

Design Issues for Peer-to-Peer Massively Multiplayer Online Games

Lu Fan, Phil Trinder, and Hamish Taylor
School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, UK
{lf16, p.w.trinder, h.taylor}@hw.ac.uk

Abstract—Massively Multiplayer Online Games (MMOGs) are increasing in both popularity and scale, and while classical Client/Server architectures convey some benefits, they suffer from significant technical and commercial drawbacks. This realisation has sparked intensive research interest in adapting MMOGs to Peer-to-Peer (P2P) architectures.

This paper articulates a comprehensive set of six design issues to be addressed by P2P MMOGs, namely interest management, game event dissemination, NPC host allocation, game state persistency, cheating mitigation, and incentive mechanisms. Design alternatives for each issue are systematically compared, and their interrelationships discussed. We further evaluate how well representative P2P MMOG architectures fulfil the design criteria.

Index Terms—P2P, MMOGs, Interest Management, Event Dissemination, Task Distribution, Distributed Storage, Anticheating, Collaboration Incentives

I. INTRODUCTION

Massively Multiplayer Online Games (MMOGs) enable thousands of players to interact simultaneously in a persistent game world over a network. Client/Server (C/S) architectures have been predominantly employed for traditional MMOGs, because they are relatively easy to secure and implement [1]. However, with widespread use of the Internet, MMOGs are becoming increasingly popular, and the participation has increased dramatically, reaching hundreds of thousands of players [2]. At such scales a C/S architecture exhibits various technical and commercial drawbacks, specifically in the area of reliability and cost, e.g. costs for server hardware, network bandwidth, housing, cooling, UPS systems and dedicated maintenance staff. These factors have engendered strong research interest in engineering Peer-to-Peer (P2P) MMOGs [3], [4], [5].

The key challenge in adapting a conventional MMOG to a P2P architecture is to fulfil the functionalities of centralised game servers in a distributed fashion. This entails addressing six essential issues, namely Interest Management (Section II-A), Game Event Dissemination (Section II-B), NPC Host Allocation (Section II-C), Game State Persistency (Section II-D), Cheating Mitigation (Section II-E) and Incentive Mechanisms (Section II-F).

Previous work has investigated an individual design issue, e.g. several interest management schemes are compared in [6], [7]. Other related work focuses on general requirements of a P2P MMOG, e.g. scalability, availability, interactivity [8],

[9]. In contrast, the most significant contributions this paper makes are to articulate a comprehensive set of six design issues to be addressed by P2P MMOGs, to survey various approaches to addressing them (Section II), and to evaluate representative infrastructures for their integration into a single system (Section III).

II. P2P MMOG DESIGN ISSUES

A. Interest Management

The primary requirement for a P2P MMOG is to maintain a consistent, shared sense of virtual space among large numbers of players without a server's support. Interest Management (IM) is a classical research topic that was initially addressed by Macedonia et al. in the mid 1990s [10]. The concept of IM originates from two observations: a single player does not need to know about what is happening in the game world as long as it does not affect the player; and a player's avatar only has limited movement speed and sensing capability. So, a player's view of the game world can be limited to a comparatively static Area of Interest (AOI), and the player only needs to be aware of game events that occur within its AOI. Generally speaking, existing IM schemes can be classified into three types - a spatial model, a region-based publish/subscribe model, and a hybrid communication model.

1) *The Spatial Model*: The spatial model uses the properties of space as the basis for mediating interaction [11]. This model is also referred to as the "aura-nimbus" model [6], [7] because of its key abstractions: aura and nimbus. The "aura" means the area that bounds the presence of an object in space, while the "nimbus" means the mutual awareness levels between two objects. In other words, object A is only able to interact with object B when their auras intersect with each other, but A is aware of B when it is in B 's nimbus. So, every object should establish communications with other objects that fall within its nimbus, in order to prepare for potential interactions.

The advantage of the spatial model is that it allows fine-grained IM in which only necessary messages are transmitted among relevant peers [6]. However, a significant drawback is that it requires all objects to exchange positional update information in order to identify when AOI collisions occur. For example, in [12] all players must share the "frame of reference" to know their location with respect to each other. Furthermore, the frequency of these updates must be sufficient to ensure that AOI collisions can be determined in a timely

fashion, which may lead to considerable communication overhead [7]. To mitigate this limitation, a Voronoi diagram can be employed to help a player find its neighbouring players in a game world [13], [14]. Each peer is required to construct and maintain a Voronoi diagram by itself, based on the spatial coordinates of neighbours. A peer only needs to keep network connections with its current neighbours, and each peer serves as the “watchman” for one another in discovering approaching neighbours. This approach is able to reduce the communication overhead induced by a pure spatial model, but it is not ideal in that:

- A Voronoi diagram is vulnerable to the “circular line-up” problem, which is the worst case when a peer has $n - 1$ neighbours in a diagram of n sites.
- The communication overhead is not minimal, because a peer still needs to receive and process messages outside its AOI.
- While the communication overhead is reduced, the computation overhead might be increased for users to construct and maintain their Voronoi diagrams.

2) *Region-Based Publish/Subscribe Model*: This model proposes to support a P2P MMOG using coarse-grained IM by partitioning a game world into static regions. Conceptually, it is analogous to a publish/subscribe model, in which the recipient of a message is limited to only interested participants that reside within the same, or neighbouring region as the sender [7]. The main responsibility of a region-based IM mechanism is to determine the regions that intersect a player’s AOI, and to form the area-of-subscription for relevant events from the union of the intersected regions.

A region-based model offers several advantages. Firstly, it is simpler and cheaper to compute a player’s area-of-subscription than to compute AOI collisions. Secondly, a region’s publish channel maps nicely onto a multicast group, hence gaming events can be exchanged efficiently. Thirdly, because regions are predefined, players may carry out local IM without knowing the positions of other players.

