

# Evaluating Compass Routing Based AOI-Cast by MOGs Mobility Models

Emanuele Carlini  
IMT, Lucca, Italy and  
CNR-ISTI, Pisa, Italy  
emanuele.carlini@isti.cnr.it

Massimo Coppola  
CNR – ISTI, Pisa, Italy  
massimo.coppola@isti.cnr.it

Laura Ricci  
Dept. of Computer Science  
University of Pisa, Italy  
ricci@di.unipi.it

## ABSTRACT

The increasing popularity of Multiplayer Online Games (MOGs) is revealing a huge interest from the research community. Many solutions exploiting P2P technologies have been proposed in order to overtake the limits of the centralized infrastructures. These solutions are based on the definition of a proper dynamic P2P overlay and exploit the concept of *Area of Interest (AOI)* to filter the information relevant for each player. An *AOI-cast mechanism* defines a spanning tree over the AOI so that the information generated by a player is propagated to all the players located in its AOI. Although most proposals are founded on solid basis, they are often validated through players' movement patterns that result different from those of a real MOG.

This paper presents a set of mobility models for the evaluation of the AOI-cast. Together with classical mobility models, mobility patterns derived from popular MOGs, i.e. World of Warcraft (WoW) and Second Life (SL), are also presented. A novel mobility model taking into account the WoW battleground scenarios is proposed. The proposed mobility models are exploited to evaluate an AOI-cast algorithm based on the definition of a Delaunay overlay. The experiments show that the AOI-cast mechanism we propose can be exploited in real scenarios only when a set of conditions hold.

## 1. INTRODUCTION

Commercial MOGs providers currently rely on client/server infrastructures to store and dispatch the state of the virtual world. It is widely accepted that the main problems of the client/server model are the scalability and the financial cost needed to maintain such infrastructures. In the recent years, a lot of effort has been devoted to the investigation of P2P-based approaches [6, 3, 8, 7] to overcome these limitations. Since in these architectures the definition of a scalable communication support is mandatory, the concept of Area of Interest (AOI) has been introduced as a fundamental mechanism to reduce the traffic on P2P overlays.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DISIO 2011, March 21, Barcelona, Spain  
Copyright © 2011 ICST 978-1-936968-00-8  
DOI 10.4108/icst.simutools.2011.245549

The implementation of the AOI requires that each peer dynamically defines either a set of connections with all the other peers in its AOI or with a subset  $S$  of them. In the first case, each peer defines direct connections with all the peers located in its AOI, while in the latter a peer  $P$  connects to peers in  $S$  and these forward the messages received from  $P$  to the remaining peers of its AOI. [19] proposes an AOI-cast routing algorithm based on the definition of a spanning tree defined on the Delaunay Triangulation [2]. This algorithm exploits compass routing [14] to reduce the number of messages required in the construction of the spanning tree. The algorithm has been evaluated in [19] by exploiting a simple but well established mobility model, i.e. the *Random Way-Point Model (RWPM)* [10] which has been exploited mostly in the area of ad hoc and cellular networks to generate simple movement patterns. This evaluation follows the general trend of most existing proposals in the field, that exploit either a model derived from the RWPM or a set of traces derived from the execution of simple ad-hoc defined games. Even if this kind of validation is surely able to provide an initial coarse evaluation of the performance of the solution, it cannot be considered an effective benchmark for the real-world workloads.

The usage of models that closely simulate the behaviour of players in real MOGs is mandatory when the goal is the evaluation of complex movements of the players and fast movements scenarios. It is worth noticing that the evaluation of these issues is fundamental for the validation of the MOG. As a matter of fact, complex movements may create inconsistencies in the local views of the players, and this is true especially for those solutions that adopt a spatial division of the virtual world. In addition, fast movements force the application to recompute the links between peers of underlying network to reconstruct the communication pattern, at the price of a communication overhead.

In this paper we describe a set of different mobility models and exploit them to evaluate the AOI-cast mechanism proposed in [19]. We consider the following mobility models (i) the *RWPM model*, (ii) the *Random Point Group Model (RPGM)* [10], which extends the RWPM by the concept of group, (iii) the *WoW mobility model (WMM)*, which is based on the analysis of the movements in a specific map of World of Warcraft (WoW) [18] and (iv) the *Second Life Model (SLM)* which is based on the proposal of [16]. The evaluation of the AOI-cast mechanism has been carried out by considering two main measures, the *network consistency* and the *messages overhead*.

In summary, the main contributions of this paper are the

following:

- the design and implementation of a new mobility model based on the analysis of players' movements in a WoW battleground scenario;
- an extensive validation of a compass based AOI-cast mechanism by an evaluation of its adaptivity in several contexts.

The paper is structured as follows: Section 2 explores the main contributions related to Voronoi-based architectures for MOGs as well as the main mobility models to evaluate such architectures. Section 3 defines the AOI cast strategy based on compass routing and discusses its extension to reduce network inconsistencies in a highly dynamic environment. Section 4 presents the mobility models. Experimental setup and an analysis of the results are presented in Section 5. Finally Section 6 concludes the paper.

