Virtual Worlds, Real Traffic: Interaction and Adaptation

Iain A. Oliver
School of Computer Science
University of St Andrews
iao@cs.st-andrews.ac.uk

Alan H. D. Miller
School of Computer Science
University of St Andrews
alan@cs.st-andrews.ac.uk

Colin Allison
School of Computer Science
University of St Andrews
ca@cs.st-andrews.ac.uk

ABSTRACT

Metaverses such as Second Life (SL) are a relatively new type of Internet application. Their functionality is similar to online 3D games but differs in that users are able to construct the environment their *avatars* inhabit and are not constrained by predefined goals. From the network perspective metaverses are similar to games in that timeliness is important but differ in that their traffic is much less regular and requires more bandwidth.

This paper contributes to our understanding of metaverse traffic by validating previous studies and offering new insights. In particular we analyse the relationships between application functionality, SL's traffic control system and the wider network environment. Two sets of studies have been carried out: one of the traffic generated by a hands-on workshop which used SL; and a follow up set of controlled experiments to clarify some of the findings from the first study. The interplay between network latency, SL's traffic throttle settings, avatar density, and the errors in the client's estimation of avatar positions are demonstrated. These insights are of particular interest to those designing traffic management schemes for metaverses and help explain some of the oddities in the current user experience.

Categories and Subject Descriptors

C.2 [**Computer Communication Networks**]: General—*Data communications*; C.4 [**Performance of Systems**]: Performance attributes; H.5 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Artificial, augmented, and virtual realities*

General Terms

Design, Experimentation, Measurement, Performance

Keywords

Multi media, traffic measurement, virtual worlds, QoS

MMSys'10, February 22–23, 2010, Phoenix, Arizona, USA.

Figure 1: A Byzantine Spartan Basilica

1. INTRODUCTION

Metaverses are 3D computer generated environments which users, often called *residents*, populate through the proxy of an avatar. They allow people to interact socially and express themselves in new ways. Unlike their cousins, 3D games, there are no predefined sets of goals for users to aim towards in order to progress. Residents are free to imagine their own activities and engage in pursuits which run the gambit from setting up their own business through staging a collaborative theatrical installation to simply playing a traditional computer game.

Second Life (SL) is a popular metaverse - over a million residents logged in between Mid August and Mid September 2009 [1]. SL supports a wide range of activities which include, but are not limited to: theater, education, live music, art exhibitions, online sex, virtual embassies and games. The diverse functionality and rich variety of activities found in metaverses such as SL is possible because they enable residents to go beyond simply interacting with the environment and support them in the creation of content for the world. Residents can build mountains, excavate valleys, create simple geometric shapes and combine them into complex structures. These structures may also be made interactive by embedding scripts in them. In a sense metaverses do for 3D environments what Web 2.0 did for the original WWW - they allow the consumer to become a creator.

This paper focuses on the network communications of SL. Our interest in this topic was motivated in part by experience in using SL to support novel approaches to exploratory learning in the realms of virtual field work, virtual laboratories and Human Computer Interaction.

In collaboration with archaeologists we have designed a virtual archaeological excavation [14], based upon a real ex-

---

[1] `http://secondlife.com/whatis/economy_stats.php`

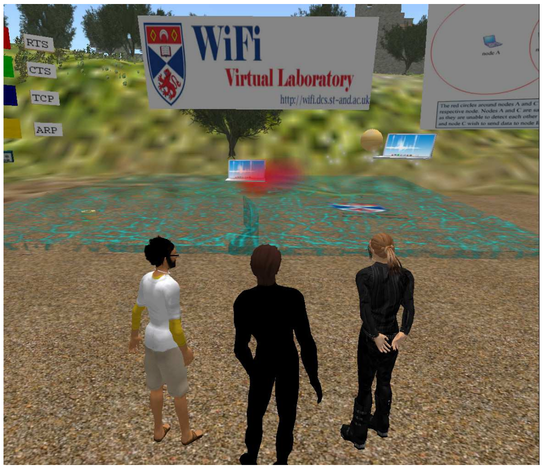


Figure 2: WiFi Virtual Laboratory

cavation of a Byzantine Spartan Basilica. In this students are able to conduct surveys of the site in 3D, participate in a collaborative virtual excavation, construct a virtual exhibition and explore a 3D reconstruction of the Basilica as shown in Fig. 1.

In order to support honours and masters modules in computer networking, we have constructed a WiFi Virtual Laboratory (WiFiVL) in SL. Students are able to construct 802.11 wireless protocol scenarios by placing networking components around an Island and configuring a traffic matrix [34].

Thirdly, as part of a Human Computer Interaction course students have designed and implemented tools for learning about Dijkstra's shunting algorithm [10].

The experience of using SL has been a positive one. In each case we received enthusiastic feedback from students who found the environment to be engaging. This positive feedback is complemented by the impressive quality of work produced and the grades achieved in assessments [31].

In the process of utilising SL a number of issues arose. IT staff shared a widespread concern that SL is a greedy application which impacts badly upon the peer traffic it shares the network with. Another problem is that compared to state of the art computer games, metaverses are considerably less "slick". Whilst moving around the environment, buildings often take a while to appear, and when they do bits may be missing. Errors in avatar positioning can create odd effects: when bumping into walls it sometimes appears as though the avatar is halfway through the wall and on many occasions movement is not as smooth as desired. Metaverses rely upon network communication to allow distributed users to interact through the proxy of avatars. For an immersive experience it is important that network communication is timely[2]. Unlike 3D games the client does not hold a model of the entire world. As avatars move around, their immediate environment is downloaded dynamically to the client. Consequently, metaverses require significantly more network bandwidth than 3D games.

Metaverses are not usually limited in size or number of simultaneous users. To support this, the world is split up into segments which are distributed across servers. Where there is one instance of a metaverse for the entire user base[3], as in the case of SL, significant Round Trip Times (RTTs) for some users are unavoidable. The combination of the importance of: low latency for avatar control; the increase in bandwidth required for dynamic downloading; and the larger RTTs implied by a single world, make delivering the desired Quality of Service to a metaverse a live and challenging subject.

In the remaining sections of this paper we look at SL system structure and traffic in order to discern and analyse the different elements that interact to perform network communication. This is followed by a discussion of related work in the fields of network measurements for online computer games and metaverses. Reports on two studies are then made. One is based upon traffic measurements taken while SL was used in a hands-on workshop for thirty academic staff. The second is a series of controlled experiments exploring the parameter space. The network measurements are analyzed in order to better understand how SL traffic affects the fidelity of avatar interaction and how it relates to other network traffic.

2. SECOND LIFE STRUCTURE

In this section the main features of SL and its communication system are outlined. SL uses the Havoc 4 physics engine, which applies Newtonian-like physics to objects in SL. *Primitives* (prims) are the basic *in-world* building blocks, they provide the framework around which structures can be created. *Textures* are essentially image files that can be applied to prims to give them the desired look. There is a particle-based visualisation system for the rendering of fire, smoke, clouds and similar. Other elements that contribute to the environment include *Wind*, which makes a noise and stimulates movement from trees and grass, and *Clouds*, which move across the sky and the *Land*. In-world elements can be animated and their response to interactions controlled by attaching code snippets written in the Linden Scripting Language (LSL). These scripts are written and activated in-world.

Users are represented by avatars which have an *inventory* associated with them. The inventory is a Pandora's box which may contain clothes, cars, buildings and any amount of weird and wonderful objects. Avatars move around the world, by walking, running, flying and teleporting. Avatars move in response to input from their user. User input is sent to the server, where the movement of the avatar is simulated before the server returns the avatar's position and motion vector to the client. In the absence of server updates and in order to allow the avatar to move smoothly the client continues the movement and acceleration of the avatar but without simulating any physics. Errors can therefore occur in the client's estimation of the avatar's position.

