Multimed Tools Appl (2009) 45:163–190
DOI 10.1007/s11042-009-0304-x

Avatar mobility in user-created networked virtual worlds: measurements, analysis, and implications

Huiguang Liang · Ransi Nilaksha De Silva · Wei Tsang Ooi · Mehul Motani

Published online: 21 May 2009

Abstract We collected mobility traces of avatars spanning multiple regions in Second Life, a popular user-created virtual world. We analyzed the traces to characterize the dynamics of the avatars' mobility and behavior, both temporally and spatially. We discuss the implications of our findings on the design of peer-to-peer architecture, interest management, mobility modeling of avatars, server load balancing and zone partitioning, caching, and prefetching for user-created virtual worlds.

Keywords Networked virtual environment (NVE) · Mobility traces user behavior · Peer-to-peer · Caching · Prefetching · Second Life

1 Introduction

The possibility of multiple users communicating and interacting with each other in a networked virtual environment (NVE) over the Internet has excited many

H. Liang (✉)
Institute for Infocomm Research,
1 Fusionopolis Way, # 21-01 Connexis (South Tower), Singapore, Singapore 138632
e-mail: liang@alumni.nus.edu.sg

R. N. De Silva · W. T. Ooi
Department of Computer Science, National University of Singapore,
10 Kent Ridge Crescent, Singapore, Singapore 119260

R. N. De Silva
e-mail: ransidesilva@nus.edu.sg

W. T. Ooi
e-mail: ooiwt@comp.nus.edu.sg

M. Motani
Department of Electrical and Computer Engineering, National University of Singapore,
10 Kent Ridge Crescent, Singapore, Singapore 119260
e-mail: motani@nus.edu.sg

researchers in the past 20 years. Scaling an NVE to many users, while maintaining interactivity requirements, however, remains a difficult challenge. Much research effort has gone into reducing communication overhead, maintaining state consistency, and managing server resources.

These research efforts were previously handicapped by a lack of deployed, open, large-scale NVE systems, on which the researchers could evaluate the effectiveness of their proposed solutions. In particular, the effectiveness of many of these solutions depends heavily on the avatar behavior, movements, and interactions within the NVE. Without such data, previous research mainly based their evaluations either on simulations with a simple model of avatar behavior (such as random way-point mobility model), or on collected traces from small-scale games and NVEs.

In recent years, however, the increase in connection bandwidth to home and availability of powerful graphics capability in commodity PCs have led to the development of several NVEs targeted at the Internet masses. Among the notable NVEs are Second Life, There, Active Worlds, and HiPiHi, all of which can be categorized as *user-created networked virtual worlds*. In contrast to massively multi-player online games (MMOGs), the other class of currently-popular NVEs, user-created virtual worlds are not task-oriented (e.g., no quest to complete) and are driven by content created by its users, instead of game providers.

Second Life is currently the most popular user-created networked virtual world available, with about 70,000 peak concurrent users spending 100 million man-hours in the third quarter of 2008 alone.[1] Furthermore, the Second Life client is open source, providing opportunities for reverse-engineering of the protocols on which Second Life runs. We believe that the availability of such large-scale, open virtual worlds provides exciting opportunities for researchers to evaluate their solutions using a large number of real traces under realistic scenarios. This belief drives our work in this paper.

This paper presents our effort in collecting and analyzing avatar traces from Second Life. By using a custom Second Life client, we collected the identities, actions, positions, and viewing directions of 122,147 avatars over a 1-year period at recurrent days. The data span 33 regions on Second Life, giving a total of 102 million records. In this paper, we chose to present results on selected days from three of the most populous regions with different characteristics, as well as one region with medium population for contrast.

We analyze our traces to study the temporal and spatial dynamics of avatars. For temporal dynamics, we look short-term dynamics (in the order of hours and days), and long-term dynamics (in the order of months). For short-term dynamics, we ask the following questions: (i) How does avatar population vary over time? (ii) How long does an avatar stay in a region? (iii) How often does an avatar return to the same region? (iv) If an avatar does return, how much time has passed before the avatar returns? and (v) How do avatar arrival and departure rates vary over time? For long-term dynamics, we study how one of the region changes, if at all, over the course of the year (between March 2008 and March 2009).

The traces also provide a rich amount of information on the spatial distribution of avatars in a region and their movement patterns. We divide the region into cells, and

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

ask: Given a cell, (i) how many times does an avatar visit the cell? (ii) How long does an avatar stay in the cell? (iii) How fast does an avatar move in the cell?

We also analyze the contact patterns among the avatars. In particular, we are interested in characterizing (i) the number of avatars within an avatar's area-of-interest, (ii) the duration two avatars stay within each other's area-of-interest, and (iii) how dynamic is the set of avatars within an avatar's area-of-interest.

Our traces and analysis are useful in many ways. First, the traces, which we share with the research community, capture the actual movements and activities of a large number of avatars. It can be used in trace-based simulations of user-created networked virtual worlds, allowing new and existing designs and algorithms to be evaluated under realistic conditions.

Second, the traces can be used to derive and verify new mobility models for user-created networked virtual worlds. Most existing research assumes mobility models based on random walk [23, 32], random waypoint or their variations [1, 2, 6, 9]. Research, however, has shown that the evaluation results based on these simple models are significantly different from those based on actual traces. A new avatar mobility model for user-created networked virtual worlds is therefore needed. Our traces and findings serve as a crucial first step towards that goal. The traces can also help in deriving and verifying appropriate models for the spatial distribution of avatars in the virtual world, where past research has assumed that avatars are uniformly distributed [18] and distributed in clusters [22].

Finally, our analysis provides insights into how avatars behave and move in an user-created networked virtual world such as Second Life. This knowledge can lead to the design of new and more effective algorithms for user-created networked virtual worlds. For instance, in designing a peer-to-peer architecture for user-created networked virtual worlds, it is important to understand the expected churn rate, identify peers that stay in the system for a long time, and understand if (and how) the avatars move and congregate. In designing load balancing and zone partitioning schemes for virtual world servers, knowing the expected spatial distribution of avatars and their tendency to moves across zones is helpful.

We do not intend to evaluate previous work using our traces, nor propose new mobility and avatar behavioral models in this paper. These are important research directions that we believe need to be explored, but do not fit into the scope of this paper. Instead, in this paper, we discuss our findings from our analysis of the measured traces, and how the findings will affect various aspects of user-created networked virtual world design in general.

The rest of this paper is structured as follows. We present previous work related to ours in Section 2. Section 3 briefly introduces Second Life. We explain how we collect and verify our data in Section 4. Section 5 presents our analysis of the traces. In Section 6, we show that our results are robust despite difficulties in the analysis and data collection process. We discuss the implications of our traces in Section 7. Finally, we conclude in Section 8.

2 Related work

We now describe previous efforts in collecting avatar traces from networked virtual environments and games. Rieche et al. [26] collected a 5-h trace of 400 players from an online game called FreeWar. Boulanger et al. [4] collected a trace of 28 players from a game they developed called Orbius. The focus of their work is not on the trace; rather, the trace is a way to evaluate their proposed algorithms. Rieche et al. use their trace to evaluate a load balancing scheme, while Boulanger et al. use their trace to evaluate different interest management algorithms. Beside traces collected from games, both works use randomly generated movements in their evaluation, and both observe significant differences in their results evaluated using the traces and using generated movements. Their results highlight the importance of having real mobility traces for researchers to evaluate their work.

Tan et al. [29] and Bharambe et al. [3] collected traces from Quake III, a multi-player first person shooting (FPS) game and developed mobility models for players movement. Pittman et al. [25] collected a large trace, comparable in scale to ours, of players movement from World of Warcraft, an MMOG, and analyzed the dynamics of the populations, players arrival/departure rate, session length, player distribution, and player movements. FPS games and MMOGs exhibit different characteristics from user-created networked virtual worlds. Players in fast-action FPS games tend to move around constantly. In MMOGs, players usually engage in quests to gain level and new abilities, and tend to gather for an event (e.g. new monsters to fight) and disperse afterward. Players also tend to move in groups. We observed a different pattern for user-created networked virtual worlds.

Most recently, La and Pietro have independently conducted a similar study on mobility in Second Life [19]. Their study, however, focuses on metrics relevant to mobile communications, such as graph theoretic properties of line-of-sight networks formed by the avatars, travel length and time of avatars, and contact opportunities among avatars. Their goal is to use the mobility traces of avatars to model human mobility for applications related to wireless and delay-tolerant networks. On the other hand, we focus on metrics that are of interest to systems design of user-created networked virtual worlds.

Independently, Varello et. al [31] crawled a large number of regions in Second Life during April 2008 and reported an analysis of region popularity, dynamics of objects, and, similar to our work, avatar behavior from popular regions. Our work complement each other—we conducted more detailed analysis on session behavior, mobility, contact patterns, short-term temporal variations, and longer-term temporal variations; they reported on the clustering of avatars, modes of movement (flying, running, walking), and social relationship among avatars. Interestingly, both work found similar stay time distributions, despite collecting traces on different regions in Second Life.