## 2. BACKGROUND AND RELATED WORK

In this section we discuss the main proposals of Voronoi based overalys for MOG and the main mobility models exploited for the evaluation of these proposals. In a MOG an *avatar* is the virtual representation of the *player* which is paired with a *node* of a P2P overlay. In the following we will use the terms player, avatar and node interchangeably, according to the context.

### 2.1 Voronoi Based MOGs

Our approach exploits a Voronoi based approach to define the P2P overlay supporting the MOG. Recently, several proposals based on the definition of a Voronoi tessellation [2] of the MOG have been exploited both for the definition of a Delaunay based P2P overlay and for the management of the passive objects of the virtual world. A Voronoi tessellation is a partition of the space that pairs with each node  $n$  the portion of the MOG including the points which are closer to  $n$  with respect to any other node of the MOG. A Delaunay triangulation is the dual structure with respect to the Voronoi tessellation and is defined by connecting two nodes which are neighbours in the Voronoi tessellation.

[5] exploits Delaunay Triangulations [2] to define an efficient algorithm for the management of the overlay topology which enables the reduction of the high cost of the overlay updates due to the continuous movement of the peers. The basic idea is to maximize the area where a node is allowed to move without triggering a flip operation. The problem of the maintenance cost of a Delaunay overlay is also tackled by [22] which proposes a dynamic clustering algorithm where each peer monitors this cost and triggers the creation of a new cluster when it exceeds a predefined threshold.

The work in [9] proposes a distributed scheme to maintain a overlay based on the partition of the nodes into two non-overlapping sets, which are updated in two distinct phases.

[11] defines an overlay including both Delaunay links and direct links between a peer and other peers in its AOI. The Delaunay links guarantees the connectivity of the overlay. This solution minimizes the latency, but increases the number of connections of each peer, especially in a crowding scenario where a huge amount of peers are located in the same area of the MOG. [11] tackles this problem by dynamically enlarging or shrinking the size of the AOI according to the bandwidth of the peers.

Let us now consider the mobility models exploited to evaluate these solutions. [11] and [9] exploit a Random Way Point Model (RWPM), while [5] and [22] provide a more structured approach by exploiting both RWPM and real traces from Second Life. Another common approach in this field is to implement a simple game and to exploit the traces generated by the game [5], [12, 7] to evaluate proposed overlay.

### 2.2 Mobility Models for MOGs

In the latest years, several supports for MOG have been proposed. Their evaluation is generally performed by exploiting mobility models originally designed to reproduce the movements of human beings, such as those exploited to evaluate ad-hoc wireless networks [10].

The *Random Way Point model*, *RWPM* [10] is one of the most widely exploited ones. In the *RWPM* a set of *way points* are placed uniformly at random on the map and the entities move independently from each other toward them. As soon as an entity reaches a way point, it stops there for a time interval, then it chooses another way point, and so on. The mobility patterns generated by *RWPM* are therefore very naive. *RWPM* has been adapted to describe different kind of scenarios in a MOG by tuning the spatial distribution and the number of the waypoints, the speed of the entities and the criteria to select the waypoints at each step.

While most mobility models for ad-hoc wireless mobile networks focus on the motion behaviour of each entity separately, mobility models taking into account the behavior of *group of entities* have been recently proposed. These models are particularly suitable for MOGs.

The *Reference Point Group Mobility model (RPGM)* [10] has been proposed as an extension of the *RWPM*, by introducing in the model the concept of group. The model can be exploited to simulate the behaviour of teams of players.

Although very simple to implement, the way point-based models fail to represent movements of real MOG avatars. Indeed, since players participating to large scale MOG usually have the possibility to move freely around the world, the distribution of the players in MOGs is usually not uniform[15]. Furthermore, their behaviour results to be highly non-uniform: players move slowly and chaotically within the hotspots, while the movement between hotspots is straight and fast [17]

Recently, some mobility models specifically developed for MOG have been proposed. [20] provides a design and evaluation of a mobility model based on Quake 2 traces. The model is based on a *RWPM* whose parameters are evaluated by using model fitting techniques on traces. Quake traces have been used also in [4] in order to evaluate their solution. They propose a model based on real traces where players tend to move between popular regions of the map and the popularity distribution of these regions follows a power law.

*Blue Banana* [16] provides the design of a mobility model based on Second Life [1]. This model distinguishes between desert areas and hotspots in the game map. The model assumes that the movement of the avatars is slow and chaotic in the hotspots while it is fast and predictable in the desert areas of the MOG. The model exploits an automaton defined by three states, the *halted state*, where the avatar does not move, the *exploring state* where the avatar moves within a hotspot and the *travelling state* where the avatar moves from one hotspot toward another one.

The definition of hotspot as an invariant for a MOG mobility model is also one of main finding of the work of Miller and Crowcroft [18]. They measure and analyse players movements in a WoW scenario, the Arathi Basin one which is representative of the team-oriented interaction that modern MOGs encourage into the game. Nevertheless, [18] does not define any mobility model for the WoW battleground scenario. The main findings of this work are that a way point-based model is not enough to describe complex movements of MOGs, that the level of gathering of players in groups is less than expected and that hotspots based mobility is a realistic pattern of movement in MOGs.

