Issues in Designing a Communication Architecture for Large-Scale Virtual Environments

Emmanuel Léty and Thierry Turletti

INRIA, 06902 Sophia Antipolis, FRANCE
{Emmanuel.Lety, Thierry.Turletti}@sophia.inria.fr

Abstract. This paper describes the issues in designing a communication architecture for large-scale virtual environments on the Internet. We propose an approach at the transport-layer, using multiple multicast groups and multiple agents. Our approach involves the dynamic partitioning of the virtual environment into spatial areas and the association of these areas with multicast groups. In our work, we describe a method to determine an appropriate cell-size that limits the traffic received per participant with a limited number of multicast groups.

1 Introduction

This paper describes the issues in designing a communication architecture for Large-Scale Virtual Environments (LSVE) on the Internet. Such virtual environments (VE) include massively multi-player games, Distributed Interactive Simulations (DIS) [1], and shared virtual worlds. Today, many of these applications have to handle an increasing number of participants and deal with the difficult problem of scalability. Moreover, the real-time requirements of these applications makes the scalability problem more difficult to solve. In this paper, we consider only many-to-many applications, where each participant is both source and receiver. We also make the assumption that a single data flow is generated per participant. However, we believe that most of the results and mechanisms presented in this paper can be easily adapted to more complex applications that use several media types or layered encodings [2].

Using IP multicast solves part of the scalability problem by allowing each source to send data only once to all the participants without having to deal with as many sequential or concurrent unicast sessions. However, with a large number of heterogeneous users, transmitting all the data to all the participants dramatically increases the probability of congestion within the network and particularly at the receiver side. Indeed, processing and filtering all the packets received at the application level could overload local resources, especially if the rendering module is already processor intensive [3]. [4] shows that in a group communication setting, as the number of flows of data and the number of users increase, the percentage of content received by each participant that is interesting decreases.

This useless information represents a cost in terms of network bandwidth, routers buffer occupation and end-host resources and is mainly responsible for the degradation of performance in LSVE.

We argue that the superfluous received traffic has to be filtered out before it reaches the end-host. The main difficulty in this filtering mechanism comes from the heterogeneity and the dynamicity of the receivers, not only in terms of bandwidth and processing power but also in terms of data interest, virtual and physical locations. In AIM[5], a network-layer approach is proposed that enables sources to restrict the delivery of packets to a subset of the multicast group members. However, this proposition requires modifications in the routers and is unfortunately not yet deployed in the Internet.

In this paper, we propose a filtering mechanism done at the transport-layer, using multiple multicast groups and agents. Our approach involves dynamic VE partitioning into spatial areas called *cells* and the association of these cells with multicast groups. In [6], we described a method, based on the theory of planar point processes, to determine an appropriate *cell-size* so that the incoming traffic at the receiver side remains with a given probability below a sufficiently low threshold. The simulations presented in this paper are complementary to this work and present a new evaluation of the mapping algorithm. We then propose mechanisms to dynamically partition the VE into cells of different sizes, depending on the density of participants per cell, the number of available multicast groups, and the link bandwidth and processing power available per participant.

The rest of the paper is organized as follows. Section 2 reviews the limitations of the current IP multicast model, presents the cell-based grouping strategy, and examines the tradeoff in selecting the cell-size parameter. Section 3 discusses simulation results of the impact of the cell-size and the density of participants on the traffic received at the receivers. Section 4 presents a communication architecture framework that allows a dynamic cell-based grouping strategy with a limited number of multicast groups. Finally, Section 5 discusses related works, and Section 6 concludes the paper and presents directions for future work.

2 Motivation

We now examine the different limitations in using multiple multicast groups, the cell-based grouping strategy, and how to select the best size of cell.

2.1 Multiple multicast groups limitations

There are several limitations on the use of multiple multicast groups. First, we have to consider that today, multicast groups are not inexhaustible resources: the number of available multicast groups in IPv4 is limited to 268 million Class

D addresses[1]. There is an increasing number of applications that require several multicast addresses, such as layered coding based videoconferencing, or DIS applications. The widespread use of multicast increases the probability of address collisions. A few solutions have already been proposed in the literature to solve the multicast address allocation problem. For example, a scalable multicast address assignment based on DNS has been proposed in [7]. Another option could be the use of the Multicast Address Set Claim (MASC) protocol which describes an hierarchical block allocation of Class D addresses scheme for the Internet[8]. Some alternatives to the current IP multicast model have also been proposed: [9] describes a multicast address space partitioning scheme based on the port number and the unicast host address. In *Simple Multicast*, a multicast group is identified by the pair (address of the group, address of the core of the multicast tree), which gives to each core the full set of Class D addresses space [10]. In *EXPRESS*, a multicast *channel* is identified by both the sender's source address and the multicast group [11]. Finally, with IPv6, the multicast address space will be as large as unicast address space, so this will solve the multicast address assignment problem. However, all these propositions are not yet available on the Internet and most of them are still active research areas.

