An SSM-based communication architecture for large-scale virtual environments

Laurentiu Barza

To cite this version:

Laurentiu Barza. An SSM-based communication architecture for large-scale virtual environments. Networking and Internet Architecture [cs.NI]. Université Nice Sophia Antipolis, 2004. English. NNT : . tel-00311830

HAL Id: tel-00311830
https://theses.hal.science/tel-00311830

Submitted on 21 Aug 2008

UNIVERSITÉ de NICE - SOPHIA ANTIPOLIS
UFR SCIENCES

THÈSE

pour obtenir le titre de

DOCTEUR es SCIENCES

de L'UNIVERSITÉ de Nice - Sophia Antipolis

Spécialité : INFORMATIQUE

Présentée par

LAURENTIU **BARZA**

École Doctorale: STIC

Laboratoire d'accueil: INRIA Sophia-Antipolis

UNE ARCHITECTURE DE COMMUNICATION POUR ENVIRONNEMENTS VIRTUELS DISTRIBUÉS À GRANDE ÉCHELLE BASÉE SUR LES CANAUX MULTIPOINT

Thèse dirigée par Walid DABBOUS

Soutenue publiquement le 2 juillet 2004 devant le jury composé de:

Professeur	Michel	RIVEILL	Président
Professeur	Torsten	BRAUN	Rapporteur
Professeur	Ken	CHEN	Rapporteur
M.	Walid	DABBOUS	Directeur
M.	Philippe	DAX	Examinateur

To my parents.

Acknowledgements

Writing a PhD thesis is a long and challenging task. I am happy to use this space to thanks to the people which have been around me during this period of time, helping me in my work or enjoying spending their free time with me.

First of all, I would like to thanks to Walid DABBOUS, for supervising my thesis, for giving me freedom in my work, for his advise always very helpful and precise in moments of doubt. For his scientific, pedagogical and human qualities.

I would like to thanks to Michel RIVEILL who made me the honor to accept being the president of the jury of my PhD. thesis, to Torsten BRAUN and Ken CHEN for accepting the task to be advisors of my thesis, to Philippe DAX for accepting to be member of my jury.

The accomplishment of my thesis was possible due to the direct help of: Thierry Turletti who was always available to answer at my questions and to guide me in my work, Hitoshi Asaeda who helped me understanding better the multicast and to advance with my experiments, Alexis Gourdon for all the work around V-Eye and Thierry Parmentelat for his network support. It was a pleasure for me to work with all of you and I would like to express my gratitude.

Also, I would like to thanks to all the people being members of the Planete Research Team those years for their support, for the pleasant work environment, for all the interesting discussion we had which gave me better view of the world.

Not in the end, I an grateful to all my friends which make my staying here on French Riviera a great pleasure, for all the nice moments we have spent together, for all the mountain trips and photo sessions: Marc, Nelly, Robin, Gustav, Joanna, Samuel, André, Sergiu, Mihai, Sorin and many others. I also have a special thinking for my friends who are living now in different corners of the world and which are supporting me from far away, making my days happier when they send me a "hello", from time to time.

All this was possible due to the continuous support of my parents, they have always encouraged me in my life, in my work and in my studies.

Laureniu BARZA
Sophia-Antipolis
2 July 2004

Contents

List of Figures

Chapter 1

Introduction

The subject of this thesis is situated at the confluence of the virtual environments and the IP networking. It is about the possible existence over the Internet, in the near future, of heterogeneous, large-scale, distributed virtual environments, inhabited by millions of entities.

1.1 Context and motivations

Below, we present the context and the motivations of our work, from the networking point of view and from the virtual environments point of view. From the IP networking point of view, we assist at a continuous evolution and expansion of the Internet. We are in the context where more and more houses become connected each day to the Internet and in the same time the broadband comes closer to the final users. Here we think at the increasingly faster ADSL connections which replace the slow telephone-line connections.[1] So, we can predict that the speed of the ADSL connections will continue to increase in the next years. This evolution was accompanied, in the last two or three years, by the wide-spreading of a new class of application: the peer-to-peer application [1] which provide a new way for the file transfers (such as Napster, Gnutella, Kazaa, Freenet etc...). They have influenced each other: the peer-to-peer applications have convinced more people to subscribe to ADSL connections, while the existence of a large number of faster Internet connections has lead to larger peer-to-peer networks.[2] The existence of high speed connections arriving up to the end users computers facilitates them the access to more Internet resources. Higher quality web content, online games, chat and teleconferences, on-line music or video on demand are only a few types of applications and services made accessible by the high speed network connections. If we think at video streams, for example, the higher the end-user's connection

---

[1] In march 2004, 512kbps is the common speed of the ADSL connections in France, while, in the same time, the speed is about 50Mbps in Japan.

[2] The "Fast Track" network (Kazaa) had about 3.5 millions of users connected, in march 2004.

speed, the better video quality it is able to receive. As the days go by, more and more people have the possibility to exchange multimedia flows over the Internet. IP multicast is an ideal tool for exchanging high rate data traffic between groups of users. It saves the bandwidth by sending data packets only once over the same network interface regardless the number of receivers. Despite the fact that high speed network connections are deployed over the Internet, the use of multicast is still very important, if we think, for example, at a TV station sending data towards thousands of receivers. Unfortunately, a large number of deployment issues prevent the Internet "standard" multicast model, aka Any-Source Multicast (ASM), from becoming widespread over the Internet. At the moment we have begun this work, a simplified model of multicast, the Source-Specific Multicast (SSM), was proposed to alleviate the deployment problems of the ASM model. We have chosen to build our work around the SSM model, due to its good filtering capabilities, to its better deployment chances and with the conviction that multicast is the right tool to exchange multimedia flows between groups of entities. We believe that our work helps to illustrate the interest of using multicast and it contributes to a wider deployment of the multicast over the Internet.

Virtual environments (VE) allow multiple entities to interact in real-time regardless their actual location in the world. A large variety of applications fall in the category of virtual environments, some of them incorporating realistic 3D graphics and sound. We can mention here the military team training, collaborative design and engineering, multi-player games, virtual shopping malls and showrooms, on-line conferences, distant learning etc... Among them, two markets of billions dollars are represented by the games industry and the military training. The computers games industry growth is so important, that recently its income has over-passed the movies industry income. A classical game example is the "Quake arena", in which every user controls an avatar and it tries to shoot the other users' avatars. This game has a realistic 3D graphics and sound; on the other hand, the maximum number of users supported is small. Other online games, like EverQuest[3] or The Sims Online[4], feature thousands of users simultaneously interacting in a single virtual environment. They have less sophisticated graphics and sound, other game aspects have been enhanced here. For example, the longer the users stay connected in the game, the more "experience points" they might acquire. In those games, according to the VE users' behavior, there have been identified different users profiles: achievers, explorers, killers, socializers etc... For military applications, different virtual environments standards have been defined, such as: High Level Architecture (HLA) or Distributed Interactive Simulations (DIS). The virtual environments are used to train soldiers, for flight simulators, for virtual battlefields which may replicate the enemy territory, or for using simulated weapons

---

[3] http://www.everquest.com/

[4] http://www.eagames.com/official/thesims/thesimsonline

instead of real missiles (which costs thousands of dollars and which may accidentally injure somebody during the tests).

The complexity of networked applications is steadily increasing: they have to manage larger, widely-dispersed sets of heterogeneous users, they have to support multiple types of data streams sent over a number of different unicast and multicast flows, they have to respond at real-time constraints, they have to provide access control, to be consistent, reliable etc...

This continuously increasing number of participants connected in the virtual worlds emphasizes the importance of their scalability requirements. The VEs' users are located in a precise part of the world, they are not interested in every event produced in the VE and they are not able to process everything. From their point of view, a good filtering scheme enhances their connectivity in the VE, thus the applications' scalability.

The permanent evolution of the virtual environments requires an adaptation of their underlining communication architectures. The IP networks recent evolutions (concerning in particular the multicast of the peer-to-peer) may provide appropriate answers. In this thesis, we respond to the issues mentioned above by proposing Score-SSM, a Large-Scale Virtual Environment (LSVE) architecture. Score-SSM supports a large number of heterogeneous receivers connected using a Source Specific Multicast routing infrastructure, it delivers different types of multimedia flows and it effectuates a dynamic zone-cut of the virtual environment according to the participants density.

1.2 Thesis contributions

The main contributions of this thesis are:

- The elaboration of Score-SSM communication architecture. We have designed the first VE communication architecture based on the SSM multicast. This architecture uses a dynamic two-level filtering scheme which allows efficient data dissemination. The first level of filtering (grid-based filtering) ensures the scalability of the signaling and it is implemented through an hierarchy of agents. The second level of filtering (entity-based filtering) ensures the scalability of the data received by the participants and it is realized through a filtering mechanism implemented at the participants

- We have conducted a large number of experiments, which prove the feasibility and evaluate the performances of Score-SSM architecture compared to Score, an existent VE communication architecture.