### 3. AOI-CAST ON DELAUNAY OVERLAYS

In [19] we have presented a compass routing based strategy for the definition of a *AOI cast* mechanism. Our solution is based on the dynamic distributed construction of a Delaunay P2P overlay supporting the notification of heartbeats, i.e. messages notifying the position updates of the players.

In our case, compass routing is exploited to build a tree spanning the *AOI* in order to propagate the heartbeat generated by a peer to other peers located in its *AOI*. The root of the multicast tree is the node which generates the heartbeat and the tree includes all the peers belonging to its *AOI*. The *AOI*-multicast tree is built by reversing the paths computed by compass routing.

Compass Routing [14] is a routing algorithm for *geometric graphs* which exploits only *local information* for routing messages. Compass routing finds a path between a source node and a destination node  $d$  by choosing at each routing step executed by node  $n$  the neighbour of  $n$  whose slope is minimal with respect to the segment connecting  $n$  to  $d$ . While compass routing is not cycle free for general graphs, it always finds a finite path between two nodes of a *Delaunay Graph*.

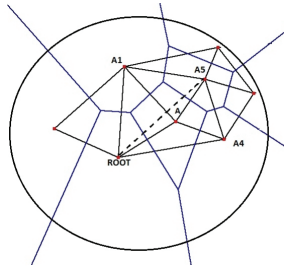


Figure 1: Spanning Tree Construction

We describe our algorithm through the example shown in Fig.1. Let us suppose that the peer *Root* generates an heartbeat, i.e. it is the root of the spanning tree, and let us consider the peer  $A$  which receives the heartbeat.  $A$  should choose among its Voronoi neighbours its children in the spanning tree. For instance, to decide if peer  $A_5$  is its child, the algorithm evaluates whether  $A_5$  would have chosen  $A$  in its path toward the root *Root*. This would happen if the angle  $\angle RootA_5A$  is smaller than  $\angle RootA_5A_1$  and  $\angle RootA_5A_4$ . Note that if this happens,  $A_5$  does not need to compare the slopes of the edges connecting it to further neighbours with respect to the segment  $RootA_5$ , since they are surely larger. This argument is reversed in order to detect the children of a peer in the spanning tree.

The definition of the *AOI Cast* mechanism must take into account the inconsistencies which may arise because of the movement of the peers. As a matter of fact two peers may have a different perception of the position of a common neighbour, due to the delay of heartbeat notifications. This implies that these peers may perceive a *positional drift* with respect to the real position of their common neighbor. Due to the *positional drift*, a peer which has received an heartbeat  $h$  may suppose that one of its neighbour  $n$  should propagate the heartbeat to a common neighbour  $p$  and neglect to propagate  $h$  to  $p$ . The *positional drift* may also generate *redundant notifications*, because two peers may decide to propagate the same heartbeat to one of their common neighbours. Note that these problems are introduced by the highly dynamic nature of the *MOG*. Furthermore, the heartbeat loss is the most serious problem, since it may lead to the overlay partition.

Compass routing has been properly modified to take into account the problem of positional drifts and to reduce the number of peers which do not receive an heartbeat. Our strategy is to define a constant network wide *tolerance angle* ( $c\angle$ ) so that a peer  $p$  routes a message both to the neighbour with the minimal slope with respect to the final destination of the message and to all the neighbours such that the difference between their slope and minimal one is lower than the tolerance angle.

In this way a *tolerance based compass routing* is defined. Note that with this modification an heartbeat may be notified to a peer by more than one neighbour. As a consequence, the resulting algorithm introduces a number of redundant messages. Anyway, it is better to send a larger number of messages instead of neglecting to send the heartbeat to some peer of the overlay. In the experimental section we will evaluate the trade off between the tolerance angle and the number of redundant messages.

## 4. MOBILITY MODELS

In this section we describe the mobility models which have been exploited for the evaluation of the compass-based AOI-cast. We omit the description of the RWPM, since it is widely investigated in the literature [10]. The definition of the *WoW Mobility model* takes inspiration from [18] where the *Arathi Basin* scenario, one of the most played battleground maps of World of Warcraft is analysed [18], while the *SL Mobility Model* has been defined in [16].

### 4.1 The WoW Mobility Model

In the *Arathi Basin* scenario [18], one of the most played map of WoW, 30 players, divided into 2 factions, compete for the control of 5 stationary interest points (i.e flags). The map has two base camps, where players are placed at the start of the game and two graveyards, where players spawn after death. Our model takes into account the team-oriented nature of the scenario, where moving in group is encouraged by the game semantics. However, an avatar may decide to move alone by itself, for instance to take the enemy by surprise.