There is a single shared public instance of the entire SL world, which is simulated by multiple colocated servers. Some servers record specific information about the entire simulated world, for example: the login server, which performs user authentication. Other servers are called Sims and simulate a part of the metaverse. Each Sim server may simulate multiple Islands, each Island measures 256 by 256 SL meters. An avatar may be able to see multiple Islands and may need to connect to multiple Sims.

---

[2] Consequently, like most 3D games, using TCP is problematic

[3] There exist testing grids to test changes to Second Life.

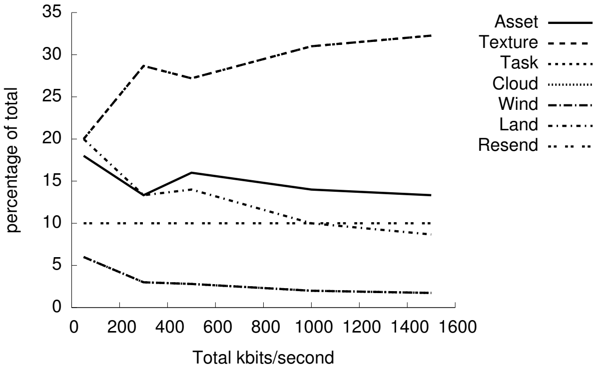


Figure 3: Bandwidth Allocation between Channels.

A *circuit* is a virtual connection, established on top of UDP, which organises communication between the client and its servers. Every client maintains one circuit for each server it is communicating with. There are over 473 types of packet. Each type corresponds to the function that the payload fulfills. Examples of packet types include: AgentUpdate which contains current camera information from the client to the server; ObjectUpdate, which sends object updates for avatars and prims from the server to the client; and LayerData, which contains information about the Land, Wind or Clouds. SL circuits use sequence numbers and timeouts to detect packet loss. The RTT is measured using SL's own ping packets. If a gap in the sequence space is detected and remains unfilled for 16 * RTT, a loss is assumed to have occurred. If a reliable packet is not acknowledged it is resent.

The downloading bandwidth to an SL client is controlled by a *throttle* system. The initial value defaults to 500 Kbps but may be set by the user to any value between 50 Kbps and 1500 Kbps. SL divides traffic into different channels which are allocated a proportion of the global throttle. Capacity that is unused by one channel may be reallocated to another. The purpose of each of the seven channels is given below along with the percentage of the total bandwidth when the global throttle is set to its default value of 500 Kbps:

1. Asset: 16%. Largely information about avatar inventory.
2. Texture: 27%. The images applied to objects.
3. Task: 27%. Contains packets relating to prims, the particle system, trees and avatar control traffic.
4. Wind: 2.8%. Moves trees etc. and makes sound.
5. Land: 14%. Land and water are represented as a height map.
6. Cloud: 2.8%. The distribution and movement of clouds.
7. Resend: 10%. Reliable resent packets from other channels.

Figure 3 shows channel allocations as percentages of the total. Note that as the total throttle increases, so does the absolute and relative differentiation between channel bandwidth allocations. The client may adjust the global throttle in response to network conditions. Initially the global throttle is set to 1.5 times the user setting. This is used as a session maximum. Packet loss is taken as an indication of congestion. If significant congestion is detected SL cuts the global throttle.

3. RELATED WORK

There have been several previous studies into SL Traffic. Two papers analyse client side traces of traffic generated for islands with different concentrations of objects and avatars, for defined avatar activities [12, 21]. Other studies have used avatar crawlers [36, 23] to collect server side statistics on loads and usage patterns. The subject of creating models of SL traffic is treated in [2].

3.1 Measurement of SL Traffic

In [12] measurements of Avatars Standing Walking and Flying in popular and unpopular places is presented. In [21] the authors add teleporting to the list of activities measured. They also separately quantify the effect of the density of objects and number of avatars on throughput. Whilst some throughput measurements are larger than [12] they are in the same order of magnitude. Walking in a busy, dense place requires average bandwidth of 775 Kbps and Teleporting requires 1164 Kbps. Both zone and avatar action have a marked effect on bandwidth. The authors note that the magnitude of the effect of zone is 2.5 in the first study and 7 to 9 in the second.

In [36] comprehensive server side statistics show that total network load follows a diurnal pattern ranging from a low of 1.7 million packets per second to a peak of 3.2 million packets per second. An average packet size of 500 bytes is assumed based on [12]. This gives a total load of 18.9 Gbps and an average of 280 Kbps per client. 50% of user sessions lasted ten minutes or less. Avatars spent 80% of their time with the rest divided between Walking, Running, Flying and Teleporting. Walking and Teleporting accounting for more than 90% of the remainder.

La and Machiardi show that avatar movement in SL [23] mirrors that of people in the real world. An analysis of textures in SL is presented in [22], individual and aggregate texture sizes are quantified, with a median size of 95 Kb and the studied regions having between 1 and 4 thousand textures.

An analysis of SL network traffic is presented in [24]. This is based upon two three minute sessions connecting to the Swedish Embassy. They observe that during the second run the SL cache significantly reduced bandwidth.

The construction of a synthetic model for SL traffic is advocated in [2]. It is suggested that the ability to model SL traffic would help game developers and network providers. Models of inter-packet arrival time and packet size distributions were derived for standing and walking in popular and unpopular places.

3.2 Comparison with Games

There have been separate studies into network traffic of First Person Shooters [11, 17], Massively Multi player Online Role Playing Games (MMORPG) [6, 7], and Real Time Strategy (RTS) games [8, 35].

In [21], an authoritative comparison of SL and 3D game traffic is given. This considers packet size, inter-arrival time and throughput. Based upon classification in [9], one example from each of three three genres is chosen. Here are the figures for bandwdith; First Person Avatar (Unreal Tournament [27]) 67 Kbps, Third Person Avatar (Madden NFL [27]) 14 Kbps and Third Person Omnipresent (WoW [33]) 5 Kbps, compared to SL 775 Kbps. The bandwidth requirements of SL are around ten times that of 3D computer games, in addition SL has larger packets. The authors conclude; *"this large turbulence suggests that meeting the quality of service requirements of Second Life, which are likely to be similar to that of third person avatar games, is a challenge."* [21].

As 3D games and SL are are both 3D environments it is not unreasonable to assume that QoS requirements of metaverses are similar to 3D games. In fact, one of the popular activities undertaken in SL is playing 3D games, examples include; New Babbage "a steampunk city ... a place they can role play and be creative"[4], "I Am Legend: Survival " a multi player first-person shooter[5] and "The Amber Raceway", which supports track racing [6].

There have been studies on latency and user satisfaction. In [27] the effect of latency on an American football game: "Online Madden NFL Football", shows that there is a negative correlation between performance in the game and latency. Henderson [18, 28] shows Half-Life players choose servers with low latency. In [7] it is shown that network latency, jitter and loss all affect the length of time a user spends online.

A comprehensive survey of the effect of latency on 3D games is given in [9]. It is shown that latency is important to the user experience [17] and to the accuracy with which tasks can be completed [9]. Environments where the avatar is controlled from a first person perspective (as metaverses are) are more sensitive than environments where the user has a global perspective. The tasks being undertaken also effects sensitivity to latency. Simple navigation through an open terrain is less sensitive than navigation through a complex environment [9]. The wide range of activities that metaverses aspire to support means latency sensitive tasks are encompassed.

3.3 Comparison with TCP

Whilst computer games are an important point of comparison with SL, there are a wide range of applications which metaverse traffic has to share the network with. A comparison with traffic that is regulated by TCP is valid and important as over 80% of Internet traffic uses TCP as a transport protocol. TCP's congestion control mechanisms come out of research conducted in the late eighties. Severe congestion stimulated research into its control. Raj Jain defined a notion of Max Min fairness [20], and designed a class of Additive Increase Multiplicative Decrease (AIMD) [32] algorithms which result in a Max Min fair allocation of resources. Van Jacobsen designed Slow Start and Congestion Avoidance algorithms, which were incorporated into TCP [19]. The combination of Congestion Avoidance and loss detection through duplicate packets is an AIMD algorithm, which achieves statistical fairness [1]. Each flow has similar opportunity to obtain network resource. This macroscopic or steady state behaviour of TCP can be represented by the TCP Fair equation [25] below 1. [7]

$$X = \frac{S}{RTT\sqrt{p}} \tag{1}$$

For applications with real time constraints TCP is often not appropriate, consequently the question of how non TCP applications should regulate their behaviour is an ongoing one. The TCP Fair equations have been used as a reference point for evaluating their behaviour [4, 3].