  Although it is widely accepted that the ASM model is the best approach for many-to-many applications, our results show that an SSM-based communication architecture can achieve better performance than an ASM-based one at the cost of a marginal overhead in signalling, and can even provide additional benefits (like "fine" grain filtering).

- The implementation of Score-SSM, which was practically realized in parallel with the patching and the debugging phase of IGMPv3 protocol's implementation, over NetBSD, made by the Kame project[5]. Furthermore, this implementation was integrated in V-Eye, a virtual world application built in the Planete research team at INRIA Sophia-Antipolis. This gave us the possibility to test in real conditions the efficiency of the Score-SSM architecture.

1.3 Dissertation outline

This thesis is organized in five chapters. It has a monolithically structure, the reading of one chapter being based on the previous chapters' content. Chapter 2 presents the state of the art of the IP multicast and of the networked Virtual Environments, the two main bricks used to construct our communication architecture. Taking into account the characteristics of the multicast and of the Virtual Environments, we explain in Chapter 3 the choices we have made in order to design our communication architecture. Once we have specified the design choices, in Chapter 4 we present in detail Score-SSM, the multicast communication architecture that we propose for group communication in LSVEs. Chapter 5 presents a large number of tests, effectuated in order to prove the feasibility and evaluate the efficacy of Score-SSM. In the first part of this chapter, we present the performances comparisons made with Score architecture; in the second part, the role of the experiments was to find the best mechanisms and parameters settings for Score-SSM. In order to realize the experiments described in Chapter 5 we have realized an implementation of Score-SSM. The last part of Chapter 5 presents the integration of this Score-SSM implementation in a virtual environment application called "V-Eye". This application was developed in the Planete research team at INRIA Sophia-Antipolis, and it is originally based on the Score architecture. We present the interface built in order to allow V-Eye application to use Score-SSM communication architecture. Chapter 6 presents a summary of the work achieved and the conclusions of the results obtained during this thesis. Some perspectives are hinted for future works in the networked VEs.

---

[5] http://www.kame.net/

Chapter 2

State of the art

2.1 Multicast routing

On today's Internet the data packets are sent using IP network-level protocol and TCP/UDP transmission-level protocol. We call unicast the communication where one source transmits data towards one receiver. In addition to this one-to-one communication model, a group communication model was introduced for the first time in 1991 by Steve Deering [2]. This is the classical IP multicast. It is a many-to-many communication model: a number of sources send data to a particular group of receivers. The purpose of multicast routing is to allow an efficient communication service for the applications wishing to send identical data to multiple receivers, i.e. to reduce at maximum the network load. In practice, in the routers and at the sources, only one entering multicast packet is transmitted on the outgoing interface instead of sending a number of packets equal to the number of receivers situated downstream this interface. A multicast "group" is associated with a group address, identified by an IP class D address. In order to receive data sent over a multicast group, a participant has to become a member of that group, by joining it. On the contrary, in order to send data over a multicast group a source doesn't have to be member of that group. Anytime during a multicast session, new members may join the multicast group while older members may leave it. At this time, after years of evolution of the multicast, we can distinguish two models of multicast: a many-to-many model, the Internet "standard" multicast, also called Any Source Multicast (ASM) and a "simplified" version, a one-to-many model, called Source-Specific Multicast (SSM). In the following paragraphs we present some ASM multicast routing protocols, then we show the need for inter-domain multicast, we present some inter-domain solutions, then we point out the problems of the ASM model and we present the SSM model as the proposed solution for the Internet multicast.

2.1.1 The classical multicast routing

If a host wants to participate in a multicast session, first, it announces that to its local router, using IGMP protocol. Then the local router uses a multicast routing protocol to communicate with other multicast routers over the Internet, in order to enable the delivery of data from the source to the local host.

Multicast routing protocols construct data delivery trees over the Internet. The leaves of these trees are the receivers corresponding routers. After the trees construction, multicast routing protocols effectively forward the traffic coming from the sources toward the receivers. Messages are replicated only where the tree branches. Members of multicast groups can join or leave at any time, therefore the distribution trees must be dynamically updated. When all the active receivers on a particular branch stop requesting the traffic for a particular multicast group, the routers prune that branch from the distribution tree and stop forwarding traffic down that branch. If one receiver on that branch becomes active and requests the multicast traffic the router will dynamically modify the distribution tree and start forwarding traffic again.

According to the data delivery mode we distinguish two types of multicast trees: shortest path trees and shared trees.

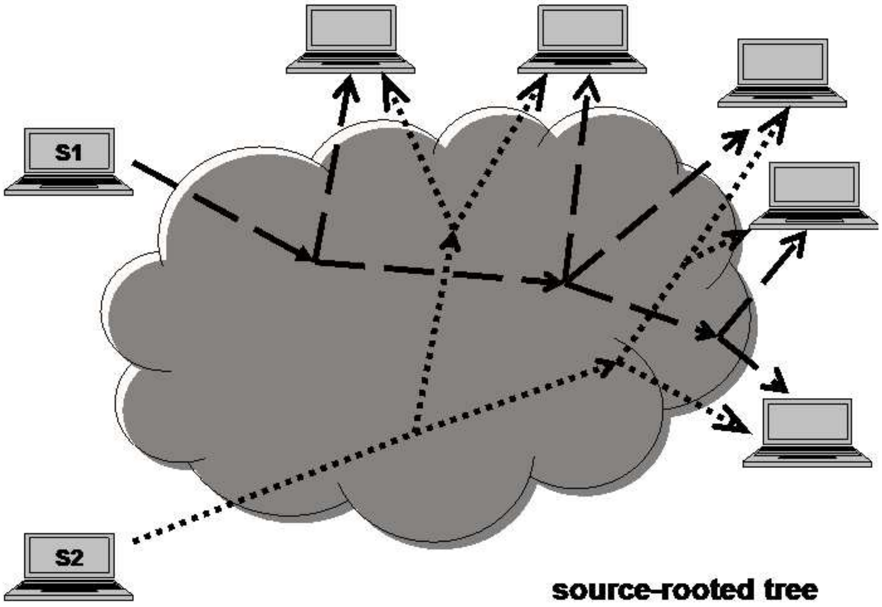


Figure 2.1: shortest path tree

Shortest path trees (figure2.1) are also called "*source-rooted trees*". These trees have the root at the source and for each member of the multicast group they contain the shortest path toward the source (this is true for a symmetric routing, i.e. the same routing path in both directions). This guarantees the minimum amount of network latency for forwarding the multicast traffic. We observe that for each source of a multicast group, a particular data delivery tree is constructed. A multicast group may have several sources, thus, more delivery trees may exist in parallel, for the same multicast group. Thus, the price to pay is that the routers must maintain path information for each source. In a network that has thousands of sources and thousands of groups, the size of the multicast routing table can quickly arise a resource issue at the routers.

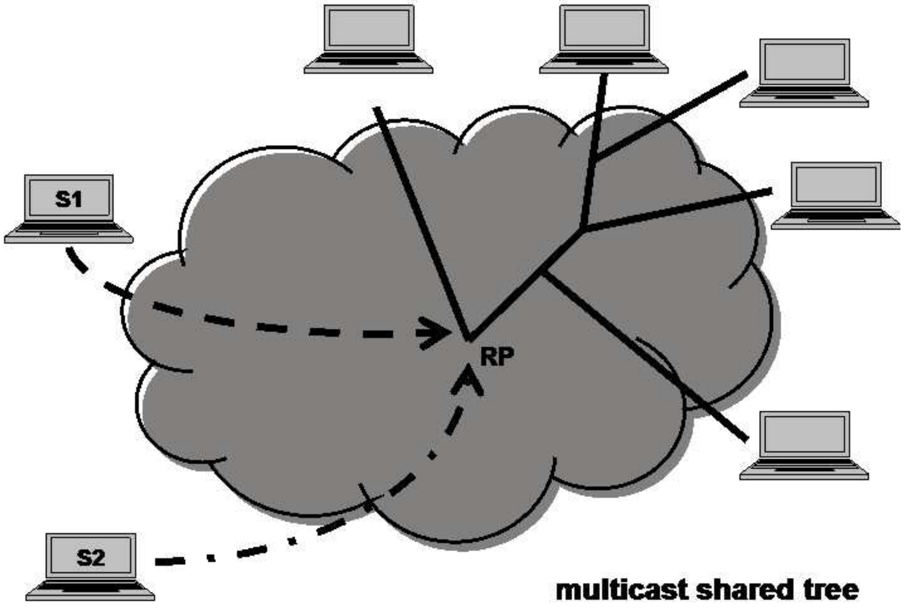


Figure 2.2: shared tree

Shared trees (figure2.2) are unique trees for a given multicast group. Thus, they require the minimum amount of state in each router. Once the members of a multicast group are known, the multicast routing protocol constructs a tree connecting all the members of a group. Any data sent towards that group uses this delivery tree to reach the group members. In this case, a concentration of the traffic will occur over certain network links (part of the shared tree). The disadvantage of shared trees is that under certain circumstances the paths between the source and receivers might not be the optimal paths-which might introduce some latency in packet delivery. In addition, the participants dynamics triggers the reconstruction of the delivery tree.