However, this model also suffers from two drawbacks:

- It is hard to determine the appropriate size of a region. A region must be of sufficient size to ensure objects are able to disseminate messages in one region before entering another [7]. On the other hand, the granularity cannot be too large, otherwise a player’s machine might be overloaded by excessive irrelevant messages.
- Region-based IM does not always work well when objects are unevenly distributed.

3) *Hybrid Communication Model*: As indicated by its name, this model is a mixture of the spatial and the region-based models. MOPAR [15] and Meta-Model [16] are representatives of recent hybrid IM schemes. Similar to a region-based model, a game world is partitioned into multiple regions, and a super-peer [17] is selected to be responsible for each region. When a player is about to join a region, it finds out the current super-peer working in that region, and updates the super-peer about its moving state. In this way, the super-peer can have a global view of the region, anticipate every peer’s

position in the near future, and analyze AOI collision events using an algorithm such as [18]. Players, whose AOIs are about to intersect, will be notified by the super-peer to establish direct P2P connections with each other, to be prepared for potential interactions.

The hybrid communication model takes advantage of both the spatial and region-based models. On the one hand, it facilitates fine-grained IM in each region and reduces more communication overhead for ordinary players than in a pure region-based model. On the other hand, it is relatively simpler to implement and more efficient than a pure spatial model.

However, it also has the following drawbacks:

- It may impose high computation and communication workloads on a super-peer for a crowded region, where dynamic zoning techniques may be needed as a remedy to distribute the total workload to a set of super-peers, each taking charge of a sub-region.
- A super-peer is potentially a single failure point in a region, so fault-tolerant mechanisms are also needed to provide suitable super-peer backups in order to achieve adequate robustness for the system.

4) *IM Discussion*: Of three IM approaches outlined above, the spatial model is the most fine-grained, and as we shall see in the next section, it is a prerequisite for disseminating game events using unicast. However, due to the lack of centralised control, the communication overhead for a player to establish a global view of the game world may be high. This can be mitigated with a hybrid model that combines the advantages of both the spatial and region-based models. As hybrid models use super-peer networks, they must address super-peer selection, load-balancing and fault-tolerance issues.

B. Game Event Dissemination

While IM focuses on finding out what information is relevant to each player, game event dissemination is concerned with how relevant information is actually delivered to the players. In fact, the choice of a game event dissemination approach is largely determined by the underlying IM mechanisms used in a P2P MMOG.

1) *Unicast vs. Multicast*: The spatial model supports fine-grained IM, which explicitly tells a small set of objects that a player may interact with them shortly. Therefore, the player may establish direct P2P connections just with these objects, and gaming events can be exchanged through unicast communications, e.g. in [12], [13], [14], [19]. Similarly, because super-peers in a hybrid model also provide fine-grained IM services, unicast applies to [15] and [16] as well.

However, a coarse-grained IM scheme only tells a player of some regions to which events he should subscribe. In these regions there might be a large number of players, so unicast becomes inefficient. Instead, each region is represented by a multicast group, which offers a single medium for any region participant to publish gaming events in, and enables the events to be received by all the other region participants. In this circumstance, multicast technology becomes crucial for game event dissemination.

Traditionally, IP multicast [20] was proposed as a efficient group communication mechanism [21]. However, due to a number of technological, practical, and business obstacles [22], IP multicast is not widely available on the Internet. As an alternative Application-Level Multicast (ALM) has been proposed to support similar functionalities, but as an application service instead of a network service [23]. Structured P2P overlays provide good communication infrastructures for building ALMs. For example, Bayeux [24] uses Tapestry [25], CAN Multicast [26] uses CAN [27], and both Borg [28] and Scribe [29] use Pastry [30]. Currently, some related work directly disseminates gaming events with a general purpose ALM system, e.g. [3], [4], [31], [32] which use Scribe. However, others like [33], [34], [35], [36] have proposed their own group member management and multicast tree construction algorithms, which aim at providing better robustness, scalability and load-balancing capabilities.

2) *Problems with ALM*: A significant problem with an ALM system is the potential latency issue, as a game event often can be relayed by multiple forwarders before it arrives at the final receiver. This process incurs unnecessary end-to-end delay, especially when the size of a multicast group is large.

To cope with this problem, it has been suggested that a multicast tree can be constructed according to the proximity of peers in the game world instead of the proximity of peers on the network [37], [38], [39]. In this way, players in the vicinity are employed as the most immediate forwarders, hence multicast messages are sent to close by peers faster, while peers that are further away receive them slower. This approach attempts to provide better game interactivity by exploiting the tolerance of distant players in a game world for weak synchronizations. In addition, an expedite event dissemination mechanism has been proposed in [40], which is able to reduce the overall time that a multicast tree takes to disseminate an event by utilizing better the time slot between a forwarder's completion of a relaying task and the generation of the next gaming event.

3) *Event Dissemination Discussion*: Disseminating game events with ALM will typically induce longer communication latency than with unicast. However, efficiency is also a consideration for unicast as a player may not have enough bandwidth to send every game event to large numbers of recipients. Hence a P2P MMOG requires either a fine-grained IM mechanism that enables a player to unicast game events only to necessary recipients, or a specially designed ALM mechanism that is able to exploit distant players' tolerance of weak synchronization.

C. NPC Host Allocation

Besides player-controlled characters (PCs), there are also considerable numbers of AI-controlled non-player characters (NPCs) in a MMOG, which either drive continuing storylines, or combat with PCs as monsters. MMOGs have to supply their game worlds with large numbers of such NPCs as required by game scenarios. Traditionally, NPCs are hosted by a game server, consuming significant processing power and network

bandwidth. Therefore, one of the prerequisites for realizing a P2P MMOG is to host such NPCs using computing resources that are available on common game participant machines.

