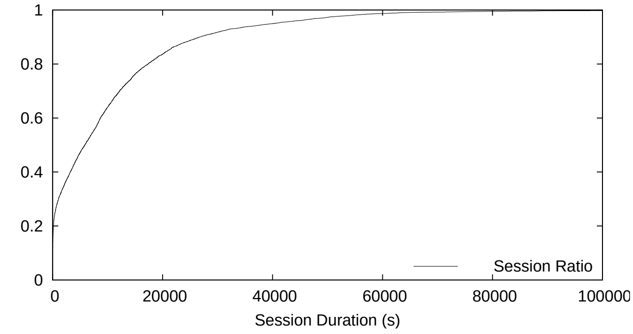

Figure 5: *Sessions Length CDF:* We see that of the 7,749 sessions we recorded, half of these sessions were less than 5000 seconds (1.3 hours). A small fraction of these (a few hundred) were over 30,000 seconds (8 hours).

Our calculations show that the shortest sessions were less than one second while the longest session was over 69 hours! However, as Figure 5 shows, for 20% of the sessions, users stayed less than 1/2 hour. In addition, for 20% of the sessions, users stayed for more than 5 hours. Thus, 60% of the sessions fell somewhere between 1/2 hour and 5 hours.

For the small fraction of sessions that were greater than 8 hours, we hypothesize that users simply did not log out of the TeamSpeak server when they were done. In the future, we would like to look at the correlation of session time with input traffic to see if long lived or very short sessions are actually sending and receiving voice traffic.

The characteristic of our curve is similar to both that can be found in [7] and in [16]. However, both of these papers analyze network traffic in on-line games, one of them focuses on a First Person Shooter (FPS) game whereas the other one focuses on a Massively Multiplayer Online Game (MMOG). On the one hand this fact validates our results but on the other it shows that it is nearly impossible to conclude what type of game is played by the users analyzing only the characteristics of the data and not the content of it.

Table 2: *Cleaning the data sets:* The effect of removing extreme outliers with the cleaning procedure on inter-talkspurt, talkspurt, and silence periods data sets.

	Inter-talkspurt arrival	**Talkspurt**	**Silence**
Original data size	188,225 (100%)	188,313 (100%)	186,626 (100%)
Filtered data size	187,884 (99.82%)	188,158 (99.92%)	186,626 (100%)
Deleted data size	341 (0.18%)	155 (0.08%)	0 (0.00%)

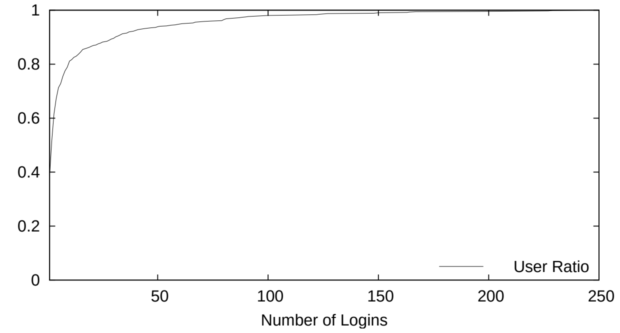

Figure 6: *Login Count CDF:* 40 % of all the IP addresses that logged to our server were unique. This is probably due to the fact that DHCP was used to assign their addresses. 17% appeared to log into the server at least once a week on average.

In the next measurements, we matched IP addresses with sessions to determine how many unique IP addresses logged into the system. In essence, we would like to determine how frequently a user logs into and uses the TeamSpeak server. We calculated the CDF of the ratio of logins versus the number of logins as illustrated in Figure 6.

Our results indicate that 40% of the users logged into the TeamSpeak server only once, while only 17% logged into it regularly (i. e. at least once every week on average). However, this result might be biased due to the fact that some users may be using DHCP to receive their IP addresses when they use the Internet. Thus, multiple IP addresses may refer to the same user and the total number of users we saw may be fewer. Although broadband Internet users usually keep their IP addresses for several days or even weeks, we are currently investigating ways to determine the correct identity of users.