4. CONTRIBUTIONS

The contributions of this work lies in four main areas. The base of SL traffic measurements is widened thereby providing context and perspective for previous studies. The data is analysed with respect to application performance. Measurements are related to SL traffic management mechanisms. The possibility of applying differential QoS and adaptive traffic management to metaverse traffic are explored.

- This is the first client side measurement study where people are using Second Life for real. It complements previous studies; where data has been collected at the server, or at the client for specific scenarios. It allows average throughput and packet size for a user session to be measured and thereby provides important context and facilitates comparison with TCP Fair behaviour.

- The accuracy with which an avatar's position is represented is measured and analysed. The factors which effects this accuracy are identified. In doing so it is shown that differential QoS which prioritises avatar traffic will improve the users experience. Furthermore our measurements establish that avatar traffic is small in bandwidth and packet size making such prioratisation a feasible endeavour.

- This is the first study to discuss, measure or critically analyse Second Life's traffic management mechanisms of, a user specified throttle, bandwidth allocation between channels, and throttle adaptation to congestion. It shows that for a given activity, within a given zone, there is not a consistent bandwidth that SL operates at. Throttle setting and congestion both impact upon bandwidth. This observation is important for correctly modelling SL traffic and opens up the possibility of intelligently adapting bandwidth usage to balance the needs of the network with that of the user.

- This paper shows that TCP Fair algorithms can be applied to SL so that the bandwidth SL uses more closely matches the congestion signal being received from the network. This has the benefit of allowing SL to automatically make use of extra bandwidth when it is available, whilst cutting back usage when congestion occurs.

5. OBSERVING SL IN THE WILD

This study is based upon measurements of SL network traffic captured during a workshop for 30 faculty staff on using metaverses for teaching. It contributes to our understanding of SL traffic as previous work has focused on measuring specific scenarios rather than actual usage. It also facilitates comparison between the demands placed upon the network by SL with TCP Fair behaviour.

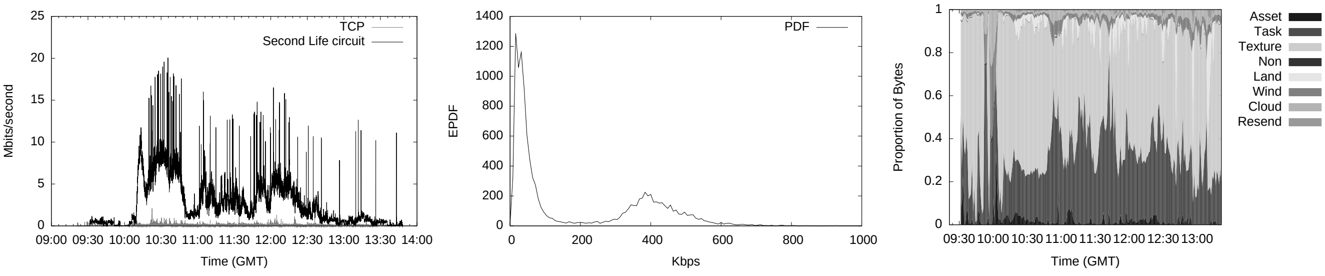
---

[4] `http://newbabbage.ning.com`

[5] `http://iamlegendsurvival.warnerbros.com`

[6] `http://www.amberracing.net`

[7] TCP Friendly Rate Control [15] uses a more complex formula [30]

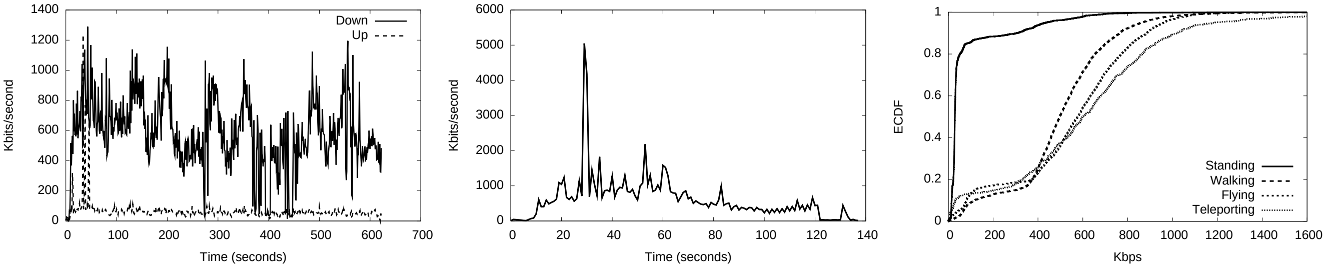


(a) TCP and UDP Throughput (b) Bandwidth Distribution (c) Channel Throughput

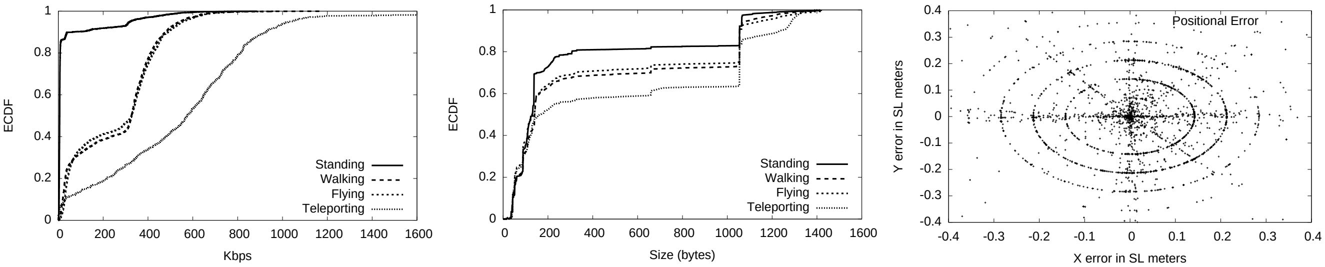
Total	**Packets**	**Bytes**	**Average Size**		**Throughput Kbps**		**% Share**
			All	**Hands On**	**Entire trace**	**Hands On**	
NoThrottle	101307	4816502	48	49	2.4	5.9	3%
Asset	3964	3246081	819	854	1.6	6,3	2%
Task	271174	53095704	195	1632	26	78	34%
Texture	84016	84671328	1007	1007	42	149	54%
Land	6397	5111161	798	906	2.5	6.4	3%
Wind	23194	2557343	110	111	1.3	2.9	2%
Cloud	9890	1967609	199	200	1.0	2.2	1%
Resend	809	845636	1044	1201	0.4	1.1	0.6%
Unknown	435	47983	110	85	0.2	0.1	0.03%
Total	501188	156359347	311	376	78	237	100%

(d) Average channel usage for each participant

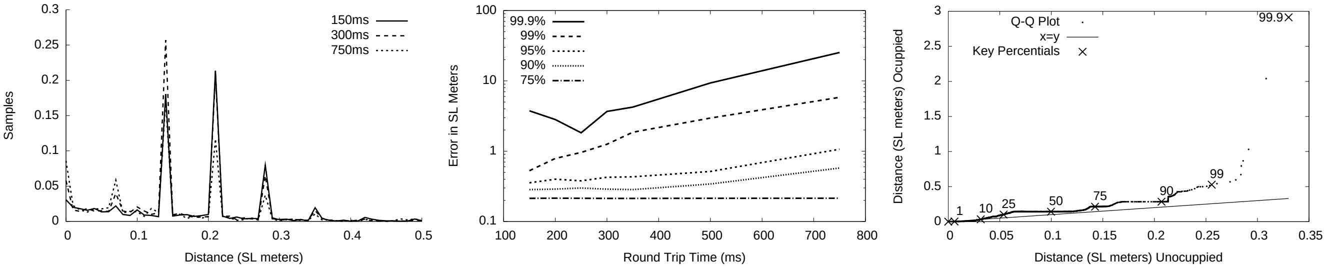
Figure 4: SL in the Wild



(a) Throughput Against Time (b) Throughput for Teleport (c) Throughput for Occupied Island



(d) Throughput for Occupied Island (e) Packet Size for Occupied Island (f) Error the XY Plane



(g) Error Probability Distribution (h) Error by Latencies (i) Error QQ:Islands

Figure 5: Standing, Walking, Flying and Teleporting

The workshop showcased some existing projects prior to a hands-on session during which all thirty participants were concurrently active within SL. The activities focused on exploration and communication. After the hands-on session, there was a break followed by some presentations.