1) *Region Based Approach*: Region based approaches [3], [32] partition a game world into multiple regions, and assign each region a super-peer, which works as an authoritative server and hosts all the NPC objects within the region. For example, in [3] a live peer whose peerId is numerically closest to the regionId is selected as the "coordinator" for that region, and in [32] a "zone owner" is selected as the first peer that joins the zone.

These approaches have several significant drawbacks:

- Because only one super-peer is selected to take charge of a region, excessive computation and communication workloads might be incurred on the super-peer.
- Their super-peer selection criteria are overly simple, as they do not take into consideration peers' actual resource availabilities.
- These approaches cannot guarantee to fulfil the QoS requirement for game interactivity.

2) *Virtual Distance Based Approach*: The key idea of virtual distance based approaches [41], [42], [19] is to allocate a NPC to the player, whose avatar is closest to the NPC in a game world. Because a player that is closest to a NPC is most likely to interact with it, if the player is hosting the NPC by itself, there is no need for the player to communicate with a remote third party. It has been suggested that this approach is optimal for minimizing interactive latency and communication overhead [41].

Colyseus [41] has demonstrated the feasibility of virtual distance based object hosting in Quake II, a well-know multiplayer first-person shooter game. The game object manager of Colyseus allocates mutable objects, e.g. NPCs, doors and weapon items, to the closest players. Similarly, AtoZ [42] allocates each player avatar a "priority field", which is analogous to the Mahalanobis distance in the domain of quadratic discriminant analysis [43] to decide which player can access a shared object in the shortest time. Furthermore, the Voronoi diagram [19] discussed in section II-A1 seems inherently suitable for virtual distance based NPC host allocation, because it is natural for each player to host the NPC objects within its own Voronoi cell.

Compared to region based approaches, virtual distance based approaches are better at utilizing the computing resources of more participant machines. However, they also have the following disadvantages:

- Though it is likely that a player closest to a NPC will interact with the NPC, it does not mean that other nearby players will not. Contrarily, it is quite usual for a group of players to interact with the same NPC in a MMOG. In this case, all non-host players need to communicate with the host, and it is not guaranteed that the latency for each player is equally small.
- The computation of accurate NPC host allocation can be expensive, and because a large proportion of the players in a MMOG are constantly moving, switches of host

may be frequent. Therefore, the overall computation and communication overhead may be still high.

- Cheating may become easier for unscrupulous players who might abuse their hosting of NPC objects to their own advantage. Even worse, because no third party is required in a local interaction, it is rather hard to detect such a breach.

3) *Heterogeneous Task Sharing*: A heterogeneous task sharing mechanism [44] distributes NPCs to game participants according to the amount of computing resource and game interactivity that they can provide. Its system model involves three parties: a work source, a set of resource providers, and multiple matchmaker super-peers. The work source is the virtual game world that constantly generates NPC tasks. The resource providers are game participants, which have spare computing resources available on their machines. Finally, the matchmakers bridge between resource requirements and resource availability using distributed task allocation.

Compared to virtual distance based approaches, a task sharing mechanism is better at load-balancing, because it takes into consideration each game participant's actual resource availability. As a result, it ensures that NPC tasks are always allocated to capable hosts, which maximises the utility of a P2P system. Furthermore, once a NPC is allocated to a game participant, the hosting relationship remains stable, unless the NPC is destroyed, or the host needs to leave the system. Therefore, NPC task migration among hosts is less frequent than in virtual distance based approaches.

However, in some circumstances it would be possible to have special NPCs like shop owners that were only present to one player at a time. Such NPCs would be best hosted by a player's own machine, incurring minimal communication latency and overhead, which is unlikely to be the case in task sharing. So, in practice it could be appropriate to combine different NPC host allocation mechanisms together, so as to attain better flexibility and efficiency.

4) *NPC Host Allocation Discussion*: Currently, virtual distance based approaches are more widely used for NPC allocation because they minimise the communication latency for NPC hosts. However, heterogeneous task sharing mechanisms also have their merits, as they optimise overall communication latency when an NPC interacts with multiple players. Potentially P2P MMOGs could utilise both strategies flexibly in different game scenarios.

D. Game State Persistency

A MMOG is also referred to as a persistent world [45], because its game world is always available to the users and game plots evolve even while some of the players are not playing their characters. In this case, a MMOG must store all players' profiles and inventories between login sessions. When a player comes back to the game, the player can retrieve its previous state information and continue to play.

1) *Distributed Storage Infrastructures*: Several distributed storage infrastructures have been proposed in the literature, which may facilitate game state persistency in P2P MMOGs.

For example, the OceanStore project [46] provides a global persistent data store utility designed to scale to billions of users. It supports consistent, highly available and durable storage atop an infrastructure comprised of untrusted hosts.

Large-scale persistent storage services have also been built upon structured P2P overlay networks, e.g. PAST [47] that uses Pastry [30]. Compared to OceanStore, PAST is preferred by more P2P MMOG systems [4], [48], [41], [32], which have already employed Scribe [29] for game event dissemination, as described in section II-B1. It is fairly easy to combine support for Scribe and PAST at the same time.

2) *Further Considerations*: Though the distributed storage infrastructures discussed above support many of the required facilities, their suitability to be applied to a P2P MMOG directly is still in doubt. One of the considerations is the efficiency for reading and writing data through such infrastructures. A MMOG may generate event updates frequently, but it takes a long time to modify the data using an overlay network. In this case, there will be a severe delay when a player wants to retrieve data that is still being modified [32]. One solution is to make super-peers provide a caching mechanism, so that real-time event updates take effect immediately, whereas a distributed storage infrastructure is only used as a slow medium for backup purposes [32].