Depending on the construction mode of their delivery trees, we distinguish two types of protocols: broadcast-and-prune and explicit-join protocols. Broadcast-and-prune protocols (also called dense mode protocols) always use a shortest path tree rooted at a source. This class of protocols assumes that the majority of the hosts within a domain are receivers. In order to construct delivery trees, they broadcast packets to all the multicast routers within a domain and they require that routers explicitly prune the undesired traffic. Explicit join protocols, (also called sparse mode protocols), use either a shortest path tree or a shared tree. A sparse mode protocol assumes that the hosts within a domain do not necessarily desire to receive multicast traffic, thus it requires explicit joins from the receivers' host routers. A shared tree uses a core or a rendezvous point (RP) to bring sources and receivers together.

Before presenting some multicast routing algorithms, we should mention that the Internet is a collection of autonomous systems (AS). In order to deploy a global multicast solution we distinguish the cases of intra-domain multicast (inside an AS) and inter-domain multicast (between ASs). First, we present some intra-domain multicast routing algorithms and then we present two inter-domain solutions.

2.1.1.1 Intra-domain multicast

These multicast routing algorithms are deployed inside the autonomous systems. We classify these algorithms in two categories, according to the distribution tree types they construct: protocols which construct shortest path trees: DVMRP [3], [4], MOSPF [5], PIM-DM [6] and protocols which construct shared trees: CBT [7], [8] and PIM-SM [9] [10].

DVMRP (Distance Vector Multicast Routing Protocol) DVMRP is a broadcast-and-prune multicast routing protocol: the first datagram for a specific source and group address (S,G) is broadcasted across the entire network. As response, the leaf routers may transmit back prune messages, if there are no group members in their leaf subnetworks. These prune messages have the result of removing all the tree branches that do not lead to group members. In this way, source-rooted shortest path trees are formed to reach all the group members; one tree for every source of the multicast group. Routers in the domain keep temporary prune states for (S,G) pairs. For a big number of (S,G) pairs this may raise scalability problems. DVMRP is a dense mode protocol, it assumes that most of the AS members desire to receive multicast packets. DVMRP was never meant to work beyond a small autonomous domain because its flooding mechanism does not scale over the entire Internet [11].

MOSPF (Multicast Open Shortest Path First) The MOSPF multicast protocol is an extension of OSPF, a "link-state" unicast routing protocol. While distance-vector algorithms use simplistic best-route metrics (RIP uses just one - the number of hops) and know only about the directly attached neighbors, a "link-state" routing protocol uses a rich variety of metrics as well as a full topological model of the network within each OSPF-enabled router. OSPF routers require that a great deal more memory be at their disposal to create these topological maps. After a topology change, the map is re-calculated using a "Dijkstra" algorithm to provide the shortest path between any two given networks/subnetworks (hence the protocol's name).

OSPF, and thus MOSPF, uses a flooding technique to advise other OSPF routers of topology information. Unlike the RIP approach of sending the entire routing table in each flood, OSPF sends only the "delta" information, i.e. what has changed in the topology. The main difference between OSPF and MOSPF is the addition of a new message that provides multicast group-specific information to the router mesh.

A MOSPF router mesh could easily become computationally overwhelmed if a large number of sources and their users came joined at the same (or nearly the same) time. The dynamicity of groups membership jeopardize the utilization of this protocol. As a result, MOSPF is currently not a very viable IP-Multicast solution.

PIM-DM (Protocol Independent Multicast - Dense Mode) While DVMRP builds a multicast routing table to calculate which are the outgoing interfaces for a given (S,G) combination, PIM-DM blindly transmits the multicast packet to all interfaces, as long as that interface has not been pruned. PIM-DM accepts this additional packet duplication in order to operate independently of the unicast routing tables and their resultant topology. In addition, no parent/child databases need to be created using this very simplistic model. It was expected to deploy PIM-DM in resource-rich environments, such as a campus LAN where group membership is relatively dense and bandwidth is likely to be readily available. This did not happened, PIM-DM it is still an ongoing work, there exists no RFC [6].

The shortest path protocols presented above are well suited for areas with high density of listeners. In addition, using these protocols, routers which don't intend to send or receive multicast packets still need to take into account the routing messages. In opposition, shared tree protocols are suitable when the multicast receivers are sparse in a wide area with large diameter.

CBT (Core Based Tree) The CBT protocol constructs a single (bidirectional) tree shared by all members of the multicast group. The multicast traffic for the entire group is sent and received over the same delivery tree, regardless the source. A CBT shared tree has a core router around which the tree is constructed. The use of shared tree can significantly reduce the multicast state information stored at individual routers. Also CBT, conserves network bandwidth since it does not periodically broadcast multicast frames to all the AS routers. But, CBT may suffer of traffic concentration and cause bottlenecks near the core routers since all the traffic traverse the same set of network links. In addition, a single shared delivery tree may create suboptimal routes resulting in increased packets delay. Also, it raises problems like core router selection and dynamic placement strategies. All shared tree based protocols suffer of the existence of a single point of failure, at the core of the tree.

PIM-SM (Protocol Independent Multicast - Sparse Mode) PIM-SM by default constructs a single spanning tree rooted at a core Rendezvous Point (RP) for all the tree members within a domain. Local sources then send their data to this RP which forwards the data down the shared tree to interested local receivers. A receiver joining a host group can only specify interest in the entire group therefore will receive data from any source sending to this group. Distribution via a shared tree can be effective for certain types of traffic, for example where the number of sources is large, since forwarding on the shared tree is performed via a single multicast forwarding entry (as discussed above). However there are many cases where forwarding from a source to a receiver is more appropriate (smaller delays) via the shortest path. PIM-SM also allows a designated router serving a particular subnet to switch to a source-based shortest path tree for a given source once the source's address is learnt from data arriving on the shared tree.

PIM-SM is currently the most popular multicast protocol. It was designed to be used in wide area, where the receivers are sparsely distributed. While used over the Internet, it has some problems as: traffic concentration (every data messages and join/leave pass through the RP), third-party resource dependency (an available RP may be located far from the source, in this case receivers, even close to the sender, may expect data packets degradation), bootstrap scalability (the election of RP is made by periodically broadcast messages). According to these problems, PIM-SM could not be recognized as an appropriate multicast routing protocol to be used in a very large network. So, PIM-SM itself is considered as an intra-domain protocol, other solutions are required for inter-domain.

2.1.1.2 Inter-domain multicast

These multicast protocols are meant to work over different autonomous systems. Two solutions were proposed: the first is the use of PIM-SM in conjunction with MSDP [12] and MBGP [13], the second is the use of BGMP [14] as the standard routing algorithm for inter-domain multicast in conjunction with a multicast addressing mechanism.

PIM-SM / MSDP / MBGP The role of these three protocols in the inter-domain routing solution is the following:

- PIM-SM is the multicast routing protocol,
- MSDP is a mechanism to connect multiple PIM-SM domains,

- MBGP carries IP multicast routes within and between BGP autonomous systems.

PIM-SM was presented above. Below, we present the other two protocols composing this solution (MSDP and MBGP).

MSDP (Multicast Source Discovery Protocol) MSDP is a mechanism to connect multiple PIM-SM domains. MSDP allows multicast sources for a group to be known to all rendezvous points in different domains. Each PIM-SM domain uses its own RPs and need not depend on RPs in other domains. An RP runs MSDP over TCP to discover multicast sources in other domains. Once an active remote source is identified, an RP can join the shortest path tree toward that source and obtain data to forward down to local shared tree on behalf of interested local receivers. Designated routers for particular subnets can again switch to a source-based shortest path tree for a given remote source once the source's address is learned from data arriving on the shared tree.

MSDP has the following benefits:

- It breaks up the shared multicast distribution tree. The shared tree can be local to a domain. The local members join the local tree, and Join messages for the shared tree never have to leave your domain.

- PIM-SM domains can rely on their own RPs only, thus decreasing reliance on RPs in another domain. This increases security because it can prevent the local sources from being known outside their domain.

- Domains with only receivers can receive data without globally advertising group membership.

- Global source multicast routing table state is not required, thus saving on memory.

The problem is that MSDP doest not scale because of its periodic flood and prune mechanism. Also MSDP prevents the use of shared tree between domains: when remote RP's receive active source messages, they join directly to the source and not the RP of the source. Even when two sources are co-located in the same domain, RP's in remote domains will form two separate per-source branches, one to each source. MSDP depends on MBGP for inter-domain operation.

Multiprotocol Border Gateway Protocol (MBGP) MBGP is run between BGP peers to exchange a second set of IP routes which are used for IP multicast reachability. Thus, MBGP allows the existence of a unicast routing topology different from the multicast routing topology. This gives to the network administrators more control over their network and resources because different policies (BGP filtering configurations) may be applied.

For PIM, MBGP is the preferred solution to achieve non-congruent multicast and unicast topologies. MBGP used together with MSDP is the current deployment solution to connect PIM-SM domains.

