

A NOVEL SCALABLE HYBRID ARCHITECTURE FOR MMOG

Christopher James Carter, Abdennour El Rhalibi, Madjid Merabti

Liverpool John Moores University
Byrom Street, L3 3AF, Liverpool, UK
{[C.J.Carter](mailto:C.J.Carter@ljmu.ac.uk); [A.ElRhalibi](mailto:A.ElRhalibi@ljmu.ac.uk); [m.merabti](mailto:m.merabti@ljmu.ac.uk)}@ljmu.ac.uk

ABSTRACT

We present a novel MMOG Hybrid P2P architecture and detail its key components, topology and protocols. We highlight the main components which lie at the heart of the proposed solution, and their roles, and describe the methods of tackling the key scenarios which are faced by the architecture during gameplay. For each role, we discuss the interactions that exist between them and describe the protocols that will be used for inter-role communication to perform the atomic actions necessary for maintaining the consistency and responsiveness of an MMOG such as peer addition, peer removal, group transfer, object change persistency and many more. We conclude the chapter with a comparison of the architecture against several existing P2P MMOG frameworks, discussing the differences which exist between them and how the novel Hybrid-P2P architecture we propose aims to address their flaws.

Index Terms — Online Games, Peer-to-Peer, Scalability

1. INTRODUCTION

The development of a modern Massively Multiplayer Online Game (MMOG) is a challenging and costly process [3][4]. MMOG are very large distributed applications, sharing substantial state data, and supporting communication between potentially thousands of player nodes. Despite the development of many solutions to define suitable architecture and communication protocols and enabling efficient deployment of these types of applications, many issues remains which still require a solution [4]. One of the fundamental challenges for MMOG developers is the need for an MMOG solution to accommodate significant growth in player numbers, without adversely affecting the performance, interactivity and ultimately, the enjoyment of the gaming experience [4]. In essence, the core challenge in developing a successful, lasting MMOG is finding workable solutions to the issue of scalability within such systems [4].

In this paper we present a novel MMOG Hybrid P2P architecture and detail its key components and algorithms. We highlight the merits of utilizing a Hybrid-P2P system in comparison with C/S and decentralized P2P models, in order to allow elements of centralization to be utilized for tertiary tasks within the framework, such as peer discovery, authentication and persistency, whilst relying on semi-structured P2P overlay networks to perform state

dissemination and the adaptive reorganization of the network.

The remaining of the paper is as follows: in section 2 we discuss the rationale of using a hybrid-P2P architecture; in section 3 we introduce our hybrid P2P architecture NAHPALM; in section 4 we address the key scenarios using our hybrid-P2P; in section 5 we contrast our work with state-of-the-art P2P MMOG architectures; and in section 6 we conclude our paper, and introduce the future work.

2. RATIONALE FOR A HYBRID-P2P

The current Client-Server model of MMOGs [4] allows platform providers to track players within the game; control the software used to connect into the game; administer users who are breaking terms and conditions; offer them advertising; utilize Web 2.0 constructs such as integration with social networking sites; store payment details and so forth. The primary reasons for adopting a different architecture to Client/Server is to alleviate the inflexibility and financial implications that are associated with the required increase of computational and networking resources in order to deal with the demands of real-time processing and synchronicity of the game-state. By eliminating centralization and prohibiting the inclusion of these existing revenue streams associated with C/S MMOGs, a fully de-centralized P2P scheme will impact on income and therefore the financial benefits become a trade-off between maintenance costs and additional revenue, instead of an inherent gain. In addition, the operation of a server for the process of authentication, authorization, and content distribution would not suffer from the same demands as a fully centralized C/S architecture, since these actions are infrequent; do not require real-time interactivity; and have much smaller data-transfer requirements. These aspects in addition to the strengths and weaknesses discussed in [4], informed the decision to design a hybrid architecture.

3. NOVEL HYBRID P2P ARCHITECTURE

The Hybrid Architecture aims to bind a loosely-coupled centralization layer to a hierarchical, semi-structured P2P overlay network for the construction of MMOG applications. The system has been codenamed “NAHPALM”. In order to address the effects of resource limitations and heterogeneity, the player’s roles within the topology and the connections that exist amongst participants

are determined based on both their resource contributions to the distributed environment and their current game play scenario. NAHPALM addresses the issues of local computational efficiency, by sharing the computational load of AoI calculation with existing game engine processing from other subsystems used in collision detection and rendering. NAHPALM aims to balance consistency and responsiveness through the re-organization of topologies and roles within the peer groups as players interact within the virtual world and resources are dynamically added/removed at runtime. NAHPALM involves four separate component systems, which are distributed among the centralized server and the game clients of the peers, (see Figure 1).

