Network Game Design: Hints and Implications of Player Interaction*

Kuan-Ta Chen
Institute of Information Science
Academia Sinica
ktchen@iis.sinica.edu.tw

Chin-Laung Lei
Department of Electrical Engineering
National Taiwan University
lei@cc.ee.ntu.edu.tw

ABSTRACT

While psychologists analyze network game-playing behavior in terms of players' social interaction and experience, understanding user behavior is equally important to network researchers, because how users act determines how well network systems, such as online games, perform. To gain a better understanding of patterns of player interaction and their implications for game design, we analyze a 1,356-million-packet trace of *ShenZhou Online*, a mid-sized commercial MMORPG. This work is dedicated to draw out hints and implications of player interaction patterns, which is inferred from network-level traces, for online games.

We find that the dispersion of players in a virtual world is heavy-tailed, which implies that static and fixed-size partitioning of game worlds is inadequate. Neighbors and teammates tend to be closer to each other in network topology. This property is an advantage, because message delivery between the hosts of interacting players can be faster than between those of unrelated players. In addition, the property can make game playing fairer, since interacting players tend to have similar latencies to their servers. We also find that participants who have a higher degree of social interaction tend to play much longer, and players who are closer in network topology tend to team up for longer periods. This suggests that game designers could increase the "stickiness" of games by encouraging, or even forcing, team playing.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications; J.7 [**Computers in Other Systems**]: *Command and control, Consumer products, Real time*; K.8.0 [**Personal Computing**]: General—*Games*

*This work is supported in part by the Taiwan Information Security Center (TWISC), National Science Council under the Grant No. NSC95-2218-E-001-001, and by the Ministry of Economic Affairs under the Grant No. 94-EC-17-A-02-S1-049.

Netgames'06, October 30–31, 2006, Singapore.

General Terms

Measurement, Human Factors

Keywords

Design Recommendations, Internet Measurement, MMORPG, Online Games, Overlay Networks, Social Interaction

1. INTRODUCTION

With an exponentially growing population and the increasing diversity of network gamers, the virtual worlds constructed by MMORPGs (Massive Multiplayer Online Role Playing Games) are gradually becoming a field for the study of social behavior [5, 15]. While psychologists analyze network game-playing behavior in terms of players' social interaction and experience, understanding user behavior is equally important to network researchers, because how users act determines how well networked systems, such as online games, perform. For example, the dispersion of players across a virtual world affects how well an algorithm performs in distributing the workload to a number of servers in terms of bandwidth usage, load balancing, and users' perceived quality of games.

To gain a better understanding of the patterns of player interaction and their implications for game design, we analyze *how players interact*. Analyzing user behavior based on network-level traces is particularly useful for our purpose, since the inferred user behavioral patterns would naturally connect to network-level factors, such as IP addresses and network latency between participating parties. Also, it is easier for commercial game operators to provide network-level traces, because unlike logging application-level activities, collecting traffic traces does not increase the load of, or require modification to, game servers. Based on an empirical analysis of player interaction inferred from network traces, this work aims to put forward architectural design recommendations for online games.

We develop an algorithm that derives patterns of player interaction from a 1,356-million-packet trace of *ShenZhou Online* [13], a commercial MMORPG. The inferred interaction patterns are then analyzed in the following aspects: the dispersion of players in a virtual world, the correspondence of network locality and in-game locality, and the "stickiness" of game-playing in terms of social interaction. Our main objective is to draw design implications of player interaction for online games, especially system-level design issues. Our major findings are as follows:

- The dispersion of players in a virtual world can be well

modeled by *Zipf-like distributions*, where 30% of players gather in the top 1% of popular places. This implies that static and fixed-size partitioning of game worlds is inadequate for both server-cluster and peer-to-peer infrastructures [10, 18], and dynamic partitioning algorithms should therefore be used [14, 2, 7, 11].

- *Players who are neighbors or teammates tend to be closer to each other in network topology.* This property is an advantage to network games with either client-server or peer-to-peer architecture, as the message delivery between the hosts of interacting players can be faster. In addition, the property improves the *fairness* of game playing, as interacting players tend to have similar latencies to their servers.

- Players who have a higher degree of social interaction tend to stay in the game world much *longer*. This suggests that game companies could increase the "stickiness" of games *by encouraging, or even forcing, team playing*. Furthermore, the duration of group play correlates with a group's size and the network distance between players. This implies that real-life relationships carry over into the virtual world, and/or real-life interaction plays a key role in game play. The latter also suggests that enriching in-game communication would encourage players to be more involved in team play.

- Larger groups generally lead to longer collaboration due to the enjoyment derived from player interaction and social bonds. This suggests that a game could be made stickier by encouraging the formation of large groups.

The remainder of this paper is organized as follows. Section 2 describes related works. We introduce the studied game and summarize the collected traces in Section 3. In Section 4, we present the methodology used in mining player-interaction from packet-level traces. Next, in Section 5, we analyze the group structure formed by players and discuss their implications for game design. Finally, Section 6 draws our conclusions.