Second, multicast addresses are expensive resources. The routing and forwarding tables within the network are limited resources with limited size. For each multicast group, all the routers of the associated multicast tree have to keep state about which ports are in the group. Hosts and routers also need to report periodically their IP multicast group memberships to their neighboring multicast routers using IGMP[12]. Moreover, some routing protocols (such as DVRMP[13]) rely on the periodic flooding of messages throughout the network. All this traffic has a cost, not only in terms of bandwidth but also in terms of join and leave latency, which should be taken into consideration for interactive applications [14]. Indeed, when a participant sends a join request, it can take several hundred of milliseconds before the first multicast packet arrives. Such costs should be obviously considered in Large-Scale Multicast Applications (LSMA) and argue in favor of a bigger cell-size, and therefore, of a limited number of multicast groups.

2.2 The cell-based grouping strategy

Grouping strategies consist in partitioning senders and receivers into multiple multicast groups, depending of their common interest. Before partitioning the entire set of participants into multiple multicast groups, the data in which users are interested have to be identified. In this paper, we define the *user interest* as the set of virtual entities that a user can interact with. Note that entities located within the domain vision of a participant should only be considered as a part of its *area of interest*. However, users interests can change during the session, in particular, new participants can join or leave a session. So, it is important to handle the dynamicity of these centers of interest during the session.

[1] IPv4 Class D addresses use 28-bits address space.

Once this identification is done, a grouping strategy has to be defined, according to several parameters, such as the number of available multicast groups, link capacities at the receiver, etc. Different grouping strategies have been proposed for LSVE [15, 16]. In this paper, we focus on the *cell-based grouping strategy* which basically consists in partitioning the VE into cells and assigning to each cell a multicast group. During the session, each participant identifies the cell it is currently "virtually" located in, and sends its data to the associated multicast group. To receive the data from the other participants included in its area of interest, each participant has to join the multicast groups associated with the cells that intersect its area of interest. Similarly, when a participant moves, it needs to leave the multicast groups associated with the cells which do not intersect anymore its area of interest.

The cell-based grouping strategy is particularly suitable on VE that can easily be partitioned into virtual areas (e.g., virtual Euclidean spaces). However, the main difficulty in this partitioning is to find the appropriated cell-size. Indeed, decreasing the cell-size increases the overhead associated with dynamic group membership whereas increasing the cell-size increases the unwanted information received per participant [17]. In the following subsection, we examine the issues involved in selecting the best size of cell.

2.3 The cell-size tradeoff

Two approaches are possible to estimate the best cell-size in a LSVE: the first approach requires the pre-calculation of a static cell-size parameter, which remains the same during the whole session. The second approach consists in dynamically re-estimating the cell-size during the session, taking into account dynamic parameters. To motivate the choice of one of these two approaches, let us first identify the parameters involved in the cell-size calculation and then, examine the impact of the dynamicity of these parameters on the appropriate cell-size.

- **The number of available multicast groups** is an important parameter to take into account for the cell-size calculation because it gives a lower bound of the cell-size. As the number of multicast groups used is inversely proportional to the size of the cell, a small set of available multicast groups will lead to a bigger cell-size.
- **The receivers capabilities** identify the link capacities and the processing power available per receiver. Typically, this parameter limits the amount of traffic that receivers can handle. Assuming each user roughly generates the same amount of traffic, the incoming traffic per receiver grows linearly with the total number of sources contained in the multicast groups to which it has subscribed. Nevertheless, some of these participants may be located outside the area of interest but inside a cell that includes this area of interest. The ratio between the corresponding number of unwanted participants and the total number of sources received represents the percentage of *superfluous* traffic received. So, the cell-size and more particularly the ratio between the cell-size and the size of the area of interest, have a direct impact on the amount of unwanted traffic.

- **The density of participants** represents the ratio between the number of participants and the size of the VE. In the cell-based grouping strategy, the area of interest is approximated by the smallest set of cells covering the area of interest. In the rest of the paper, we refer to the difference between these two areas as the *superfluous area*, see Figure 3. So, the density of participants in a VE not only has an impact on the average number of participants located in the area of interest, but also on the superfluous area. A smaller cell-size could allow a better approximation of the area of interest and a significant reduction of superfluous area and its corresponding traffic.
- **The participant velocity** can be used in a cell-based grouping VE, to estimate the bandwidth overhead generated when participants cross cells, and the mean time that the participant stays per cell. For example, assuming a straight and horizontal movement, the product of the participant velocity with the cell-size determines the mean time that the participant stays per cell, and therefore, the average frequency of join and leave messages.

3 Impact of the cell-size

In this section, we analyze the impact of the cell-size and the density of participants on the traffic received by participants. We denote s the cell-size (i.e., the distance between two adjacent horizontal or vertical cell boundaries), $CellArea$ the cell area s^2, and $IArea$ the area of interest. Furthermore, the following assumptions are made for the simulations:

- The cells form a regular square grid on the plane; the left and the right extremities, and the bottom and the top extremities are linked to each other, thus forming a torus.
- The participants are static and located on the plane according to a uniform distribution law and each participant generates the same amount of traffic.
- $IArea$ is a square of area r^2 centered on each participant.