The traffic traces were captured using passive monitoring [26] techniques. The client computers were connected to two daisy chained 100 Mbit switches with a 1 Gbps up link to a router, which connects to the Internet through JANET. The traffic was captured by mirroring the up-link port on the switch to the machine capturing the traffic. The capture was started a little after 09:00 and the introductory session started at 09:30. The participants went for lunch at 12:20. The hands-on part of the session was from 10:10 until 10:45. The capture was finished at 13:45.

A bespoke traffic analysis tool was developed and used for post processing the traces. This tracks the state of SL circuits and groups together packets that belong to the same circuit. For each circuit a data structure is created to hold statistics about the packet flow. The packet type information is used to allocate packets into their channels. Information about the packet size, throughput and inter-arrival times for different packet types, channels and circuits is maintained. The tool also infers packet loss rates and RTTs. The program outputs a textual representation of the trace and can report statistics on the variables it tracks at a range of granularities.

During the workshop 5.1 Gigabits of data were transferred in over 17 million packets. The average throughput was 2.5 Mbps. The quantity of UDP data was 50 times that of TCP. There were a large number of TCP connections, 21,851 and a comparatively small number of UDP flows 332, this averages at 11 UDP circuits per person.

The number of SL packets lost was 3,841, which is 0.02% of the total. The average round trip time was measured as 153ms. Using the TCP Fair Equation 1 with a MTU of the average packet size (304 B) a loss rate of 0.02% and a RTT of 153 ms gives a fair sending rate of 1,455 Kbps.

A time series graph of TCP and UDP throughput is shown in Fig. 4(a). At the start of the trace there is a little traffic. At 10:10 the hands-on session starts and traffic increases. The average throughput for the hands-on session is 6.8 Mbps overall or 231 Kbps per user. Peaks of two and three times these levels are reached. The average throughput for the entire trace is 2.3 Mbps or 76 Kbps for each user.

The distribution of throughput usage for a single user is shown in Fig. 4(b). The downstream UDP throughput for each second is taken as a single sample and the number of samples at each throughput level is plotted. The distribution is binomial. Reading the graph from left to right, the first peak corresponds to periods during which the avatar is stationary or inactive. The second peak is located at slightly less than 400 Kbps. It corresponds to avatar activities such as walking and flying. The proportion of bandwidth each channel uses against time is shown in Figure 4(c), the large light area corresponds to the texture channel. Task and Texture together account for 88% of all bytes. Summary statistics for channel bandwidth usage are shown in Table 4(d). The figures given are the average for each user. Most of the packets (54%) belong to the Task channel, whereas most of the bytes, again 54%, belong to the Texture channel. The Task channel has a small average packet size of 200 bytes whilst the average Texture packet size is 1007 bytes.

A comparison of the bandwidth measurements in this study with [12], shows that whilst 231 Kbps is similar to the authors' figures for walking and flying at the "Goddess of Love" place (220/240 Kbps), our measurements include a mix of activities and so show higher relative throughput. Consequently a more appropriate point of comparison would be the 400 Kbps "activity" peak in Fig. 4(b). If the activity peak is taken as the point of comparison then our measurements are consistent with [21].

The measurements presented here should allay fears about SL being a greedy application. The average throughput of 231 Kbps during the hands on session is 14% of the TCP Fair rate. This suggests that at bandwidth issues should not prevent the utilisation of virtual environments.

6. EXPERIMENTAL INVESTIGATION

In order to investigate the effect of adjusting specific parameters, e.g. throttle setting, and the relationship between traffic and in-world activities such as teleporting, a number of scenarios were designed where controlled activities with specific parameter settings were carried out. In each case the communication with the SL servers hosted by Linden Labs occurred across the Internet against the background of real traffic. A series of experiments were conducted to validate previous studies and to investigate the effect of altering system and user settings. Seven sets of experiments were conducted to:

1. Validate previous studies; client bandwidth and packet size distributions for Standing, Walking, Flying and Teleporting (SWFT) in quiet and occupied Islands were investigated.
2. Establish the accuracy with which the client represents an avatar's position; different values for latency, activity undertaken and composition of environment were measured.
3. Investigate the proportion of traffic that channels utilise and the distribution of packet sizes for individual channels.
4. Further decomposition of the Task channel to establish the throughput and inter-arrival times of avatar traffic.
5. Measure the effect of caching by looking at how traffic patterns change as data is built up in the cache.
6. Establish the relationship between throttle setting and the throughput used for the range of user settings.
7. Compare the performance of SL and a TCP Fair algorithm over a range of congestion levels.

In each experiment, the avatar performs a series of defined actions within specific scenarios. Each scenario lasts for approximately ten minutes. For experiment 1 40 runs were done 5 for each activity island pair. For experiment 2 an additional 20 runs were done to explore, different islands, activities, latency and throttle values. Experiments 3 and 4 were based on runs for experiments 1 and 2. For the cache experiment (5) 48 runs were done 6 for each activity on each island. For experiment 6 24 runs were done at different throttle settings and on different islands. For experiment 7 over 30 runs were done enabling the SL algorithm to be compared with TCP and a TCP Fair window tracking algorithm under a range of loss regimes.

Two Islands were used for the experiments. One Island (Minerva) was unoccupied and had a total of 5586 objects. The other, Miramare, had an average of 12 avatars present and 8749 objects. The client's cache was emptied between each run. A modified SL client was used for experiments to determine positional error. Both the client side position and server position were logged, thereby allowing the discrepancy between the two to be derived and analyzed. Where necessary, connections were fed through a router, which was equipped with the netem network emulator traffic [16]. This facilitated control over the level of delay. TCPDump was used to capture the packets.

The following metrics were considered for each scenario: bandwidth, packet size and inter packet arival time.

6.1 Standing, Walking, Flying, Teleporting

These experiments compare SL traffic for the activities of SWFT on a quiet and an occupied Island. First consider a single avatar walking on the occupied Island. Fig. 5(a) shows the variation in throughput, each reading being the average for ten seconds. The peaks are over 1 Mbps and the troughs below 50 Kbps. The throughput from the server to the client is over ten times that from the client to the server. As the avatar progresses around the Island different densities of prims and textures are downloaded, consequently the bandwidth required to support this downloading is also variable. The time series for a teleport is shown in Figure 5(b).

Table 1 shows average throughputs for SWFT for different islands, the first two are from [12], the next two are from [2], the next three are from [21] and the last two are from the islands looked at in this study. The average bandwidth utilisation for our measurements ranges from 55 Kbps for standing in on Minerva to 604 Kbps for teleporting. UDP traffic is over 90% of the total traffic.

The Empirical Cumulative Distribution Function (ECDF) for the occupied Island, Miramare, is shown in Fig. 5(c). The throughput rises from its lowest for standing, through walking and then to flying. The throughput for teleporting reaches a higher peak than any of the other activities, but has periods of slightly lower throughput than flying and walking.