4.5. MEASURED VOICE PATTERNS

Voice patterns in multiparty voice communication consist of talkspurts (on periods) and silence (off periods). We measured these and the inter-talkspurt arrival time to characterize voice patterns. TeamSpeak uses 100ms long frames, therefore the shortest talkspurt in our case is 100ms. To be consistent, the smallest measureable silence period must also be 100ms. The inter-talkspurt arrival time is measured as the time between any two in-sequence talkspurts observed by the server. Since the smallest talkspurt is 100ms and the smallest silence period is 100ms, then any inter-talkspurt arrival time that is at least 200ms for a given user is interpreted as silence. Note that, if we have multiple users using the server at the same time the talkspurts can overlap and thus the inter-talkspurt arrival time can be shorter than the talkspurt itself.

In order to measure the voice patterns, we captured the voice packets during the peak periods (7pm–9pm). After sorting and analyzing the data we realized that our data points did not fit on a linear curve. Therefore, we identified the extreme outliers using the method described in Section 3.2 and removed them from our data set. When we applied the cleaning procedure to the inter-talkspurt arrival times, we removed 341 data points. Table 2 shows the results of cleaning the inter-talkspurt arrival times.

Figure 7 plots the inter-talkspurt arrival times seen at the server during the peak periods. The majority (90%) of the inter-arrival times is less than 7.65 sec. However, the remaining 10% of the data forms a tail which stretches to 536.83 seconds. Note that we only include the first 100 seconds and use a log-log scale in the graph so that the CDF can be seen more clearly.

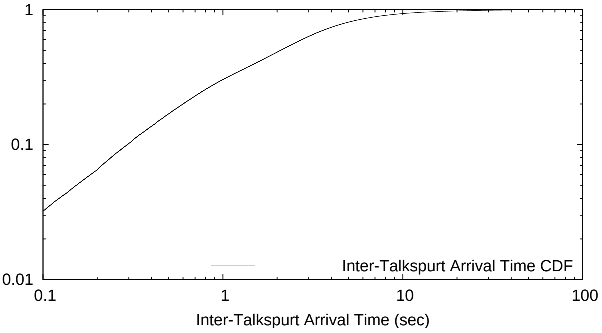

Figure 7: *Inter-talkspurt arrival time:* The majority (90%) of the inter-talkspurt arrival times are less than 7.65 sec. The resulting CDF appears to be exponential in nature.

We collected the talkspurts and silence periods for each of the users during the peak period. We then merged these sets into a single data set and found that the data was non-linear. We transformed it and deleted the extreme outliers with the results listed in Table 2.

In Figures 8 and 9, we plot the CDFs of the talkspurts and silence periods, respectively. Both CDFs appear to follow an exponential distribution, which we explore further in Section 5. The expected value of the talkspurts is much lower than the expected value of the silence periods. 90% of the talkspurts are shorter than 5.40 sec, whereas the same measure for the silence periods is 70.11 sec, which is around an order of magnitude higher. This implies that the users tend to listen more than they talk. After the filtering process, our lowest talkspurt value was .1 sec and our highest value was 96.46 sec. This can be seen in Figure 8.

When we analyzed the silence periods, the filtering process did not affect our data set (see Table 2). This is due to the fact that the expected value of our exponential-like curve was higher and thus the IQR was broader. In addition, because our measurements were only performed during the peak period, the silence periods have an upper bound of 3 hours (or 10,800 secs). The silence period data set ranged from .1 sec (the minimum possible silence period) to 7036.95 sec (almost 2 hours). However, in order to examine the curve of the CDF better, we only include the first 1000 seconds in Figure 9. In the next section, we model the data sets mathematically and discover that both the talkspurt and silence periods are better modeled by Weibull distributions.