The behaviour of a player in the WoW Mobility model is briefly summarized in Algorithm *WoW Player Behaviour*. We suppose that a *initialization* phase takes place before any player movement, where players are placed in their respective base camps. Every player chooses to participate to a group with probability  $p_{group}$  or to act by its own with

---

**Algorithm 1:** WoW Player Behaviour

---

```
while true do
  if destination is empty then
    | destination = PickFlag;
  end
  MoveTo(destination);
  WanderAround(destination);
  Battle;
  if death then
    | GoTo(graveyard);
    | WaitResurrection;
    | destination = empty;
  else
    | if Lost or (Win and def(flag) > d) then
    | | destination = empty;
    | end
  end
end
end
```

---

probability  $1 - p_{group}$ . In the following, the first type of player will be referred as team-oriented player, i.e. *TP*, while the second kind of players will be referred as stand-alone player, *SAP*. Afterwards, each player chooses a flag. Function *PickFlag* defines the flag selection which depends on the kind of the player. A *TP* player more likely chooses a flag selected by further players. A *TP* selects the flag from a set composed by: (i) the flags already selected by the members of the same team and (ii) the flags that can be conquered (i.e. they do not already belong to the team). A *SAP* chooses a flag regardless of its owner. In any case, if no flag is available to conquer (i.e. all flags belongs to the team), a flag is chosen at random from the set of all the flags.

---

**Algorithm 2:** PickFlag Function

---

```
if conquerableFlags = ∅ then
  | return random(AllFlags);
else
  if (player = TP) and (teamSelectedFlags ≠ ∅) then
    | return random(teamSelectedFlags ∩
    | | conquerableFlags);
  end
  if player = SAP then
    | return random(conquerableFlags);
  end
end
end
```

---

After the selection of a flag, the player moves toward it and once the flag is reached, the player wanders around it until both the players of the opposite team and those of the same team which have selected that flag have reached it. Once the players have reached the flag, a simulation of the battle is performed.

The outcome of the battle depends on the number of members of each team which are close to the flag. The probability to win for each teams is proportional to the ratio between the number of players of the team and the total number of players at the flag, for instance, if team A has 7 players and team B 5, the probability for the team A to win

is  $p_{win(A)} = 7/12 = 0,58$ . The winning team conquers the flag.

The same probability is also exploited to decide if a player is alive or dead after the battle.  $p_{win(A)}$  is the probability of remaining alive after the battle for the members of team A. Dead players of both factions are placed at the respective graveyards at the end of the battle and, after a predefined interval of time, players resurrect and choose another flag as their goal.

Players that are still alive after the battle and that are part of the winning faction may decide either to stay close to the flag to guard it or to move to another flag using the same criteria shown in the function *PickFlag*. The model guarantees that the number of players guarding the just conquered flag is less than  $d$  where  $d$  is a predefined threshold of the model. The players of the defeated faction that are alive after the battle select a new flag by exploiting the same function.

## 4.2 The SL Mobility Model

The SL Mobility Model [16] is inspired by the virtual world defined by Second Life. In this world, players gather around a set of *hotspots*, which usually correspond to towns, or in general to points of interest of the virtual world. In our model we exploit  $H$  hotspots [21], that are defined by a circular area characterized by a center and by a radius.

The initialization step defines the position of both hotspots and players. The hotspots are placed in the map uniformly at random. Each player may be placed either inside an hotspot or outside it, according to a  $p_{dens}$  probability. If the player is placed outside hotspot, its position is chosen uniformly at random on the whole map. Otherwise, an hotspot for the player is randomly selected and the player is positioned inside the hotspot by considering a power law distribution, which ensure an higher density of players near the center of the hotspot.

The movements of the players are defined according to a Markov chain [16]. The possible states for the players are the following.

- *Halt(H)*: the player remains in place.
- *Exploration(E)*: the player explores a specific area. If the player is moving inside an hotspot, the new position is chosen according to a power law distribution. Otherwise, the new position is chosen at random.
- *Travelling(T)*: the player moves straight toward another point, and the  $p_{dens}$  constraint is guaranteed.

Initially each player is in the state H. At each step, the model decides the new state according to the probability of moving between states.

## 4.3 Reference Point Group Mobility Model

In this model each player belongs to a group and it moves by following the movement of its group, in order to model the players habit to gather in teams. Each group has a virtual *center* which defines the motion behaviour of the players of the group, i.e. next location, the speed and the direction of the movement. Members of a group move together by following the center of the group. In addition to the group behaviour, the RPGM allows independent behaviour for each avatar. More formally, each group has a vector  $V_{group}^t$  which defines the direction of the center of the group

at time  $t$ . At each step the new position  $P_{t+1}$  of the center is computed by exploiting such a vector. Then each member computes its own position by adding a motion vector  $RM$  chosen uniformly at random to  $P_{t+1}$ . The length of the vector  $RM$  is uniformly distributed at random and its direction is uniformly distributed between 0 and 360 degrees. The direction of  $V_{group}$  is chosen so that the corresponding group moves toward an hotspot chosen from a predefined list. When a group reaches the hotspot it computes a new  $V_{group}$  according to the position of the current and of the next hotspot.

## 5. METHODOLOGY AND RESULTS

The AOI-cast algorithm described in Section 3 has been evaluated by Peersim [13], a simulator supporting large scale simulations of P2P overlays. We have developed a set of experiments for each mobility model presented in Section 4.