BGMP (Border Gateway Multicast Protocol) The second solution for inter-domain multicast is BGMP. It is a hierarchical multicast routing protocol that uses border routers acting as nodes in the routing tree. It constructs bi-directional shared trees between domains and it relies on MAAA (Multicast Address Allocation Architectures) [15] [1]protocols or GLOP (static addressing) [20] to solve the address allocation and to designate the core-domains of the multicast groups. Such conceptual change implies a big deployment effort. When it was first introduced, BGMP was presented as the long-term solution for the inter-domain multicast, a solution which will avoid the scalability issues of MSDP. Meanwhile, the proposal of the SSM model, with smaller complexity and bigger chances of deployment, have postponed (if not definitively suspended) any implementation efforts for BGMP.

Deployment status of the inter-domain multicast The PIM-SM/ MSDP/ MBGP architecture has been deployed in IPv4 and IPv6 networks. It is particularly effective for groups where sources are not known in advance, when sources come and go dynamically or when forwarding on a common shared tree is found to be operationally beneficial. Two networks that deploys this solution are GÉANT[2] and RENATER[3]. The GÉANT project is a collaboration between a consortium of 26 National Research and Education Networks across Europe, the European Commission, and DANTE. The project's purpose is to create a new backbone at gigabit speeds - the GÉANT network. Since the beginning of operational service on GÉANT, a range of services has been made available on the network as: IPv6, IP Quality of Service, IP Multicast or Virtual Private Networking.

2.1.1.3 Problems of the ASM model

There are various problems in deploying the ASM model [11] [19] [21] some of them coming from the architecture itself, while others are due to the complexity of the protocol architecture.

- Security. In ASM model, while joining a group, a receiver cannot specify which specific sources it would prefer to receive data from. A receiver gets all the data sent over the group it has subscribed. This lack of access control can be exploited by

---

[1] MALLOC is three layered, comprising a host server protocol (MADCAP) [16] an intra domain protocol (Multicast AAP) [17] and an inter domain protocol (MASC) [18]. This architecture turned out to be extremely complex, not being capable to guarantee address availability in any address range. [19]

[2] http://www.dante.net

[3] http://www.renater.fr/

malicious transmitters to disrupt data transmissions from authorized transmitters. Security should be assumed at higher levels.

- Deployment complexity. The ASM protocol architecture is complex and difficult to manage and to debug. Most of the complexity arises from the RP-based infrastructure needed to support the shared trees, and from the MSDP protocol used to discover sources across multiple domains. These challenges often make network operators reluctant to enable IP multicast capabilities in their networks, event though most of today's routers support the IGMP/PIM-SM/MSDP/MBGP protocol suite. As we have seen above, some networks already deploy these protocols.

- Address allocation. This is one of the core deployment challenges posed by the ASM service model. The current multicast architectures do not provide a deployable solution to prevent address collisions among multiple applications. A static address allocation scheme, GLOP has been proposed as an interim solution for IPv4; however GLOP addresses are allocated per AS and in practice the number of sources may exceed the AS numbers available for mapping. Proposed long-term solutions such as the Multicast Address Allocation Architecture (MAAA) are perceived as being too complex (with respect to the dynamic nature of multicast address allocation) for widespread deployment.

- Inter-domain scalability. MSDP has weakness in terms of security and scalability. For security, it is susceptible to denial-of-service attacks by domains sending out a flood of a source announcements. For scalability, MSDP is not well designed to handle large numbers of sources. The primary reason is because the source announcements were designed to be periodically flooded throughout the topology. With the increasing number of sources over the Internet, MSDP will generate greater amounts of control traffic.

- Single point of failure. The use of shared trees in some multicast routing algorithms may raise the problem of failure of the core which can lead to complete breakdown of multicast communication. In the ASM protocol architecture a receiver is always grafted on to an RP-based shared tree when it first joins a multicast group. This reliance on the shared tree infrastructure makes the ASM protocol architecture fundamentally less robust. Additional cores may alleviate this problem, at the price of more overhead and complexity.

All the above ASM model problems have lead to the proposition of a new multicast model, a simplified version, which is the Source Specific Multicast (SSM).

2.1.2 Source-Specific Multicast

The network layer service provided by SSM is a "channel", identified by an SSM destination IP address G and a source IP address S. An IPv4 (also IPv6) address have been reserved for use by SSM service. A source "S" transmits IP datagrams to an SSM destination address "G". An end-host can receive these datagrams by subscribing to the multicast channel (Source, Group). Channel subscription is supported by the version 3 of the IGMP protocol [22] for IPv4 (MLDv2 for IPv6 [23]). The inter-domain distribution tree which forwards the IP multicast datagrams is rooted at the source "S" and is constructed using PIM-SM. SSM is well-suited for dissemination style applications with one or more senders, whose identities are known before the application begins. The SSM service model alleviates the following deployment problems encountered in the ASM case:

- Security. SSM provides an elegant solution to the access control problem. Only a single source S can transmit to a channel (S, G), where G is an SSM address. This makes is significantly more difficult to spam an SSM channel than an ASM host group. In addition, data from un-requested routers are not forwarded by the network, thus reducing the network consumption.

- Deployment complexity. The distribution tree for an SSM channel (S, G) is always rooted at the source S. Thus there is no need for a "shared" tree infrastructure. This makes the SSM protocol architecture less complex than that for ASM, making it easy to deploy. In addition, SSM is not vulnerable to RP failures or denial-of-service attacks on RP(s). The "sources" are expected to be more reliable to the failures. Anyway, if the source fails there will be no data flow.

- Address allocation. SSM defines channels on a per-source basis, so these addresses are local to each source. This avoids the problem of global allocation of multicast groups' addresses and makes each source independently responsible for resolving address collisions for the various channels that it creates.

- Inter-domain scalability. The domains are "transparent" in the SSM model: a receiver directly sends a "subscribe" request to the source. MSDP is not required in the SSM model in order to advertise the sources IP addresses. The receivers have to know the IP addresses of the sources before subscribing to an SSM channel.

- Single point of failure. The SSM model doesn't use intermediate "rendezvous" nodes or core nodes which have the potential to become single points of failure. The distribution trees are rooted at the sources: of course if the source fails, no data will arrive to the receivers, because there is no data sent.

In order to deploy the SSM service model, besides PIM-SM (in fact only a small subset of the full PIM-SM protocol functionality is needed), the IGMPv3 protocol is required.

IGMPv3 (Internet Group Multicast Protocol) allows end-hosts to register their interest in a multicast channel to their corresponding routers. IGMPv3 supports "source filtering" i.e. the ability of the end systems to express interest in receiving data packets from only a set of specific sources, or from all except a set of specific sources. Thus, IGMPv3 provides a superset of the capabilities required to realize the SSM model. The routers periodically query the LAN to determine if known group members are still alive. If there are more than one router on the LAN performing IP multicasting, one of the routers is elected "querier" and it assumes the responsibility of querying the LAN for group members. Based on the group membership information learned from the IGMP, a router is able to determine which (if any) multicast traffic needs to be forwarded to each of its "leaf" subnetworks.

There are 3 versions of the IGMP protocol :

- In the first version, the host leaves a group without announcing the router, which in exchange, transmits periodically a "querry" to see if there are any members of the multicast group. If there are no other members of the multicast group, the router prunes its corresponding branch of the forwarding tree.
- The second version [24] introduces "explicit leave" operation from the part of the leaving hosts, in order to lower the leave latency. In addition, it introduces a group-specific "querry" message, and a procedure for election of the multicast querier for each LAN.
- The third version, which is required for the SSM model, introduces group-source "report" messages: a host has the possibility to specify the sources (of a particular multicast group) from which it desires to receive traffic.

SSM deployment status We cite from the Internet 2[4] initiative: "The participants in this effort have agreed to first target support for Source Specific Multicast. The primary reason is that SSM support involves support for fewer protocols than Any Source Multicast. SSM support requires IPv6 support for MBGP, PIM, and MLDv2. ASM additionally requires MSDP and PIM RP support." This illustrates the option of the Internet community to deploy the SSM model as the first step in deploying the multicast over the Internet. We have seen in the ASM section that PIM-SM and MBGP are already deployed in some networks. For the hosts point of view, IGMPv3 becomes widely available: it is delivered with Windows XP and Linux Red Hat operating systems. Also an implementation

---

[4] http://www.internet2.edu/

of IGMPv3 exists for NetBSD and FreeBSD. In most of the Internet routers, the existing IGMPv2 implementations have been replaced by IGMPv3.

2.2 Networked virtual environments

A Large Scale Virtual Environment (LSVE) application connects participants on the Internet from different locations by offering them a virtual overlay "world". The LSVE applications are characterized by a shared sense of space, a shared sense of presence, a shared sense of time, a way to communicate, a way to share, and a way to get in touch. The figure 2.3 shows the evolution in time of the VEs and of the users' implication in the virtual worlds, starting with the very first groups of user communicating online in MUDs[5] and going until the extreme represented in the film Matrix, where the users are totally immerged in the VE.