Availability is another significant consideration. It has been argued that data should always be available for retrieval, since players would not be satisfied if their characters were unavailable because the persons that were storing their characters were not playing the game [48]. This related work addresses the problem of separating storage needs into two categories: ephemeral and permanent data. The former can be stored using a distributed storage infrastructure, indexed by its geographical area in the virtual world. The latter should be stored locally, as it exists only when the player is in the game. Players can also backup their permanent data periodically using the distributed storage infrastructure in case their local data becomes corrupted.

3) *State Persistency Discussion*: Game state persistency is a major challenge for P2P MMOGs as existing P2P storage infrastructures are designed to support file sharing, and seldom fulfil the performance and security requirements of a MMOG. Compared with the previous three design issues that have been heavily researched, the persistency area is still immature with many problems waiting to be investigated.

E. Cheating Mitigation

The C/S architecture is easier to secure, because a server is able to validate every action request sent by a client before carrying it out. However, without the existence of such an authority, prevention of cheating becomes a challenging problem in P2P MMOGs. In the literature, some propose to prevent cheating from happening by reinforcing game event ordering and state exposure protocols, whereas others just aim at detecting and remedying inconsistent simulation results after suspicious game sessions. In this section, the

former are termed proactive approaches, and the latter reactive approaches.

1) *Proactive Approaches*: Generally speaking, there are two main ways for an unscrupulous player to cheat during a game session - unfair knowledge acquisition and unwanted event suppression. A player may gain extra advantages unfairly by peeking at other players' current status, which is especially useful in strategy games. Advanced information exposure protocols such as [49] can be applied to reinforce the fair payout of a game in these cases. A more significant security weakness in a P2P MMOG is "suppress-correct" cheating, which allows a player to gain an advantage by purposefully dropping update messages. Lockstep [50] is the first event-ordering protocol to address fixed-delay and timestamp cheats. Lockstep orders events by rounds and increments a round only after every player has committed its move for that round. A drawback of Lockstep is that the total ordering of events suffers from the largest delay between any two players. NEO [51] improves this design by bounding the length of each round with a maximum latency, where voting is used to form a consensus on whether a given player has sent an update within a round. However, the tradeoff is that a player who is slow to most nearby players will not be able to play in that area of the virtual world.

NEO requires event updates to be signed and encrypted before being sent to other players, so that a player cannot modify its own action after it has learned of others' actions. In order to achieve better performance, SEA [52] replaces NEO's encryption with a cryptographic hash function as the commitment method. Later on, a more efficient signature scheme EASES [53] further improves SEA by computing a message's digest before signing the message. Similarly, many other infrastructures have also been devised, e.g. FPS [54] and Hack-Proof [55], which actively minimize the opportunity for cheating by requiring players always to react upon the same game state information for every frame.

2) *Reactive Approaches*: Instead of applying sophisticated information exposure and event-ordering protocols, reactive approaches just aim at detecting unfair game payouts afterwards and rolling them back. For example, Log Auditing (LA) [56] partitions a game world into multiple regions, and a super-peer called a "region controller" (RC) is selected in each region. In LA, each player sends their commands as a signed sequence of packets, and the RC responds with signed game state updates. In this process, significant game events are logged by both the player and the RC. When given the same initial game state and player commands, the correct output can be reproduced by rerunning the log on a trusted machine, so that cheating can be detected. Likewise, [57] introduces multiple "monitor nodes" in each region, which calculate the latest game state from the previous game state respectively according to the game events that happened during the current timeslot. Hash values of game states are compared periodically, so as to identify potential cheats.

Both approaches mentioned above can be classified as referee based mechanisms, in which the selection of non-

colluding referees from untrusted peers is critical. Two secure referee selection algorithms have been proposed in [58], which not only emphasizes the fairness issue, but also the communication latency among referees and players.

DaCAP [59] and FreeMMG [60] rely on mutual monitoring among all players concerned, rather than a limited number of referees. They organise players into "legal groups" according to their locality in the game world. All members of a legal group have to compute and record all actions and status of the other members in the same group. Once cheating behaviour is detected, the cheating player is reported to a "check server" with related evidence. To make collaboration to falsify player data hard, DaCAP randomly chooses a number of players from other areas of the game world to join the group, while FreeMMG allocates each group a server simulated player that can always be trusted.

Last but not least, a novel behavioural monitoring mechanism has been proposed in [61]. This approach differs from any other methods in that it does not rely on knowledge about specific vulnerabilities and their method of exploitation in order to protect the system. Instead, it relies on the real-time monitoring of players' movements and behaviours in the game world for indications of cheating play. This concept is based on the hypothesis that players engaged in cheating will exhibit characteristic behaviour which is clearly distinguishable from normal play.

3) *Cheating Mitigation Discussion*: It is widely accepted that P2P MMOGs are more difficult to secure than conventional C/S architectures. Moreover, security issues present themselves at all stages in the design and implementation of P2P MMOGs. Hence it is reassuring to see the intensive research into proactive and reactive cheating mitigation mechanisms starting to bear fruit.

F. Incentive Mechanisms

P2P applications are by nature voluntary resource sharing systems, in which there is often a tension between individual concerns and collective welfare. As the benefits of these systems are rooted in cooperation, they are inherently vulnerable to non-cooperative behaviour, and it is necessary for such systems to be designed so that participants are induced to cooperate. The mechanisms that are embedded in the system for this purpose are called Incentive Mechanisms [62]. A P2P MMOG requires an incentive mechanism to convince its participants to contribute their resources. For example, network bandwidth is needed in game event dissemination, storage capacity is needed in game state persistency, and CPU cycles are needed in interest-management, NPC hosting and cheating mitigation, as discussed in previous sections.