Figure 5(e) gives the distribution of packet sizes for SWFT in the occupied Island. In each case most of the packets are smaller than 150 bytes and there are few packets between this size and 1100 bytes. For teleporting over 40% of packets are larger than 1100 bytes, for walking or flying 30% are larger where as for standing less than 20% are large packets.

These throughput values for walking on Miramare are greater than the average for the hands on session: 466 Kbps compared to 231 Kbps. This is accounted for as the hands on session would have included a mix of activities, with the avatar standing and chatting or figuring out what to do next.

The results confirm that the throughput of SL varies considerably with time and is orders of magnitude greater than that for 3D games. The throughput required increases with the number of avatars and with the number of objects in a location. Furthermore, the activity being undertaken effects throughput, with a rising rank order of standing, walking and flying. For the environment these experiments were conducted in, it remains a fraction of the TCP Fair value.

6.2 Representation of Avatar's Position

The accuracy with which the system can represent an avatar's position is important for the usability of the system. As latency increases discrepancies are introduced between the user's, client's and server's estimation of where an avatar should be. When the discrepancy grows large it diminishes the fidelity of the metaverse and the accuracy with which tasks can be completed. In this section we present measurements of the discrepancy between server and client estimations of the avatar's location.

Figure 5(f) is a scatter plot of the discrepancy between the client and server's estimation of avatar position in the XY plane. The plot is for an average Round Trip Time of 150 ms. 99% of readings have an error of under half a meter. This is sufficient to enable avatars to appear to partially walk through walls or to interfere with other activities requiring precision, such as a paint ball game. However, it allows many activities that would be undertaken in SL to proceed smoothly.

The concentric circles visible in 5(f) occur where the avatar is moving on the flat without obstruction and the client believes it is stationary, or the client believes there is movement and it is stationary.

The empirical probability distribution function of the absolute error in three dimensions for RTTs of 150, 300 and 750 ms is shown in Fig. 5(g). The distributions extend beyond the x-axis at low concentrations. At all latencies error is concentrated in several regularly spaced peaks. This corresponds to the concentric circle discussed above and reflects the error that can accumulate between packets from the server to the client.

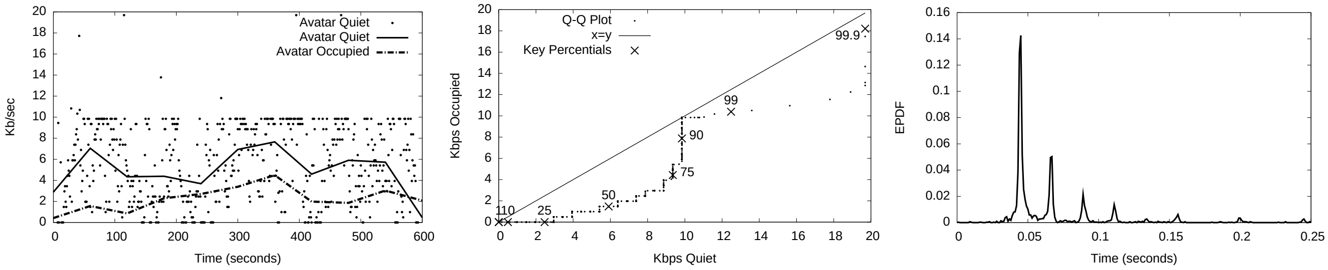
The error at the 75th, 90th, 95th, 99th and 99.9th percentiles, for RTTs ranging from 150ms to 750ms is shown in Fig. 5(h). There is no measurable increase in error below the 90th percentile. This suggests that error in a small proportion of samples has a disproportionate effect on usability. Note that the X axis is to a log scale and the increase in error for the 95th, 99th and 99.9th percentile is approximately linear on this scale.

Table 2 compares the errors for walking with flying on the occupied and the quiet Island. The error for flying is systematically higher than that for walking. At the 90th percentile it is five times greater on the quiet Island and four times greater on the occupied Island. The error is systematically larger for the occupied Island. For walking and flying it is almost twice that found on the quiet Island as shown by Figure 5(i).

		Downstream (Kbits/s)			
SL island	**Country**	**St**	**W**	**F**	**T**
Goddess of Love	BZ	150	240	220	-
Magdelen	BZ	25	70	75	-
Hippie pay	BZ	255	330	250	-
Tuskany IV	BZ	20	77	75	-
Isis	US	192	703	1164	877
Cyclops	US	141	278	445	821
Solaris	US	10	31	448	27
Minerava	UK	55	295	349	604
Miramare	UK	102	466	589	528

Table 1: Throughput SWFT different Islands.

(a) Channel Throughput (b) Channel Throughput (c) Channel Packet Size Distribution



(d) Avatar Traffic (e) QQ Avatar (f) Avatar Inter Arrival Time

Figure 6: Breakdown of SL Traffic

The measurements presented here show that the size of the error depends upon latency, the activity being undertaken and how busy the environment is. At 150 ms RTT the error in avatar representation is within the range where it has an impact on the usability of the system. Consequently, it is preferable to control, and where possible reduce, this error.

6.3 Decomposition by Channel

SL both places a cap on total throughput and allocates bandwidth between eight channels. Allocation and utilisation may however differ as the demand for bandwidth varies with avatar activity. This section presents and discusses measurements for each of the channels, for SWFT in the quiet and the occupied Island. The user setting for the global throttle is set to its default of 500 Kbps. Bandwidth, packets per second and the distribution of packet sizes are examined in turn.

The decomposition by channel of throughput, for SWFT on the quiet and occupied Islands is shown in Fig. 6(a). The largest component comes from the Texture channel, with the Task channel second. For teleporting there is a large Land component. There is also a small amount of traffic in the Asset channel. Cloud and Wind have very little traffic. The Resend channel is negligible for each of these cases except when teleporting to the occupied Island. This suggests that teleporting stresses the network more than other activities.

Scenario	Error in SL meters for given percentile					
	50	75	90	95	99	99.9
W Quiet	0.10	0.14	0.21	0.21	0.25	0.33
Walking	0.18	0.21	0.28	0.36	0.53	3.75
F Quiet	0.62	0.71	1.05	1.07	1.41	3
Flying	1.03	1.39	1.77	2.10	3.5	8.21

Table 2: Error in SL meters at 150ms RTT

Figure 6(c) shows the wide distribution of packet sizes for each of the channels. The Asset channel has 90% of its traffic with packet sizes over 1000 bytes; the Texture channel has a similar distribution. The Task channel is mostly made up of small packets, 60% of them are less than 150 bytes. The Land texture has a wide spread of packet sizes approximately evenly distributed over the range from 40 to 1400 bytes. The remaining Wind and Cloud channels have very little variation in packet size with averages of 112 and 204 bytes respectively.

The distributions of packet sizes within each channel varies little for Standing, Walkig, Flying and Teleporting or between the different Islands, however there is significant change in the overall packet size distributions. This suggests that changes in the proportion of traffic belonging to each channel is important in determining the packet size distribution.

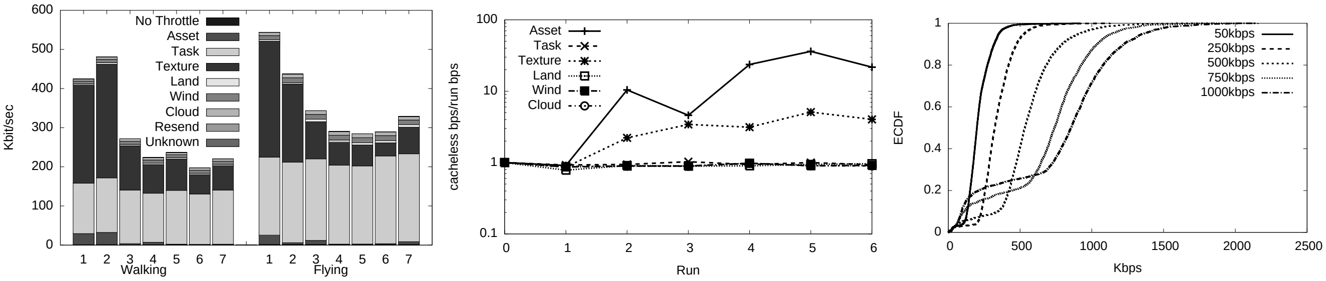
The small packet sizes for the task channel means that although most data bytes belong to the Texture channel most of the packets belong to the Task channel.