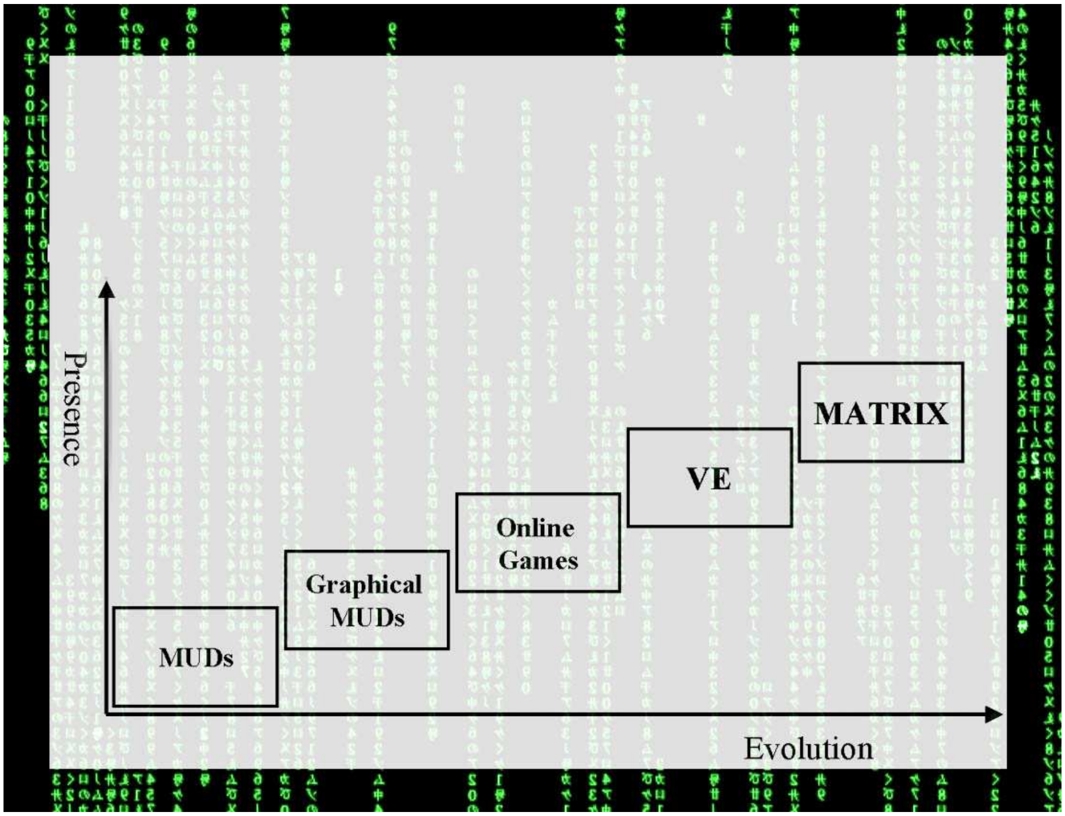


Figure 2.3: The evolution of the virtual environments and the implication of the users

In our work we are focusing on a particular case of LSVEs where the emphasis is more on collaboration between users than on simulation (in the literature it is known as Collaborative Virtual Environments). In this section we present some VEs architectures, different data filtering approaches and VE scalability issues.

---

[5] A MUD (Multiple User Dimension, Multiple User Dungeon, or Multiple User Dialogue) is a computer program which users can log into and explore. Each user takes control of a computerized persona/avatar/incarnation/ character. They can walk around, chat with other characters, explore dangerous monster-infested areas, solve puzzles, and even create their very own rooms, descriptions and items.

2.2.1 VE architectures

The state of a VE at a given moment depends on the states of all the participating elements at that moment. The information (or knowledge) in a VE, at a precise moment, can be global, centralized or distributed, according to the specific architecture of the VE. In the global case, each participant has the knowledge of all current states in the VE. This is not feasible for large number of participants, or for dynamic worlds, or for virtual worlds where a big amount of data is exchanged, due to the scalability and limited network resources (large amount of unnecessary data will flow to the participants). Centralized worlds assume that a central server has the knowledge of the virtual world, but this supposes that the server will be in contact with all the participants in order to update VE state: this is not scalable with the number of the participants due to the centralized architecture and limited network resources. Distributed knowledge assumes that the participants are interested only by the communication held in a part of the VE and they keep the current state of the VE only for their area of interest. This also supposes that, during participant's movements in the VE, the area of interest will be updated according to it. This leads to the problem of area of interest definition (how do we do it?) and how large can be this area? (A larger area of interest means additional unnecessary traffic, a small area of interest means more "handovers", so additional control traffic). So a trade-off arises between trying to reduce unnecessary data traffic and trying to reduce the signaling traffic. We can imagine that the knowledge of a particular area of the VE is kept by the participants who are present in that area (or in proximity of the area). In fact, those are the participants who "modify" that area, with their movements and their actions. A complete distributed architecture will not scale for simple tasks like: finding if a particular participant is connected in the VE at a precise time.

2.2.2 VE performance

As VEs grow, in terms of number of participants and network latency, the scalability and the interactive performance become key issues. Network bandwidth becomes the keystone of the networked VEs [25]. The bandwidth use increase with every new participant arriving in the VE: each participant receive updates from the participants already connected in the VE, each new participant introduces more shared state in the VE (at least its position, orientation, graphical representation) and new interactions with the participants connected in the VE. In [26] is introduced the Networked-VEs Information Principle:

The resource utilization of a net-VE is directly related to the amount of information that must be sent and received by each host and how quickly that information must be delivered by the network.

The parameters used to compute the resource utilization are: the number of messages sent, the average number of receivers for each message, the average message size, the frequency of these messages and the computing power required to process these messages. In order to reduce the resource utilization, a large variety of techniques have been developed, by modifying one or more of the above parameters. Among them we cite: area of interest filtering, the use of multicast, packet aggregation and compression, reducing the level-of-detail etc...

2.2.2.1 Filtering in the VEs

As the VEs grow in size, the filtering of the data of no interest for the receivers become very important. This filtering is also known as "interest management" and its purpose is to reduce the number of messages exchanged among the users by limiting their scope in the VE. Different interest management techniques have been proposed. We classify these techniques in tree big categories:

- **Spatial-based.** A participant communicates with the entities located within its range (filtering according to the proximity in the VE). We distinguish three types of spatial-based filtering:
    - ***Grid-based*** filtering. The world is divided into cells. Each cell has an associated multicast group. Each entity has a "sending" and a "receiving" set of cells. Examples of grid-based VE are: NPSNET [27], SPLINE [28], and SCORE [29].
    - ***Entity-based*** filtering (or aura-based). Each entity performs a relevancy test in order to select which data it will receive (for example, using a range-based relevancy test it may choose to receive data only from the entities situated closer than a given distance). It is possible to receive data only from a given entity, because each entity uses its own multicast group in order to send data. Examples of aura-based VE are: MASSIVE [30][31].
    - ***Hybrid.*** They combine grid-based filtering with entity-based filtering. An example of hybrid VE is VELVET [32].
- **Class-based.** The objects of the VE are partitioned into classes. The participants register their interest in some classes and thus they communicates only with objects from the registered classes. An example is the High Level Architecture (HLA) [33]. It specifies functional requirements, but not the hardware, software or the network architecture. HLA filters object and interaction classes and the attributes of object classes. In addition, HLA defines a value-based filtering which use routing spaces and regions.

- **Hybrid.** In [34] is presented a three-tiered interest management system which partition the VEs data flows into fine grain multicast groups. Three filtering schemes are combined: a group-per-region scheme, a group-per entity scheme and a per-protocol scheme.

2.2.2.2 Grid-based vs. Entity-based filtering

As cited above, many VE propositions implement a spatial-based filtering. Many research works evaluated the efficiency of the grid-based and the entity based filtering [35] [36] [37] [38]. Below, we describe the main characteristics of these two filtering approaches.

Grid-based filtering improves the scalability by keeping the update messages within a region of a VE.

A multicast group address is assigned for each region. State updates are transmitted to the multicast address of the region in which an entity is located. Thus, updates are received from an area of the VE by joining the multicast groups for the regions that compose the area.

This leads to traffic concentration for every region of the VE and superfluous traffic is received by the users (the area of interest of an user doesn't necessarily coincide with the region of the VE in which it is located). The receivers have no explicit knowledge about the sources, the relevant information is determined by the cells of the grid that fall within the area of interest. In grid-based VE, the users have limited possibilities to express their interest [6], so their heterogeneity requirements are only partially fulfilled.

If the ASM model is used, the number of group addresses is limited, thus the number of regions composing the VE is limited. This has an impact over the scalability of the VE.

In **entity-based** filtering, the relevant data is selected via a relevancy test. An example of relevancy test is the distance in the VE: every user has its area of interest surrounding its location in the VE. When the area of interest of two users collide they start to communicate.

A key concept is the existence of a subscription agent [35]. Its role is to perform relevancy tests on behalf of the users. In addition, the agent may arrange the delivery of the relevant data to the receivers (for example, by providing multicast group addresses to the receivers). The subscription agent must have some degree of global knowledge of the simulated environment in order to perform relevancy tests. For example, the information about the location of every entity in the VE, if we consider the Euclidian distance as relevancy test.