3 Second Life

Before we describe how we collected our traces, we briefly introduce Second Life in this section. Second Life is an user-created networked virtual world launched by Linden Lab in 2003. It is a so-called *metaverse*, where users participate in creating the virtual world by constructing buildings and authoring objects. Furthermore, users control avatars that can interact with each other, socialize, and trade user-created objects. Because Second Life supports socialization of avatars, it is also known as a *social virtual world* in the literature [31].

Unlike popular MMOGs such as World of Warcraft, the virtual world in Second Life is highly dynamic—users can create objects, place them into the virtual world, and write scripts to program the behavior of the objects. In comparison, the world or game maps in MMOGs are mainly static and are built by game publishers. As such, it is possible to distribute the data describing the game world on DVDs. In Second Life, however, only a viewer program is distributed to the users. Data pertaining to the virtual world, such as terrain, objects, behavioral scripts, and textures, are downloaded on demand as the user explores and interacts with the virtual world.

The virtual world in Second Life is made up of *regions*. Each region is a 256m × 256m piece of land, managed by a Second Life server process known as simulator that maintains the states of all avatars and objects within the region. The virtual world in Second Life is not seamless—a user cannot walk seamlessly between regions, but rather has to *teleport* from region to region. As the user teleports, the state of the avatar is transferred to the destination's simulator (which usually completes in an order of seconds), while interactivity between the user and avatar ceases until completion. Each region has a teleportation point called the *landing point*, where all arriving avatars will appear at.

An activity unique to Second Life is *camping*, where avatars can earn free virtual money by engaging in certain activities (e.g., get paid by the hour to sit on a chair). Region owners typically use camping to boost the popularity of the regions.

Within a region, a user can walk, run, or fly, or teleport from one place to another. In this paper, we are interested in collecting the mobility traces of the avatars as they move around within a region. We describe how we collect the traces next.

4 Data collection

To collect the traces of avatars, we developed a client for Second Life based on an open source library called libsecondlife.[2] Our client visits selected regions in Second Life using a bot, and by parsing update packets from the servers, we can obtain information about other avatars in the regions. We log information about all detected avatars at 10 s intervals.

Second Life uses an architecture that stores the states of all avatars in a centralized server. The server pushes information about other avatars and objects in the region to the clients, depending on the client's avatar position. How the server decides which information to push is unfortunately proprietary and unknown to us. It was therefore not clear to us whether our bot is able to track the positions of every avatar on the region. For instance, if the server uses interest management techniques, then only positions of other avatars within the area-of-interest (AoI) of our bot will be updated, while those outside are culled.

We inserted several avatars into Second Life to determine how interest management is done. We observe that culling indeed does occur for avatars outside of the AoI, but, once the bot starts detecting an avatar, updates about that avatar will no longer be culled, even if the avatar moves outside the bot's AoI. To track the movement of as many avatars as possible, we therefore place our bot at the landing

[2] www.libsecondlife.org

point. As such, all incoming avatars will be immediately detected by the bot and subsequently our bot will receive update of these avatars regardless of where they are in the region. The existing avatars that are already in the region when we started our bot, however, may not be tracked. Fortunately, our measurements show that 95% of the avatars stay less than an hour in the region (see Section 5.1). Thus, after the first couple of hours, our bot can track nearly all of the avatars in the region.

Our bot does not track movements of avatars between regions. Second Life does not provide information about the destination of an avatar when it leaves a region. In fact, we cannot differentiate between an avatar logging out and teleporting to another region.

4.1 Difficulties encountered

We now briefly explain several issues we ran into during our data collection process, which lead to imperfect traces.

One source of imperfection in our traces is related to avatars that are sitting on objects. Second Life reports the locations of these avatars relative to those objects. To recover the position of a sitting avatar, we need the position of the object being sat on. We observe that information on objects are sometimes culled if they fall outside the bot's AoI. In this case, we are unable to compute the position of the avatars sitting on these objects. Such records are removed from the trace, but only when we compute metrics that require the positions of the avatar. About 8% of our records have unknown avatar positions.

Occasionally, our bots are kicked out by the region owner, as the owner may not take kindly to bots. Each region allows a limited number of avatars inside at any one time, due to server's resource constraints. An inactive bot occupies a valuable slot that could have been filled by an actively contributing avatar. On a few occasions, region owners went as far as to ban our bots from returning, creating temporal breaks within the data set.

Temporal breaks also happen when our client crashes due to insufficient memory. The libsecondlife library maintains a staggering amount of states about each object and avatar in a region (at least 1 GB in densely populated regions). We have a script to automatically restart the client in such situations, but the bot may not be able to log back to the same region because the region becomes full during the crash, again creating temporal breaks in our traces. We address the impact of these breaks in Section 6.

4.2 Verification

Positional prediction techniques, such as dead reckoning, are commonly used in NVEs to reduce the update frequency between the server and clients at the cost of reduced consistency in avatar position. We are concerned with how consistent the reported positions of the avatars are and if any predictions are done at the client.

An experiment was conducted to verify the consistency of the reported position. We placed seven bots on a region called Freebies, with four static bots at each corner of the region, one static bot at the center of the region, and two bots that walk around the region following a random walk model. We log the positions of other avatars in the region as seen by each bot. For each avatar at each time instance, seven records of the positions are obtained. We compute the standard deviations of x and y positions of the avatars seen by these bots. We found that only 1.14% of the reported x positions and 1.39% of the reported y positions have a standard deviations of more than 10 m, while 9.00% and 10.15% of the records have standard deviations of more than 1 m for x and y positions respectively. There are instances where the two bots detected the same avatar with reported positions more than 100 m apart—this happens when the avatar teleports within the region. Delay in receiving updates from the server causes these discrepancies.

Another concern we have is whether our bot consistently observe the same set of avatars. To verify this, we place two bots close together at the landing point on Freebies. Each bot recorded the number of avatars it knows every 10 s for 30 min. We found only small differences in the number of avatars reported by the two bots. Each bot detected an average of 71.22 avatars, and the mean difference between the reported number of avatars at each scan is 1.16 (1.63%). This difference is caused by one to two avatars (who entered in the region before our bots) moving into the AoI of one bot but not the other.

The verification experiments above indicate that our collected data contain some errors, caused by state synchronization delay and interest management techniques used by the server. With no access to the server states, we can only collect the traces at a client, where these errors are unavoidable. Fortunately, the errors are found to be reasonably small and we believe they will not affect the general conclusions obtained from the trace analysis.

4.3 Traces

While we collected data from 33 regions at recurrent dates over a year (March 2008 to March 2009) we focus our analysis on four regions in this paper, namely *Isis*, *Ross*, *Freebies*, and *The Pharm* and on selected days. These traces are summarized in Table 1.

Three of the regions, Isis, Freebies, and The Pharm, which we studied are consistently among the most popular regions in Second Life in March 2008. Isis has a mature adult theme. Residents can participate in paraphiliac activities, buy adult-novelty items, and camp. Freebies gives away free objects, clothes, accessories, and other inventory items to any resident. It also features a very small camping area. The Pharm is a region focused only on camping. We chose to analyze these popular regions since they provide insights that are most relevant to the design of large-scale user-created networked virtual worlds. They represent the typical case with a large

Table 1 Summary of traces analyzed

Trace	Region name	Number of avatars	Dates
1	Isis	2,735	28 Mar 2008, Fri
2	Ross	560	11 Mar 2008, Tue
3	Freebies	3,153	11 Mar 2008, Tue
4	The Pharm	1,537	5 Mar 2008, Wed
5	Isis (Long)	8,795	28–31 Mar 2008, Thu to Sun
6	Freebies (Days)	9,934	11 Mar 2008, Tue, 24 Oct 2008, Fri, 25 Dec 2008, Thu, 9 Jan 2009, Fri, 27 Mar 2009, Sat

number of avatars in the region. Under these scenarios, design techniques such as load balancing, zone partitioning, caching, and prefetching would make the most significant impact.

Ross, a moderately popular region, is chosen to contrast the results from popular regions. Ross is an information hub and serves as meeting place for avatars. We did not choose any unpopular regions to study as they usually attract fewer avatars and we found that such traces do not provide useful insights to design of virtual worlds.

Besides popularity and variation in themes, the choice of these four regions is also due to the completeness of their traces. As mentioned in Section 4.1, we encountered temporal breaks in our traces. The set of 1 day traces from these regions are the most complete, with only an average of 3 breaks per day. The average break is 6.5 min long, while the longest is 16.5 min. Out of the traces, only 1.3% of the records are lost.

We have collected traces on the regions over multiple days. In this paper, we first presents our analysis for 1-day traces for each of the four regions (Traces 1–4 in Table 1). We found that analyzing longer traces only leads to negligible differences in the distribution for most of the metrics we are interested in. The justification is exposited in detail in Section 6.4 using a 4-day trace for Isis (Trace 5 in Table 1) The same trace is used to study hour-by-hour variations of certain metrics in Section 5.4.