6.4 The Task Channel and Avatar Control

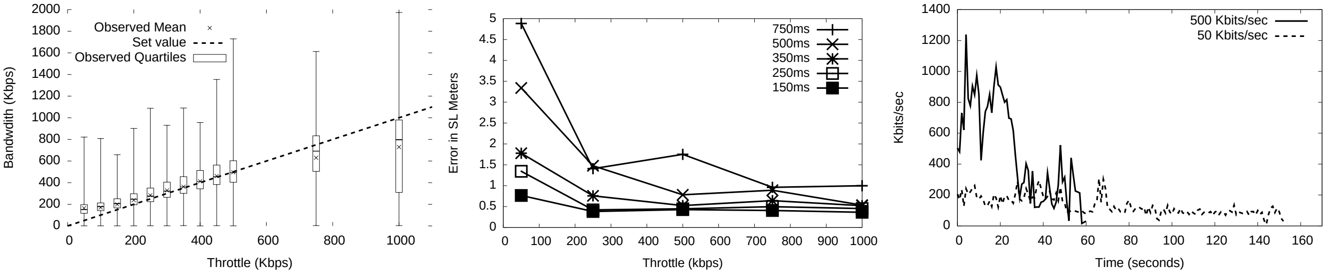
The Task channel contains packets which update: prims, avatar control, the particle system, trees and grass. We look at the composition of the Task channel for Walking on the occupied and quiet Islands. The total bandwidth for the Task channel is 110 Kbps on the occupied Island and around 20 Kbps on the quiet Island. In both cases traffic for trees, grass and the particle system is negligible. Avatar control traffic averages less than 10 Kbps and the majority of traffic is for prims.

The bandwidth used by avatar control traffic is shown in Fig. 6(d). The scatter plot gives the average throughput for a second. The line graphs give the throughput average for each minute. With the exception of a few outliers the maximum throughput in each second is 10 Kbps.

The amount of avatar traffic for unoccupied places is higher than for occupied places. This is despite the overall Task



(a) Sequential Runs, Effect of Caching (b) Channel Sensitivity to Caching (c) ECDF for Throttle Values



(d) Boxplot for Throttle Values (e) Accuracy by Throttle Value (f) Teleport Delay for Throttles

Figure 7: Effect of Caching and Throttles

traffic for the occupied Island being more than five times that of the quiet Island and the activity undertaken by the avatars being the same on both Islands.

The distribution of inter-arrival times for avatar control traffic is shown in Fig. 6(f). Over half the packets have an inter-arrival time of slightly less than 50 ms. There are then regular but decreasing peaks in the distribution up to slightly more than 150 ms. These peaks correspond to the circles in Fig 5(f). The walking speed of an avatar is 30 SLM/sec. The distance travelled by a walking avatar in the time between one peak in inter-arrival time and the next approximates to the distance between peaks in Fig. 5(g). Thus regularities in the distribution of inter-arrival times provides an explanation for regularities in the accuracy of avatar representation.

The reduction in bandwidth for avatar control traffic correlates with a decrease in the accuracy of avatar positional representation. The reduction in avatar traffic for the occupied Island explains the decrease in accuracy of representation. It follows that if it was possible to provide a QoS guarantee for the avatar control traffic then accuracy of avatar representation and consequently the usability of the system would be improved. *This could be achieved through the introduction of an Avatar Control traffic channel.* If the network supported end to end differential service [5] SL would be able to classify Avatar Control Packets as requiring priority service.

6.5 The Second Life Client Cache

SL clients maintain a cache of data between sessions, which can be set by the user to between 10 MB and 1 GB. It defaults to 500 MB. Here the effectiveness of the cache is evaluated by conducting a sequence of scenario runs on the same Island without flushing the cache between the runs. The runs labelled “1” in Fig. 7(a) were conducted immediately after flushing the cache. The run labelled Walking “2” was done next again with an empty cache. From here the cache was allowed to accumulate. Flying “2” was conducted next, followed by Walking 3, Flying 3 etc. Each of these pairs of runs was carried out on consecutive days. As the cache builds up, the average throughput required falls to about half of that when the cache is empty. This is accounted for mainly in the reduction in bandwidth required by the Texture channel. There is little change in the bandwidth requirement for the Task channel.

The sensitivity of each channel to caching can be determined by comparing the throughput required when the cache is empty to the throughput required in consecutive runs as shown in Fig. 7(b). The Asset channel’s bandwidth requirements is reduced by over 90% and Texture’s by over 75%. The other channels are unaffected.

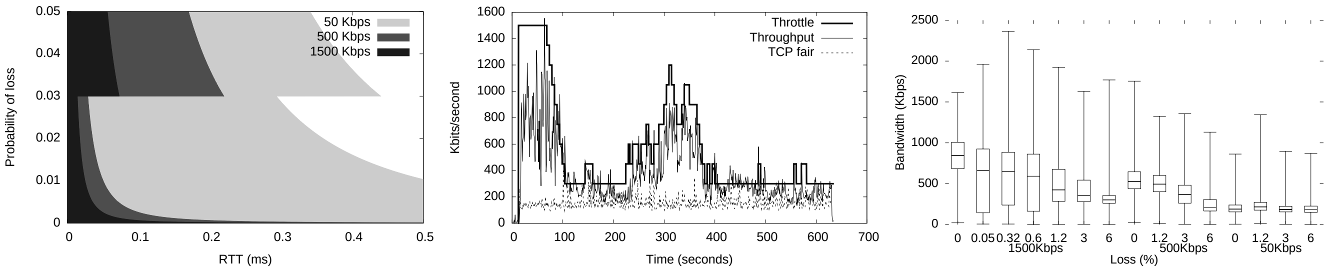
In summary, when a user revisits the same area or Island caching reduces the bandwidth. The Asset and Texture channel are both helped by caching, however the effect on other channels is minimal.

6.6 Throttle Settings

The SL client contains a dialog box for network settings, where the user may adjust a slider to set “bandwidth” to any value between 50 Kbps and 1500 Kbps. The default setting is 500 Kbps. This section seeks to find out what effect, if any, changing the settings on the bandwidth slider have on SL.

At the start of a session and periodically throughout its life the SL client communicates a throttle value to the server. The server uses this value to limit the UDP bandwidth it uses when sending to the client. The throttle value sent to the server is one and a half times the user selected value or 1,500 Kbps whichever is smaller.

In each experiment the avatar was walked around a set course on the occupied Island. Each run lasted ten minutes and throttle settings between 50 Kbps and 1000 Kbps were used. Runs were also conducted where extra latency was introduced giving RTTs of 150ms to 750ms. The distribution



(a) TCP and SL Comparison (b) TCP and SL Throughput (c) SL Throttle Trace

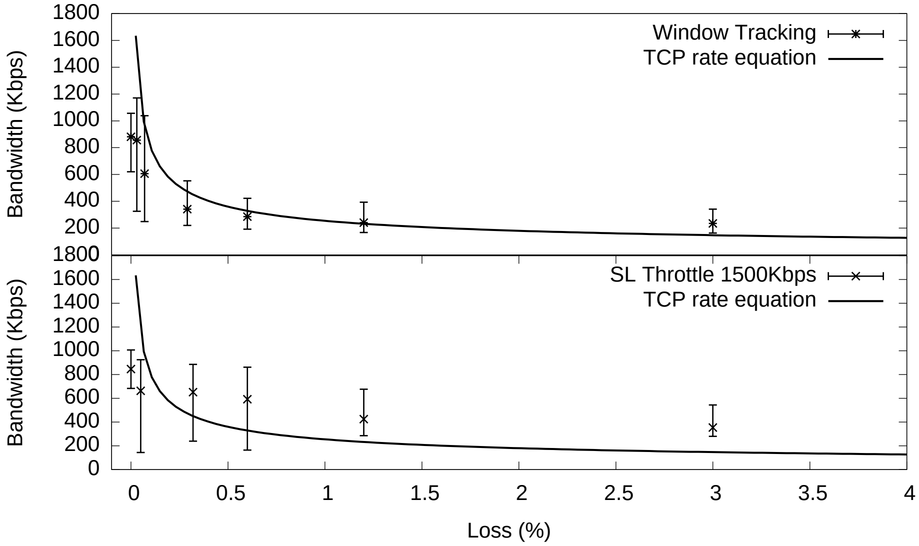


Figure 8: SL Adaptation to Loss

of bandwidth utilisation, the accuracy of avatar representation and the time it takes for a download to complete after a teleportation are discussed below.