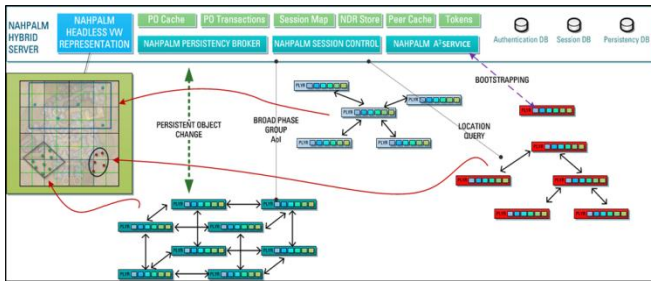


Figure 1: Hybrid-P2P: Component Architecture

3.1. Hybrid-P2P Architecture Components

This section details the specifics of each of the four components which comprise NAHPALM.

3.1.1. Using Semi-Structured Overlays for P2P Communications

NAHPALM constructs a hierarchical network composed of several sub-graphs formed by peer groups. Sub-graphs are loosely connected via the services provided by the Hybrid server. Peer groups are determined via the initial bootstrapping of the virtual world, where initial entry is assigned via the Hybrid Server and via the aforementioned AoI techniques. Unlike Spatial AoI-based architectures such as VAST/VON [3][6], which relies on the AoI to determine the topology of the network in an unstructured fashion, NAHPALM imposes a degree of order, by employing a variety of traditional network topologies. The initial design of NAHPALM incorporates four distinct topologies for sub-graph connectivity: Hub and Spoke (Star), Multi-Hub and Spoke, Tree and Hypercube. Each topology has been extensively researched within the domain of network theory and their properties are well known [7]. Each of these topologies requires utilization of roles in order to assign nodes to a particular position inside the topology.

3.1.2. Adaptive Role Switching Mechanisms

In order to apply the hierarchical overlays defined in section 3.1.1, NAHPALM cannot rely on homogeneous processing among peers and thus uses a dynamic role assignment system, where each peer adopts a role within their group. This can be seen as a form of Super Peer system, where peers who contribute greater computational resource and

efficient networking capabilities are given more tasks. NAHPALM defines six high-level roles in the network:

Peer Group Coordinator (PGC): The PGC is the primary and most resource intensive role within the network. The PGC is responsible for maintaining the overlay topology. The coordinator determines which topology is suited to the current gameplay scenario and peer group and collates the AoI information detailing the virtual world coverage of the group. Each peer group must only contain a single PGC, and is considered authoritative within the group.

Gameplay State Coordinator (GSC): The GSC is responsible for the dissemination of gameplay messages between peers. A GSC is informed by the PGC as to which peers within the group it is responsible for updating and coordinating. Depending on the topology applied, there may be more than one GSC within the network.

State Replication Coordinator (SRC): The SRC is responsible for acting as a backup for the GSC. The SRC needs awareness of all peers coordinated by the GSC and is responsible for determining whether the GSC is still live. The SRC will become the GSC should the existing GSC leave the group (planned/unplanned).

Role Allocation Coordinator (RAC): The RAC is responsible for ensuring that all roles are fulfilled within the group and initiates any transference of roles between peers.

Persistent Object Coordinator (POC): The POC is responsible for managing all mutable persistent objects which are relevant to the entire group. The POC liaises with the hybrid server in order to record any changes to the state persistence within the area of the virtual world inhabited by the group. The POC reacts to changes made by the GSC.

Generic Personal Coordinator (GPC): The GPC is the least intensive role within the group. The GPC is expected to communicate any updates with the GSC and apply the resulting state changes to their local view of the application. This role is analogous to being an edge peer within a traditional super peer-based network [4].

Roles are assigned based on the topology and resources available with the peer group. Each peer is assigned a NAHPALM Diagnostic Report (**NDR**), initially computed by the Hybrid Server. This models both the computational and network resources available at each peer. Each Peer maintains a local view of the other group member's NDR and updates it based on their own experiences (via internal log analysis). Further clarification regarding the expected tasks performed by each role and the relationships which exist between them is discussed in section 4.