To show how our particular choice of days affect the analysis, we further analyze the traces from three random days over the course of a year (March 2008 to March 2009) from Freebies (Trace 6 in Table 1). In this analysis, we also include the results from Christmas Day of 2008, a day with low activity, for contrast. The results are shown in Section 5.5.

5 Characterizing avatars

5.1 Session behavior

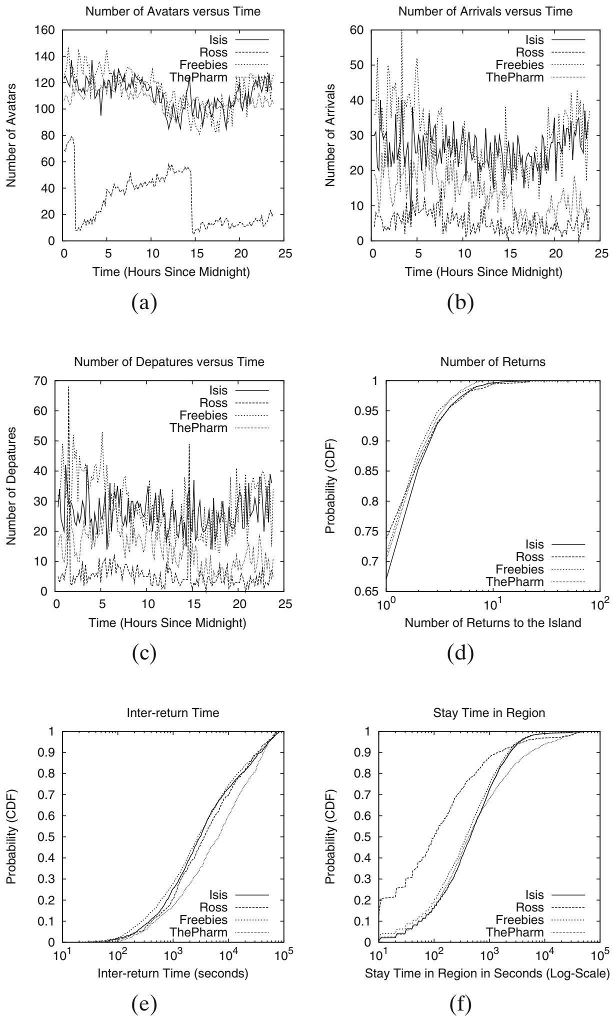
We now report on the session behavior of avatars based on the 1-day traces (Traces 1–4).

Population over time Figure 1a shows how region population changes over time. The three popular regions, Isis, Freebies, and The Pharm exhibits similar patterns in the population over time, observing a slight drop in population from 12 noon to 6 pm.[3] The population from Ross exhibits a different pattern, where population increases steadily but suddenly drops. These drops could be due to server reset, causing all avatars (include our bot) to be logged out.

The diurnal pattern is not as obvious as observed in MMOG [25]. We observe similar patterns when studying our traces from other days. Our 4-day Isis trace, for instance, exhibits similar drop, surprisingly, even on weekends. One possible explanation for the lack of obvious diurnal pattern is that Second Life users span many parts of the world. According to the economy statistics published by Linden Lab, about 40% of the users come from North America. Another 40% come from

[3] All time mentioned in this paper are based on local (Singapore) time (UTC +8).

Fig. 1 Session behavior (**a**–**f**)



Europe. The rest are from Asia, South America, and Oceania. Users log onto Second Life at all times of the day. The slight drop during 12 noon to 6 pm local time corresponds to wee hours in North America, suggesting that many avatars that visited the regions we analyzed are from North America.

Arrivals and departures Figure 1b and c show the number of arrivals and departures over time. These figures show that the population on the regions is highly dynamic, as expected from the maximum number of avatars in a region (Fig. 1a) and number of unique avatars observed (Table 1). We can see a high churn rate for Freebies, as avatars tend to drop into the region, picks up free items, and leaves. The churn rate is especially high from 12 midnight to 6 am (up to 60 churns per hour). The churn rates for The Pharm and Ross are lower, due to the camping activities and low popularity of the region, respectively.

Returning to the same region We are interested in how many times an avatar revisits a region. Surprisingly, even within a day, we observed multiple visits by the same avatar. The maximum number of revisits observed is 55 (for Freebies). We speculate that these avatars might be hopping from region to region. About 25–35% of the avatars revisited the same region within a day. The CDF for this metric is shown in Fig. 1d. Figure 1e shows the CDF for the time that passed between an avatar leaving the region and returning to the region. We call this metric *inter-return time*. The median inter-return time for the three regions on the day observed ranges from 45 min to an hour. 90% of the inter-return time observed is less than 10 h.

Stay time Figure 1f shows the cumulative distribution of how long an avatar stays in a region. We call this the *stay time* of an avatar. We note that this does not correspond to the time an avatar stays in Second Life, since the avatar could have just teleported to another region rather than leaving Second Life. We compute the stay time by logging the time between the arrival and departure of an avatar, excluding all avatars that are already in the region at the beginning of our trace. The distribution of stay time is highly skewed, close to a power law distribution. An obvious observation is that the stay time at Ross is lower (median of 92 s) than that of Isis and Freebies (median of 448 s and 373 s). The distribution of stay time at The Pharm is skewed towards higher values (despite a median of only 427 s) than Isis and Freebies, since avatars have incentives to stay in the region. The periodic reset on Ross could explain the shorter stay time.

5.2 Mobility

We now characterize how avatars move: where they visit, how long do they pause, how fast do they move, and whether they stay in groups. We quantize the regions into 256×256 equal size *cells*, and compute a set of metrics for each cell. Figures 2, 3, 4, and 5 show the choropleth maps of the regions for various metrics. Figure 6 plots the CDF of the same metrics (x-axis in log-scale). Figure 6b and c shows the distribution over all cells visited.

We ignore the z-coordinates of avatars in our analysis. Each avatar position gives the coordinate of the avatar in a 3D space. We observe, however, that most of the time the avatars stay on the ground (see Section 6.1). We therefore focus only on the x- and y-coordinates. As a result, an avatar that hovers in the air is considered to be at the same position as another avatar standing on the ground if they have the same x- and y-coordinates.

Number of visits An indication of cell popularity is how many times the avatars visit a cell. We count the number of times an avatar enters a cell. If the same avatar enters and leaves the cell multiple times, it is counted as multiple visits. If an avatar logs out and logs in again, Second Life will place the avatar at the previous position when it logs out. We do not count this entry as a new visit.

Figure 2a to d show the choropleth map of the number of visits to a cell in log-scale. The number of visits to cells is highly skewed. There are many cells that are not visited by any avatars in our traces, and a small number of cells are visited many times. Figure 6a shows the CDF for number of visits for all four regions. Freebies has the most visited cells (52% are visited at least once), as its activities spread to all

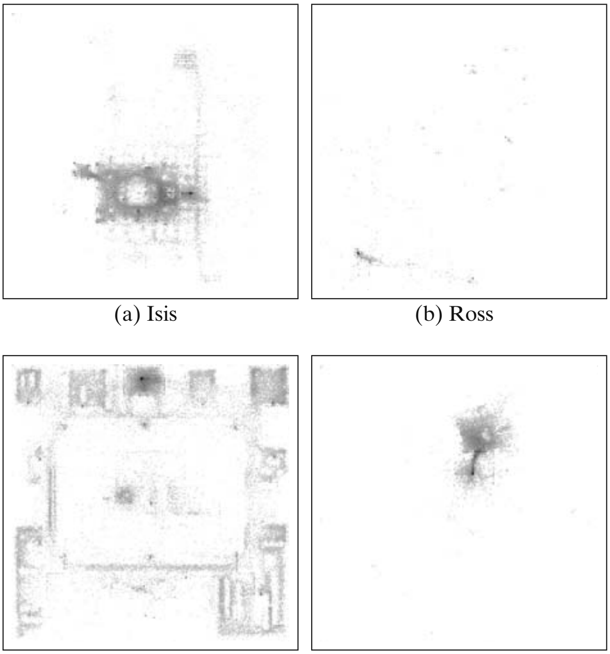


Fig. 2 Log-scale choropleth map for number of visits. Darker colors mean higher value (**a**–**d**)