In order to be as generic as possible, we focus on the impact of the ratio between $CellArea$ and $IArea$ (i.e., $\frac{s^2}{r^2}$). Figure 1 plots the average percentage of superfluous traffic out of the total traffic received by a participant. Since the participants are located on the plane according to a uniform distribution law, the percentage of superfluous traffic is equal to the ratio between the superfluous area and the area including all the cells that intersect $IArea$. We observe that when $CellArea$ is larger than $IArea$, more than 70% of the traffic is superfluous. This figure also suggests that when $CellArea$ is smaller than $IArea$, a slight diminution of $CellArea$ decreases significantly the traffic received by receivers. However, it is important to notice that 70% of superfluous traffic is acceptable compared to the situation in which all the users communicate on a single multicast group [3]. Indeed, this 70% represents the percentage of superfluous traffic out of the total traffic received and not out of the total superfluous traffic. With a single multicast group and a large number of participants, almost all the traffic received would be superfluous.

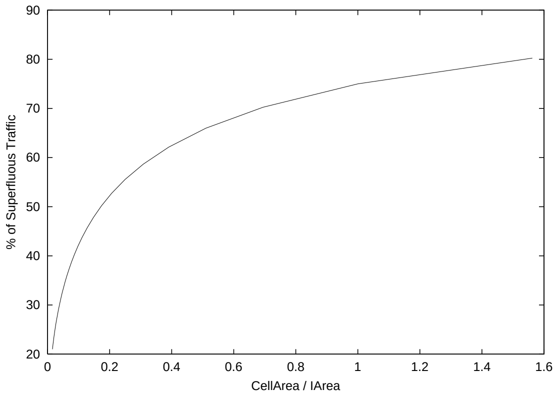

Fig. 1. Percentage of Superfluous Traffic

In order to evaluate the maximum traffic received per participant with a given probability, we performed a set of extensive simulations with different densities of participants and different $CellArea$ values. In each simulation, we kept track of the traffic received by each participant for 10,000 different distributions of participants on the grid. We then sorted an array containing the different traffic received for all the distributions and all the participants. In order to determine the maximum traffic with a probability p, we read the element whose index is equal to p times the number of elements in the sorted array. This allows us to assert that the traffic received per participant is less than the value of this element with the probability p. In [6], we presented a model based on the theory on planar point processes, to determine the probability that the receiving traffic stays below a given threshold. However, the following simulations are complementary to this model and confirm the impact of the cell-size on the traffic received by end-users.
The left side of Figure 2 shows the maximum traffic received by a participant 50% of the time, depending on the density of participants in the VE, and on the $\frac{s^2}{r^2}$ ratio. Here, the density of participants represents the average number of participants per $IArea$. Such a way to express the density of participants in a VE is very useful, as it allows us to modify the $CellArea$ without having an impact on the value of the density. The simulation shows that for a given value of participant density, it is possible to find the largest ratio between $CellArea$ and $IArea$, so that this traffic remains under a sufficiently low threshold. Note that the maximum traffic received with a probability of 0.5 does not represent the average traffic received but the median value of the traffic received.

Finally, the right side of Figure 2 probably shows the most interesting results. In order to satisfy the participants in a VE, it is better to determine an ap-

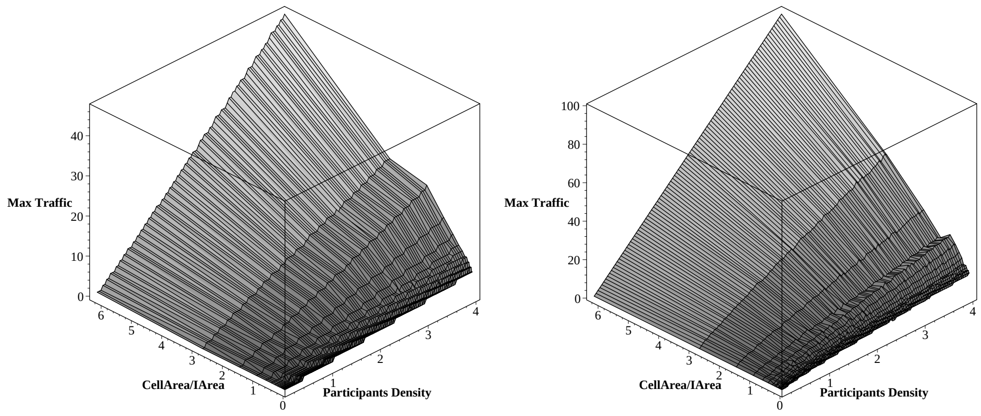

Fig. 2. Max Traffic Threshold with p=0.5 and p=0.99

propriate *CellArea* so that the incoming traffic remains with a high probability below the maximum traffic that they can handle. This value of probability reflects the tradeoff between the satisfaction of the users and the required number of multicast groups. Figure 2 shows that for a given density of participants, it is possible to find the largest *CellArea* (i.e., the smallest number of multicast groups), so that the incoming traffic remains below a sufficiently low threshold with a probability of 0.99 (i.e., 99% of the time). Moreover, for a given *CellArea*, we observe that this traffic increases linearly with the density of participants.