3.1.3. Usage of the Hybrid Server

We introduce the duties expected of the Hybrid Server (HS) component of the NAHPALM framework. The HS is designed to imply a flexible SOA method for incorporating

centralized tasks into the MMOG environment. These services address several key scenarios:

Initial Peer Discovery and Virtual World Bootstrapping: As a player enters their particular game’s virtual environment their credentials are authenticated via service requests to the HS. The player enters the game as described in section 4.1.

Initial Peer Performance Diagnostics: When a player starts the NAHPALM client, a test is made to run the application with all subsystems except the Networking Layer, with a static memory pool allocation test to determine the profile of the machine running the application. The results are serialized as key-value pairs which report the local resources that can be provisioned. At the bootstrapping phase, a bandwidth and RTT test is done to determine the peer’s initial performance ratio. This is combined with the performance test to produce the initial NDR.

Group Approximation Queries: NAHPALM constructs a series of peer groups, which hierarchically form the overall semi-structured overlay network. The PGC periodically reports this information to the Hybrid Server, and maintains this information inside the resident Session Cache. Other PGCs can query the rough locations of the group in order to determine the nearest groups within the world. This information is next used to transfer a particular participant between groups as they interact with the world. As groups expand, the PGC has the ability to merge/split a peer group; this information is then communicated to the other group managers via the approximation queries.

Persistency Brokering: In order to keep consistency, an element of centralized persistency is added. A mapping of persistent game objects is maintained on the server, using a transactional access scheme using an in-memory cache, backed by a permanent storage system. As players are authenticated they are sent an updated version of their cache (e.g. weapons, inventory etc.). Any items which are shared amongst participants within the world and are subject to change are modified through the application of transactions which have occurred whilst the player is offline. The peers responsible for group persistence consistency periodically transfer game object changes to the HS, which wraps them as a transaction and applies them to the cache.

3.1.4. A Two-Phase Approach to AoIM

NAHPALM utilizes a two-phase approach to determining area of interest. The first phase involves a broad-phase mechanism using a flexible hierarchical geographic AoI scheme [4]. This is utilized on a peer group basis to determine the bounding extents which are encapsulated by a group within the virtual world and identify surrounding groups which may be of interest as participants, as they navigate the environment. Peer groups utilize a narrow-phase spatial AoI scheme to determine the AoI relationship between the group’s members using a suite of generic bounding volumes. A hierarchical bottom-up approach is

used to merge bounds in order to construct the overall extents of the group, for use with the broad-phase algorithm.

4. KEY SCENARIOS USING HYBRID-P2P

We present how the NAHPALM architecture is designed to handle key scenarios which arise during the course of game execution.

4.1. Player Bootstrapping and Initial Addition

The first aspect is the player authentication process, where the initial conditions for entry are raised and the player is assigned their unique identifying credentials by the HS, (see Figure 1).

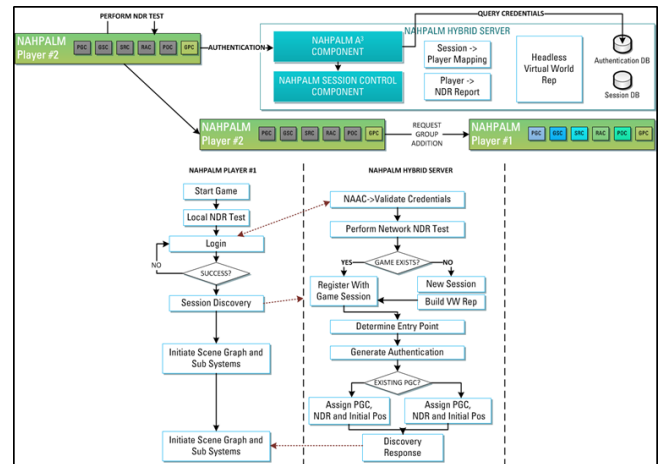


Figure 1: Bootstrapping Process

There are three major scenarios: If a player is the initial participant then HS builds its own view of virtual world, which is held in-memory and constructs mapping systems for storing player credentials and NDR reports, initializes the persistency database and registers the player. The player’s initial entry point is determined and is assigned all six roles to begin with. The player then enters the game. If the game is already in place, the player is registered with the session. If the entry point is located in an area with no PGC, the player is assigned all six roles. If a PGC is in place, the player is given the discovery credentials of this player and is informed that this player must be contacted to start into the game. In all cases, an entry confirmation is sent to HS.