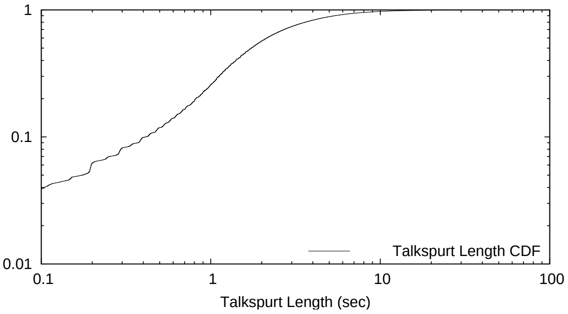

Figure 8: *Talkspurts:* The majority (90%) of the talkspurts are less than 5.40 sec.

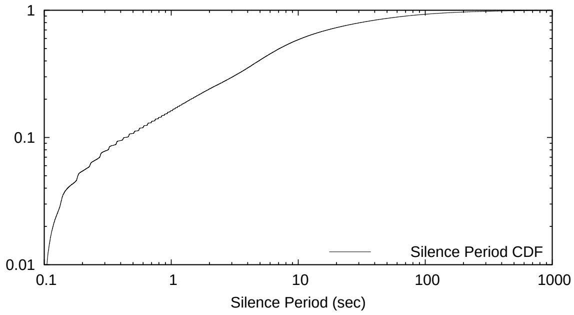

Figure 9: *Silence periods:* The majority (90%) of the silence periods are less than 70.11 sec, which is much higher than the talkspurts.

5. MODELING MULTIPARTY VOICE COMMUNICATION

We now model multiparty voice communication. We have three primary factors that we need to model mathematically: talkspurts, silence and group sizes. With these models, we can simulate and predict the characteristics of multiparty voice communication, regardless of whether a client/server, peer-to-peer or hybrid architecture is used.

5.1. METHODOLOGY

Initially, we thought that the data appeared to follow some kind of exponential distribution, but as we analyzed the data further, we discovered that it fits a Weibull distribution better. Note that this differs from previous research in classical telephony and VoIP conversations which showed that the data followed an exponential distribution.

In order to model the conversations, we first estimated the parameters of the exponential and Weibull distributions. We looked at other distributions, but found that these two distributions had the best fit with our data. We then validated our estimation by calculating the mean and standard deviation of the residuals and by using the λ^2 test.

5.2. PARAMETER ESTIMATION AND ERROR CALCULATION

For parameter estimation, we used the least-squares method to minimize the square of the sum of the residuals for both the Exponential and Weibull distributions.

In order to justify the correctness of our estimation, one could perform a goodness-of-fit test. However, traditional tests, such as Chi-square (χ^2) and Kolmogorov-Smirnov (KS) are not suitable for data from Internet traffic [12]. Moreover, these tests are biased against large data sets [6], such as the ones that we have.

We use two methods to determine if the data fits a particular distribution. After we have used the least squares method to estimate the parameters for a distribution, we plot the residuals and examine their mean and standard deviation. These values give us an idea of how well our model predicts the data. In addition, we use the λ^2 method as a discrepancy tool [13]. We describe how we used the λ^2 method and how we binned our data in Subsection 5.3. With the λ^2 method, we can compare the fit between two possible distributions.

5.3. USING λ^2 FOR NETWORK MODEL EVALUATION

The quantity λ^2 is the discrepancy between an actual and an assumed statistical model, which is the measure of the goodness-of-fit of the estimated curve. However, this method can be applied to data in different ways. Here, we present the details of how we applied it.

The λ^2 metric is defined as follows:

$$\lambda^2 = \frac{\chi^2 - K - df}{n-1} \tag{2}$$

where n is the total number of datapoints and df is the number of degrees of freedom of the test.

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{3}$$

and

$$K = \sum_i \frac{|O_i - E_i|}{E_i} \tag{4}$$

Since this discrepancy is based on Pearson's χ^2 test, it requires the binning of the data. Here O_i is the observed number of datapoints in bin i and E_i is the estimated number of datapoints in bin i. Please note that not just χ^2 but K is also dependent on the number of bins and therefore determining this parameter can be crucial. If the parameter is too large, the estimate will be too rough; on the other hand if the parameter is too small than the distribution of the datapoints will be too smooth, equivalent statistically to imprecise estimation.