(a) Isis (b) Ross

(c) Freebies (d) The Pharm

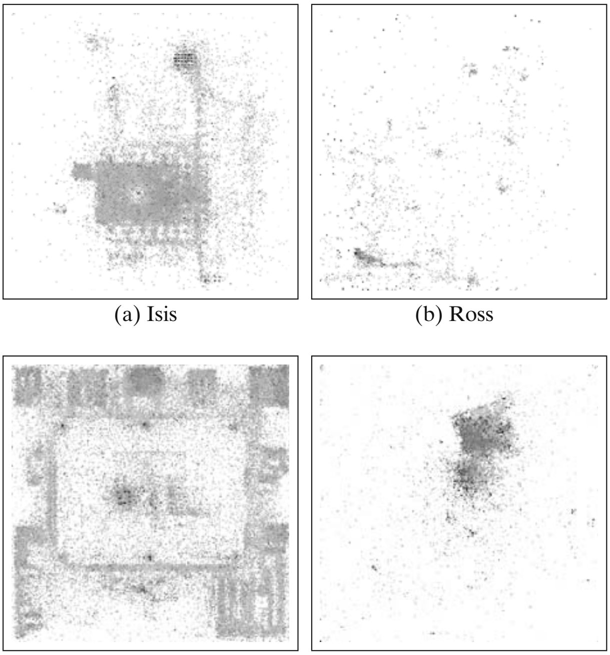
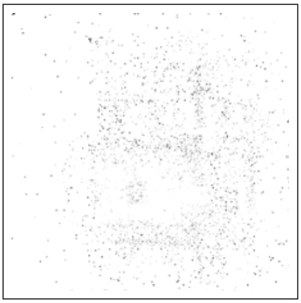


Fig. 3 Log-scale choropleth map for average pause time. Darker colors mean higher value (**a**–**d**)

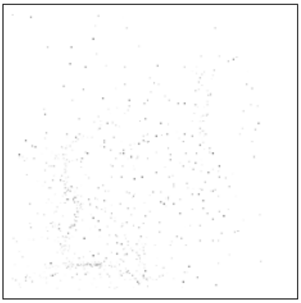
(a) Isis (b) Ross

(c) Freebies (d) The Pharm

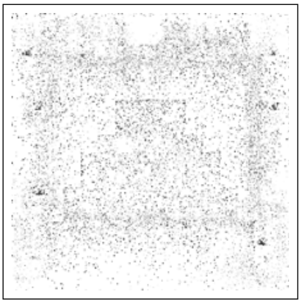
Fig. 4 Log-scale choropleth map for average speed. Darker colors mean higher value (**a**–**d**)



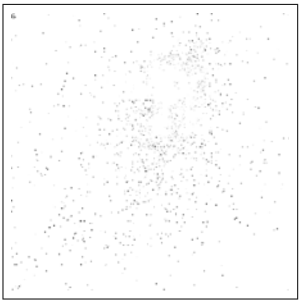
(a) Isis



(b) Ross

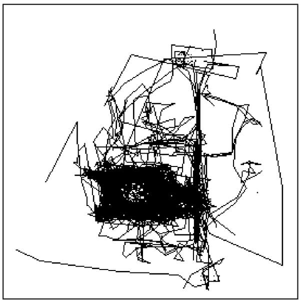


(c) Freebies

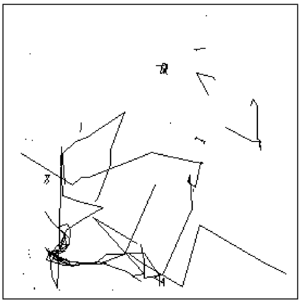


(d) The Pharm

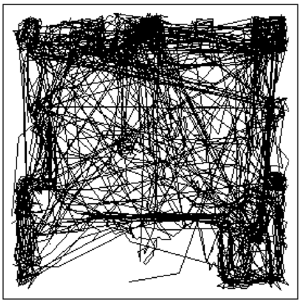
Fig. 5 Movement trails from 6 am to 7 am (**a**–**d**)



(a) Isis



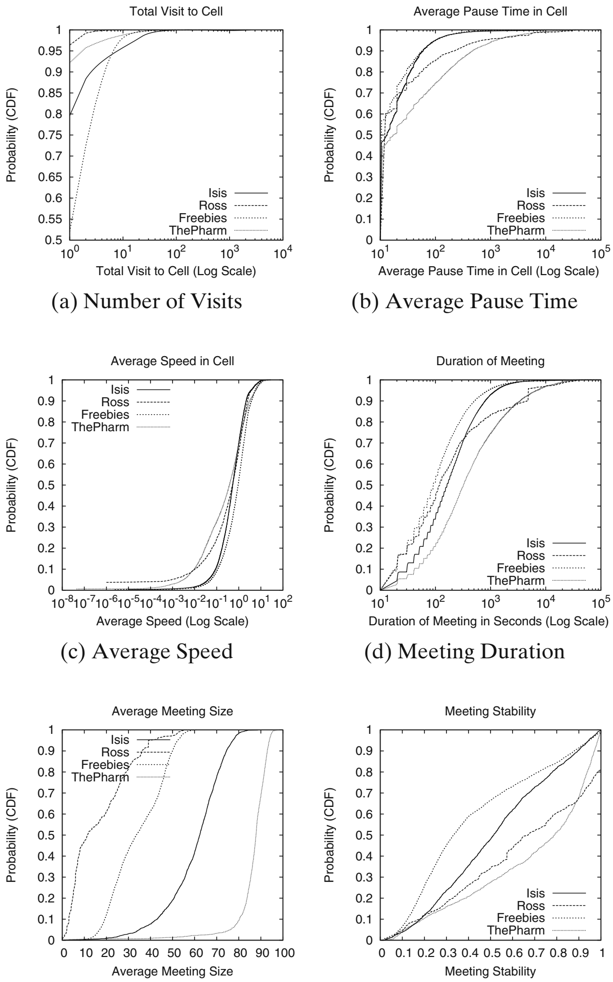
(b) Ross



(c) Freebies

(d) The Pharm

Fig. 6 Distributions of mobility-related metrics (**a**–**f**)



(a) Number of Visits (b) Average Pause Time

(c) Average Speed (d) Meeting Duration

(e) Meeting Size (f) Meeting Stability

corners of the region (Fig. 2c). The regions have 2 (Ross) to 17 (Isis) cells that are visited more than 100 times, with the most visited cell having been visited 2487 times (in Isis). This cell corresponds to the landing point on Isis, which can be seen as the dark black spot in Fig. 2a. Other regions show similar spots at their landing points.

Average pause time Another indication of the popularity of a cell is the duration an avatar stays in a cell (until it moves to another cell, teleports to another region, or logs out). The total pause time of a cell c is the sum of the pause time of all avatars that have ever been to c. Note that since we log the movement of avatars every 10 s, the pause time has a minimum resolution of 10 s. The average pause time of a cell is thus total pause time divided by number of visits. This metric indicates the *stickiness* of a cell.

Figure 3a to d show the choropleth map of the average pause time of a cell in log-scale. Comparing these maps to Fig. 2a to d shows there exist cells with high average pause time that are not visited often (for instance, the group of cells at the top right quadrant of the map in Fig. 3a). The distribution of average pause time is again highly skewed (Fig. 6b). On Isis and Freebies, avatars pause for less than 100 s in 95% of the visited cells but 11 and 5 avatars pause more than 3 h on Isis and Freebies respectively. In the camping region, The Pharm, 40 avatars paused more than 3 h. The longest average pause time observed is just over 14 h in The Pharm.

Average speed in a cell For each pair of consecutive recorded avatar positions, we note the time between the records t and the distance traveled d. The speed of an avatar movement in cell c is then d/t, where c is the cell the avatar is in after moving. We average the speed over all avatar movements in that cell to get the average speed in a cell.

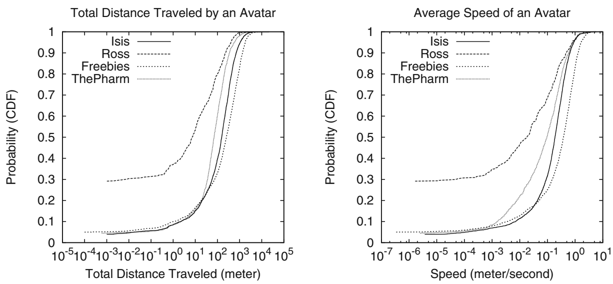
The map for average speed is shown in Fig. 4a to d, with the corresponding CDF shown in Fig. 6c. Comparing these maps to Fig. 2a to d shows high average speed outside of the frequently visited cells, confirming the intuition that avatars move quickly (either run or fly in Second Life) in non-interesting regions but move normally (walk) within interesting regions. This is confirmed in Fig. 5a to d, which show the trail of the avatars in a 1-h sub-trace (6am) of the four 1-day traces in Table 1. A long and straight line means the avatar is moving at a fast speed (including intra-region teleport).

Figure 7a and b show the distribution of total distance traveled by an avatar (sum over all visits to the same region by the same avatar) and average speed (total distance traveled over total time in region) of an avatar. The figures indicate something interesting: about 30% of avatars have a speed of zero and never moved in Ross and 10% of avatars move less than a meter in the other regions. Likely these users log in and leave the session running, exploiting camping facilities to earn virtual money (Note that Second Life leaves the avatar at their previous location when the user logs in).

5.3 Contact patterns

To characterize the spatial relationship among avatars, we look at *meetings* among the avatars. Two avatars meet if their distance in a region is within a certain

Fig. 7 Distributions of distance traveled and average speed per-avatar (**a**, **b**)



(a) Total Distance Traveled (b) Average Speed threshold. We use 64m as our threshold since this is the default AoI distance in Second Life.