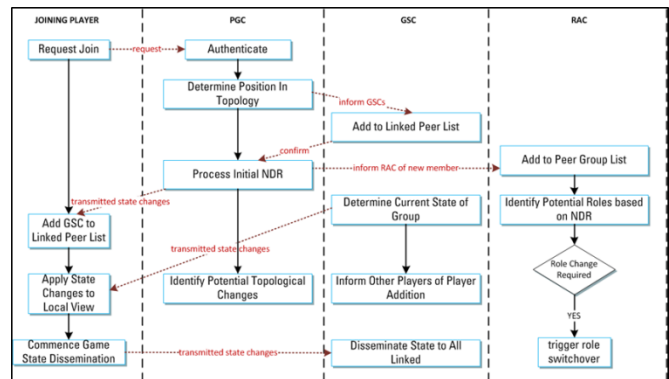


Figure 2: Peer Group Addition

4.2. Player -> Peer Group Addition

This scenario occurs when a player first joins the game and is told to join an existing peer group, (see Figure 2).

Usually a player will partake for a few ticks before a RAC consider the role change. The other case for addition is when player wish to transfer from one peer group.

4.3. Player -> Peer Group Removal

This scenario occurs when a player wishes to leave the peer group, (see Figure 3).

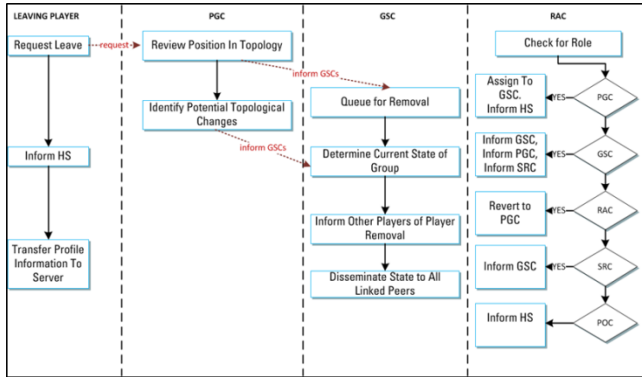


Figure 3: Peer Group Removal

The RAC plays the most important role, determining the effects on the group. The two most severe cases are if it is a PGC or GSC that is leaving, due to their importance. The GSC is the best candidate for PGC to assume this role since it is topology-aware. The GSC has a backup mechanism in place with the SRC, who is swapped into the GSC role. The RAC's role reverts to the GSC, since this requires no state transference as the PGC is aware of role assignments. The HS is informed by the ROC in the cases of PGC and POC removal, in order to update the session and persistence data.

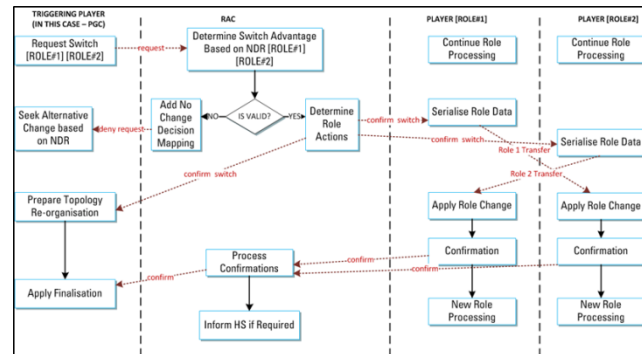


Figure 4: Role Transference

4.4. Role Transference

Role transference can be triggered by the PGC, GSC and RAC based on topological modifications, this is due to players with better NDRs; load balancing requests due to poor performance; and peer removal, (see Figure 4).

4.5. Peer Group Transference

As player's traverse the virtual world, they will require exchange between peer groups. NAHPALM is designed to

proactively detect transference, based on the broad phase AoIM system, with assistance from the HS, (see Figure 5).