2. RELATED WORK

In [5], the authors observed player-to-player interaction in two locations in the game *Star Wars Galaxies*. They analyzed user interaction patterns, mainly gestures and utterances, and discussed how they are affected by the structure of the game. In [15], the authors conducted an online survey to study the demographics of groups. The relationship between demographic factors and the willingness to participate in a group, or be a group leader, was analyzed. The main result was that older players were more likely to prefer playing solo, either because of schedule constraints or because they simply preferred to play alone. In the PlayOn project [16], the same authors analyzed the relationship of grouping time with leveling time and the class and levels of characters based on a set of population data for *World of Warcraft*.

Similar to the above studies, this work also analyzes the interaction between players in the game world. However, our study is different from previous works in that it mainly focuses on system design issues, rather than psychological factors or design of game content. Our approach is also distinctive because it infers players' group structure based on out-of-game observations only. This strategy enables us to correlate in-game user behavior with out-of-game variables, such as network latency, so that we can evaluate system-level design alternatives from a high-level perspective of user interaction.

Figure 1: A screen shot of *ShenZhou Online*

3. TRACE COLLECTION

ShenZhou Online is a mid-scale, commercial MMORPG that is popular in Taiwan [13], where there are thousands of players online at any one time. To play, the participants purchase game points from a convenience store or online. A screen shot of *ShenZhou Online* is shown in Fig. 1. The character played by the author is the man in the center of the screen with a smiling face above him. He is in a typical market place, where other players keep stalls. As is normal in MMORPGs, a player can engage in fights with random creatures, train himself in special skills, participate in marketplace commerce, or take on a quest.

With the help of the *ShenZhou Online* staff, we set up a traffic monitor beside the game servers. The monitor was attached to a layer-4 switch upstream of the LAN containing the game servers (we call it the "game LAN"). The port forwarding capability of the tapped layer-4 switch was enabled so that all inbound/outbound game traffic was forwarded to our monitor as a copy. To minimize the impact of monitoring, all remote management operations were conducted via an additional network path, i.e., the game traffic and management traffic did not interfere with each other. The network topology and setup of the game servers and the traffic monitor are shown in Fig. 2.

The traffic monitor was a FreeBSD PC equipped with 1.5 GHz Pentium 4 and 256 MB RAM. We used *tcpdump* with the kernel built-in BPF to obtain traffic traces. We randomly chose a subset of game sets in each trace, and only packets belonging to the selected game sets were logged. A game set, which is logically a "game server" from a player's viewpoint, comprises an entry server, several map servers, and a database server. All game sets are equivalent in content, but *isolated*. The reason for providing identical game sets is to distribute the workload over a number of servers with limited game content, e.g., terrain, missions, and creatures in the virtual world. We took two packet traces, $\mathcal{N}_1$

Table 1: Summary of Game Traffic Traces

Trace	Sets	Date	Time	Period	Drops†	Session	Pkt. (in / out / both)	Bytes (in / out / both)
$\mathcal{N}_1$	3	8/29/04 (Sun.)	15:00	8 hr.	0.003%	7,597	342M / 353M / 695M	4.7TB / 27.3TB / 32.0TB
$\mathcal{N}_2$	2	8/30/04 (Mon.)	13:00	12 hr.	?‡	7,543	325M / 336M / 661M	4.7TB / 21.7TB / 26.5TB

† This column gives the kernel drop count reported by *tcpdump*.

‡ The reported kernel drop count was zero, but we found that some packets were actually dropped at the monitor.

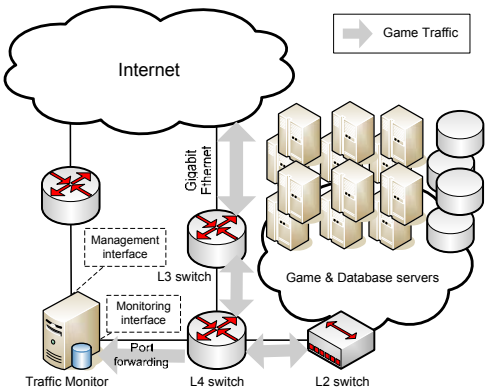

Figure 2: Network setup for traffic measurement

and $\mathcal{N}_2$, which recorded traffic for two and three game sets, respectively. The collected two traces, $\mathcal{N}_1$ and $\mathcal{N}_2$, which spanned 8 and 12 hours respectively and contained more than 1,356 million packets, are summarized in Table 1.

4. DERIVING PLAYER INTERACTION

In this section, we discuss how player interaction is derived from packet traces. As logging in-game activities of all game characters would place an extra burden on game servers, which are already busy, inferring user behavior from network-level traces offers a number of advantages:

- The sampling does not increase the workload of game servers, since packet traces can be recorded by a separate monitor.
- No modification to game servers is required.
- Player behavior inferred from traffic traces naturally connects to network-level factors, such as IP addresses and network latency between conversational parties.

The first two benefits are of particular importance for commercial games, where overloading and possible instability caused by "inessential tasks" (from the viewpoint of game operators) are not tolerated. The last point, on the other hand, is especially relevant to system-level design issues for online games. Because of these advantages, we believe methodologies that derive in-game user activities from out-of-game observations are valuable and merit investigation.