1) *Accounting Mechanisms*: An accounting mechanism maintains the viability of a P2P MMOG by quantifying the amount of computing resource a player has contributed to the system. On the one hand, it keeps a record of a player's historical contribution, and on the other hand it entitles the player to consume roughly equivalent resources from other players. In this way, selfish players can be identified and discouraged,

| P2P MMOG Architectures | Interest Management | Game Event Dissemination | NPC Host Allocation | Game State Persistency | Incentive Mechanism | Overall Evaluation |
|------------------------|---------------------|--------------------------|---------------------|------------------------|---------------------|--------------------|
| P2P Support '04 [3] | Region-based | ALM | Region-based | None | None | Simple |
| Distributed '04 [31] | Region-based | Unicast | Distance-based | Distributed | None | Moderate |
| OPeN '05 [5] | AOI-based | Unicast | None | Centralized | None | Moderate |
| P2P Arch '06 [4] | Region-based | ALM | None | PAST | None | Simple |
| VAST '07 [69] | Voronoi | Unicast | Distance-based | Centralized | REPS | Complete |
| Mediator '07 [44] | Hybrid | Unicast | Task Sharing | Distributed | DCRC | Complete |

TABLE I
COMPARISON OF REPRESENTATIVE P2P MMOG ARCHITECTURES

and a sufficient level of reciprocity can be ensured to make use of a P2P MMOG beneficial.

DCRC [63] is a fully distributed accounting system that applies to general P2P applications. The key idea in DCRC is a Debit/Credit platform using a virtual currency. By tracking a user's activities in a P2P system, DCRC bills the user according to the amount of resources that the user has consumed (i.e. Debit), and rewards the user according to the time and quality of a service that the user has offered to others (i.e. Credit). A user that stays in credit for long can be further encouraged in many different ways, e.g. service quality differentiation and application-specific privileges.

2) *Reputation Mechanisms*: Merely quantifying a peer's contribution to a P2P system is sometimes inadequate in discouraging certain disadvantageous behaviours. For example, a player may have worked as a super-peer in a region for a long time and have contributed a lot of resources to the application. However, the player may also disconnect from the system abruptly when it decides to leave, and thus put the system into an inconsistent state which takes much time and inconvenience to recover from. In this case, a reputation mechanism becomes valuable for qualifying a peer's dependability, honesty and overall manner towards P2P collaborations.

Many distributed reputation management systems can be used in a P2P MMOG. EigenTrust [64] and REPS [65] are representatives of mutual rating based approaches. In these systems, after each interaction peers produce either positive or negative feedbacks for each other, and keep the feedbacks in their own storage. Reputation query algorithms are provided for a peer to aggregate such local trust values from its direct friends, friends of friends, or arbitrary numbers of unacquainted peers, so that the peer can estimate approximately the trustworthiness of any other stranger peer.

In contrast, approaches like proactive [66] and local [67] reputation do not depend on ratings from third parties. Instead, they provide various means for a peer to evaluate the trustworthiness of a target peer directly, hence they are inherently immune to bad-mouthing or collusion attacks. These approaches focus on addressing the challenge of anonymous reputation requests inside application traffic, because once an unscrupulous peer determines that the purpose of a request is to measure its reliability, it will be tempted to process these requests to boost its reputation.

3) *Incentive Discussion*: The success of a P2P MMOG relies on an effective incentive mechanism that facilitates

the collection of resources, and a reputation mechanism that minimises antisocial behaviours. Such incentive mechanism are a key design issue that is often unjustly ignored in the literature. As we shall see in the next section some P2P MMOG infrastructures may need to improve their incentive mechanisms to make them more practical.

III. COMPARISON OF P2P MMOG ARCHITECTURES

In this section, six representative P2P MMOG architectures are selected and compared. Table I summarizes their features and illustrates how they address the essential issues discussed previously. However, cheating mitigation is not included in this table, because it is a relatively separate issue, and all the architectures are potentially compatible with existing cheating mitigation techniques.

a) *P2P Support '04 [3]*: This early architecture partitions a virtual game world into multiple regions, and interest-management is carried out by region-based publish/subscribe. Each region is associated with a publish channel, to which all region participants subscribe, and gaming events are delivered using Scribe. A super-peer called a "region coordinator" is selected in each region, which hosts all the NPC objects in that region.

While representing a number of good design decisions, this architecture is evaluated as *simple* for the following reasons. The IM scheme is coarse-grained and its event dissemination relies on a general purpose ALM middleware, which may induce high communication latency. Also, its NPC host allocation mechanism is intuitive, and neither specific game state persistency nor incentive mechanisms are provided.

b) *Distributed '04 [31]*: This architecture adopts a region-based interest management scheme, where the size of each region is quite small. In this case, the number of players in a region is limited, so players in the same region can communicate with each other using unicast. However, it is likely that a gaming event that takes place in one region may also affect the players in neighbouring regions. Therefore, a super-peer is selected in each region to propagate local gaming events to neighbouring super-peers when necessary. Furthermore, the architecture suggests each player storing its own permanent data, while public ephemeral data is stored by a Distributed Hash Table [68].