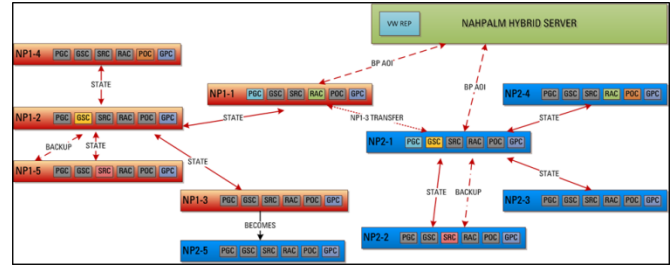


Figure 5: Peer Group Transference

The HS has identified that Groups 1 and 2's Broad Phase AoI bounds are close to intersecting and proceed to inform each PGC of the two groups. They initiate an exchange which defines that Peer 3 in Group 1 will need to be moved. The PGC of group 1 (NP1-1) starts the removal of Peer 3 and starts the process described in 4.3. Also, the PGC of group 2 (NP 2-1) triggers the addition of this peer as NP2-5 (as described in 4.2).

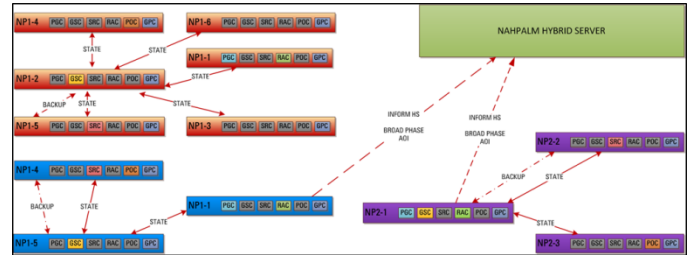


Figure 6: Topology Splitting

4.6. Topology Split

When a group's PGC detects divergence between players, a topological split may be initiated, creating two peer groups from one. In order for this process to be successful, each group must be assigned at PGC and all six roles must be assigned across the new group, as demonstrated in Figure 6.

4.7. Topology Merging

As two groups converge within the virtual world, it may be prudent to merge them, in order to limit the number of overlapping roles. This is the reverse of the process described in section 4.6. In this case, the primary problem which needs resolving is to assign the new PGC role to one of the two existing PGCs. This is determined by comparison of the NDRs.

4.8. Game Play State Dissemination

The primary and most recurrent task within the framework is the dissemination of game state across the group. This is highly dependent on the topological arrangement and role assignments, the examples in Figures 5 and 6 illustrate single GSC state transfer using a star network arrangement within the group. Figure 7 provides an overview of some potential multi-GSC configurations and their impact.

Topology 1 utilizes four GSC roles, with the eight player arranged in a 3-dimensional hypercube arrangement. This

scenario requires three virtual hops in order to transmit state, using inter-GSC communication to merge the results from each dimensional axis and therefore can increase latency and subsequently reduce responsiveness. Topology 2 utilizes two GSC roles and therefore reduces the virtual hop count to two and therefore may improve responsiveness in comparison to Topology 1.

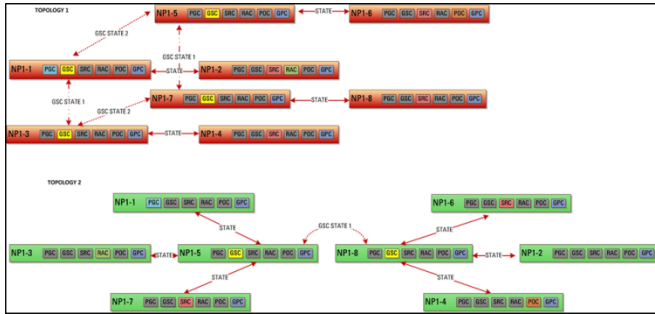


Figure 7: Game State Dissemination

4.9. Handling Persistency Changes

NAHPALM uses the centralized HS to aid persistency. When players are bootstrapped into a peer group and given the role of POC, the HS determines the mutable persistent object which is relevant to this group based on its Broad AoI and communicates the current state of these objects. The POC is responsible for communicating any subsequent changes to these objects to the HS, which are then handled by the HS as described in section 3.1.3, (see Figure 8).