In our paper we used $1 + 2 \times 2 \times \log_{10} n$ equiprobable bins [9]. If this was not a whole number we took the floor of it. This method ensures that if we have at least one datapoint the denominator of neither χ^2 nor K can be zero. Our experience is that these parameters were accurate and worked well with our data because they were in accordance with what we expected based on the graphs.

5.4. MODELING TALKSPURTS AND SILENCE

To model the talkspurts and silence periods, we looked at the packets sent and received during the peak period on the server (from 7pm to 9pm). We focus on this period because the model needs to be able to predict the behavior under peak loads. After looking at the data, graphed in Figure 8, we hypothesized that the data followed the exponential distribution.

Table 3: *Experimental values:* The Mean, Min and Max are calculated from the data sets. Using our parameter estimation methods, we calculated the parameters for the CDFs of the exponential and Weibull distributions. The λ^2 values are the results of using the λ^2 test to determine the accuracy of our fit (smaller is better). For both the talkspurt and silence data sets, the Weibull distribution is a better fit.

	Talkspurt	**Silence**
Mean	2.74s	35.90s
Min	0.1s	0.1s
Max	96.46s	7036.95s
Exponential estimated parameters	$\lambda = 0.4185$	$\lambda = 0.0877$
Weibull estimated parameters	$\lambda = 2.3002$ $k = 1.1846$	$\lambda = 13.5275$ $k = 0.6168$
λ^2-test for exponential	0.0999	0.2739
λ^2-test for Weibull	0.0769	0.0636

Table 4: *Residuals from Model:* The max, min, and standard deviation of the residuals between the modeled CDFs and the talkspurts and silence periods.

	Talkspurt	**Silence**
Max	0.0401	0.0269
Min	-0.0350	-0.0382
Std.Dev.	0.0190	0.0180

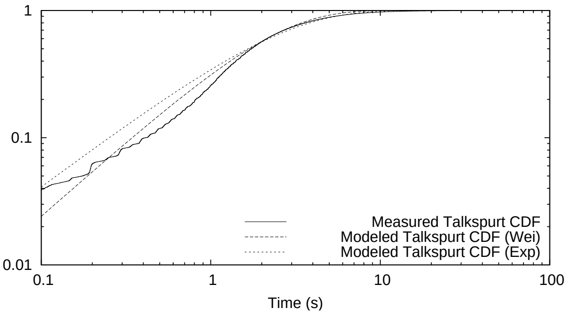

Figure 10: *Modeling talkspurts:* Visually, we see that the Weibull CDF ($k = 1.1846, \lambda = 2.3002$) slightly overestimates the number of short talkspurts around the 10s range but otherwise it is a better fit than the exponential ($\lambda = 0.4185$).

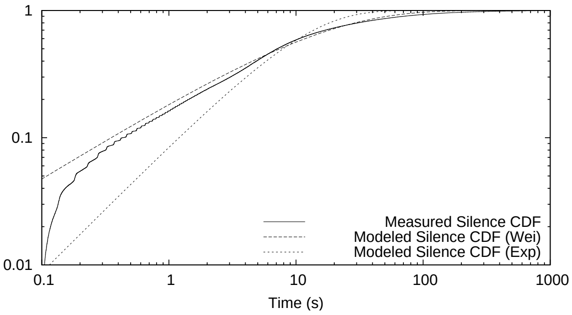

Figure 11: *Modeling silence:* We model the silence period using a Weibull CDF ($k = 0.6168, \lambda = 13.5275$). We see that like the talkspurts, the Weibull distribution fits better than the exponential distribution ($\lambda = 0.0877$).

Our first attempt at modeling the talkspurt and silence periods was to try an exponential distribution. Table 3 lists the means and parameters we estimated using the least-squares method for the exponential distribution.