In [6], we have shown that the mean residence time in a cell decreases exponentially as the mean velocity approaches 1 cell-size per second. This result argues in favor of a limited velocity in LSVE, so that the residence time per cell remains higher by orders of magnitude than the join and leave latency. Indeed, a participant needs to anticipate its join request by subscribing to the multicast groups which map the cells where it can go during the time corresponding to a join latency. Hence, its velocity and the cell-size impact on the number of multicast groups it needs to join by anticipation, and therefore on the IGMP traffic generated.

4 Framework for a Scalable Communication Architecture

In this section, we describe a framework for a scalable communication architecture for LSVE. We believe that today, such many-to-many applications, with potentially thousands of users, require minimal management and administration support. Indeed, the number of participants in such applications is too large to transmit all data to all users and to let them filter out the part of the data they are interested in. This would not only waste network and end-hosts resources, but also result in the fast degradation of the application performance at the end-user. Moreover, the aim of this paper is not to propose a new IP multicast model neither to come up with a network-layer approach, adding new mechanisms in the routers. Alternatively, we present a transport-layer solution with multiple agents, assuming that all the users are capable of receiving multicast transmissions. The goal of this architecture is to make LSVE scalable with thousands of heterogeneous users on the Internet. Moreover, we claim that this solution works with a limited number of available multicast groups.

In order to allow participants to select the information they would like to receive, we propose mechanisms using multiple multicast groups. Each participant joins and leaves multicast groups, depending on its interest in the content of the data transmitted.

This section is organized as follows. First, we introduce a user satisfaction metric and present the rôle of the *agents* in our architecture. Then, we describe the information exchange process between participants and agents. Finally, we present our mapping algorithm with a first evaluation.

4.1 User satisfaction metric

An ideal situation from the end-user viewpoint can be defined as a situation where the traffic received contains no superfluous data. However, this situation is far from being realistic, considering the cost of multicasting, and therefore, the limitation in the number of available multicast groups (see Section 2.1). For this purpose, we define the metric of the user satisfaction S as:

$$S = \frac{U_r}{\min(T, C)} \tag{1}$$

where U_r stands for the interesting data rate received and processed (in the case of a homogeneous Poisson point process of intensity λ, this would be proportional to λr^2); T represents the global data rate (received or not received), in which the user is interested; and C stands for the receiver capability, which is the maximum data rate that the receiver can handle (limited by its network connectivity and/or processing power). When a participant receives and processes all the data it is interested in, this satisfaction metric is maximal whatever the superfluous traffic rate. Notice that for a particular user, S is also maximal when U_r is equal to C. This is true even though only a part of the interesting and useful data is received by the application. We justify the choice of this metric by the necessary tradeoff between the superfluous data rate received, the network state, and the overhead associated with dynamic group membership. Note that the goal of our architecture is not to maximize the satisfaction of the worst receiver in terms of network connectivity and processing power, but of the receiver that is the least satisfied. This approach often referred as *max-min fairness* is described in [18].

4.2 Agents responsibility

Let us define *agents* as servers or processes running at different parts of the network (e.g., on a campus LAN, hosted by an ISP or by LSVE developers).

Administrators of LSVE are responsible for deploying such agents on the Internet and for positioning them as close as possible to their potential users. Our approach requires the dynamic partitioning of the VE into cells of different sizes, and the association of these cells with multicast groups. Agents have to dynamically determine appropriate cell-size values in order to maximize the users satisfaction.

Before any participant is connected, the VE has to be partitioned into *start-zones*, according to its intrinsic structure (e.g., walls, rooms, etc.). Each start zone is then associated with a single multicast group. During the session, four successive operations are required:

- Partition each start-zone into several *zones*, according to the distribution of users within the start-zone.
- Compute the appropriate cell-size for each zone, according to the parameters listed in Section 2.3 (see Figure 3).
- Divide each zone into cells, according to its computed cell-size, and assign a multicast group address to each cell of each zone.
- Inform the participants of which multicast groups they need to join in order to interact with participants located around them.

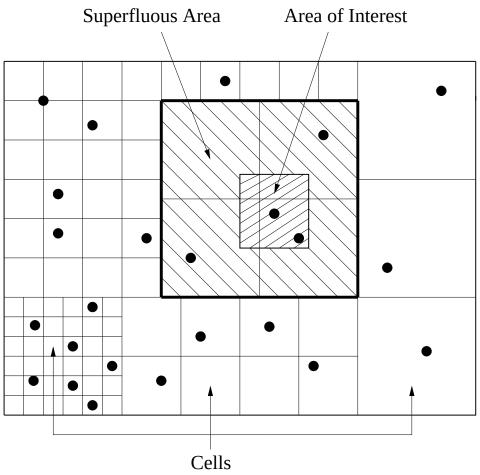

Fig. 3. Partitioning with different cell-sizes

In the rest of the paper, we refer to the three first operations as the *mapping algorithm*. We also designate the results of these three operations as the *mapping information*. Concerning the fourth operation, it is necessary to distinguish between two different situations: the first situation happens when a participant is moving in the VE and is about to enter in an area where it does not have the mapping information. The second situation occurs when agents decide that the cell-size of a part of the VE is no longer appropriate; for example if the density of participants in this area suddenly increases. In this case, a new cell-size needs to be computed and the participants who are currently located in this area need to update their group memberships. Moreover, participants need to keep interacting between each other without suffering from this *remapping*. From now, we refer to this critical operation as the *handover management* (see Section 4.5).