We infer player interaction based on *the correlations between packet arrival processes*; so that the decision about whether two players interact is based on observations over a period of time. To make the analysis tractable, we adopt a *session-based correlation analysis* to determine the degree of interaction between a pair of players. That is, two players are deemed to be *interacting* if they maintain contact throughout the overlapped time of their sessions; otherwise, they are considered unrelated. This strategy is conservative, since some players may interact with other players for a period of time, and be independent hereafter. However, as our objective is to draw hints and implications from player interaction for network game design, the session-based analysis is sufficient as it is competent to elicit the group structure formed by players.

In the following, we describe the terminology and methods used to determine the relationships between players. Hereafter, we use the terms "players" and "sessions" interchangeably.

4.1 Terminology

- **Co-presence Time**: The overlap of the sessions of two or more players. Players are **co-present** if their co-presence time is longer than ten minutes.
- **Neighbor**: Two players are neighbors if they are co-present and their characters are close throughout their co-presence time. A player is called **social** if s/he has a neighbor(s), and **independent** otherwise.
- **Partner**: Two players are partners if they are neighbors and act in synchrony throughout their co-presence time. A player is called **grouped** if s/he has a partner(s), and **solo** otherwise.
- **Horde**: A set of players in which every two players are *neighbors* if they are co-present. Each player in a horde must have at least one neighbor in that horde. Players in a horde always stay together—either at a site, or moving as a cohort in the virtual world.
- **Group**: This term is similar to "horde," except that the interacting subjects are "partners" instead of "neighbors." A group corresponds to a team of players, which is formed for journey or certain quests. Players in a group always move and act in unison.
- **Place**: A relaxed definition of "horde." This is a superset of one or many hordes that stay at the same location; therefore, a place semantically corresponds to a location in the game world.

To clarify the above terminology, the relationships between social players, independent players, and group players are depicted in Fig. 3. Note that social players and independent players are complementary, and group players are a subset of social players.

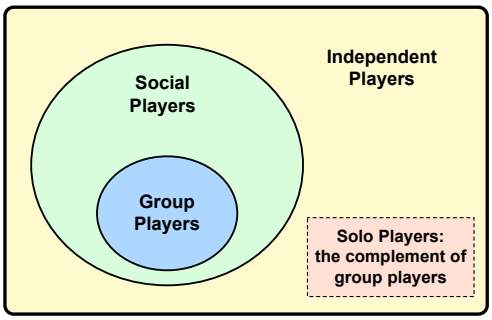

Figure 3: The Venn diagram of player classification

4.2 Identification of Player Interaction

We have developed an algorithm to identify how players interact in a virtual world from packet traces. The only input of this algorithm is each session's packet arrival processes, which are formed by counting the number of client data packets and server data packets observed in successive 30-second periods. For the sake of brevity, hereafter, the term "packets" refers to "data packets" unless otherwise stated. Server packets primarily convey the state and position updates of nearby characters; therefore, they possess the property of *spatial locality* [3]. In other words, game servers release an approximate number of server packets for the sessions whose characters are in the vicinity of each other. We exploit this property to detect players who are close to each other. On the other hand, client packets convey user commands; therefore, the packets' transmission rate reflects *the degree of player activity*. Based on this property, we treat a cohort of players as a team if their activity levels are synchronous during their co-presence time.

The pseudo code for determining the neighbors and partners of all players is listed in Algorithm 1. The procedures ensure that the relationships of neighbors and partners follow the law of communication and the law of transition. The identification of relationships proceeds in a pairwise fashion. Two players who have server packet arrival processes $\mathcal{S}_a$ and $\mathcal{S}_b$ respectively during their co-presence time are deemed to be neighbors if and only if they meet all the following conditions: 1) there is a high Pearson's correlation coefficient between $\mathcal{S}_a$ and $\mathcal{S}_b$; 2) the ratio between $mean(\mathcal{S}_a)$ and $mean(\mathcal{S}_b)$ is small; 3) the ratio between $sd(\mathcal{S}_a)$ and $sd(\mathcal{S}_b)$ is small, where $sd(\cdot)$ denotes the standard deviation; 4) the 95% percentile of $|\mathcal{S}_a - \mathcal{S}_b|$ is small; and 5) when the law of transition cannot be satisfied, neighbors with higher correlation coefficients are respected. Based on observation, we set the threshold of the correlation coefficient condition to 0.6, the threshold of all ratio tests to 2, and the threshold of the deviation test to 2 pkt/sec. Even the choice of thresholds affects the decision about relationships, the algorithm has been re-run with different thresholds and similar results were obtained. We argue that the desired result—the *qualitative property* of player interaction—holds as long as the thresholds are kept within a reasonable range.