We then tried the Weibull CDF which is a generalization of the exponential distribution. We estimated the parameters for the Weibull distribution for both the talkspurts and silence periods (Table 3). We then plotted the talkspurt and silence data sets along with both exponential and Weibull CDFs using their estimated parameters. The talkspurt graph with its models can be seen in Figure 10. Visually, the Weibull CDF appears to be a better fit for the talkspurt data set. We plotted the residuals (not shown) to examine their mean and standard deviations and summarize them in Table 4.

We then used the λ^2 test on the CDFs and validated that the Weibull CDF fits better than the exponential as shown in Table 3. Thus, *unlike prior results which showed that an exponential distribution better modeled talkspurts, we found that the Weibull CDF more accurately models the talkspurts of multiparty voice communication.*

For the silence periods, we repeated our method of plotting the data set with both the exponential and Weibull CDFs and their estimated parameters, as shown in Figure 11. To further validate the results, we plotted the residuals (not shown), which are the differences between the predicted values and observed values. The residuals shows us that the model is off by at most 4%, with a standard deviation less than .02 as shown in Table 4. Using the λ^2 test, we see that our estimated Weibull CDF is indeed a better fit than the exponential distribution (Table 3). Thus, *the silence periods are more accurately modeled with Weibull CDF for multiparty voice communications*.

5.5. GROUP EFFECTS ON TALKSPURT AND SILENCE

Besides the talkspurt and silence distributions, we wanted to understand how group sizes affect these distributions. We hypothesized that as the number of people in a group increased, the mean talking time decreased while the mean silence time increased. To study this, we plotted the mean talkspurt and silence times versus the group sizes observed during our measurement period.

As TeamSpeak does not use a group identifier in the messages, it is impossible to identify the groups with 100% accuracy. However, for modeling the behavior of the groups with different sizes it is not essential to associate the messages to a particular group. Simply knowing the size of the group that a message was sent to would be sufficient if this method was also capable of grouping the silent periods based on the group size. Thus, we counted the number of replications for each of the incoming messages from a given user. Next, we used this group size to determine the group size for the following silence period. This way we could associate a group size to both the talkspurts and the silence periods. The only time when our method fails is when a player leaves or joins a group during a silence period. However, this event is very unlikely and therefore our solution is capable of providing an accurate result.

Figure 12 shows the mean talkspurt and silence times versus the group size. We only show groups of up to 8 people due to the fact that while we did observe groups with up to 24 people, the number of data points in these larger groups were too few to be statistically meaningful.

Looking at this graph, we see that the mean talking and mean silence time do not change significantly, regardless of the group size, contradicting our hypothesis. To investigate this unexpected result, we ran a script which looked at the number of people talking in a group and found that as the group size increases, the number of completely silent people increases (e.g., they only have headphones, but not a mic to speak on). In essence, *a single group appears to support a maximum amount of conversation, regardless of the group size.* We expect that future architectures and codecs may be able to take advantage of this information.

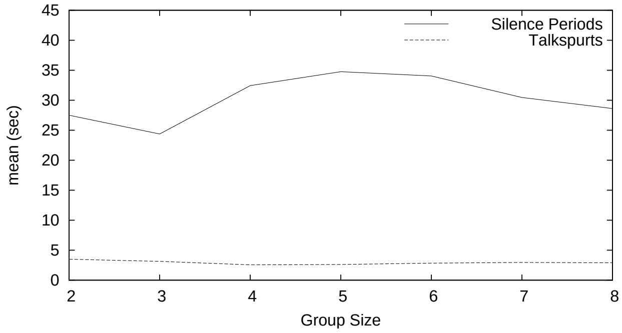

Figure 12: *Talkspurts and silence periods among groups:* Note that the mean talkspurt and silence times are fairly constant.

6. CONCLUSION AND FUTURE WORK

We have presented the first work that examines the characteristics of multiparty communication for games. While VoIP has been successful for point-to-point communication, and research has looked at the feasibility of VoIP over the Internet, our work is the first to address multiparty voice communications on the Internet.