The architecture is evaluated as *moderate*, because its IM, event dissemination and state persistency mechanisms are well designed. However, it assumes that there are always adequate

peers donating computing resources, and thus when a NPC becomes active, a random capable peer is selected to host that NPC. Such NPC host allocation and incentive mechanisms are overly simple.

c) *OPeN '05 [5]*: The OPeN architecture proposes a distributed spatial data index service, which is built on top of a structured P2P overlay network. With this service, players can register their current locations in a game world, and query about other entities in their AOIs. Nearby entities establish direct UDP packet flows with each other in order to exchange gaming events. Persistent game data is stored and managed by a centralized database server.

The architecture is evaluated *moderate*, because its IM and event dissemination mechanisms seem to be adequate. However, the infrastructure still depends on game servers for state persistency, and NPC host allocation and incentive mechanisms are not supported.

d) *P2P Arch '06 [4]*: This architecture is purely Pastry based, as it uses Scribe for game event dissemination, and PAST for game state persistency. The architecture employs coarse-grained region-based interest-management, but does not provide details about NPC host allocation and incentive mechanisms.

The architecture is evaluated as *simple*, because it directly employs ALM and distributed storage middlewares built on top of Pastry. As these middleware components are designed for general P2P applications their performances may not be adequate for a P2P MMOG. No arguments are provided for the suitability of the middleware nor demonstrations of its effectiveness.

e) *VAST '07 [69]*: A unique Voronoi assisted interest-management mechanism is employed in the VAST project. The Voronoi diagram is also inherently suitable for virtual distance based NPC host allocation. Furthermore, the architecture provides a native incentive mechanism as well.

Though currently the architecture still requires a game server for peer bootstrapping, load-balancing, fault-tolerance and state persistency purposes, it is evaluated as *complete*, because its application of the Voronoi technology is quite remarkable, which offers a consistent way of fine-grained IM, efficient game event dissemination and convenient NPC host allocation.

f) *Mediator '07 [44]*: The Mediator framework is the authors' ongoing research. The framework employs a hybrid IM scheme like MOPAR [15] and disseminates gaming events through unicast communication. A novel task mapping mechanism, Deadline-Driven Auctions (DDA), is devised to support the sharing of real-time NPC tasks. DDA is inherently compatible with reactive cheating mitigation approaches, e.g. Log Auditing [56], and also supports a DCRC-like [63] incentive mechanism that motivates application participants to contribute their resources to the system. Furthermore, a membership-aware multicast mechanism (MAMBO) [70] is developed for maintaining game zone infrastructures efficiently. It is also convenient to support game state persistency using PAST [47], as both PAST and MAMBO use the same overlay network.

The architecture is evaluated as *complete*, because it addresses all of the design issues surveyed by this paper. In particular, its NPC host allocation mechanism is good at minimising communication latency among NPC hosts and ordinary players.

IV. CONCLUSION & FUTURE WORK

To adapt MMOGs from conventional C/S architectures to P2P architectures is a challenging and active research area. This paper articulates a comprehensive set of six key issues for the design of P2P MMOGs. Design alternatives for each issue are systematically compared, the relationships between design decisions are discussed, and areas for further research are identified (Section II). Finally, representative P2P MMOG infrastructures are evaluated against the design issues (Section III).

The authors are currently refining the Mediator framework (Section III) and intend to evaluate it for a prototype MMOG in the near future.

REFERENCES

- [1] J. Mulligan and B. Patrovsky, *Developing Online Games - An Insiders Guide*. New Riders Pub., ISBN: 1592730000, 2003.
- [2] B. S. Woodcock, "An analysis of MMOG Subscription Growth," www.mmogchart.com, 2005.
- [3] H. Lu, B. Knutsson, W. Xu & B. Hopkins, "P2P Support for Massively Multiplayer Games," in *Proc. of INFOCOM*. IEEE, 2004, pp. 7–11.
- [4] T. Hampel, T. Bopp & R. Hinn, "A P2P Architecture for Massively Multiplayer Online Games," in *Proc. of NetGames*. ACM, 2006.
- [5] S. Douglas, E. Tanin, and A. Harwood, "Enabling Massively Multi-Player Online Gaming Applications on a P2P Architecture," in *Proc. of ICIA*. IEEE, 2005, pp. 7–12.
- [6] J.-S. Boulanger, J. Kienzle, and C. Verbrugge, "Comparing Interest Management Algorithms for Massively Multiplayer Games," in *Proc. of NetGames*. ACM, 2006.
- [7] G. Morgan, F. Lu, and K. Storey, "Interest Management Middleware for Networked Games," in *Proc. of i3D*. ACM, 2005.
- [8] G. Schiele, R. Suselbeck, A. Wacker, J. Hahner, C. Becker, and T. Weis., "Requirements of Peer-to-Peer-based Massively Multiplayer Online Gaming," in *Proc. of CCGRID*. IEEE, 2007, pp. 773–782.
- [9] J. Simon, W. Klaus, F. Marc, N. Heiko, P. Leo, and C. Georg, "Peer-to-Peer-Based Infrastructure Support for Massively Multiplayer Online Games," in *Proc. of CCNC*. IEEE, 2007, pp. 763–767.
- [10] M. M., Z. M., P. D., B. P., and Z. S., "NPSNET: A Network Software Architecture for Large Scale Virtual Environments," *Presence*, vol. 3, no. 4, pp. 265–287, 1994.
- [11] S. Benford and L. E. Fahlen, "A Spatial Model of Interaction in Large Virtual Environments," in *Proc. of ECSCW*. IEEE, 1993, pp. 107–123.
- [12] N. Matsumoto, Y. Kawahara, H. Morikawa, and T. Aoyama, "A Scalable and Low Delay Communication Scheme for Networked Virtual Environments," in *Proc. of GLOBECOM*. IEEE, 2005, pp. 529–535.
- [13] E. Buyukkaya and M. Abdallah, "Data Management in Voronoi-Based P2P Gaming," in *Proc. of CCNC*. IEEE, 2008, pp. 1050–1053.
- [14] S.-Y. Hu and G.-M. Liao, "Scalable Peer-to-Peer Networked Virtual Environment," in *Proc. of NetGames*. ACM, 2004, pp. 129–133.
- [15] A. P. Yu and S. T. Vuong, "MOPAR: a Mobile Peer-to-Peer Overlay Architecture for Interest Management of Massively Multiplayer Online Games," in *Proc. of NOSSDAV*. ACM, 2005, pp. 99–104.
- [16] A. E. Rhalibi & M. Merabti, "Interest Management & Scalability Issues in P2P MMOG," in *Proc. of CCNC*. IEEE, 2006, pp. 1188–1192.
- [17] B. B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network," in *Proc. of ICDE*. IEEE, 2003, pp. 49–60.
- [18] K. Storey, F. Lu, and G. Morgan, "Determining Collisions between Moving Spheres for Distributed Virtual Environments," in *Proc. of CGI*. IEEE, 2004, pp. 140–147.
- [19] S.-Y. Hu, S.-C. Chang, and J.-R. Jiang, "Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games," in *Proc. of CCNC*. IEEE, 2008, pp. 1134–1138.