The metrics exploited to evaluate our solution are the following ones:

- *network consistency*, which evaluates the level of consistency of the overlay, i.e. the consistency among the local views of the overlay at each peer. Network consistency is computed by comparing the local views of the peers with a global view of the whole MOG. To enable this comparison, the simulation builds, at each simulation cycle, a global Voronoi diagram including all the peers of the MOG. The simulator is able to build the global view of the MOG because it knows the position of all the peers, at each simulation cycle. The network consistency is defined as the ratio between the number of links which are present both in the global view and in the local view of the peers and the total number of links in the global view.
- *message overhead*, which evaluates the amount of messages per node required to build the AOI-cast spanning tree.

These two measures are evaluated by varying a number of common parameters listed in Table 1.  $N$  defines the number of players in the virtual world and the number of nodes in the overlay. We vary  $N$  as well as the size of the map according to the semantics of the particular simulation genre. Each simulation lasts  $T$  ticks, and the players perform an action at each tick.  $S$  is the length of the movement done by each player during each tick, and  $d\angle$  represents the deviation range with respect to the trajectory of each player. The deviation range has been introduced to avoid that all the players move along the same line toward their destination.  $c\angle$  is the *tolerance angle* for the compass routing algorithm (see Section 3). Model specific parameters are discussed in the following sections.

### 5.1 The World of Warcraft Model

In this section we describe the results of the evaluation of the AOI-cast algorithm obtained by a set of traces generated by the WMM. We tune the parameters of the model by following the evidence presented in [18]. The probability  $p_{group}$  has been set to 0.4, since the players move mostly alone rather than in group when they are heading toward flags. We set also  $d_{30} = 1$  and  $d_{70} = 2$  respectively for the 30 and 70 player scenarios.

The graphs in Figure 2 refer to simulations run with  $c\angle = 6$  and  $d\angle = 3$ . The runs last 500 cycles and are performed for 30 rs. 70 nodes. It is worth noticing that even if the number of online players in WoW may be quite large, the number of players present in the same area of the virtual world is generally no more than a hundred.

The speed of the players varies from 1 to 6. The experiments show that, in general, the level of consistency decreases when the speed of the players increases. This result is expected since when new players suddenly appear in the AOI of a player, the local views of the peers may be not updated properly for a transient period of time. However, when increasing the speed, the number of avatars plays an important role. Indeed, with  $S \geq 3$  the consistency remains roughly on the same order with 30 nodes (the 99th percentile consistency goes from 0.50 with  $S = 3$  to 0.59 with  $S = 6$ ). However with an overlay including 70 nodes the 99th percentile goes from 0.69 with  $S = 3$  to 0.12 with  $S = 6$ . This trend suggests that our approach cannot be applied in a WoW-like scenario with high-speed avatars. On the other hand, the analysis of communication overhead suggests that the number of messages does not depend on speed. With the only exception of the 99th percentile with  $S = 3$ , the number of messages maintains roughly a constant trend in the interval  $[1, 2]$  both for the average and the worst case.

Simulation in Figure 3 refer to  $d\angle = 3$  and  $S = 1$ . The run lasts 500 cycles and has been repeated for 30 and 70 nodes. The  $c\angle$  ranges from 2 to 20. The consistency graph 3a shows how the increment of the tolerance angle provides an improvement on the overall consistency. Further, a small increase of the value of the tolerance angle (from 0 to 4) generates a burst of consistency in all the scenarios. While the values for 70 nodes are the expected ones, the same does not hold for 30 nodes. In the worst case (30 nodes, 99th perc.) the consistency remains very low even with an high value for tolerance angle. This trend is not visible on the average curve, but only from the 99th percentile curve. A possible explanation for this behaviour lies in the fact that the AOI-cast is based on geometric proprieties of the positions of the avatars. Due to this, it is possible that with a low number of avatars, the angle needs to be wider in order to maintain topological consistency. This ratio is supported by the slight increment of consistency when using a very wide compass angle (20 degrees).

The analysis of the communication overhead shows that an increase of the tolerance angle corresponds to a slight increase of the number of messages of the same order both for the average and for the worst case. The comparison between this graph and the consistency graph suggests that a good value for the compass angle lies between 2 and 6. In fact, the increase of communication overhead with  $c\angle > 6$  does not correspond to a valuable gain in terms of consistency.

### 5.2 The Second Life Model

In this section we evaluate the AOI-cast algorithm by the SLM. The number of hotspots is  $H = 3$ , their radius  $H_{rad} = 60$  and their density  $p_{dens} = 0.8$ . The probabilities to move among the automata states have been taken from the original work.