Once all neighbor-pairs have been identified, we determine whether a pair of neighbors are partners according to their activity levels. By the observation that the distribution of client packet rates is bi-modal with a trough around 1.5 pkt/sec, in each epoch (30 seconds), a session is considered "active" if the average client packet rate is above 1.5 pkt/sec, and considered "idle" otherwise. Two players with co-presence time t epoches are treated as partners if both are active for at least $\lceil t/2 \rceil$ epoches, and their activity levels are the same for at least $\lceil 0.9 \times t \rceil$ epoches. The rationale behind this judgement is that a team of players should be active or inactive synchronously—it is unlikely that one player would rest while his/her teammates are involving in fierce combat. Because many players remained idle during most of their online time, we avoid misidentifying them as a large "idle-team" by not including them in any group. This is reasonable since players join groups for a journey, not for a rest.

Algorithm 1 Identification of neighbors and partners

for all session i **do**
  compute packet counting processes $\mathcal{C}_i$ and $\mathcal{S}_i$ for client data packets and server data packets, respectively (sampled every 30 seconds)
  $\mathcal{AI}_i \leftarrow \mathcal{C}_i > 1.5$ pkt/sec ▷ activity indicator
end for
$\Omega \leftarrow$ all session pairs (a, b) that are co-present and have Pearson's correlation coefficient $\rho > 0.6$
for all pairs (a, b) in Ω with descending ρ **do**
  if is.neighbor(a, b) **and**
    is.neighbor$(a, b$'s all neighbors$)$ **and**
    is.neighbor$(b, a$'s all neighbors$)$ **then**
    set a and b as neighbors
  else continue
  end if
  if is.partner(a, b) **and**
    is.partner$(a, b$'s all partners$)$ **and**
    is.partner$(b, a$'s all partners$)$ **then**
    set a and b as partners
  end if
end for

function IS.NEIGHBOR(a, b)
  ▷ check coincidence between $\mathcal{S}_a$ and $\mathcal{S}_b$
  ▷ $\mathcal{S}_a$ and $\mathcal{S}_b$ refer to their intersectional portion
  $ratio.avg \leftarrow mean(\mathcal{S}_a)$ / $mean(\mathcal{S}_b)$
  $ratio.sd \leftarrow sd(\mathcal{S}_a)$ / $sd(\mathcal{S}_b)$
  $result \leftarrow quantile(|\mathcal{S}_a - \mathcal{S}_b|, .95) < 2$ **and**
    $(ratio.avg < 2$ **and** $ratio.avg > .5)$ **and**
    $(ratio.sd < 2$ **and** $ratio.sd > .5)$
end function

function IS.PARTNER(a, b)
  ▷ $\mathcal{AI}_a$, $\mathcal{AI}_b$ refer to their intersectional portion
  $\tau \leftarrow$ co-presence time of session a, b
  $result \leftarrow count(\mathcal{AI}_a = true) > 0.5 \times \tau$ **and**
    $count(\mathcal{AI}_b = true) > 0.5 \times \tau$ **and**
    $count(\mathcal{AI}_a = \mathcal{AI}_b) > 0.9 \times \tau$
end function

After the relationship between each pair of players has been determined, we divide players into clusters: players who are neighbors (must be mutually) form a horde, while players who are partners (must be mutually) form a group. Note that a player may simultaneously belong to a number of groups, because s/he can form a group with some players and form another group with some other players in different periods. Finally, hordes containing the same players are

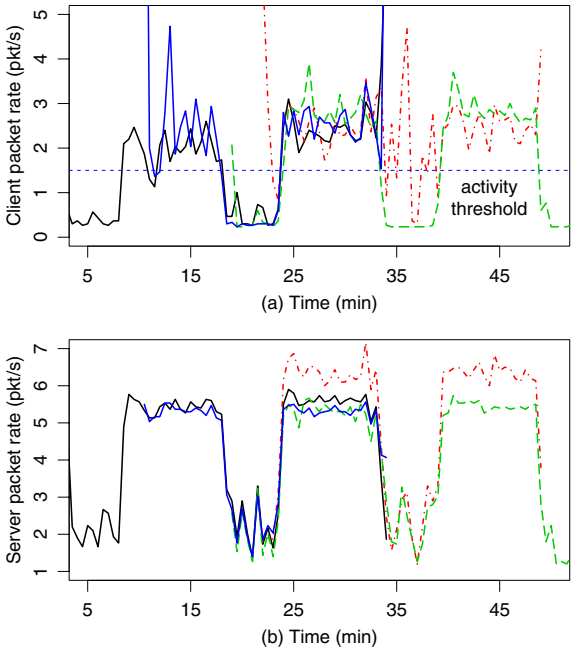

Figure 4: Packet arrival processes of a four-player group

merged into a place. This operation forms disjoint places, where each player uniquely belongs to a place.

To demonstrate the principle behind the identification algorithm, in Fig. 4, we depict the packet arrival processes of a four-player group. We can see that the fluctuations of server packet arrivals follow very similar patterns during the co-presence time, which is longer than 40 minutes. Meanwhile, the client packet arrivals exhibit similar patterns in their activity levels. The figure also shows that the duration of a group may be longer than the session time of any member. In other words, even the members in a group come and go, a group remains if at least two members are still in the game.

5. ANALYSIS OF PLAYER INTERACTION AND ITS IMPLICATIONS

In this section, we analyze players' interaction patterns and their implications for network game design. We first present a summary of the group structure, which is inferred using the Algorithm 1 in Section 4. We then analyze player interaction patterns and discuss their implications in a number of aspects, including player density, network proximity, session duration, and interaction levels.