- [20] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems*, vol. 8, no. 2, pp. 85–110, 1990.
- [21] S. Fiedler, M. Wallner, and M. Weber, "A Communication Architecture for Massive Multiplayer Games," in *Proc. of NetGames*. ACM, 2002, pp. 14–22.
- [22] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network*, vol. 14, pp. 78–88, 2000.
- [23] A. El-Sayed, V. Roca, and L. Mathy, "A Survey of Proposals for an Alternative Group Communication Service," *IEEE Network*, vol. 17, no. 1, pp. 46–51, 2003.
- [24] S. Q. Zhuang, B. Y. Zhao & A. D. Joseph, "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination," in *Proc. of NOSSDAV*. ACM, 2001, pp. 11–20.
- [25] B. Zhao, J. Kubiawicz, and A. Joseph, "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing," UC Berkeley, Tech. Rep., 2001.
- [26] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Application-Level Multicast Using Content-Addressable Networks," in *Proc. of NGC*, Nov. 2001, pp. 14–29.
- [27] S. Ratnasamy, P. Francis, M. Handly, R. Karp, and S. Schenker, "A scalable content-addressable network," in *Proc. of SIGCOMM*. ACM, 2001, pp. 161–172.
- [28] R. Zhang and Y. C. Hu, "Borg: A Hybrid Protocol for Scalable Application-Level Multicast in Peer-to-Peer Networks," in *Proc. of NOSSDAV*. ACM, 2003, pp. 172–179.
- [29] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron, "SCRIBE: A Large-scale and Decentralized Application-level Multicast Infrastructure," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 8, pp. 1489–1499, 2002.
- [30] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location and Routing for Large Scale Peer-to-Peer Systems," in *Proc. of Middleware*. ACM, 2001, pp. 329–350.
- [31] C. G. Dickey, D. Zappala, and V. Lo. "A Fully Distributed Architecture for Massively Multiplayer Online Games," in *Proc. of NetGames*. ACM, 2004, p. 171.
- [32] T. Iimura, H. Hazeyama, and Y. Kadobayashi, "Zoned Federation of Game Servers - a Peer-to-peer Approach," in *Proc. of NetGames*. ACM, 2004, pp. 116–120.
- [33] S. Yamamoto, Y. Murata, K. Yasumoto, and M. Ito, "A Distributed Event Delivery Method with Load Balancing for MMORPG," in *Proc. of NetGames*. ACM, 2005, pp. 1–8.
- [34] S. Rooney, D. Bauer & R. Deydier, "A Federated P2P Network Game Architecture," *Communications*, vol. 42, no. 5, pp. 114–122, 2005.
- [35] F. Chen and V. Kalogeraki, "Adaptive Real-Time Update Dissemination in Distributed Virtual Simulation Environments," in *Proc. of ISORC*. IEEE, 2005, pp. 233–236.
- [36] E. Léty, T. Turletti, and F. Baccelli, "SCORE: A Scalable Communication Protocol for Large-Scale Virtual Environments," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 247–260, 2004.
- [37] S. Xiang-bin, W. Yue, L. Qiang, D. Ling, and L. Fang, "An Interest Management Mechanism Based on N-Tree," in *Proc. of SNPD*. IEEE, 2008, pp. 917–922.
- [38] C. GauthierDickey, V. Lo, and D. Zappala, "Using N-Trees for Scalable Event Ordering in Peer-to-Peer Games," in *Proc. of NOSSDAV*. ACM, 2005, pp. 87–92.
- [39] A. Schmiege, M. Stieler, S. Jeckel, P. Kabus, B. Kemme, and A. Buchmann, "pSense - Maintaining a Dynamic Localized Peer-to-Peer Structure for Position Based Multicast in Games," in *Proc. of P2P*. IEEE, 2008, pp. 247–256.
- [40] D. T. Ahmed and S. Shirmohammadi, "An Expedite State Dissemination Mechanism for MMOGs," in *Proc. of i-span*. IEEE, 2008, pp. 199–203.
- [41] A. Bharambe, "Colyseus: A Distributed Architecture for Online Multiplayer Games," in *Proc. of NSDI*. USENIX, 2006, pp. 3–6.
- [42] T. Yonekura, Y. Kawano, and D. Hanawa, "Peer-to-peer networked field-type virtual environment by using atoz," in *Proc. of CW*. IEEE, 2004, pp. 241–248.
- [43] T. W. Anderson, *An Introduction to Multivariate Analysis 2nd Edition*. John Wiley & Sons, 1984.
- [44] L. Fan, H. Taylor, and P. Trinder, "Mediator: A Design Framework for P2P MMOGs," in *Proc. of NetGames*. ACM, 2007, pp. 43–48.
- [45] D. James and G. Walton, "2004 Persistent Worlds Whitepaper," IGDA Online Games SIG, www.igda.org, 2004.