4.3 Mapping information

In order to communicate mapping information to users, i.e., the association between cells and multicast group addresses, it is necessary to find a way to identify and name these cells within the VE. Moreover, the VE could be a structured environment with walls and rooms of different sizes. Two participants can be very close to each other but as a wall is separating them, there is no possible interaction. This specific information should be taken into account before partitioning a VE into different zones. Note that cells have the same size within each zone and that all cells are squares (at the exception of a few of them located at the border of the zones). To refer to a virtual position in the VE in a permanent way, we divide the VE in *area units*. So, a cell contains an integer number of area units and a zone an integer number of cells. The area unit is chosen according to the maximal participant velocity, the number of available multicast groups, the average size of the participant area of interest, and the join and leave latency. Once this division is done, each area unit is referenced by its position in the VE. A matrix of "probability of interaction" between area units is built according to the structure of the VE. Agents use this matrix to dynamically define the different zones of the VE and the mapping information.

4.4 Participants-to-Agent communication

Figure 4 shows the different levels of communication in our architecture :

- Each participant subscribes to one or more multicast groups but sends data packets on a single group.
- Each participant is connected to a single agent, using a unicast connection.
- Agents communicate with each other on a single multicast group: the Agent Multicast Group (AMG).
- Agents subscribe to users' multicast groups during handover operations.
- New participants send *Hello packets* on the agent's multicast group.

Let us now describe the way participants enter the VE and the different messages they send to their agents.

Connection to the Virtual Environment

We assume that before starting a session, participants have already downloaded the VE description and know the agent's multicast group address. When a new

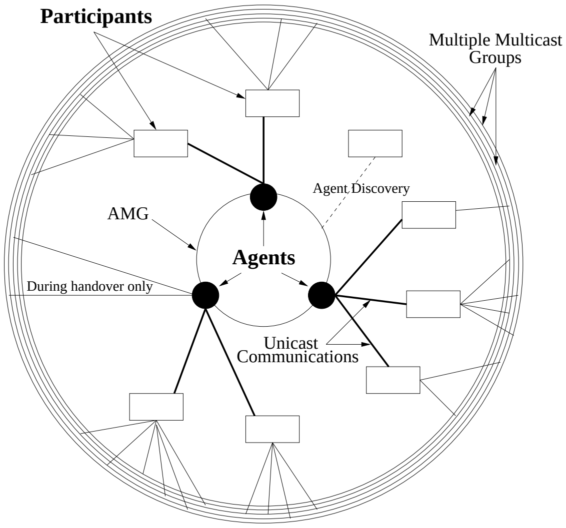

Fig. 4. Communication architecture

participant wants to enter the VE, it first needs to find the "closest" agent before registering and starting a login process. In our architecture, end-users discover agents by sending Hello packets on the agent multicast group address (they do not need to request membership to that group). This agent discovery is done using either an incremental TTL-based mechanism or an RTT-based mechanism, depending on the distance metric we decide to choose. As soon as an agent receives a Hello packet from a new participant, it opens a TCP connection with it and starts the login process. Afterwards, an optional authentication process can start. Only the mapping information concerning the virtual area where the new participant is located, is transmitted during this connection. Indeed, the mapping information of other parts of the VE might change before the participant needs to use them.

Map request **message**

As participants move in the VE, they enter into new virtual areas and require the associated mapping information in order to keep interacting with other participants. Therefore, when participants are about to enter in a new part of the VE, they send a unicast map request to their agent. These requests contain their current position in the VE, so that agents can send back to them the right mapping information. Note that participants have to anticipate their map request in order to obtain the mapping information before they enter in the new area. The anticipation time depends on the round-trip time to the associated agent, the participant velocity, and the size of its area of interest. Moreover, a mapping information is considered valid by a participant only for a short duration after its reception. Indeed, if a participant receives the mapping information, but finally decides to stay out of the corresponding area for a while, the mapping information for that area may change during that time (see Section 4.5). So, before entering a virtual area, a participant needs to check if its mapping information is still valid. If the difference between the current time and the reception time of the mapping information exceeds the *validity period*, a new map request is necessary. In case of a remapping, the agents which have sent a mapping information during the time corresponding to a validity period, re-send the new mapping information to the corresponding users.

Remapping request **message**

In LSVE, heterogeneity at the receivers implies that some users are able to interact simultaneously with a large number of other participants whereas other users are much more limited. However, both of them can be confronted with the situation in which the data rate received is about to exceed the maximum data rate they can handle. Two different reasons may lead to this situation:

- When the number of participants located in its area of interest exceeds the maximum number of participants it can handle.
- When the sum of the number of participants located in its area of interest and in its superfluous area (see Figure 3) exceeds the maximum number of participants it can handle.