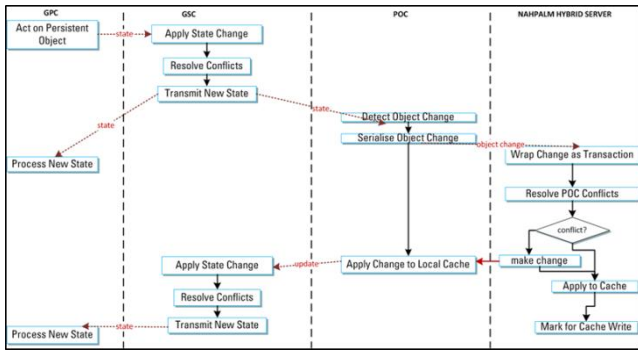


Figure 8: NAHPALM – Persistency Changes

At a macro level, P2P architectures can be considered more robust than a C/S architecture, due to the elimination of single-point of failure and delegation of tasks across multiple independent processing units. However, on a micro-level, peers lack the robustness that dedicated servers provide, since their availability is managed externally by the end-user, both in terms of willingness to participate and hardware faults. NAHPALM’s distribution of roles across multiple peers attempts to mitigate this threat; nevertheless, the roles of PGC, GSC and POC are fundamental to the delivery of consistency and responsiveness. Therefore rapid identification and response are required in order to maintain these characteristics and achieve scalability within the system, (see Figure 9). The response mechanism is like to the peer removal mechanism described in section 4.3.

4.10. Handling Peer Failures

The most damaging threat to consistency and responsiveness within a P2P-based MMOG architecture can arise due to peer failures.

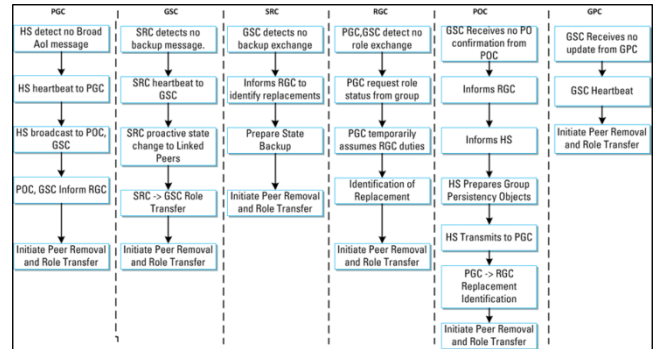


Figure 9: Peer Failover Mechanism

5. COMPARISON WITH RELATED WORK

We contrast NAHPALM against the existing MMOG P2P architectures and highlight the key differences amongst them and how NAHPALM addresses issues related with them.

5.1. Comparison with DESD

DESD [8] is similar to NAHPALM as it employs a P2P overlay network constructed using a Role-based Super-Peer network. Peers known as responsible peers are allocated to control a region within the virtual world, using a centralized lobby server for bootstrapping. Intersection between players and regions is determined via a bounding rectangle. The overlay network, for each sub space, is distributed across a load-balancing tree, for event dissemination. The server is used for responsible peer failover. However, in comparison to NAHPALM, DESD provides only three role types which are all based on network maintenance and therefore provides less flexibility when distributing processing load amongst players and has no flexibility in determining the geographic AoI scheme. DESD statically assigns a player to a region, regardless of their position within the virtual world, meaning peers effectively have to maintain multiple views on the Virtual World. The load balancing tree is the only topology choice and therefore leads to multi-hop state distribution in all situations. DESD also does not account for persistency and only handles responsible peer failure whilst event dissemination utilizes fixed timeslots and therefore does not balance consistency and responsiveness across the network.

5.2. Comparison with QuON

QuON [1] is a fully-decentralized P2P system using mutual notification for event dissemination. QuON also uses a two-phase approach to AoI. Players connect directly to their neighbors within their spatial AoI locality, using unstructured point-to-point communications, whilst global connectivity with players outside the AoI is maintained by connecting to binding neighbors, which are determined

using a 2D QuadTree partitioning scheme across the virtual world. In comparison to NAHPALM, all players are treated homogeneously and therefore does not account for local resources. The full decentralization of the network means that each peer must incur the overhead of maintaining global connectivity since there is no other mechanism for discovery. Players are bootstrapped into the game via an arbitrary position choice and therefore cannot be matched with friends. In addition, QuON does not account for persistency across the network and does not provide mechanisms to handle authentication and authorization.