5.1 Basic Statistics

Of the 16,528 players who had potential to interact with others, i.e., players with session time longer than ten minutes, 67% were social players, and 17% were group players. On average, a social player interacted for 70% of his online time, while a group player teamed up with others for 50% of his/her online time. Meanwhile, a social player had 3.5 neighbors and a group player had 1.4 partners on average, as shown in Table 2.

Table 2: The average number of neighbors and partners of a player

	All Players	Social Players	Group Players
# of Neighbors	2.4	3.5	2.3
# of Partners	≈ 0	0.5	1.4

Fig. 5 shows how players were scattered in different places and groups. We can see that 40% had at least four neighbors during the game, while 20% stayed in places where more than one hundred players had gathered. The most popular place, where 1,816 players gathered, shows that more than 10% of players spent the whole session in the same location. On the other hand, groups are frequent but mostly small-sized. Among 1,389 groups, 85% were "duos," and only 2% had more than four members. We believe the prevalence of "duos" is not unique to this game, as it reflects the fact that some players are not really social and usually interact primarily with one other person in real life. This echoes the following comments by gamers reported in [15]:

> *"Distinct from 'grouping' in that [sic] (as I see it), the 'duo-group' is most often the same two people (perhaps with a relationship offline), who don't typically group with people other than each other."*

> *"My answers to questions in this vein would show I often group, but it [sic] would likely be grouped with others who are honestly much more social than I, since I predominantly duo with my spouse."*

5.2 The Skewness of Place Populations

As the number of concurrent players in a MMOG are usually exceeding tens of thousands, even a powerful server could not handle the vast computational requirement of an entire game. A straightforward solution for solving the scalability problem is to distribute the workload among multiple servers. There are two common approaches: the server-cluster approach [1, 20], and the peer-to-peer approach [10], both of which normally divide the game world into a number of regions with each of which may be managed by different servers. With this approach, the key to an efficient load balancing depends on whether it can *correctly delegate a region (of the game world) to an appropriate server*, while minimizing the overall transmission latency and bandwidth utilization and subject to the constraint that no server is overloaded.

Furthermore, one of the deciding factors that affects the outcome of region delegation is *how players are distributed across the virtual world*. If players tend to be uniformly distributed, then a straightforward fixed-size partitioning of the game world is sufficient, given the players in each region is not too many to overload any single server. On the contrary, if players tend to aggregate in certain hotspots that change from time to time, then a dynamic and adaptive partitioning algorithm must be used.

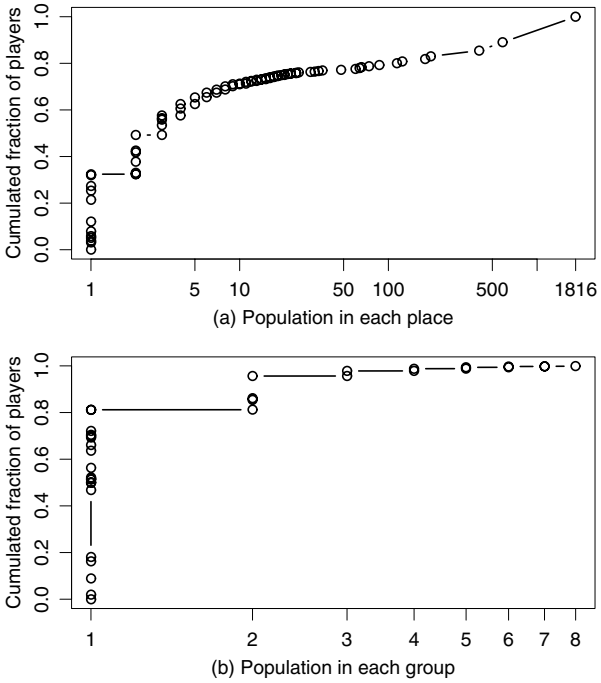

Figure 5: The distribution of groups and places

From the distribution of place popularity (Fig. 5), there were a number of crowded places, e.g., places where more than fifty players gathered, which indicate the existence of hotspots. Further analysis shows that the distribution of populations in different places can be well modeled as the concatenation of two Zipf-like distributions [19]. Suppose a place with rank i and population p_i follows a Zipf-like distribution; then $p_i = C/i^{\alpha}$, where C is a normalizing constant and $\alpha > 0$ is the power-law exponent, i.e., the population of a place is inversely proportional to its rank with a log-log transform. The smaller the α, the heavier tail the distribution will have.