In the first case, there is no way for our architecture to increase the satisfaction of the participant. The only thing that the participant can do is leave the multicast groups which map the "least interesting part" of its area of interest. In the second case, the participant could claim for a better mapping, i.e., a better cell-size. Indeed, with a smaller cell-size, its area of interest will be better approximated, and therefore, its superfluous area will be reduced. In this case, the participant sends to its agent a *remapping request* containing its virtual position.

4.5 Agents Algorithm Overview

Now, we describe the communication between agents, the mapping and the handover algorithms, then, evaluate performance of the mapping algorithm.

Agent-to-Agent communication

We assume that, at the beginning, agents only know the maximal velocity of the participants. Note that this assumption is realistic, considering that most of the time, the maximal velocity is an "application dependent" parameter. In our architecture, two kinds of information are used by the agents to partition the VE into zones and cells: map requests and remapping requests. Since a map request contains the virtual position of its sender, each agent is able to track the location of its connected users in the VE. In order to evaluate the density of participants within each zone, agents exchange information on the AMG multicast group. However, agents do not need to send the exact virtual position of their associated users. Only the number of users per zone is necessary to allow agents to compute periodically the density of participants per zone.

Remapping requests inform agents of the possible dissatisfaction of some of their connected users. As remapping requests also contain the virtual positions of their respective senders, agents can use these messages to define new zones where a more appropriate cell-size should be computed. In order to process all the remapping requests received per zone, each agent sends all its remapping requests received on the agent multicast group. Using these messages, agents can jointly decide when and where to modify the mapping in the VE.

Mapping algorithm

The same mapping algorithm is used by each agent. Basically, throughout the session, agents periodically compute the average density of participants per multicast group, by dividing the number of connected participants with the number of available multicast groups for the application. We refer to this density as the *remapping threshold* of the mapping algorithm.

Since a VE disposes of a limited set of multicast groups, the number of cells in the VE is also limited. So, the density of participants in the VE should be limited in order that agents are able to maximize the users satisfaction. In order to make a VE scalable for a large number of participants, the following solutions are possible:

- Build an extensible VE whose size adapts to the number of users so that the average density of participants in the VE always remains under a maximal threshold.
- Limit the maximum number of participants connected to the VE, and build a VE large enough such as the average density of participants in the VE never exceeds a maximal threshold.
- Use protocols such as the ones defined in MASC, to dynamically allocate more multicast groups to the LSVE, whenever the density of participant exceeds a maximal threshold. However, this solution only solves part of the problem by reducing the average size of superfluous area: as the density of participants increases, more and more participants have to reduce their area of interest in order to avoid packet loss or CPU overload.

As participants arrive and move in the VE, agents keep track of the density of participants in each of these zones. Two possible reasons can lead to the division of a zone into smaller cells:

- It is possible to find a smaller cell-size where the average density of participants per cell exceeds the remapping threshold.
- Remapping requests are sent by participants located in a given zone.

In the first case, agents can use the density of participants in the zone to compute a more appropriate cell-size, taking into account the number of available multicast groups. In the second case, agents first determine the distribution of remapping requests within the zone. If agents detect a concentration of "unsatisfied users" in only a part of the zone, then the zone is divided into smaller zones and a new cell-size is evaluated for each zone. However, before proceeding to a remapping, agents have to check if there are still enough available multicast groups.

Conversely, agents can decide to remap a zone using bigger cells. This remapping occurs when the density of participants per cell is smaller than the remapping threshold. Moreover, if neighboring zones contain only one cell (i.e., each zone is associated with a single multicast group), then agents can decide to merge these zones into a single one. This situation can occur if the resulting zone contains less participants than the remapping threshold. Nevertheless, two start zones can never be merged.

Handover management

This operation is certainly the most critical operation in LSVE. When agents decide to change the mapping in a zone, the participants located in that zone need to keep interacting with each other while they update their groups memberships. Here are the successive operations required to realize an handover:

- Agents elect a *designated agent*, which takes care of this operation. The designated agent is the one with the highest number of connected users involved in the handover. If several agents are candidates at the same time, a selection can be made based on their IP addresses.
- The designated agent joins all the current multicast groups associated with the cells of the remapping zones. Since each group maps a cell, the agent only sends in each group the mapping information relative to the neighboring virtual area of that cell.
- When a participant receives the new mapping information, it joins the new groups which map its area of interest. However, the participant waits for the time corresponding to the join latency [14], before sending in the new groups. Thus, when the first packet is sent to the group, the new multicast tree is already established between the participants.
- As the designated agent keeps receiving information on the old groups (the new mapping information might have been lost before reaching some end-user), it periodically resends the new mapping information on the old multicast groups.
- When the agent detects that no packet has been received for a given period of time on an old multicast group, it sends a *Free packet* to that group.
- When a participant receives a Free packet, it leaves the corresponding multicast group.