---

[6] They receive aggregated flows from the users located in the same region, but the existence of several aggregated flows (ex. one video flow, one audio flow) in the same region of the VE is possible.

The existence of the agents and their requirement for global knowledge presumes a low-rate broadcast of the state of the entire simulated world to support relevancy testing (for example, the existence of a "radar" entity in the VE will require information about large areas of the VE). Another drawback is that the relevancy tests may be computationally intensive (for example the Euclidian distance is an O(N*N) problem). However these problems may be alleviate by using other techniques, as grid-based filtering, to reduce the number of irrelevant entities.

If every entity uses a multicast group(or more) to send its data traffic, it is clear that the total the number of multicast groups allocated is more important than the number of multicast groups used in grid-based approaches.

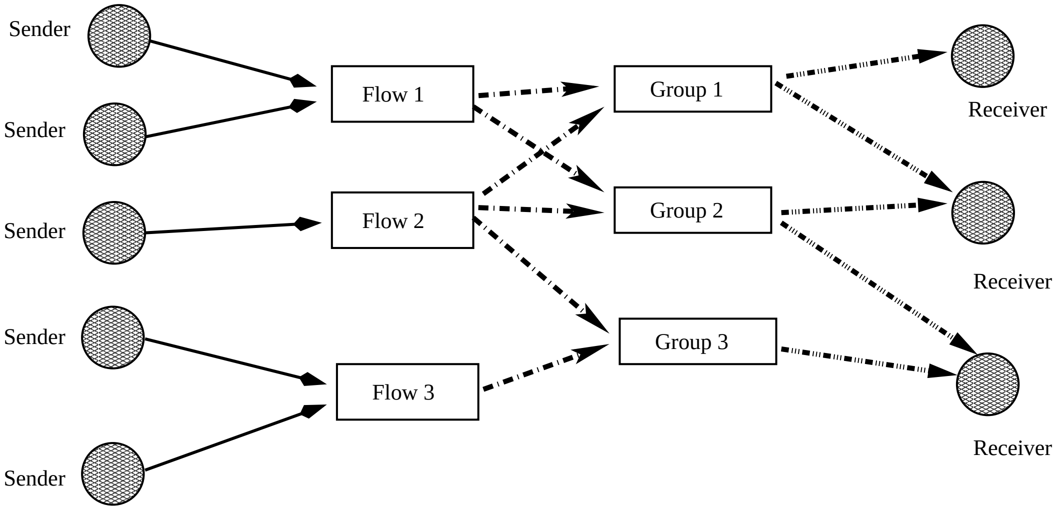


Figure 2.4: The "channelization" problem

2.2.2.3 Mapping data flows to multicast groups

In VEs, a large number of information flows must be delivered to a large number of information receivers. However, the receivers have different interests: not all receivers are interested in all of the information flows. Multicast offers the opportunity to deliver a subset of the information flows to a subset of the receivers. Having a limited number of receivers available, the "channelization" problem is to find an optimal mapping of information flows to a fixed number of multicast groups and a subscription mapping of receivers to multicast groups in order to minimize a cost function involving the total bandwidth consumed and the amount of unwanted information received by the receivers (figure 2.4). In [39] it was conjectured that the channelization problem is a computationally hard problem. In [40] two versions of the channelization problem were defined: a given flow can be assigned to multiple multicast groups or to only one multicast group. This paper proves that both problems are NP-complete. The paper also treats the "subscription problem" i.e. a component of the channelization problem in which the information-flow-to-multicast-group mapping is predetermined, and only the receiver-to-multicast-group subscription question is considered. It is shown that the subscription problem is NP-complete when one flow could be assigned to multiple multicast groups, while the problem can be solved in linear time when one flow is restricted to belong to only one multicast group. So, the channelization approach provides only a roughly linear improvement in delivery efficiency while it potentially scales exponentially with the number of users.

Different approaches, which try to reduce the delay caused by the "join/leave" to multicast groups operations, have been proposed.

SANDS [41] is based on the use of "active interest" filtering, which is a publish/subscribe mechanism that uses active networks technology to install and to control dynamically-established content-based filters, in intermediate active routers in an IP multicast distribution tree. It provides scalable and efficient mechanisms for dynamic information dissemination within applications such as distributed simulation. In addition, it offers receiver-driven data filtering, the rapid change of the multicast content sent over a forwarding tree by modifying a filter, congestion avoidance by to lowering the received data traffic through the use of filters.

Router level filtering (RLF) [42] is a solution that allows receivers to customize the data they receive without the drawback of joining a new group. It assumes that a LSVE session relies on few stable groups and that a participant will have to execute fewer joins and leaves. On the other hand, the receiver asks the routers supporting the distribution of a particular multicast group to filter data. This approach is faster than the classic join procedure and is independent of the multicast protocol architecture. It provides a solution for flow and congestion control because the filters are determined by the receivers, at the application level.

2.2.2.4 The communication module in several existing VE

Several LSVE architectures using multicast to implement a spatial filtering have been proposed.

NPSNET-IV [27] was the first DIS application to use the IP multicast. The VE is statically divided into hexagonal regions and a group leader maintains a list of all the entities within the region. Each participant sends data to the multicast group associated with the zone that covers his current location. To receive data, the participant subscribes to the multicast groups associated with the zones intersecting his area of interest.

Spline [28] uses "beacon servers" to learn about the participants located nearby in the VE. Each beacon server has a multicast group associated in order to receive data about the participants located in its corresponding region.

MASSIVE [43][44][45] implements a self-configuring hierarchy of multicast groups with aggregated data transmissions from children to parents in the hierarchy.

In DIVE [46] the objects of VE are composed in a hierarchy. A set of hierarchical multicast groups is associated with this hierarchy of objects.

Score [47] is one of the first communication architectures proposed for LSVE applications in the Internet. It uses a mechanism to dynamically partition the VE into cells of different sizes, based on: the density of participants, the link capacity and processing resources of each participant, and the available number of multicast groups. This dynamic partitioning of the VE is done by a transport-layer filtering mechanism using multiple agents. SCORE makes the assumption that the participants use ASM model, and that they generate a single type of data flow. Score will be presented in details in the Chapter 4.

In VELVET [48] each avatar is able to "see" whatever is located within its area of interest. The avatars can dynamically enlarge or reduce their area of interest, taking into account, for example, the density of the objects around the avatar. The area of interest is composed by two rings: a smaller one (the area of interest Check-In) and a larger one (the area of interest Check-Out). Their role is to avoid multiple "check-in", "check-out" operations for the objects crossing the border of an area of interest. Each avatar uses a personalized set of metrics (there are two area of interest rings for each metric), the objects are placed in these rings according to the metric chosen by that participant. The VE is divided into areas, each o these area has a multicast address. In addition, each object which generates a flow of data has assigned a multicast address. VELVET may experience scalability problems due to the static partitioning of the VE into area, which doesn't take into account the local participant density.

In this chapter, we have presented various technics for reducing the resources utilization in the VEs, more precise, the use of multicast to implement spatial-based filtering approaches (grid-based and/or entity-based filtering). Most of the proposed VEs use only one of these filtering mechanisms, having scalability problems and not taking into account the dynamics of the participants. In addition, the specificity of each VE leads to different requirements, in term of communication, heterogeneity, real-time constraints etc... The participants requirements are different in an on-line game compared to a military training VE, or compared to a multimedia chat VE, or to a virtual shopping mall. This specificity influences the design of the communication layer for every VE. In the next chapter we study the requirements in building a VE capable of supporting millions of simultaneously connected participants exchanging multimedia flows (this kind of flows have bandwidth and real-time constraints).

Chapter 3

Design issues

In this chapter we explain the problems that we have taken into account, in order to design a LSVE communication architecture which, in the same time, is scalable, supports very heterogeneous participants and responds at the real-time requirements of the multimedia flows.

3.1 Overview

We present the hypothesis that we have assumed in order to construct our communication architecture and an LSVE application implementing this architecture:

a. **Multicast.** The first option was to take the advantage of SSM as data delivery model and as network-level filtering mechanism. In fact, one of the roles of our LSVE application is to prove to the VEs users (and owners) that the use of multicast for data delivery leads to more scalable architectures (i.e. more users can be simultaneously connected, the users receive less superfluous data, the signaling is reduced etc...). In order to provide the same services and functionalities to the final users, a LSVE communication module over SSM will be more complex than a communication module over ASM.

b. **Heterogeneity.** According to the characteristics of the participants connected to the Internet, the LSVE application should support different quality flows depending upon the capabilities of the participants (network bandwidth and computing power). In other words, a strong participant's heterogeneity support must be provided by the communication architecture.

c. **Filtering and scalability.** The communication architecture should efficiently integrate the specific filtering demands of the participants with the global scalability requirements. The participants need a way to express or to change their own momentary interest in a zone of the VE or in the communication quality level with a particular neighbor. In the same time, the purpose of the network architecture is to simultaneously accept as many participants as possible without degrading the communication quality of the participants already connected in the VE.

d. **Real-time constraints.** One main role of an LSVE application is to support the information exchange between the participants. Audio and video flows, for example, are among the most used data flows types in the communications of the participants. These are real-time communication flows which require low communication delays and a certain amount of available bandwidth. This imposes the requirement for a communication architecture which builds shorter routing paths.