- [46] J. Kubiawicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummedi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "OceanStore: An Architecture for Global-Scale Persistent Storage," in *Proc. of ASPLOS*. ACM, 2000, pp. 190–201.
- [47] A. Rowstron and P. Druschel, "Storage Management and Caching in PAST, A Large-scale, Persistent Peer-to-peer Storage Utility," *Symposium on Operating Systems Principles*, pp. 188–201, Oct. 2001.
- [48] M. Assiotis and V. Tzanov, "A Distributed Architecture for MMORPG," in *Proc. of NetGames*. ACM, 2006.
- [49] C. Chambers, W. chang Feng, W. chi Feng, and D. Saha, "Mitigating Information Exposure to Cheaters in Real-Time Strategy Games," in *Proc. of NOSSDAV*. ACM, 2005, pp. 7–12.
- [50] N. E. Baughman, M. Liberatore, and B. N. Levine, "Cheater-Proof Payout for Centralized and Peer-to-Peer Gaming," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 1–13, 2007.
- [51] C. GauthierDickey, D. Zappala, V. Lo, and J. Marr, "Low Latency and Cheater-Proof Event Ordering for Peer-to-Peer Games," in *Proc. of NOSSDAV*. ACM, 2004, pp. 134–139.
- [52] M. Gorawski & K. Stachurski, "A Secure Event Agreement Protocol for Peer-to-Peer Games," in *Proc. of ARES*. IEEE, 2006, pp. 34–41.
- [53] M.-C. Chan, S.-Y. Hu & J.-R. Jiang, "An Efficient and Secure Event Signature Protocol for Peer-to-Peer Massively Multiplayer Online Games," *Computer Networks*, vol. 52, no. 9, pp. 1838–1845, 2008.
- [54] B. D. Chen and M. Maheswaran, "A Fair Synchronization Protocol with Cheat Proofing for Decentralized Online Multiplayer Games," in *Proc. of NCA*. IEEE, 2004, pp. 372–375.
- [55] Y. S. Fung, "Hack-Proof Synchronization Protocol for Multi-Player Online Games," in *Proc. of NetGames*. ACM, 2006.
- [56] P. Kabus, W. Terpstra, M. Cilia & A. Buchmann, "Addressing Cheating in Distributed MMOGs," in *Proc. of NetGames*. ACM, 2005, pp. 1–6.
- [57] T. Izaiku, S. Yamamoto, Y. Murata, N. Shibata, K. Yasumoto, and M. Ito, "Cheater Detection for MMORPG on P2P Environments," in *Proc. of NetGames*. ACM, 2006.
- [58] S. D. Webb, S. Soh, and J. Trahan, "Secure Referee Selection for Fair and Responsive Peer-to-Peer Gaming," in *Proc. of PADS*. IEEE, 2008, pp. 63–71.
- [59] H.-I. Liu and Y.-T. Lo, "DaCAP - A Distributed Anti-Cheating Peer to Peer Architecture for Massive Multiplayer On-line Role Playing Game," in *Proc. of CCGRID*. IEEE, 2008, pp. 584–589.
- [60] F. R. Cecin, R. Real, R. de Oliveira Jannone, C. F. R. Geyer, M. G. Martins, and J. L. V. Barbosa, "FreeMMG: A Scalable and Cheater-Resistant Distribution Model for Internet Games," in *Proc. of DS-RT*. IEEE, 2004, pp. 83–90.
- [61] P. Laurens, R. F. Paige, P. J. Brooke, and H. Chivers, "A Novel Approach to the Detection of Cheating in Multiplayer Online Games," in *Proc. of ICECCS*. IEEE, 2007, pp. 97–106.
- [62] M. Zghaibeh & K. Anagnostakis, "On the Impact of P2P Incentive Mechanisms on User Behavior," in *Proc. of NetEcon*. ACM, 2007.
- [63] M. Gupta, P. Judge, and M. Ammar, "A Reputation System for Peer-to-Peer Networks," in *Proc. of NOSSDAV*. ACM, 2003, pp. 144–152.
- [64] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks," in *Proc. of WWW*. ACM, 2003, pp. 640–651.
- [65] G.-Y. Huang, S.-Y. Hu, and J.-R. Jiang, "Scalable Reputation Management for P2P MMOGs," in *Proc. of MMVE*. IEEE, 2008.
- [66] G. Swamynathan, B. Y. Zhao, and K. C. Almeroth, "Exploring the Feasibility of Proactive Reputations," *Concurrency and Computation: Practice & Experience*, vol. 20, no. 2, pp. 155–166, 2008.
- [67] S. Liu, J. Li, and X. Wang, "Local Reputation for P2P MMOG Design," in *Proc. of PDCAT*. IEEE, 2007, pp. 523–528.
- [68] I. Stoica, R. Morris, D. Karger, and F. Kaashoek, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proc. of NetGames*. ACM, 2001, pp. 149–160.
- [69] S.-Y. Hu, J.-F. Chen, and T.-H. Chen, "VON: A Scalable Peer-to-Peer Network for Virtual Environments," *IEEE Network*, vol. 20, no. 4, pp. 22–31, 2006.
- [70] L. Fan, P. Trinder, and H. Taylor, "MAMBO: Membership-Aware Multicast with Bushiness Optimisation," in *Short Paper for DEBS*. DEBS, 2008.