Performance of the mapping algorithm

In order to evaluate the performance of the mapping algorithm, we compared it with a static partitioning strategy by simulating a square VE with 511 participants and 144 multicast groups. First, the VE was partitioned into 3x3 square start-zones. For the static partitioning strategy, we divided the VE into 12x12 squares cells of the same size (i.e. 4x4 cells per start-zone). With the mapping algorithm, each start-zone was dynamically divided into square cells, depending on its density of participants. Note that as cells and zones are both squares, the number of cells per zone takes its value in $\{1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121\}$. However, the total number of cells in the VE was always less than or equal to the number of available multicast groups. For each partitioning strategy, we ran two major sets of simulations. In the first set, the 511 participants were randomly distributed in the VE according to a uniform distribution law. In the second set, we randomly distributed participants in each start-zone according to the same law, but we fixed the number of participants in each zone: the first zone contained 256 participants, the second zone 128 participants, the third zone 64 participants and so on. During these simulations, we considered that participants had different capabilities uniformly distributed between 15 and 35 sources. For example, if a participant was able to handle a maximum of 20 sources, but 40 participants were located in the cells intersecting its area of interest, then only half of its incoming traffic was received and processed. The mapping algorithm first computed an appropriate cell-size for each zone based on the density of participants, and then kept dividing the zones where the participants with the lowest satisfactions were located (see Section 4.1).

Figure 5 shows the distribution of participants in the different multicast groups, i.e., in the different cells of the VE. As expected, Figure 5 shows that with a

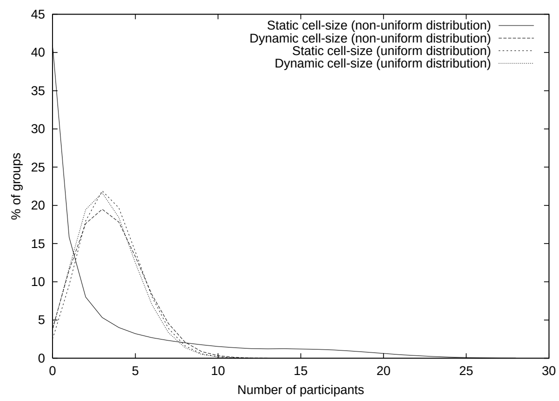

Fig. 5. Distribution of participants in multicast groups

uniform distribution, the static partitioning performs as good as our algorithm: in both cases, more than 95% of the groups contain between 1 and 10 users, with an average of 3.5 users per groups. This average number is equal to the ratio between the total number of participants and the total number of available multicast groups. Moreover, Figure 5 shows that the two strategies completely differ during the second set of simulation. Indeed, the static strategy reveals that more than 40% of the multicast groups contain no user and some other groups contain up to 25 users. However, the mapping algorithm allows approximately the same distribution of participants in multicast groups as in the first set of simulation. Therefore, it shows its adaptive capacity to VE, even with a limited number of available multicast groups.

In order to evaluate the satisfaction, we use the metric S defined in Section 4.1. The first simulation, done using a uniform distribution of participants, shows that more than 95 % of participants have a maximal satisfaction ($S = 1$) independently of the mapping strategy (static or dynamic). The second simulation, done with a non-uniform distribution of participants, is shown in Figure 6.

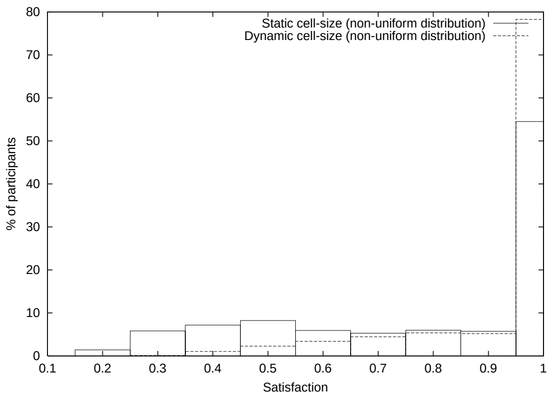

Fig. 6. Percentage of satisfaction

We note that about 80 % of participants have a maximal satisfaction using the mapping algorithm compared to less than 55 % with the static partitioning. Moreover, it is important to notice that in the case of a non-uniform distribution of participants, some participants receive more interesting data-rate that they can handle, especially the participants located within the zone with the highest density of participants. However, as the mapping algorithm computes a smaller cell-size in this zone, participants can gradually reduce their number of subscribed multicast groups by leaving the groups which map the "least interesting part" of their area of interest.

5 Related Work

There has been relatively little published work on the issue of evaluating grouping strategies for LSVE. [19] analyzes the performance of a grid-based relevance filtering algorithm that estimates the cell-size value which minimizes both the network traffic and the use of scarce multicast resources. However, the paper shows specific simulations done using different granularity of grids for several types of DIS entities, but the generic case is not studied. [16] compares the cost of cell-based and entity-based grouping strategies using both static and dynamic models but the paper does not propose any solution to calculate the cell-size value.

Different architectures using multiple multicast groups have already been designed for LSVE such as NPSNET[15], SPLINE[20], and MASSIVE-2[21]. [22] also suggests an approach for interest management using multicast groups. However, none of them have presented an architecture to dynamically partition the VE into multicast groups taking account the density of participants per cell and the participants capabilities.