3.2 Design hypothesis

In the following paragraphs we enter into details and explain the choices we have made.

a. Multicast support for LSVE. The multicast has been proposed, from long time, to be used in LSVE applications to deliver the high bandwidth consumption flows. The ability to send only once a video packet, for example, and this packet to be received by 10 different participants, brings important gains in the network bandwidth used by the participants. At the moment we have started our work, the ASM model was contested, the deployment and the scalability problems were preventing it, for years, to become widespread over the Internet. In the same time a simplified multicast model, the SSM model, was newly proposed. This model promises better chances of deployment due to its simplified structure.

From the network point of view, the SSM model is situated between the ASM model and the peer-to-peer communication (i.e. one-to-many is situated between many-to-many and one-to-one). SSM model is a simplified service of the classical ASM. While in ASM everybody may send data to a multicast group (it is not necessarily to be a group member), in SSM a group is constructed around a particular source. The SSM's reduced service fits perfectly for some types of applications (ex. video on demand), but for other types of applications (ex. whiteboard, chat...) additional communication modules are required. So, this network layer simplicity comes at the price of additional complexity at the transport and application layers. An important difference between the ASM model and the SSM model is the rendezvous point requirement. To become member of an ASM group "G", an user only needs to get the IP multicast address of that group. To become member of an SSM channel "(S,G)", an user needs to get the IP address of the source "S", in addition of the multicast group address "G". In order to receive more sources for a given multicast group address "G", an user has to get the addresses of each of these sources. So, the use of the SSM model instead of the ASM model requires the implementation of additional rendezvous capabilities in the applications.

In order to provide the same services and functionalities to the final users, the complexity of the LSVEs' communication architectures (we refer here at the application-level part) varies according to the underlying network model. As we have seen above, an LSVE over the SSM requires the implementation of more functionalities in the communication module than an LSVE over the ASM. In the same time, the use of peer-to-peer requires the complete implementation of the communication architecture at the application level, the network level provides only the unicast packets delivery.

In the state of the art chapter, we have seen that some of the ASM model problems are avoided by the use of new SSM model. We just enumerate these problems here: access control, address allocation, deployment complexity, inter-domain scalability, single point of failure. In addition, SSM builds source-rooted delivery trees. They avoid the traffic concentration and they are in the same time shortest-path routing trees which is beneficial for the end-to-end delays (very important for real-time flows like video and audio flows).

b. Participants heterogeneity The LSVE may contain a large number of participants connected from all over the Internet. These participants are connected using different network bandwidths links and they may have different computing power machines. This heterogeneity in terms of bandwidth and computing power may be completed by the "heterogeneity" of the quantity of details desired, the desired consistency-level and the data flows types a participant communicates.

This important heterogeneity of the participants must be taken into account by the VEs in order to lead at an efficient use of the resources of the participants. The VE architecture and mechanisms must find the best solution between the satisfaction of the participants, the bandwidth used and the number of entries memorized in the routing tables. Two parameters characterize the profiles of the participants in the VE: the area of interest and the capacity. The participant's area of interest expresses the presence of this participant into the VE. The participant's capacity expresses the limit in receiving data and processing in real-time the information received. During the participant's connection time in the VE, these parameters may change several times, the area of interest may shrink or it may increase, and also the participant's capacity. Thus LSVE architecture should take into account this dynamicity. In addition, at any time participants (with different quality requirements) can connect or leave their LSVE session. So, the heterogeneity coupled with the dynamicity is a very important issue in designing a LSVE communication architecture. Let's consider in a VE, a source which sends data toward two receivers with different capabilities. If the data rate overpasses the capacity of the low capacity receiver, this receiver will not be able to get the data flow without losses. In the same time, if the data rate is lower than the capacity of the high capacity receiver, this receiver will not benefit of the full information quality, as its capacity give it the possibility to do. So, we observe that a unique data flow does not satisfy both receivers. The solution is to send the data using different quality flows, for example sending layered video. Each participant chooses the data flow quality according to its capacity.

c. Filtering and scalability. The filtering of the data received by the participants is directly linked with the scalability, with the area of interest management and with the satisfaction of the participants. Filtering helps a participant to receive data closed to its interest in other participants flows and the total amount of this interesting data doesn't overpass its capacity. The participant satisfaction has a metering role about how efficient is the filtering. A "good" filtering preserves the scalability: the amount of data received by every participant should not be influenced by the current number of participants connected in the VE. In addition, the filtering depends according to the communication architecture: it may take place at a central server or distributed at the participants level. Solutions for filtering have been proposed at the transport level (architectures based on the multicast) and at the routing level (by adding states in the routers). We can also imagine hybrid solutions, by combining the two approaches. The routing-level filtering requires that some mechanisms should be deployed in the routers, in addition to the routing protocols. These solutions have the inconvenient to be difficult to deploy, all the routers in the Internet should be modified! Referring at the transport level filtering, we have seen in the previous chapter that filtering using multicast may be grid-based, entity based or hybrid. We have seen that each of these solutions presents some advantages and some disadvantages. The scalability of a VE architecture is related to the scalability of the data traffic and the scalability of the signaling. With the growth in the number of participants simultaneously connected in the VE the data and the signaling traffic increase accordingly. We have seen above that filtering plays a very important role in the scalability. In addition, the dynamicity of the participants also affects the VE scalability. During the VE lifetime it may happens that a lot of participants are concentrated in a small area of the VE. If the filtering doesn't takes into account this high participant-density area, the VE entities located in this area may be seriously affected by the increased signaling and data traffic; thus rising scalability issues. In addition, this "peak-density" spot may move in time to another VE coordinates. So, the filtering must take into account the dynamicity of the participants and it must implement some mechanisms to responds in the real-time to the new VE conditions. In addition to the data scalability, the dynamicity of the participants also affects the signaling scalability. The number of participants located in a VE area managed by an agent may overpass that agent's capacity. In this case new agents (more computer power) are required to process the VE signaling.

d. Real-time characteristics. Two participants located next to each other in the VE could be geographically situated on different continents. When they communicate, the data flow has to traverse different networks with different speeds. [1] At this propagation delay we should add the time spent in the routers queues. In the same time, a delay of 250 milliseconds may cause quality deterioration in audio communications. [2] Different types of flows may have different delay requirements. This requires the use of different mechanisms of transmission control to reduce the packets delay impact.

3.3 Our choice

Above, we have presented the issues that we think they are very important in designing a communication architecture for LSVE. After analyzing these issues, we propose our solution, the Score-SSM architecture.

a. From the multicast point of view we propose the use of the SSM model.

b. In our architecture each participant supports different data flow types and each type of data flow is sent on a different multicast SSM channel. In addition, each participant may have a different area of interest for each type of data flow it receives. It may dynamically modify the diameters of these areas of interest according with the momentary preferences.

c. To provide filtering and good scalability we propose a hybrid filtering solution based on:

   – an entity-based filtering which gives the possibility of the participants to express their particular interest,
   – a grid-based filtering which distributes the global VE information into smaller zones.

   The entity-based filtering, first determines the neighbors candidates to communicate (located in participant's area of interest) and then it chooses which type of flow to exchange for every neighbor. The grid-based filtering is hierarchical. First, the virtual world is split into smaller areas, each of these areas is managed by a machine called "agent". Second, each agent divides its managed area into smaller zones according to the local participant's density. This hierarchical filtering is meant to enhance the scalability of the communication model by taking into account the participants

---

[1] The packets which traverse an optical fiber between Europe and Australia have a delay of at least 140 milliseconds.

[2] In DIS standard is recommended to fix this delay at maximum 150 milliseconds. In the networks games, this maximum delay is about 200 milliseconds.

characteristics (the size of a VE zone is determined according to the capacities of the participants) and the agents' characteristics (each agent manages an area of the VE according to its capabilities).

d. Our communication architecture is designed to support multimedia flows like audio and video streams. It uses the SSM model, which constructs source-rooted trees. This will reduce the packets delays because the participants receive data from the preferred sources over the shortest routing path.