Meeting durations Figure 6d shows the distribution of meeting duration. Meeting durations are long—over 50% of the durations are over 82 s (for Freebies) and 303 s (for The Pharm). On The Pharm, 9% of the meetings are over an hour.

Average meeting size A closely related measure we compute is average meeting size over time. For every 10 s, we computed the *meeting size*—the number of avatars within their AoI. We then compute, for each avatar, the average meeting size over time. We found that the meeting size is generally large in the regions we studied. On The Pharm, the average meeting size for avatars is above 40 for 99% of the avatars. Even on Ross, a medium popularity region, the median meeting size is 11.4.

Meeting stability To see how the avatars in a meeting change over time, for each avatar *a*, we take the ratio of average meeting size over the number of unique avatars ever met by *a* in the region. We call this ratio *meeting stability*. If an avatar's stability is 1, then the avatar always meets with the same set of avatars while she is in the region. The distribution of this ratio is shown in Fig. 6f. Avatars in The Pharm and Ross have a high meeting stability, with 50% or more having a stability of 0.67 and 0.79 respectively. 19% of avatars have a stability of 1.0 in Ross. Avatars in Freebies show much more dynamic behavior, with a median stability of only 0.33.

5.4 Short-term temporal variation

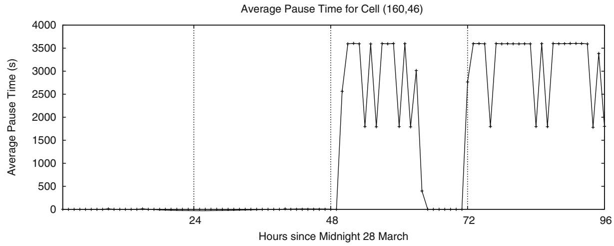
To see if the general observations we made above change over time, we analyze a 4-day trace from Isis, hour-by-hour. We pick the number of visits to a cell as the metric to study, as it is a good indication of whether a cell is popular. For each cell, we sample the number of visits to that cell in each hour, and computed the standard deviation of the samples.

We found very little variation in the number of visits to the cells, hour-by-hour, over the course of 4 days. 99% of the cells have a standard deviation less than 1.05. The largest standard deviation, 20, is observed at the landing point. This is expected since it corresponds to the number of arrivals, which fluctuates as seen in Fig. 1b.

We repeated the same experiment using average pause time in the cells. 98% of the cells have a standard deviation in average pause time of less than 1 min, while the largest standard deviation is about 27 min, occurring in cell (160, 46). While the variation on this cell seems large, a further analysis on this cell reveals what happened.

Figure 8 plots the hour-by-hour variation of average pause time in cell (160, 46) and reveals that the cell is almost empty in the first 2 days, while in the next 2 days (Saturday and Sunday) the average pause time is close to 60 min. Since this cell is located at an outdoor amphitheater, we speculate that, during weekend, the region hosts some events that attracted avatars to stick in the cell. In this case, despite the high standard deviation observed, the hour-by-hour average pause time is still highly predictable.

Fig. 8 Average pause time, computed hourly, in cell (160, 46) from 4-day Isis trace



5.5 Long-term temporal variation

To study the long-term evolution of avatar movement and behaviour, we analyze the traces collected from Freebies on four additional days, including three randomly selected days and Christmas Day of 2008 (Trace 6 in Table 1). We only present the results of Freebies, since The Pharm no longer exists and the content of Isis has changed significantly. In contrast, the design and function of Freebies are unchanged during the year-long period.

We found that, over a one-year period (March 2008 to March 2009), the session behavior, mobility, and contact pattern distributions remain relatively unchanged. Figure 9a shows that the population of Freebies has declined from March 2008 to March 2009, with Christmas Day attracted fewest avatars. Despite this, we found distributions of metrics we studied remain relatively unchanged (Fig. 9b to e show some examples), except for metrics that depends on region population (such as average meeting size, shown in Fig. 9f). Interestingly, despite changing distribution of meeting size, the distribution of meeting durations (Fig. 9d) remains the same.

It is interesting to note that close to 1% of avatars who visited Freebies in March 2008 revisited Freebies on the five selected days. Further, 6%, or 76 avatars, visited Freebies on both 25 December 2008 and 9 January 2009. Such revisits are likely due to avatars returning to shop for new free items.

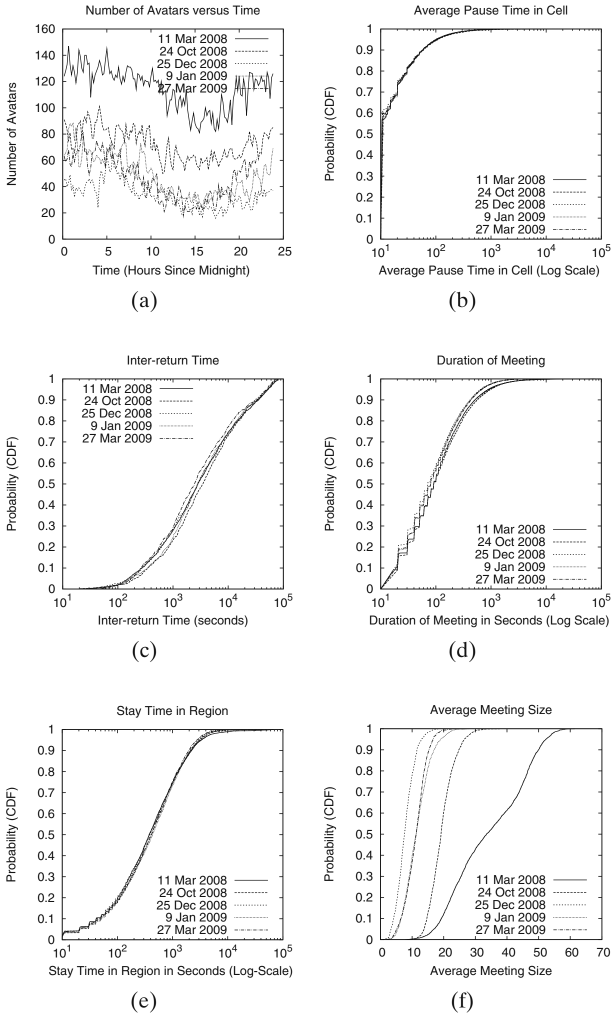
6 Validation

In this section, we validate that our analysis is robust against our assumptions, limitations in data collection, and choice of traces. We start by verifying the assumption that most avatars stay close to the ground in Section 6.1. We then address a concern that temporal breaks in our traces can bias our analysis results in Section 6.2. We show in this section that the analysis is robust against up to seven temporal breaks, at random locations, with up to 30% of data loss, a large amount compared to what we obtained in the traces.

Another concern readers might have is our choice of 1-day trace starting from 12 midnight for analysis. Do the length of the trace and starting time affect the distribution obtained? In Sections 6.3 and 6.4, we validate that the distributions obtained generally do not depend on the starting time and the length of the trace.

Even though we repeated our analysis on all metrics reported in Section 5, we show only three key metrics in this section: average pause time, stay time, meeting

Fig. 9 Distributions of session behavior, mobility and contact patterns for Freebies over March 2008 to March 2009 (**a**–**f**)



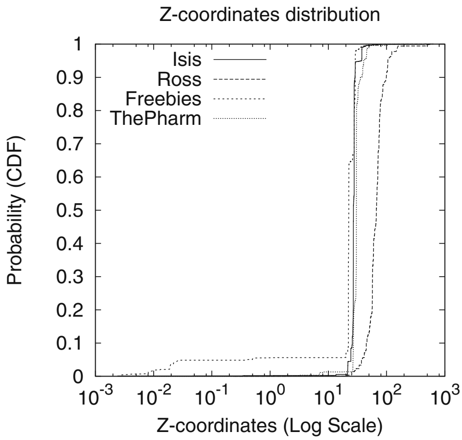
stability, and meeting durations. These metrics are representatives since they have implications on design of user-created networked virtual worlds (see Section 7).

6.1 Ignoring z-coordinates

We have ignored z-coordinates in our analysis (Section 5.2) as we observed that most avatars stay near the ground in Second Life. To validate this, we plot the distribution of z-coordinates of all avatars positions recorded in the log from the four regions in Traces 1 to 4.

Figure 10a shows the CDF of the z-coordinates. We can see that 95% of positions in Freebies, Isis, and The Pharm has z-coordinates smaller than 40m. Ross, which has elevated terrain, has 95% of observed z-coordinate positions lower than 105m. This validates our choice to ignore z-coordinates during our analysis.

Fig. 10 CDF of z-coordinates



(a)

6.2 Effects of temporal breaks

As mentioned in Section 4.1, due to the nature of the data collection process, temporal breaks are unavoidable. We address the concern of how these breaks would affect the analysis of the traces in this section. To evaluate the effects of temporal breaks, we use a simplified error model to introduce temporal breaks into our traces by artificially deleting records from the traces. We vary three parameters: (i) the number of breaks, (ii) the location of breaks and (iii) the amount of data lost due to temporal breaks. The full day trace with no temporal breaks from Freebies on 11 March 2008 (Trace 3) was used as the basis for the evaluation.