The consistency graph (Figure 4a) is pretty clear. If no tolerance angle is introduced in compass routing, the consistency is very close to zero both in the average and in the worst case. This suggests that in the SLM, where players

Symbol	Description	WMMè	SLM	RPGM	RWPM
$N$	size of voronoi overlay (nodes)	30-70	10-150	100	200
$(X, Y)$	size of the map	601 x 615		700 x 680	
$T$	simulation time (in peersim cycles)	500			
$S$	speed of the entities	1-6	2	1-6	1-6
$d\angle$	max deviation for the entities movements	3			
$c\angle$	tolerance angle for the compass routing	0-20	0-6	0-6	0-16

Table 1: List of parameters with values for the experiments for the different models

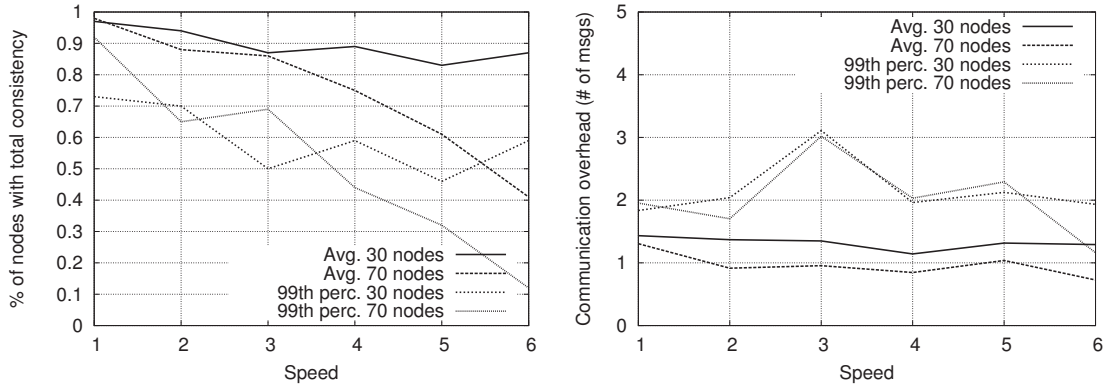


Figure 2: WMM's consistency and communication overhead with variable speed

tend to gather within an hotspot, a tolerance angle avoid inconsistency. It is also clear that even a small tolerance angle introduces a burst of consistency. With  $c\angle \geq 2$  the consistency is always above the 0.9 even in the worst case. This behaviour is due to the smooth movements of the SLM with respect to more chaotic genres.

The graph in Figure 5 considers several scenarios characterized by different number of players (from 10 to 150 nodes). In the graph (Figure 5a), the consistency remains at a good level in all the runs. In details, a slight decreasing of the consistency is noticeable when the overlay size is increasing. However in the worst case with a size of 150 nodes, the consistency remains around 90%. As regards the communication overhead, the amount of messages is linear with respect to the increase of the number of players. This trend is present both in the worst and in the average case. This happens because the number of hotspots is fixed, so the increase of the number of players generates scenarios of crowding inside the hotspots. This however does not increase inconsistency, due to the fixed speed of the players ( $S = 2$ ) and to the presence of the tolerance angle ( $c\angle = 3$ ).

### 5.3 The RWPM and RPGM Models

Figures 6c and 6d show the consistency values when varying  $c\angle$  from 1 to 6 degrees for RPGM and from 1 to 16 degrees for the RWPM. The consistency remains above a good level in the whole set of experiments. For example, in the RPGM, consistency reaches a value of 0.9 in the worst case when  $c\angle = 6$ , and it never falls under 0.6.

Figures 6a and 6b still show the evaluation of consistency when speed is variable (from 1 to 6) for both models. The behaviour is the same in both models as the consistency decreases more or less linearly when the speed increases.

We skip the presentation of the RWPM and RPGM message overhead, since it keeps a constant trend (under 1 message per node) in every scenario.

### 5.4 Discussion

The comparison between consistency in WMM and SLM returns similar results. Indeed, the consistency suddenly increases when  $c\angle \geq 2$ , reaching almost the maximum. For example, in the worst case with  $c\angle = 2$  the consistency is 0.82 for WMM and 0.91 for SLM. On the other hand, in RWPM and RPGM a high level of consistency exists even when no tolerance angle is exploited. In this sense the RPGM, due to the presence of groups has a trend slightly closer to the WMM and SLM, but still far to be comparable. The analysis of the consistency reveals that the tolerance angle is the factor that greatly affects consistency when a mobility model generating complex movements is exploited. On the other way round, the impact of the tolerance angle is not evident for simpler models.

Conversely, when analysing the impact of the speed on the consistency level, we note that all the models exhibit similar behaviours. This may be due to the fact that the overlay is built according to spatial relationship in the MOG, where fast changes in the avatar configurations generate a great level of inconsistencies in any case.