Fig. 6 depicts the populations of places versus their ranks on a log-log plot, where the points are scattered along two line segments. This evidences that the populations of the most popular twenty places follows a Zipf-like distribution with $\alpha \approx 1.4$, while the populations of the less popular places follows the same distribution with $\alpha \approx 0.5$. Even the populations of popular places have a lighter tail than a standard Zipf distribution ($\alpha = 1$), the most crowded hotspot has about 10% of the players aggregated, and the most popular 1% of places (out of $7,712$) contain about 30% of the players. This manifests that *the dispersion of players is far from uniform and is closer to a power-law distribution.* The high variability in player density across a virtual world implies that static world partitioning strategies [10, 18] are inadequate for MMOGs. On the other hand, we remark that dynamic partitioning algorithms, which adaptively adjust the number, size, and ownership of regions based on the constantly changing player density, should be employed for online games with either server-cluster [14, 2] or peer-to-peer

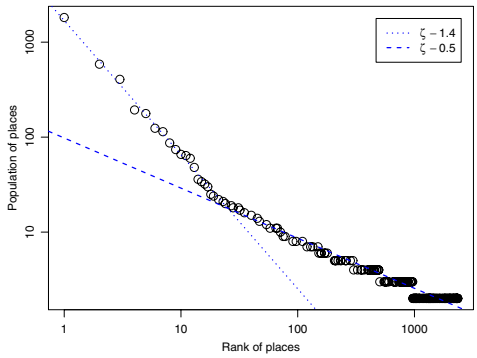

Figure 6: Popularity versus rank of places

architecture [7, 11].

5.3 Network Proximity of Interacting Players

A number of peer-to-peer infrastructures for online games and network virtual environments (NVEs) have been proposed in recent years. However, a fundamental design question remains unsolved: *How should the participating hosts interconnect for better networking performance?* Currently, there are three common approaches for constructing overlay networks: 1) numeric-ID based [7]; 2) network topology-based [6], e.g., optimizing network latency; and 3) in-game location-based [18, 8]. Although the first two approaches are adequate for general purpose applications, such as file sharing, they may be less appropriate for NVEs. For example, with a latency-optimized overlay, a message sent from a host can be delivered to *any other host* in the network with generally low latencies. However, if *game players only communicate with their neighbors in the virtual world*, then a naive approach, where the hosts of each pair neighbor are interconnected, would achieve better performance in terms of message transmission latency. A complete evaluation of the design alternatives (i.e., different strategies to connect overlay nodes) requires a detailed trace involving inter-host measurements and is beyond the scope of this study. Instead, using a *comparative* approach, we investigate *the degree of correspondence between the locality in the network and the locality in the virtual world*, which implies the similarity of the overlay networks built with the second approach and the third approach.

5.3.1 Similarity in Network Addresses

Fig. 7 shows the proportion that a pair of players with a certain kind of relationship share the same IP address prefix. We found that about 7% of partner-pairs co-located in the same /16 network, whereas only 1.7% of unrelated-pairs were in the same network. At the same time, approximately 5% of partner-pairs used the same IP address, while very few unrelated-pairs did so. This indicates that a proportion of group members played at the same physical location (in an apartment or Internet café) through a NAT device. Comparing the three levels of relationship, we can see that players with a higher level of interaction tend to be closer in network topology. These findings suggest that *overlays built on in-game localities correlate with topology-aware overlays to some extent*, i.e., whether the overlay is based on in-game

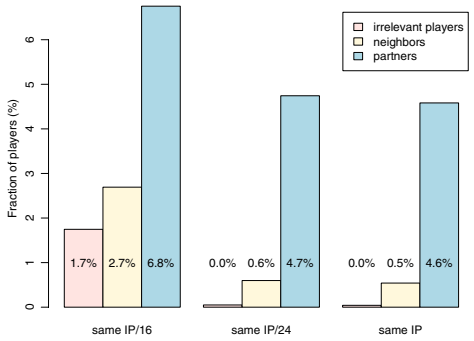

Figure 7: The correspondence between IP addresses and in-game closeness

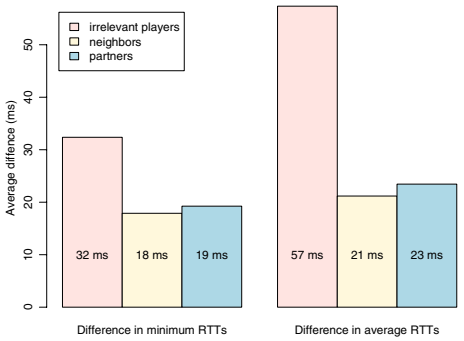

Figure 8: The correspondence between round-trip times and in-game closeness

closeness or network topology does not make so much difference.

5.3.2 Similarity in Network Latencies

Fig. 8 provides another view to examine the network proximity of players. We use the difference in minimum RTTs (round-trip times) and the difference in average RTTs between the players' hosts to gauge their network distance from the perspective of game servers. The figure shows that interacting players are much closer to each other compared with unrelated players. This echoes the agreement between in-game locality and network locality based on IP addresses.

5.3.3 Advantages of the Property

Client-server based games would benefit from the network proximity property of interacting players in terms of *proxy placement* and *fairness of game playing*. A client-server architecture with proxy support is a design that alleviates the single bottleneck problem of, and retains the manageability of, the centralized server architecture [12]. In such architecture, players connect to their nearest proxy server, which processes and responds to user commands in place of game servers. Proxy servers are usually close to users; thus, the transmission latency and the user-perceived response time could be significantly reduced. However, additional proxy-to-proxy communication is required when players who connect to different proxy servers interact. In this case, the overall message delivery latency could be longer due to detours between proxys. To provide high responsiveness and interactivity, two seemingly contradictory conditions need to be met: 1) players should connect to their nearest proxy server, and 2) players who are neighbors in the game should connect to the same proxy server. Fortunately, the network proximity property of interacting players resolves the contradiction between the above conditions. The reason is that players are likely to connect to the same proxy as their in-game neighbors (with a non-significant probability), even if proxy selection is only based on the network distance.