Figure 11a to d show the effects of varying the number of temporal breaks from four to seven on the distributions of average pause time, stay time, meeting stability, and meeting contact time. We fix the total number of deleted record at 20% and distribute the deleted records evenly over each break, whose locations are selected randomly. The figures show that the number of breaks has no significant effect on these metrics.

Next, we consider the effects of the amount of data loss on the distribution of our metrics. We fixed the number of breaks to five and varied the amount of data loss from 10% to 30%. Figure 12a to c show no significant effects even when 30% of data is lost, which is far more than the data loss experienced in our traces.

Finally, we evaluate the effects of break locations. We randomly vary the locations of five breaks with a total 20% data loss, approximating our traces. Figure 13a to d show little variation of metric distributions as the location of breaks are varied.

In sum, we validate that the temporal breaks in our traces do not significantly affect the distributions of metrics we analyze in this paper.

6.3 Effects of starting time

In our characterization in Section 5, we use traces that start from midnight. In this section, we show that the choice of trace starting time has no effect on the final distributions of the metrics. To evaluate the effects of starting time on the

Fig. 11 Distributions of metrics with varying number of breaks and constant data loss of 20% (**a**–**d**)

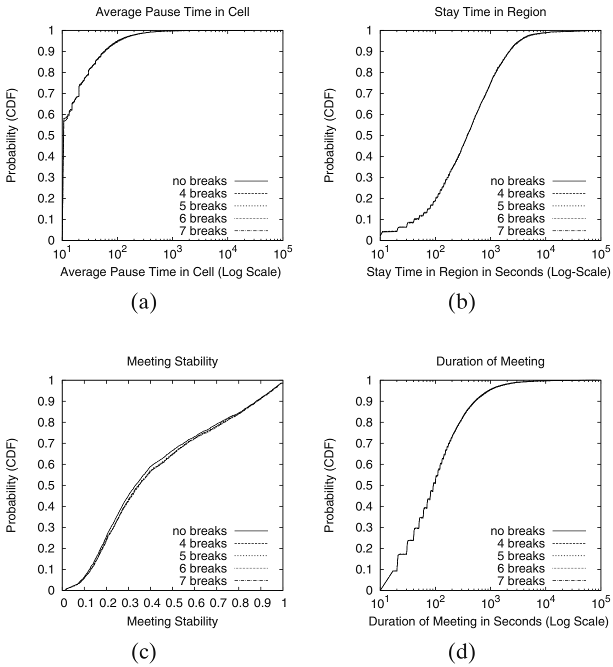
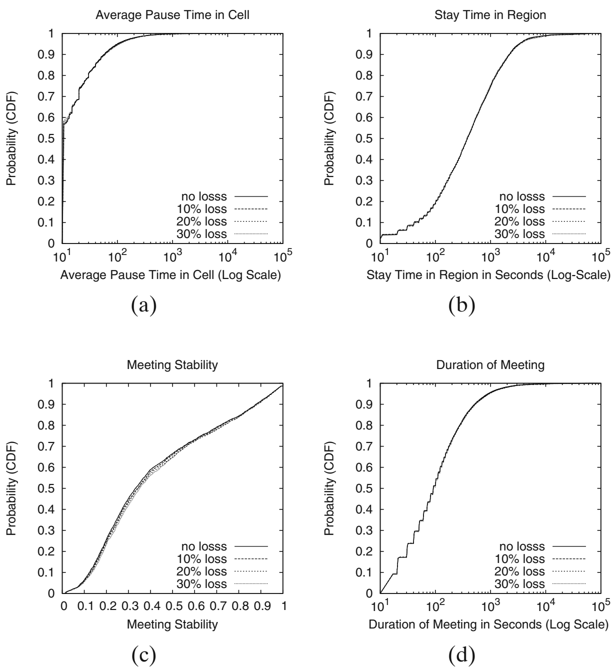


Fig. 12 Distributions of metrics with varying amount of data loss and 5 breaks (**a**–**d**)



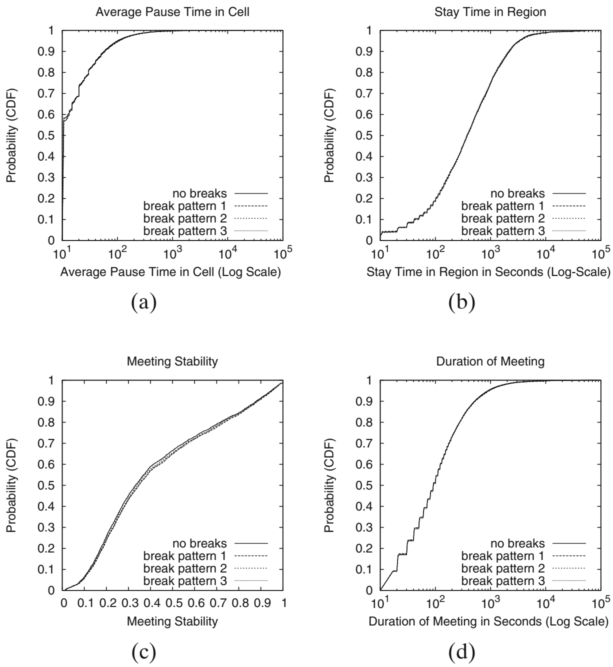


Fig. 13 Distributions of metrics with varying location of breaks with 20% loss and five breaks (**a**–**d**)

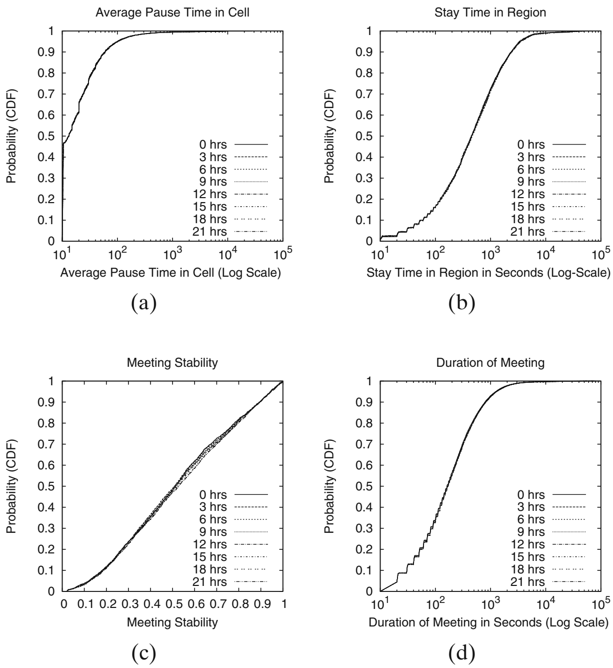
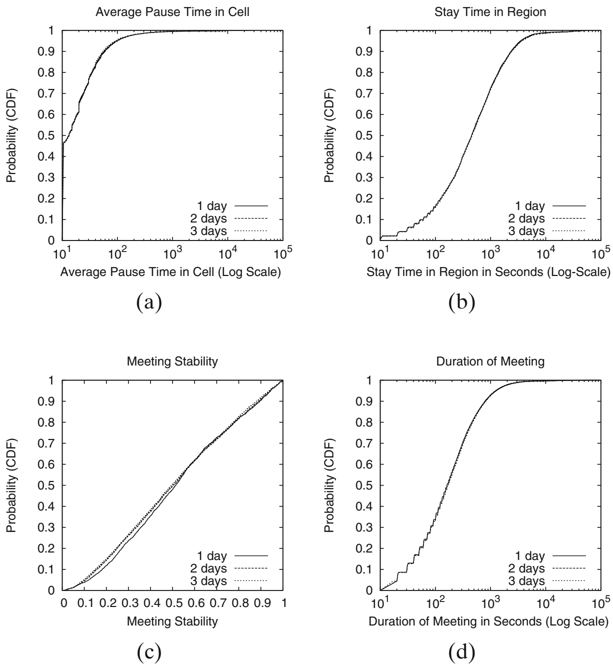


Fig. 14 Distributions of metrics with varying shift in starting times (**a**–**d**)

Fig. 15 Distributions of metrics with varying amount of data lengths (**a**–**d**)



distributions, we use the 4-day trace from Isis (Trace 5 in Table 1). We select 24-h sub-traces with a varying starting time, obtained by shifting the sub-trace "window" by 3 h each time. We obtained seven 24-h traces with starting time ranging from midnight to 9pm of 28 March. Again, we found that the distributions are not affected by the starting time. (Fig. 14a to d).

6.4 Effects of trace length

We used traces of 1-day length in our analysis. To see the effects of trace length on our analysis, we created sub-traces of varying length from the 4-day Isis trace. We fix the starting time and choose trace windows of length varying from 1 day to 3 days. Figure 15a to d show only a slight variation when we consider trace lengths longer than a day.

Certain time-dependent metrics such as inter-return time, however, are inherently effected by trace lengths. A shorter trace will miss future returns to a region, biasing towards lower inter-return time. We should be mindful of this when interpreting such metrics.