Empirical distribution functions for throughput are shown in Fig. 7(c). The mean throughput values and the user throttle settings along with box plots of the median, lower quartile, upper quartile, minimum and maximum value, are shown in Fig. 7(d). As the throttle level increases so too does the mean throughput, which ranges from 170 Kbps to 780 Kbps. However, the user does not get exactly what they asked for. At low throttle settings mean throughput is above the setting, at high throttles it is less than the setting and at 500 Kbps the user setting and mean coincide. At 50 Kbps throttle the throughput is over three times the setting. The variation in throughput increases with the throttle value.

Figure 7(e) shows the 95th percentile of error in the client position of the avatar. As discussed previously error in a relatively small percentage of samples has an important effect on user experience. This graph shows that avatar representation is more accurate at higher throttle settings and at lower RTTs.

Time series which illustrate the delay between a teleportation starting and all relevant data being downloaded are shown in Fig 7(f). Delay is reduced at the higher throttle level. It is also the case that the time it takes for elements of the environment to download is reduced at higher throttle levels.

In summary, changing the user bandwidth setting has an important effect upon the bandwidth utilisation. At higher bandwidth settings accuracy was improved and delay reduced. This is not a setting which should be ignored either by end users or those studying metaverse traffic.

6.7 Throttle Adaptation

When connecting to SL a user may experience a range of network conditions. RTT and level of loss may vary because of factors such as location, time of day and access technology. This is increasingly the case as the range of hand held devices and the popularity of mobile computing increases.

Second Life tracks the level of network congestion and adapts its bandwidth utilisation. Initially the cap on bandwidth may be set by the user to any value between 50 and 1500 Kbps. The initial value used is the smaller of 1500 Kbps and 1.5 times the user setting. During the session the bandwidth cap may range between 0.1 and 1.5 times the initial user setting.

- When the loss rate is less than 0.5% and the current cap is less than the starting value the rate is increased by 0.1 times the set rate.
- When the loss rate is greater than 3% the sending rate is reduced by 0.1 times the set rate.

In this section the behaviour of the SL bandwidth adaptation algorithm is evaluated against a range of congestion regimes, and compared against TCP Fair behaviour. A window tracking algorithm that has been implemented in the SL client is introduced and evaluated.

Under the assumptions that the data source is not application limited, loss is deterministic and for long lived flows, the mean bandwidth produced by the TCP Fair equation and SL adaptation algorithm can be calculated. For RTTs up to 500 ms and for loss regimes up to 5% whether the TCP or SL algorithm will generate more bandwidth is shown in Figure 8(a). Three shaded areas are shown for the minimum, default and maximum user bandwidth settings. The leftmost shaded area represents the region where, with the user setting is 1500 Kbps, SL uses less bandwidth than TCP. At lower user settings the SL algorithm is less aggressive. So, there are a wide range of RTTs and loss regimes where it is less aggressive than TCP, but at the cost of degraded application performance.

Next the behaviour of SL under a range of loss conditions was evaluated. Using netem, runs were done with added loss of 0, 1.2, 3, and 6 percent for the min (50), max (1500) and default (500) throttle values. The loss rate, throughput and throttle values were logged each second.

A time series of throttle values, the measured throughput and the TCP Fair bandwidth is shown in Fig. 8(b), the mean loss rate is 3%. Two things are striking, the throughput is effectively capped by the throttle value, the throttle level increases from 200 time to 400 time seconds, without there being an identifiable reduction in congestion.

The box plots in Fig. 8(c) show that a throttle of 1500 Kbps adapts more to congestion than other settings. However, throughput is higher than the other settings both at low and high loss levels. When the cap is set to 50 Kbps there is little variation in thoughput.

Alternative algorithms for dynamically adjusting the throttle were considered. DCCP's [13] CCID 2 window tracking and CCID 3 equation based algorithms were evaluated using a User Level DCCP [29] library designed for the purpose. A Window Tracking Rate Reporting (WTRR) algorithm has been implemented in the SL client. This mimics TCP's congestion avoidance behaviour by increasing the congestion window additively when packets are successfully received and reducing it multiplicatively (by half) when loss events are detected. RTT is tracked so that the window size can be converted to a rate. The rate is communicated to the SL server as a throttle value.

No claim is made about the optimality of this algorithm. On the contrary further investigation is needed into the applicability of TCPFair algorithms and user perception of QoS. They are presented to show progress from the status quo and to motivate further work in this direction.

The upper graph in Fig. 8(d) shows the box plots of SL throughput using the WTRR algorithm. A guideline of TCP Fair behaviour is provided. The lower graph shows the standard client's behaviour, with 1500 Kbps initial setting. From visual inspection it appears that WTRR more closely tracks the guideline. The SL behaviour is too aggressive at higher levels of loss.

To confirm this correlation the Spearman rank correlation test was carried out for both the Window Tracking algorithm and SL. The correlation between the throttle value and TCP Fair rate as well as throughput value and TCP Fair rate, were tested. The TCP Fair rate was calculated from the measured loss and RTT each second. For Window Tracking the correlation was 6.7 for the throttle and 0.32 for throughput. For SL the correlation was 0.54 and 0.038.

This demonstrates that it is possible to increase the adaptiveness of SL so that it can take advantage of the improved performance available in high bandwidth low congestion environments and automatically adjust to a lower bandwidth utilisation in the presence of congestion. This is achieved without the need to alter the server at all.

7. CONCLUSION

Metaverses are a relatively new class of application. They provide computer based simulated environments where multiple users interact using avatars. The functionality they offer differs from computer games in that users are free to think up their own goals and are empowered to contribute to the development of the environment. This paper has analysed the network traffic associated with Second Life (SL), one of the most popular metaverses.

SL has sophisticated traffic management mechanisms, it performs application level framing, provides reliability for some packets, tracks RTTs and congestion levels, and reduces bandwidth at high loss levels. SL also has seven channels each of which correspond to data that fulfil a distinct functionality e.g. Texture, Asset and Wind. This differentiation enables the system to ensure that no single component is starved of resource. SL network traffic is not homogeneous. Different packets and channels support different functionalities within the system. Avatar control traffic determines the movement of avatars, Prim traffic communicates the structure of objects and texture traffic communicates images which define how the surfaces of objects should look.

Each of these types of traffic has its own footprint in terms of bandwidth, packet size and inter-arrival time. Avatar control traffic is low bandwidth and is made up of small packets, which is similar to First Person Shooter computer game traffic. Prims and Textures contain parts of the environment that are being downloaded. They are made up of bigger packets and together account for most of the bandwidth used by SL. Each of these types of traffic has its own Quality of Service requirements.