Our results show familiar and new trends. First, as we modeled the talkspurts and silence periods, we found that both types of data fit a Weibull CDF, which differs from previous work on traditional telephony and VoIP that shows talkspurts following exponential distribution. Moreover, we showed that the length of the talkspurts and silence are always the same regardless either of the game played or the group size. On the other hand, the distribution of our daily traffic was similar to other works in both games and VoIP, where server usage peaked during evening hours and on weekends.

Finally, human protocols seem to be at work here as our measurements indicate. The increase in group sizes does not increase the amount of input traffic linearly, though output traffic is necessarily linear in the number of packets received. This is simply due to the fact that humans best process voice information when only one person is talking at the same time. Thus, if more than one person starts talking, other speakers naturally back-off and wait for their turn.

As for future work, we plan on using our models for simulation of client/server and peer-to-peer multiparty voice communication systems.

REFERENCES

[1] BASET, S., AND SCHULZRINNE, H. An analysis of the Skype peer-to-peer internet telephony protocol. In *Proceedings of IEEE Infocom* (April 2006), pp. 1–11.

[2] BORELLA, M. Source models of network game traffic. *Computer Communications 23*, 4 (February 2000), 403–410.

[3] BRADY, P. A statistical analysis of on-off patterns in 16 conversations. *Bell Systems Technical Journal 47*, 1 (January 1968), 73–91.

[4] CHEN, K.-T., HUANG, C.-Y., HUANG, P., AND LEI, C.-L. Quantifying Skype user satisfaction. In *Proceedings of ACM SIGCOMM* (2006), pp. 399–410.

[5] FÄRBER, J. Network game traffic modelling. In *Proceedings of the 1st workshop on Network and system support for games* (2002), pp. 53–57.

[6] GLESER, L. J., AND MOORE, D. S. The effect of dependence on chi-squared and empiric distribution tests of fit. *Annals of Statistics 11*, 4 (1983), 1100–1108.

[7] HENDERSON, T., AND BHATTI, S. Modelling user behaviour in networked games. In *MULTIMEDIA '01: Proceedings of the Ninth ACM International Conference on Multimedia* (New York, NY, USA, 2001), ACM Press, pp. 212–220.

[8] JIAN, W., AND SCHULZRINNE, H. Analysis of on-off patterns in VoIP and their effect on voice traffic aggregation. In *Proceedings of Computer Communications and Networks* (Oct. 2000), pp. 82–87.

[9] LARSON, H. J. *Statistics: An introduction*. Wiley, New York, NY, USA, 1975.

[10] MARKOPOULOU, A. P., TOBAGI, F. A., AND KARAM, M. J. Assessing the quality of voice communications over internet backbones. *IEEE/ACM Trans. Netw. 11*, 5 (2003), 747–760.

[11] PAPP, G., AND GAUTHIERDICKEY, C. Characterizing multiparty voice communication for multiplayer games (extended abstract). In *to appear in ACM Sigmetrics* (June 2008).

[12] PAXSON, V. End-to-end routing behavior in the internet. *IEEE/ACM Trans. Netw. 5*, 5 (1997), 601–615.

[13] PEDERSON, S. P., AND JOHNSON, M. E. Estimating model discrepancy. *Technometrics 32*, 3 (1990), 305–314.

[14] PITTMAN, D., AND GAUTHIERDICKEY, C. A measurement study of virtual populations in massively multiplayer online games. In *Proceedings of ACM NetGames* (September 2007).

[15] SRIRAM, K., AND WHITT, W. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Selected Areas in Communications 4*, 6 (1986), 833–846.

[16] SVOBODA, P., KARNER, W., AND RUPP, M. Traffic analysis and modeling for world of warcraft. *2007. ICC '07. IEEE International Conference on Communications* (24-28 June 2007), 1612–1617.

[17] WU-CHANG FENG, CHANG, F., WU-CHI FENG, AND WALPOLE, J. Provisioning on-line games: A traffic analysis of a busy counter-strike server. In *Internet Measurement Workshop* (2002).