## 6. CONCLUSION

This paper presents an extensive evaluation of an AOI-cast algorithm based on compass routing. We have run a large set of experiments by considering different mobility models derived from the most popular MOGs, World of Warcraft and Second Life. We have compared these mea-

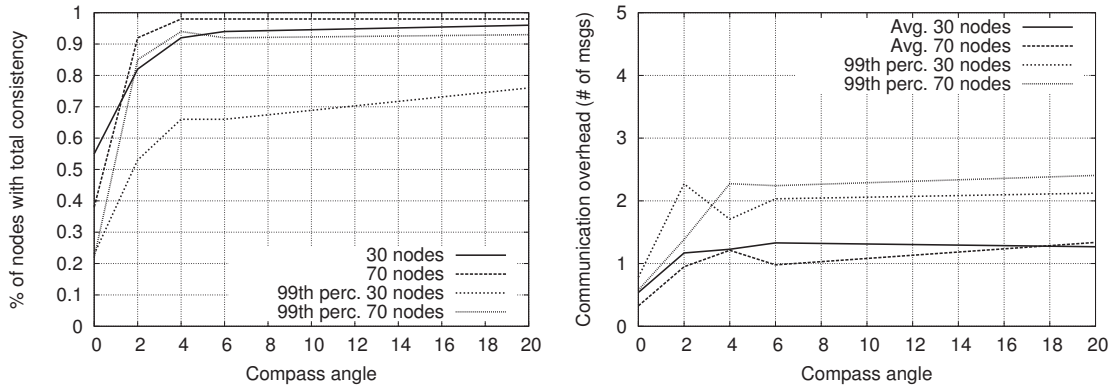


Figure 3: WMM's consistency and communication overhead with variable tolerance angle

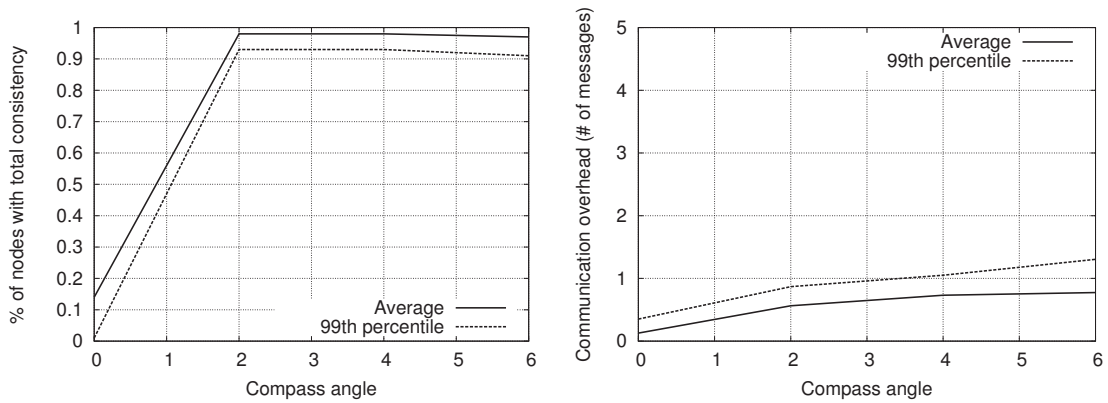


Figure 4: SLM's consistency and communication overhead with variable tolerance angle

surements with those obtained by simpler mobility models. The comparison has shown that these simpler models fail to be effective benchmarks for the MOGs.

As a future work, we plan to investigate further mobility models. An interesting research field is that of community based mobility models which have been mainly proposed in the area of MANET and delay tolerant based networks. We plan to investigate their effectiveness for modelling the behaviour of the players in a MOG.

## Acknowledgment

The authors acknowledge the support of Project FP7-257438, Contrail: Open Computing Infrastructures for Elastic Services (2010-2013). The authors would thank Antonio Quartulli, Francesca Gaddini, Davide Agrifoglio and Luca Medone who conducted part of the experiments. The authors would also thank the anonymous reviewers for useful comments.

## 7. REFERENCES

- [1] Second life website. <http://secondlife.com/>.
- [2] F. Aurenhammer. Voronoi diagrams - A survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, September 1991.
- [3] A. Bharambe, J. Douceur, J. Lorch, T. Moscibroda, J. Pang, S. Seshan, and X. Zhuang. Donnybrook: Enabling large-scale, high-speed, peer-to-peer games. *ACM SIGCOMM Computer Communication Review*, 38(4):389–400, 2008.
- [4] A. Bharambe, J. Pang, and S. Seshan. Colyseus: A distributed architecture for online multiplayer games. In *NSDI '06: 3rd Symposium on Networked Systems Design & Implementation*, pages 155–168, 2006.
- [5] E. Buyukkaya and M. Abdallah. Efficient triangulation for P2P networked virtual environments. *Multimedia Tools and Applications*, 45(1-3):291–312, May 2009.
- [6] E. Buyukkaya, M. Abdallah, and R. Cavagna. VoroGame: A Hybrid P2P Architecture for Massively Multiplayer Games. *6th IEEE Consumer Communications and Networking Conference (CCNC 2009)*, pages 1–5, 2009.
- [7] L. Chan, J. Yong, J. Bai, B. Leong, and R. Tan. Hydra: a massively-multiplayer peer-to-peer architecture for the game developer. In *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, pages 37–42. ACM, 2007.
- [8] D. Frey, J. Royan, R. Piegay, A. Kermarrec,