5.3. Comparison with GROUP

GROUP [2] maintains two overlay topologies and utilizes both geographic and spatial AoI to disseminate state and handle persistency. Players are allocated as managers of both objects and avatars. Object state is maintained via an arbitrary Grid-based dynamic partitioning scheme, where players who are intersecting with the grid are broadcast updates by the manager, who is assigned to a particular grid. Avatar state is disseminated via a mutual notification protocol using Voronoi Spatial AoI [4] to form the second overlay. This overlay is also used for peer discovery. In this system, the virtual world is arbitrarily split into grids and therefore has no utility for matching with existing PVS/Culling schemes used in games. Each player must maintain a Voronoi diagram and also have awareness of the grid partitions, incurring a large local computation cost. Each peer also has the communication overhead of keeping global connectivity via bounding neighbors, defined by the Voronoi AoI. Bootstrapping is done via gate peers, who are a set of peers that have publicly known IP addresses. The mechanism for maintaining and broadcasting this set of gate peers is not specified and thus discovery cannot be ensured. GROUP also does not account for peer failure.

5.4. Comparison with Mediator

The Mediator Framework [5], combines structured DHT-based overlays with hierarchical super peer roles. Mediator uses a multi-role approach to distribute tasks amongst players and uses a peer's resource contribution when determining the amount of processing and communication to be assigned. The hierarchical nature of Mediator allows for particular peers to take the role of state computation and dissemination amongst peer groups. Whilst NAHPALM defines the protocol for role mapping and allocates a supervisory role to coordinate this, Mediator relies on an incentive deadline driven auction approach (DDA) to give tasks to specific peers, using advertisements to publish tasks in a distributed manner, which incurs an added overhead. Event dissemination is reliant on ALM across the DHT, which also adds computational overhead for keeping the multicast tree. Mediator does not define a specific mechanism for handling AoIM amongst peers, instead relying on an adaptation of MOPAR's [9] master node

scheme and thus is lacking the local computational optimizations NAHPALM provides.

6. CONCLUSION

We have introduced a novel Hybrid P2P architecture and detailed the key components and their roles. We have also compared the architecture against several existing architectures and shown where the novel approach can address the flaws that are inherent within these designs. Further analysis of the various combinations of grouping and topology is required in order to deduce the efficacy and the determination of the gameplay scenarios they may be suited to. These will be proposed as future work.

7. REFERENCES

- [1] H Backhaus and S Krause, "QuON - a Quad-Tree Based Overlay Protocol for Distributed Virtual Worlds". 2009, 2nd International Workshop on Massively Multiuser Virtual Environments (MMVE09).
- [2] E. Buyukkaya, M. Abdallah, R. Cavagna, and S. Y. Hu, "GROUP: Dual-Overlay State Management for P2P NVE". 2008, 14th IEEE International Conference on Parallel and Distributed Systems, Melbourne, pp. 1-6.
- [3] C. Carter, A. El Rhalibi, M. Merabti, "A survey of AoIM, distribution and communication in peer-to-peer online games". 2012, 21st International Conference on Computer Communications and Networks, ICCCN.
- [4] A. El Rhalibi and M. Merabti, "Agents-Based Modeling for a Peer-2-Peer MMOG Architecture," *Computers in Entertainment (CIE)*, vol. 3, no. 2, pp. 3-3, April 2005.
- [5] L. Fan, P. Trinder, and H. Taylor, "Deadline-Driven Auctions for NPC host allocation in P2P MMOGs". 2010, *International Journal of Advanced Media and Communication*, vol. 4, no. 2, pp. 140-153.
- [6] S. Y. Hu and G. Liao, "Scalable Peer-to-Peer Networked Virtual Environment". 2004, 3rd ACM SIGCOMM workshop on Network and system support for games.
- [7] K. Kimura, S. Amamiya, T. Mine, and M. Amamiya. 2008. "A semi-structured overlay network for large-scale peer-to-peer systems". 7th International conference on Agents and Peer-to-Peer Computing.
- [8] S. Yamamoto, Y. Murata, K. Yasumoto, and M. Ito, 2005. "A Distributed Event Delivery Method with Load Balancing for MMORPG". 4th ACM SIGCOMM workshop on Network and system support for games.
- [9] A. P. Yu and S. T. Vuong, "MOPAR: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games". 2005, International workshop on Network and operating systems support for digital audio and video, Washington, pp. 99-104.