The timely delivery of avatar control traffic is critical to the user experience. Loss and latency result in unresponsive interactions for the avatar and inaccurate representation of its position. Consequently, we can confidently assert that SL would benefit from protecting the bandwidth of avatar traffic, through allocating it a dedicated channel. Furthermore, expedited forwarding and protection against loss could be provided by a *DiffServ* architecture. Prims and Textures both enable the environment an avatar moves through to be represented to the user. Prims provide the structure and Textures the surface. There is more leeway in the timeliness with which these can be delivered. A late arriving Texture will not have the same impact as the late arrival of an instruction to turn left. Hence in contrast to game traffic SL can and does remain usable for a range of different throughput values.

The characteristics of metaverse traffic mean that, within limits, when network resource is scarce they can make do, yet if more is available it can usefully be utilised. This suggests that adaptive traffic management could, and should, be used, which would allow metaverses such as SL to provide a better user experience.

This work leads us to hold the position that the combination of adaptive congestion control and differential QoS can increase the range of conditions under which metaverses are both usable and fair to competing traffic whilst improving the service that the application receives.

8. REFERENCES

[1] Allman, M., Paxson, V., and Stevens, W. TCP Congestion Control. RFC 2581 (Proposed Standard), Apr. 1999. Updated by RFC 3390.

[2] Antonello, R., Fernandes, S., Moreira, J., Cunha, P., Kamienskia, C., and Sadok, D. Traffic Analysis and Synthetic Models of Second Life. *ACM/Springer Multimedia Systems Journal* (2008).

[3] Balakrishnan, H., and Seshan, S. The Congestion Manager. RFC 3124 (Proposed Standard), June 2001.

[4] Bansal, D., and Balakrishnan, H. Binomial Congestion Control Algorithms. In *IEEE Infocom 2001* (Anchorage, AK, April 2001).

[5] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W. An Architecture for Differentiated Service. RFC 2475 (Informational), Dec. 1998. Updated by RFC 3260.

[6] Chen, K.-T., Huang, P., Huang, C.-Y., and Lei, C.-L. Game traffic analysis: an MMORPG perspective. In *NOSSDAV '05: Proceedings of the international workshop on Network and operating systems support for digital audio and video* (New York, NY, USA, 2005), ACM, pp. 19–24.

[7] Chen, K.-T., Huang, P., and Lei, C.-L. How sensitive are online gamers to network quality? *Commun. ACM 49*, 11 (2006), 34–38.

[8] Claypool, M. The effect of latency on user performance in real-time strategy games. *Computer Networks 49*, 1 (2005), 52–70.

[9] Claypool, M., and Claypool, K. Latency and player actions in online games. *Commun. ACM 49*, 11 (2006), 40–45.

[10] Dijkstra, E. Making a Translator for ALGOL 60. *APIC Bull 7* (1961), 3–11.

[11] Feng, W.-c., Chang, F., Feng, W.-c., and Walpole, J. A traffic characterization of popular on-line games. *IEEE/ACM Trans. Netw. 13*, 3 (2005), 488–500.

[12] Fernandes, S., Antonello, F., Moreira, J., Sadok, D., and Kamienski, C. Traffic Analysis Beyond This World: The Case of Second Life. In *7th International workshop on Network and Operating Systems Support for Digital Audio & Video (NOSSDAV) 2007* (2007).

[13] Floyd, S., Handley, M., and Kohler, E. Problem Statement for the Datagram Congestion Control Protocol (DCCP). RFC 4336 (Informational), Mar. 2006.

[14] Getchell, K., Miller, A., Allison, C., and Sweetman., R. *Exploring the Second Life of a Byzantine Basilica*, vol. Serious Games on the Move. SpringerWien, 2009, pp. 165–180.

[15] Handley, M., Floyd, S., Padhye, J., and Widmer, J. TCP Friendly Rate Control (TFRC): Protocol Specification. RFC 3448 (Proposed Standard), Jan. 2003.

[16] Hemminger, S. Network emulation with NetEm. In *Linux Conf Au* (2005).

[17] Henderson, T. Latency and user behaviour on a multiplayer game server. In *NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication* (London, UK, 2001), Springer-Verlag, pp. 1–13.

[18] Henderson, T., and Bhatti, S. Networked games: a QoS-sensitive application for QoS-insensitive users? In *RIPQoS '03: Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS* (New York, NY, USA, 2003), ACM, pp. 141–147.

[19] Jacobson, V. Congestion Avoidance and Control. *ACM SIGCOMM Comput. Commun. Rev. 18*, 4 (1988), 314–329.

[20] Jain, R., Chiu, D. M., and Hawe, W. A quantitative measure of fairness and discrimination for resource allocation in shared systems. Tech. Rep. DEC TR-301, Digital Equipment Corporation, Littleton, MA, 1984.

[21] Kinicki, J., and Claypool, M. Traffic Analysis of Avatars in Second Life. In *Proceedings of the 18th ACM International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)* (Braunschweig, Germany, May 2008).

[22] Kumar, S., Chhugani, J., Kim, C., Kim, D., Nguyen, A., Dubey, P., Bienia, C., and Kim, Y. Second life and the new generation of virtual worlds. *Computer 41*, 9 (2008), 46–53.

[23] La, C.-A., and Michiardi., P. Characterizing user mobility in second life. In *Proceedings of WOSN* (Seattle, USA, August 2008).

[24] Liang, H., Motani, M., and Ooi, W. T. Textures in second life: Measurement and analysis. In *ICPADS '08: Proceedings of the 2008 14th IEEE International Conference on Parallel and Distributed Systems* (Washington, DC, USA, 2008), IEEE Computer Society, pp. 823–828.

[25] Mathis, M., Semke, J., Mahdavi, J., and Ott, T. The macroscopic behavior of the TCP Congestion Avoidance algorithm. *SIGCOMM Comput. Commun. Rev. 27*, 3 (1997), 67–82.

[26] Mogul, J. Efficient use of workstations for passive monitoring of local area networks. In *SIGCOMM '90: Proceedings of the ACM symposium on Communications architectures & protocols* (New York, NY, USA, 1990), ACM Press, pp. 253–263.

[27] Nichols, J., and Claypool, M. The effects of latency on online madden NFL football. In *NOSSDAV '04: Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video* (New York, NY, USA, 2004), ACM, pp. 146–151.

[28] Oliveira, M., and Henderson, T. What online gamers really think of the Internet? In *NetGames '03: Proceedings of the 2nd workshop on Network and system support for games* (New York, NY, USA, 2003), ACM, pp. 185–193.

[29] Oliver, I., Miller, A., and Allison, C. A User Space Implementation of DCCP. In *PGNET* (2008).

[30] Padhye, J., Firoiu, V., Towsley, D., and Kurose, J. Modeling TCP Throughput: A Simple Model and its Empirical Validation. Tech. rep., Amherst, MA, USA, 1998.

[31] Perera, I., Allison, C., Nicoll, J. R., and Sturgeon, T. Towards Successful 3D Virtual Learning – A Case Study on Teaching Human Computer Interaction. In *4th International Conference for Internet Technology and Secured Transactions* (2009), IEEE Press.

[32] Ramakrishnan, K. K., and Jain, R. A binary feedback scheme for congestion avoidance in computer networks. *ACM Trans. Comput. Syst. 8*, 2 (1988), 158–181.

[33] Sheldon, N., Girard, E., Borg, S., Claypool, M., and Agu, E. The effect of latency on user performance in Warcraft III. In *NetGames '03: Proceedings of the 2nd workshop on Network and system support for games* (New York, NY, USA, 2003), ACM, pp. 3–14.

[34] Sturgeon, T., Allison, C., and Miller, A. 802.11 wireless experiments in a virtual world. *ACM SIGCSE Bull. 41*, 3 (2009), 85–89.

[35] Suznjevic, M., Matijasevic, M., and Dobrijevic, O. Action specific Massive Multiplayer Online Role Playing Games traffic analysis: case study of World of Warcraft. In *NetGames '08: Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games* (New York, NY, USA, 2008), ACM, pp. 106–107.

[36] Varvello, M., Picconi, F., Diot, C., and Biersack, E. Is There Life in Second Life? In *Proceedings of ACM CoNEXT 2008* (December 2008).