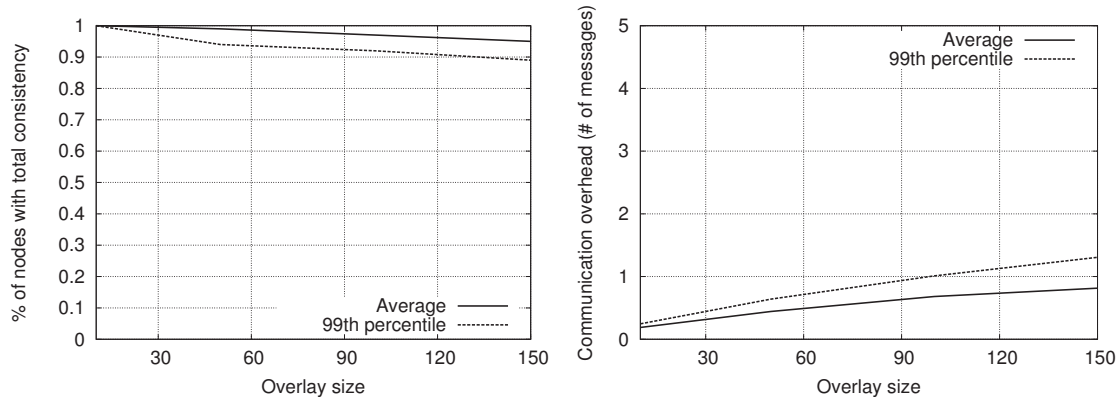


Figure 5: SLM's consistency and communication overhead with variable overlay size

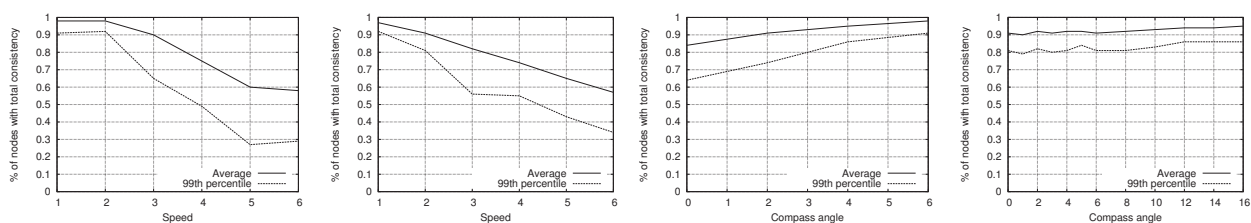


Figure 6: Consistency in RPGM and RWPM with variable speed and variable tolerance angle

- E. Anceaume, and F. Le Fessant. Solipsis: A decentralized architecture for virtual environments. *Proceeding of 1st International Workshop on Massively Multiuser Virtual Environments (MMVE'08)*, pages 29–33, 2008.
- [9] M. Ghaffari, B. Hariri, and S. Shirmohammadi. A delaunay triangulation architecture supporting churn and user mobility in MMVEs. In *Proc. 18th Int. workshop on Network and operating systems support for digital audio and video*, pages 61–66. ACM, 2009.
- [10] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems - MSWiM '99*, pages 53–60, 1999.
- [11] S. Hu, J. Chen, and T. Chen. VON: a scalable peer-to-peer network for virtual environments. *Network, IEEE*, 20(4):22–31, 2006.
- [12] J. Jardine and D. Zappala. A hybrid architecture for massively multiplayer online games. *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games - NetGames '08*, page 60, 2008.
- [13] M. Jelasity, A. Montresor, G. P. Jesi, and S. Voulgaris. The Peersim simulator. <http://peersim.sf.net>.
- [14] E. Kranakis, H. Singh, and J. Urrutia. Compass Routing on Geometric Networks. In *11th Can. Conf. on Computational Geometry (CCCG)*, pages 1–4, 1999.
- [15] C.-A. La and P. Michiardi. Characterizing user mobility in second life. *Proc. of the first Workshop on Online social networks - WOSP '08*, page 79, 2008.
- [16] S. Legtchenko. Blue Banana: resilience to avatar mobility in distributed MMOGs. *Networks*, pages 171–180, 2010.
- [17] H. Liang, I. Tay, M. Neo, and W. Ooi. Avatar mobility in networked virtual environments: measurements, analysis, and implications. *CoRR abs/0807.2328*, 2008.
- [18] J. L. Miller and J. Crowcroft. Avatar movement in World of Warcraft battlegrounds. *2009 8th Annual Workshop on Network and Systems Support for Games (NetGames)*, pages 1–6, Nov. 2009.
- [19] L. Ricci, M. Albano, A. Quartulli, and L. Genovali. AOI-cast by Tolerance Based Compass Routing in Distributed Virtual Environments. In *ACM/IEEE Netgames, Network and System support for Games*, 2009.
- [20] S. Tan and W. Lau. Networked game mobility model for first-person-shooter games. *on Network and system support for games*, page 1, 2005.
- [21] R. Thawonmas, M. Kurashige, and K. Chen. Detection of Landmarks for Clustering of Online Game Players. *The International Journal of Virtual Reality*, page 6, 2007.
- [22] M. Varvello, E. Biersack, and C. Diot. Dynamic clustering in delaunay-based P2P networked virtual environments. *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games - NetGames '07*, pages 105–110, 2007.