7 Implications

We now discuss how our observations from the traces relate to existing research in NVEs. Much of existing work on NVEs proposes general techniques for all class NVEs. As mentioned, user-created networked virtual world such as Second Life exhibit different characteristics and avatar behavior. Our observations, which is based on Second Life, are useful in providing insights and hints on applying the general NVE techniques on this class of NVEs. The implications of our observations are discussed below.

7.1 Peer-to-peer NVEs

Centralized server architectures, such as those employed by Second Life, do not scale well to a large number of players. This challenge of scalability has motivated research into alternative architectures, one of which is the peer-to-peer architecture, where clients communicate directly with other clients through an overlay, without going through the server [13, 14]. A client may also share some responsibility of the server (such as maintaining states) [3, 18].

Our traces show evidence of high churn rate, averaging about one every 2 min on the popular regions, and could have drastic effect on the efficacy of peer-to-peer NVEs. Such high churn rates imply that the system has to continuously configure the peer-to-peer overlay and sufficient redundancy needs to be built-in to prevent loss of information [3]. Our analysis shows that the stay time is highly skewed. This observation supports peer-to-peer NVEs that employ super-nodes to store states and manage other peers [6, 18]. The avatars who stayed for a long time in a region may be good candidates to be super nodes.

Our traces also suggest a novel and interesting way to identify the potential super nodes. We observe that there are cells within the regions that are "sticky"—avatars tend to pause within these cell for much longer time than other cells. Thus, avatars that pause at these sticky cells are more likely to pause for a long time. Further, we observe that such stickiness are highly predictable.

Many peer-to-peer NVE schemes build an overlay by connecting peers within an AoI as neighbors [8, 14, 17]. Our traces support this design in regions with low popularity and low mobility (such as Ross and The Pharm), where the meeting stability is relatively high. For regions such as Isis and Freebies, where the AoI neighbors change frequently, reducing overhead in establishing and tearing down connections among neighbors remains a challenge. Furthermore, we found that meeting size can be large (Fig. 6e) in popular regions. Designers should thus pay attention to the connection overhead and its trade off with latency (e.g., do not connect a node with all its neighbors directly [15]).

Esch et. al proposed a hybrid NVE scheme [12] in which a node engages in peer-to-peer communication with its neighbors if its AoI fringe is completely covered by its neighbors' AoI, since this condition ensures detection of any new neighbor. Such nodes are called cluster nodes and their existence helps reduce communication overhead to servers. Our analysis of meeting size have hinted that AoI overlaps significantly and thus, such cluster nodes are common. Indeed, an analysis by Markus Esch using our traces reveals that cluster nodes constitute 70%–80% of all nodes in The Pharm, Isis, and Freebies, implying that the proposed scheme works well in these popular regions on Second Life.

7.2 Zone partitioning and load balancing

Besides peer-to-peer architectures, another architecture that can improve the scalability of NVEs is server cluster. The game world is divided into zones, each managed by a server. This architecture is similar to what Second Life employs today. The research challenge is to make the game world seamless by making zoning transparent to the users.

A consequence of seamless zone partitioning is that we can dynamically repartition the zones to balance the load on the servers [7, 9, 11, 22]. Two important factors that affect any load balancing scheme are population in a zone, and movement across zones.

Our traces show that, unsurprisingly, the spatial distribution of avatars is far from uniform, especially for Isis and The Pharm. This observation supports the need for more sophisticated partitioning scheme, beyond grid-based partitioning.

We also observed from our traces that avatars tend to move quickly in lowly populated, non-interesting areas within a region. This creates another issue in which such fast-moving avatars have to be handed off from one server to another as they move across the zones, increasing server overheads. The existence of such behavior also points to the importance of load balancing schemes that dampen sensitivity to minor increase in load—to avoid frequent triggers of the load balancing mechanism due to such fast-moving avatars. Our traces show that the popularity of cells does not show much variation over time, suggesting that dynamic load re-balancing of zone would occur only rarely.

7.3 Interest management

Interest management techniques suppress updates from one avatar to another avatar if the two avatars are deemed to be irrelevant to each other. For peer-to-peer architectures, there is no centralized authority that stores the states of all avatars. Researchers have previously proposed novel, distributed schemes to determine the interest between two avatars [23, 28]. The ideas behind these schemes are that, two avatars exchange their locations and each computes a safe zone,[4] based on occlusions and visibility information, where they can remain in without needing to update each other. If one avatar moves out of this safe zone, they exchange their location again and recompute the safe zones.

Our traces mean both good news and bad news for these schemes. On one hand, we found that in interesting cells, typically with many occlusions, avatars move slowly. Thus avatars need to update each other rarely. On the other hand, in non-interesting cells, there are few occlusions and fast movements, thus avatars need to frequent update their locations.

Since determining visibility between avatars can be expensive, another popular technique uses AoI to determine relevancy between two avatars. Two peers update each other as long as their avatars are in the other's AoI. Our traces, however, indicate large average meeting size, (Fig. 6e), suggesting non-negligible updates overhead with this technique.

7.4 Mobility modeling

Our traces are useful in designing and verifying new mobility models for user-created virtual worlds. We observed that on popular regions such as Isis and Freebies, avatars

[4]The term zone here has no relationship to that in Section 7.2

tend on congregate in interesting places and move at a slower speed. They also move faster in non-interesting places, perhaps exploring and looking for interesting things, or moving from one interesting place to another (See Fig. 5a–d). This observation suggests that simple mobility models such as random walk and random way-point [16] are insufficient in modeling mobility of all regions in Second Life.

The movement within high density areas in a region seems to suggest a pathway model—where avatars move along constrained paths (such as corridors or bridges) and visit various rooms [30]. This model, however, does not account for the high speed movements when avatars move in non-interesting cells. Thus, our measurements seem to suggest that a hybrid mobility model that incorporates both random way-point mobility model (for outdoor) and pathway mobility model (for indoor) would be more suitable. Our analysis also suggests the mobility model should incorporate skewed distributions in movement speed and pause time.

7.5 Prefetching

Prefetching is commonly used to reduce the object access latency in NVEs [5, 10, 20, 24]. The key to successful prefetching is predicting accurately which objects are needed, so that bandwidth is not wasted in retrieving objects that will not be eventually used.

Second Life prefetches the data within a circular region of an avatar. We found from our Isis and Freebies traces, however, that avatars only spent 18% of their time rotating around a point. We obtain this by counting the number of instances where an avatar has changed its viewing angle without changing its cell.[5] Thus, Second Life's prefetching algorithm is not ideal. Our traces, however, shows that avatars do tend to rotate around when they first enter a region at the landing point to discover the surrounding (up to 28% of the time). This discovery, coupled with the fact that the area around the landing point contains the most visited cells, suggest that prefetching the data around the landing point when the avatar first teleports in would be more beneficial.

Our extended analysis of the multiple day traces reveals that the popularity of the cells in a region do not change over days, or even over hours. This observation suggests that we can use short-term, historical information about the popularity of a cell as an input to the prefetching algorithm to help with predictions.

7.6 Texture caching

Liang et. al shows that textures constitute up to 88% of all traffic between Second Life server and client [21], pointing to the importance of texture caching. Second Life currently uses a simple local cache replacement algorithm in its client,[6] which simply replaces 90% of the disk cache during startup or when the cache is full. There is clearly room for improvement. Besides employing classic cache replacement algorithms such as LRU, our traces point to several possible directions of improving texture caching in Second Life.

[5] Unfortunately we cannot distinguish between clockwise and anti-clockwise rotation. Otherwise we can compute the average rotation angle, which will provide more insights.

[6] http://wiki.secondlife.com/wiki/Texture_cache

Our traces show that even within a day, there are multiple revisits to the same region by the same avatar. This pattern suggests the possible benefits of *region-aware caching*, which considers access patterns to a region, in addition to access patterns to textures, in the cache replacement algorithm. By considering access pattern to regions, when a user is hopping from region to region, textures from a region he repeatedly visits are not evicted from the cache when they visit other regions.

Further, the skewed distribution of cell popularity points to the benefits of proxy caching. Since few cells are frequently visited by many avatars, caching textures from these cells in a proxy server, shared by multiple users, can lead to a good hit ratio. Like prefetching, we can exploit the short-term historical information to predict cell popularity in the future.

8 Conclusion

This paper presents our effort in collecting and analyzing large amount of traces of avatar mobility in Second Life. We focus on four regions with different characteristics in this paper. It would be interesting to capture such traces on regions which hold transient events (such as parties), where popularity could vary both spatially and temporally.

We computed and analyzed baseline metrics related to system design in NVEs. Our analysis has give us new insights into how avatars in Second Life behave, which could lead to design of new and novel schemes for building virtual worlds, some of which are discussed in this paper. We also provide concrete supporting evidences from a real, popular virtual world that affirm conventional wisdoms such as the failure of random waypoint model and skewness of avatar distributions. One could mine the traces for answers to other interesting questions (such as whether an avatar tends to move towards cells with other avatars). We plan to continue analyzing our traces to reveal interesting patterns in avatar mobility.