Fairness is also an important issue in network game design. Normally there is a bias against players with longer latencies (between clients and servers), since their screens tend to update later than others if the server releases an update message for all participants at the same time. Furthermore, the actions of players who are far from servers are likely be processed (by servers) later than others, even if all commands are issued simultaneously (on their respective computers). In the case of client-server based games, a set of players can have *fairer play* if their network latencies to game servers are similar. It is because that servers will process their actions approximately at the same time if they trigger commands simultaneously. Thus, the network proximity for interacting players makes their interaction, such as fighting each other, fairer, as their network latencies to game servers are similar.

For the above reasons, we consider that the network proximity of interacting players is a "benign" property, as it is favorable to the design of network games with both peer-to-peer and client-server architectures. We remark that as player interactions are generally *spatially localized*, the mapping of spatial locality between the game world and the network topology would be rather important and could be a key to the optimization of network latency in online games.

5.4 Game Playing Time and Social Interaction

As social interaction is one of main attractions of MMORPGs [17], a factor that encourages time investment and personal attachment, one may ask: Does social interaction "tie" players to a game and encourage them to play longer than independent players who do not have any interaction with others?

To answer this question, we plot the survival curves [9] for sessions grouped by interaction levels, and check the significance of the differences between the groups. In Fig. 9, the survival curves for three groups of sessions are plotted. Visually, the curves diverge significantly from each other; and the log-rank test reports $p = 1 - \Pr_{\chi^2,2}(1532) \approx 0$, which indicates the sessions in different groups are far from equivalent. The median session time for independent players, social (but solo) players, and group players is 40, 112, and 190 minutes, respectively, which gives a high ratio of nearly 5:1 for group players and independent players. This figure implies that there is a strong relationship between social interaction and time investment in a game. Game designers, therefore, may provide more facilities to encourage teamwork, or even force players, to form groups for quests, which would increase the "stickiness" of a game and eventually generate more revenue through subscription fees.

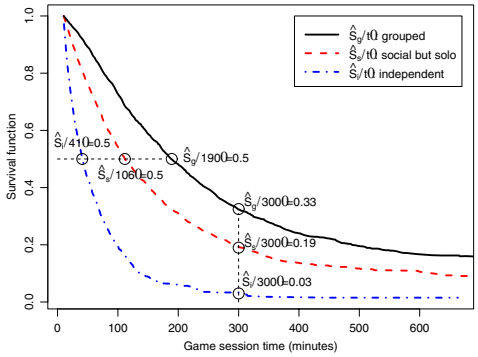

Figure 9: Comparison of game session duration between players with different levels of interaction

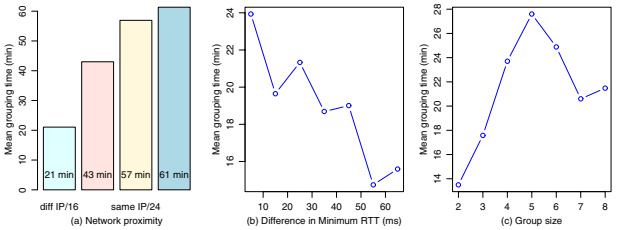

Figure 10: The relationship between grouping time and various related factors

5.5 Degree of Group Interaction

Now that the existence of network proximity and the tendency of group players to be "tied" in a game have been established, in this subsection, we focus on group players and investigate the relationship between the level of group interaction and some potentially relevant factors. We will show that the former is related to *the degree of network proximity* and *group size*. To quantify the level of group interaction, we define the "group time" of a group as the average pairwise co-presence time among its members.

We plot the relationship between group time and three potentially relevant factors in Fig. 10. As can be seen in Fig. 10(a), *the closer group members were in network topology, the longer they played together*. The property of network proximity can also be examined in terms of the network latency between clients and servers. The smaller the difference in minimum RTT experienced by group members, the longer the group playing time they shared, as shown in Fig. 10(b). Assuming that network proximity reflects geographical locality, the above relation implies that 1) real-life relationships carry over into the virtual world, and/or 2) real-life interaction plays a key role in game playing. The latter explanation suggests that enriching in-game communication supports, e.g., audio and video chat facilities, would encourage players to be more involved in team cooperation.

Fig. 10(c) shows that players in larger groups tend to stay longer. This phenomenon can be explained by the *enjoyment derived from interaction* and *social bonds*. For example, the latter may prevent a player leaving a game prematurely, as it would affect the group's adventure. This suggests that encouraging players to form larger groups would also increase the "stickiness" of the game. However, the effect diminishes for groups with more than five members, possibly because the role of each player in a large group is less important. In other words, the social bond is weak when a player leaves a large group than leaving a smaller group.