6 Conclusion and future work

Although the current IP multicast model has a lot of imperfections to handle LSVE applications, we have described a communication architecture that enables us to run such applications on the Internet today. We have proposed and simulated mechanisms to dynamically partition a VE into cells of different sizes, depending on the density of participants per cell, the number of available multicast groups, and the link bandwidth and processing power available per participant. We believe that our framework is flexible enough to be easily adapted to new approaches to provide multicast such as *EXPRESS*[11] or *Simple Multicast*[10] and also to benefit from new functionalities like the future support for source filtering in IGMP[23].

Directions for future work include the use of a congestion control scheme for multicast UDP streams described in [24], the extension of the architecture framework to multi-flow sources, and the implementation and experimentation with a real LSVE application on the Internet.

References

1. J.M.Pullen, M.Myjak, and C.Bouwens, "Limitations of internet protocol suite for distributed simulation in the large multicast environment," *Request for Comments 2502*, Feb 1999.
2. Steven McCanne, Van Jacobson, and Martin Vetterli, "Receiver-driven layered multicast," in *Proc. of ACM SIGCOMM*, August 1996.
3. E. Lety, L. Gautier, and C. Diot, "Mimaze, a 3d multi-player game on the internet," in *Proc. of 4th International Conference on VSMM (Virtual Systems and MultiMedia)*, Gifu, Japan, November 1998.

4. B.N. Levine, J. Crowcroft, C. Diot, J.J. Garcia-Luna Aceves, and J. Kurose, "Consideration of receiver interest for ip multicast delivery," *Submitted for publication*, July 1999.
5. B.N. Levine and J.J. Garcia-Luna-Aceves, "Improving internet multicast with routing labels," in *Proc. of IEEE International Conference on Network Protocols*, October 1997, pp. 241–250.
6. E. Lety, T. Turletti, and F. Baccelli, "Cell-based multicast grouping in large-scale virtual environments," Tech. Rep. RR-3729, INRIA, July 1999.
7. M. Sola, M. Ohta, and T. Maeno, "Scalability of internet multicast protocols," in *Proc. of INET*, 1998.
8. S. Kumary, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, and M. Handley, "The masc/bgmp architecture for inter-domain multicast routing," in *Proc. of ACM SIGCOMM*, sept 1998.
9. S. Pejhan, A. Eleftheriadis, and D.Anastassiou, "Distributed multicast address management in the global internet," *IEEE Journal on Selected Areas in Communications*, pp. 1445–1456, October 1995.
10. R Perlman, C-Y Lee, A. Ballardie, J. Crowcroft, Z. Wang, T. Maufer, C. Diot, and M. Green, "Simple multicast: A design for simple, low-overhead multicast," *Internet Draft*, February 1999.
11. H.W. Holbrook and D.R. Cheriton, "Ip multicast channels: Express support for large-scale single-source applications," in *Proc. of ACM SIGCOMM*, August 1999.
12. S. Deering, "Host extensions for ip multicasting," *RFC-1112*, August 1989.
13. D. Waitzman, C. Partridge, and S. Deering, "Distance vector multicast routing protocol," *RFC-1075*, November 1988.
14. L. Rizzo, "Fast group management in igmp," in *Proc. of Hipparch'98 workshop*, UCL, London, UK, June 1998.
15. Michael R. Macedonia, Michael J. Zyda, David R. Pratt, Donald P. Brutzman, and Paul T. Barham, "Exploiting reality with multicast groups: A network architecture for large-scale virtual environments," *IEEE Computer Graphics and Applications*, vol. 15, no. 5, pp. 38–45, September 1995.
16. L. Zou, M.H. Ammar, and C. Diot, "An evaluation of grouping techniques for state dissemination in networked multi-user games," *Submitted for publication*, May 1999.
17. D.J. Van Hook, S.J. Rak, and J.O. Calvin, "Approaches to relevance filtering," in *Proc. of 11th DIS Workshop*, September 1994.
18. D. Bertsekas and R. Gallager, *Data Networks*, chapter 6, pp. 524–529, Prentice-Hall, 1987.
19. S.J. Rak and D.J. Van Hook, "Evaluation of grid-based relevance filtering for multicast group assignment," in *14th DIS workshop*, 1996.
20. John W. Barrus, Richard C. Waters, and David B. Anderson, "Locales and beacons: Efficient and precise support for large multi-user virtual environments," *IEEE Virtual Reality Annual International Symposium*, March 1996.
21. C. Greenhalgh, "Dynamic, embodied multicast groups in massive-2," Tech. Rep. NOTTCS-TR-96-8, Department of CS, The University of Nottingham, UK, 1996.
22. Howard Abrams, Kent Watsen, and Michael Zida, "Three-tiered interest management for large-scale virtual environments," in *Proc. ACM Symposium on Virtual Reality Software and Technology*, Taipei, Taiwan, 1998.
23. B. Cain, S. Deering, and A. Thyagarajan, "Internet group management protocol, version 3," *Internet Draft*, February 1999.
24. A. Clerget, "Tuf : A tag-based udp multicast flow control protocol," Tech. Rep. RR-3728, INRIA, July 1999.