We then presented the implications of our observations from the traces on several design aspects of user-created networked virtual worlds, including peer-to-peer architecture, interest management, mobility modeling of avatars, server load balancing and zone partitioning. We also discussed how our findings could impact other aspects of NVE design and suggested several potentially beneficial ideas (e.g., identifying super-nodes, region-aware caching, and popularity-based prefetching). We plan to investigate these ideas in depth and evaluate their effectiveness using our traces.

We have released our traces to the community through our website.[7] Since the release of our traces in July 2008, they have been used in several research projects, including integration into TopGen [27], a simulator for NVE, and evaluation of VON [14].

Acknowledgements We thank Ian Tay and Ming Feng Neo from National University of Singapore and Vikram Srinivasan of Bell Labs Research India for useful discussions throughout this project. We thank Markus Esch from University of Luxembourg for contributing to the analysis of cluster nodes in Section 7.1. This work is partially supported by National University of Singapore Academic Research Fund R-252-000-306-112.

[7] http://nemesys.comp.nus.edu.sg/projects/secondlife

References

1. Backhaus H, Krause S (2007) Voronoi-based adaptive scalable transfer revisited: gain and loss of a voronoi-based peer-to-peer approach for MMOG. In: NetGames '07. Melbourne, pp 49–54
2. Bharambe A, Agrawal M, Seshan S (2004) Mercury: supporting scalable multi-attribute range queries. In: SIGCOMM '04. Portland, pp 353–366
3. Bharambe A, Pang J, Seshan S (2006) Colyseus: a distributed architecture for online multiplayer games. In: NSDI '06. San Jose, pp 12–12
4. Boulanger JS, Kienzle J, Verbrugge C (2006) Comparing interest management algorithms for massively multiplayer games. In: NetGames '06. Singapore, p 6
5. Chan A, Lau RWH, Ng B (2001) A hybrid motion prediction method for caching and prefetching in distributed virtual environments. In: VRST '01. Baniff, pp 135–142
6. Chen A, Muntz RR (2006) Peer clustering: a hybrid approach to distributed virtual environments. In: NetGames '06. Singapore, p 11
7. Chen J, Wu B, Delap M, Knutsson B, Lu H, Amza C (2005) Locality aware dynamic load management for massively multiplayer games. In: PPoPP '05. Chicago, pp 289–300
8. Chen JF, Lin WC, Bai HS, Dai SY (2005) A message interchange protocol based on routing information protocol in a virtual world. In: AINA '05. Taipei, pp 377–384
9. Chertov R, Fahmy S (2006) Optimistic load balancing in a distributed virtual environment. In: NOSSDAV '06. pp 74–79
10. Chim JHP, Green M, Lau RWH, Leong HV, Si A (1998) On caching and prefetching of virtual objects in distributed virtual environments. In: ACM MULTIMEDIA '98. Bristol, pp 171–180
11. De Vleeschauwer B, Van Den Bossche B, Verdickt T, De Turck F, Dhoedt B, Demeester P (2005) Dynamic microcell assignment for massively multiplayer online gaming. In: NetGames '05. Hawthorne, pp 1–7
12. Esch M, Botev J, Schloss H, Scholtes I (2009) P2p-based avatar interaction in massive multiuser virtual environments. In: Proceedings of the 1st international workshop on virtual environments and network-oriented applications (VENOA 2009). Fukuoka
13. Gautier L, Diot C (1998) Design and evaluation of MiMaze, a multi-player game on the internet. In: International conference on multimedia computing and systems, pp 233–236
14. Hu SY, Chen JF, Chen TH (2006) VON: a scalable peer-to-peer network for virtual environments. IEEE Netw 20(4):22–31
15. Jiang JR, Huang YL, Hu SY (2008) Scalable AOI-cast for peer-to-peer networked virtual environments. In: International workshop on cooperative distributed systems (CDS). Beijing
16. Johnson DB, Maltz DA (1996) Dynamic source routing in ad hoc wireless networks. In: Imielinski T, Korth H (eds) Mobile computing, vol 353. Kluwer, Deventer
17. Keller J, Simon G (2003) Solipsis: A massively multi-participant virtual world. In: PDPTA '03. Las Vegas, pp 262–268
18. Knutsson B, Lu H, Xu W, Hopkins B (2004) Peer-to-peer support for massively multiplayer games. In: INFOCOM '04
19. La CA, Pietro M (2008) Characterizing user mobility in Second Life. Technical Report RR-08-212, Institut Eurecom
20. Lau RWH, Chim JHP, Green M, Leong HV, Si A (2001) Object caching and prefetching in distributed virtual walkthrough. Real-Time Syst 21(1/2):143–164
21. Liang H, Motani M, Ooi WT (2008) Textures in Second Life: measurements and analysis. In: P2P-NVE'08. Melbourne
22. Lui JCS, Chan MF (2002) An efficient partitioning algorithm for distributed virtual environment systems. IEEE Trans Parallel Distrib Syst 13(3):193–211
23. Makbily Y, Gotsman C, Bar-Yehuda R (1999) Geometric algorithms for message filtering in decentralized virtual environments. In: I3D '99. Atlanta, pp 39–46
24. Park S, Lee D, Lim M, Yu C(2001) Scalable data management using user-based caching and prefetching in distributed virtual environments. In: VRST '01. Baniff, pp 121–126
25. Pittman D, GauthierDickey C (2007) A measurement study of virtual populations in massively multiplayer online games. In: NetGames '07. Melbourne, pp 25–30
26. Rieche S, Wehrle K, Fouquet M, Niedermayer H, Petrak L, Carle G (2007) Peer-to-peer-based infrastructure support for massively multiplayer online games. In: The 4th IEEE consumer communications and networking conference, pp 763–767
27. Scholtes I, Botev J, Esch M, Höhfeld A, Schloss H, Zech B (2008) TopGen—internet router-level topology generation based on technology constraints. In: SIMUtools'08. Marseille

28. Steed A, Angus C(2005) Supporting scalable peer to peer virtual environments using frontier sets. In: IEEE virtual reality. 12–16 March 2005, pp 27–34
29. Tan SA, Lau W, Loh A (2005) Networked game mobility model for first-person-shooter games. In: NetGames '05. Hawthorne, pp 1–9
30. Tian J, Haehner J, Becker C, Stepanov I, Rothermel K (2002) Graph-based mobility model for mobile ad hoc network simulation. In: ANSS' 02. Washington, DC, p 337
31. Varvello M, Picconi F, Diot C, Biersack E (2008) Is there life in Second Life? In: ACM CoNEXT '08. Madrid
32. Vik KH, Griwodz C, Halvorsen P (2006) Applicability of group communication for increased scalability in MMOGs. In: NetGames '06. Singapore, p 2

Huiguang Liang received his Bachelor of Engineering degree with First Class Honours in computer engineering from the National University of Singapore in 2008. For outstanding scholastic achievements at an undergraduate level, he was awarded amongst others the National Semiconductor Book Prize and the A*STAR Pre-Graduate Award in 2007. In 2008, he was awarded the A*STAR Graduate Scholarship by Singapore's Agency for Science, Technology and Research (A*STAR) to study at Carnegie Mellon University, Pittsburgh, PA, for the pursuit of a Ph.D. degree in electrical and computer engineering. His research interests are primarily in wireless network protocols.

Ransi Nilaksha De Silva received his B.Sc. degree with First Class Honours in Computer Science from the University of Colombo, Srilanka. He is currently in his final semester of study in Msc in Electrical Engineering at the National University of Singpore(NUS). He is also working as a full time Research Assistant in the Computer Science Department at NUS. His research interest are in the areas of computer architecture and networking.

Wei Tsang Ooi received his B.Sc. degree from the National University of Singapore, Singapore and Ph.D. degree from Cornell University, Ithaca, NY. He is currently an Assistant Professor with the Department of Computer Science in the National University of Singapore, where he does research in multimedia systems, distributed systems, and computer networking.

Mehul Motani received the B.S. degree from Cooper Union, the M.S. degree from Syracuse University and the Ph.D. degree from Cornell University, all in Electrical and Computer Engineering. He is an assistant professor in the Department of Electrical and Computer Engineering at the National University of Singapore. He has also been a research scientist at the Institute for Infocomm Research in Singapore for three years and a member of technical staff at Lockheed Martin in Syracuse, New York for over four years. His research interests are in the area of wireless networks. He has recently been working on research problems which sit at the boundary of information theory, networking and communications, including the design of wireless ad hoc and sensor network systems. He was awarded the Intel Foundation Fellowship for work related to his Ph.D. in 2000 and won the Telecom Italia Mobile prize at SIMAGINE in 2003. He has served on the organizing committees of ISIT and ICCS and the technical program committees of MobiCom, InfoCom, SECON and various other conferences. He participates actively in the IEEE and ACM and has served as the secretary of the IEEE Information Theory Society Board of Governors. He is a member of the IEEE and ACM.