6. CONCLUSION

In this paper, we have analyzed player interaction for *ShenZhou Online*, a mid-sized commercial MMORPG, and drawn out a number of hints and implications for network game design. The analysis reveals that the dispersion of players in a virtual world is heavy-tailed, which implies that static and fixed-size partitioning of game worlds is inadequate. We have shown that neighbors and teammates tend to be closer to each other in terms of network topology, a property that is beneficial to both client-server and peer-to-peer based games, because message delivery between the hosts of interacting players is faster. In addition, this property makes games fairer, as interacting players tend to have similar latencies to their servers.

We have also found that participants who have a higher degree of social interaction tend to play much longer than independent or solo players. This suggests that game companies could increase the "stickiness" of games by encouraging, or even forcing, team playing. Furthermore, the duration of group play correlates with the size of the group and the network distance between the players. Specifically, players who are closer in network topology tend to team up for longer periods. Larger groups generally lead to longer collaboration due to the enjoyment derived from interaction and social bonds, which suggests that games could also be made stickier by encouraging the formation of such groups.

We believe that only through observation can we fully understand what users desire and therefore be able to design appropriate network systems for them. For instance, via an inspection of how players react to network conditions, we can understand how players perceive unfavorable network QoS (e.g. packet loss and delay jitters are more intolerable than network latency). Such understanding could be very helpful in the design of network game systems, especially in the message delivery subsystem, to diminish the impact of unavoidable network impairment on players as possible [4]. Designing more efficient network gaming systems which take account of user behavior and perception and be optimized for user satisfaction will be one main theme of our future work.

7. REFERENCES

[1] Butterfly.net, Inc. The butterfly grid: A distributed platform for online games, 2003. http://www.butterfly.net/platform.

[2] J. Chen, B. Wu, M. Delap, B. Knutsson, H. Lu, and C. Amza. Locality aware dynamic load management for massively multiplayer games. In *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*. ACM Press, June 2005.

[3] K.-T. Chen, P. Huang, and C.-L. Lei. Game traffic analysis: An MMORPG perspective. *Computer Networks*, 51(3), 2007.

[4] K.-T. Chen, P. Huang, G.-S. Wang, C.-Y. Huang, and C.-L. Lei. On the sensitivity of online game playing time to network QoS. In *Proceedings of IEEE INFOCOM'06*, Barcelona, Spain, Apr. 2006.

[5] N. Ducheneaut and R. J. Moore. The social side of gaming: a study of interaction patterns in a massively multiplayer online game. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 360–369. ACM Press, 2004.

[6] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of DHT routing geometry on resilience and proximity. In *Proceedings of SIGCOMM'03*, pages 381–394. ACM Press, 2003.

[7] S.-Y. Hu and G.-M. Liao. Scalable peer-to-peer networked virtual environment. In *Proceedings of ACM SIGCOMM 2004 workshops on NetGames '04*, pages 129–133. ACM Press, 2004.

[8] T. Iimura, H. Hazeyama, and Y. Kadobayashi. Zoned federation of game servers: a peer-to-peer approach to scalable multi-player online games. In *Proceedings of ACM SIGCOMM 2004 Workshops on NetGames '04*, pages 116–120. ACM Press, 2004.

[9] E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53:437–481, 1958.

[10] B. Knutsson, H. Lu, W. Xu, and B. Hopkins. Peer-to-peer support for massively multiplayer games. In *Proceedings of IEEE INFOCOM'04*, Hong Kong, China, Mar. 2004.

[11] J. C. S. Lui and M. F. Chan. An efficient partitioning algorithm for distributed virtual environment systems. *IEEE Trans. Parallel Distrib. Syst.*, 13(3):193–211, 2002.

[12] M. Mauve, S. Fischer, and J. Widmer. A generic proxy system for networked computer games. In *Workshop on Network and System Support for Games*, Braunschweig, Germany, Apr. 2002.

[13] UserJoy Technology Co., Ltd. ShenZhou Online. http://www.ewsoft.com.tw/.

[14] B. D. Vleeschauwer, B. V. D. Bossche, T. Verdickt, F. D. Turck, B. Dhoedt, and P. Demeester. Dynamic microcell assignment for massively multiplayer online gaming. In *NetGames '05: Proceedings of the 4th Workshop on Network and System Support for Games*, 2005.

[15] N. Yee. The demographics of groups and group leadership. The Daedalus Project. http://www.nickyee.com/daedalus/archives/000553.php.

[16] N. Yee. Grouping and leveling time. PlayOn: Exploring the social dimensions of virtual worlds. http://blogs.parc.com/playon/archives/data/wow_data/grouping/.

[17] N. Yee. A model of player motivations. The Daedalus Project. http://www.nickyee.com/daedalus/archives/001298.php.

[18] A. P. Yu and S. T. Vuong. MOPAR: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games. In *NOSSDAV'05: Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 99–104. ACM Press, 2005.

[19] G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison Wesley, Reading, MA, 1949.

[20] Zona Inc. Terazona white paper, 2002. http://www.zona.net/whitepaper.