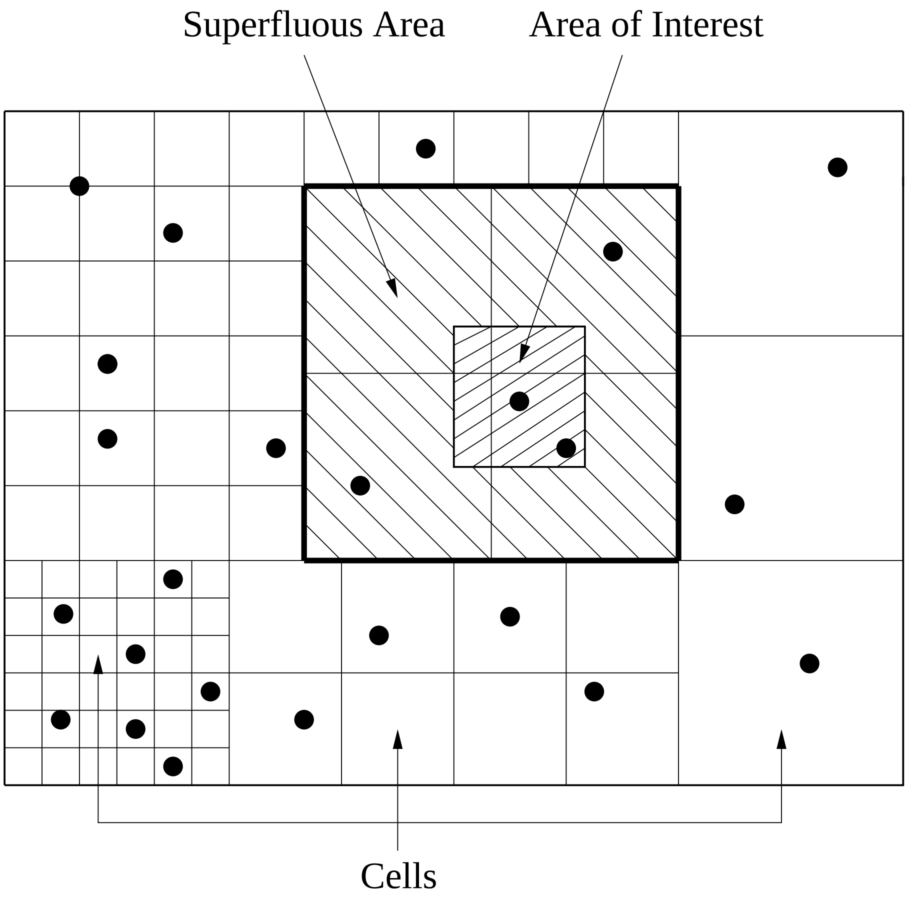
Chapter 4

The Score-SSM architecture

In this chapter, we present two communication architectures for the large-scale virtual environments. The first of them is called Score and it was designed in 1998, in Rodeo Research Team at INRIA Sophia Antipolis. It uses the ASM multicast model and it is the first to introduce a dynamical partitioning of the VE into multicast groups taking into account the density and the capabilities of the participants. We have chosen Score as a representative architecture for the ASM grid-based filtering. The second communication architecture presented in this chapter, it is called Score-SSM and it was designed with the intention to create an SSM multicast-based version of Score. Because the SSM model is a simplified case of the ASM model (as we have seen in the second chapter of this thesis), additional complexity is required at the transport and application levels to overcome the network layer limitations and to provide the same functionalities to the users. Score-SSM was designed to accomplish this task. In this chapter, we present the two communication architectures followed in the next chapters, by the experimental results between Score and Score-SSM, as well as performance comparisons between an application implementing both Score and Score-SSM.

4.1 The Score architecture

Score is an agent-based communication architecture for large-scale virtual environments (LSVE) which uses a limited number of multicast groups. The purpose of this architecture is to help the scalable deployment of the VE over the Internet by allowing thousands of heterogeneous participants to be connected at the same time. In Score, the virtual environment is dynamically partitioned into spatial areas. Each of these areas is associated with a multicast group. Score uses a method based on the theory of planar point process to determine an appropriate cell-size, so that the incoming traffic at the receiver side remains within a given probability, under a sufficiently low threshold. The participants dynamically join/leave the multicast groups, according to their interest on the data transiting those



groups. The performance evaluations showed that Score allows to significantly improving the satisfaction of the participants while adding a very low overhead. In order to describe the Score architecture we introduce the satisfaction metric and present the role of the agents. Then, we describe the information exchange process between the participants and the agents and finally we present the mapping algorithm and the handover management mechanism.

4.1.1 The user satisfaction metric

An ideal situation from the end-user point of view can be defined as a situation where the received traffic contains no superfluous data. It requires a very "fine grain" filtering and this uses a large number of multicast groups. But in the ASM model, the multicast groups are a limited resource. Moreover, participants have limited network resources and CPU processing power. If the participant area of interest is so large that the traffic it receives cannot be real-time processed, no other mechanism could enable it to receive all the data it is interested in. Indeed, in this case, even if the subscribed cells exactly match its area of interest, the received traffic exceeds the capacity. For this purpose, the user satisfaction metric S is defined as:

$$S = \frac{U_r}{min(U_t, C)}$$

Where:

- U_r stands for the interesting data received and processed,
- U_t represents the data rate (received or not received) in which the user is interested
- C stands for the receiver capacity, which is the maximum data rate that the receiver can handle.

We can observe that when a participant receives and process all the data it is interested in, this satisfaction metric is maximal regardless the superfluous traffic rate. This metric was chosen taking into account the necessary tradeoff between the superfluous data rate received, the network state, and the overhead associated with dynamic group membership. Note that with this satisfaction metric, the goal of Score is not to adapt to the worst receiver in terms of network connectivity and processing power, but to maximize the satisfaction of the receiver with the lowest S value.

4.1.2 Agents role

Score define agents as servers or processes running at different parts of the network (e.g. on a company LAN, hosted by an ISP or by an ASP). Administrators of LSVE are responsible for deploying such agents on the Internet and for positioning them as close as possible to their potential users. Agents are not servers, i.e. they do not aim to process any global state for the VE, and so they do not receive "data" traffic sent between participants. Actually, agents dynamically determine zones within the VE by considering the distribution of participants and calculate appropriate cell-sizes according to the density of participants in each zone. Agents also have to process periodically the satisfaction of each participant according to their capacities, the size of their area of interest and the density of participants within the current zone. Once this computation is done, the agents may rearrange the VE zones by dividing or aggregating them or they modify the cell-sizes for the zones where the participants with the lowest satisfaction are located. Therefore, this approach dynamically partition the VE into cells of different sizes, and the association of these cells with multicast groups. Agents have to dynamically determine the appropriate cell-sizes in order to maximize users' satisfaction.

4.1.3 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 between them. This specific information should be taken into account before partitioning a VE into different zones. First, the VE is statically partitioned into several large parts that are called start-zones. These start-zones are actually defined according to the intrinsic structure of VE (e.g. rooms, floor, walls, etc...). Each start-zone is statically partitioned into indivisible zone-units which are the smallest unitary zones that compose the start-zone. During the session, start-zones are dynamically divided into zones with the same cell size.

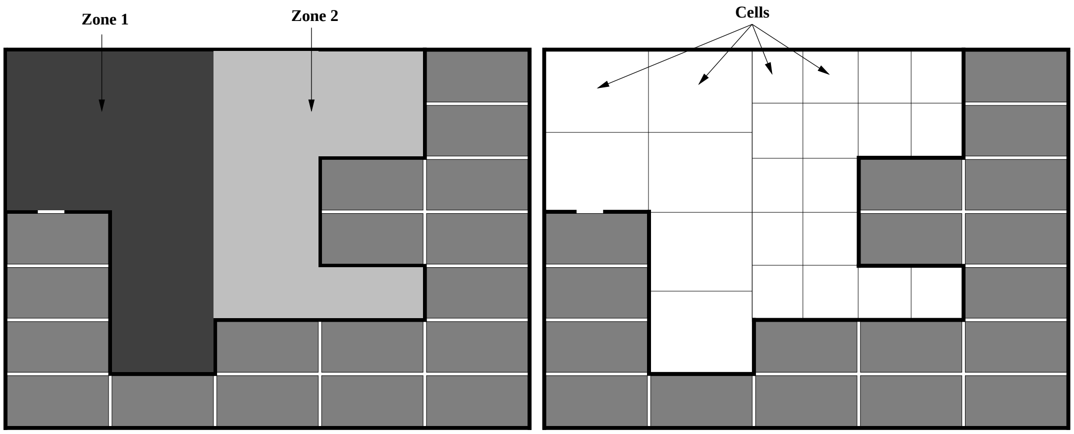


Figure 4.1: The division of the world in zones and cells

So, cells (figure 4.1) are mapping of multicast groups to a number of zone-units. As agents decide to define new zones in order to take into account changes in the distribution of the participants, they identify these zones as sets of one or more adjacent zone-units belonging to the same start-zone. To summarize, a zone is a subset of a start-zone and is composed by n contiguous zone-units. Within a given zone, all cells have the same size but two distinct zones could have different cell sizes.

The mapping algorithm consists in three successive operations:

- At first, a calculation is done in order to define a cell-size for each zone by only taking into account the distribution of participants in the VE.
- Then, the participants with the lowest satisfaction are identified as well as their distribution in the VE. If agents detect a concentration of unsatisfied participants within a part of a zone, this zone is divided into two new zones in order to isolate these participants. If not, the zone remains unchanged. A smallest cell-size is computed in the zones which contain the participants with the lowest satisfactions, so that they can better approximate their area of interest and therefore improve their good put.
- The final operation is less frequently performed: the agent can decide to aggregate contiguous zones, if the cell-sizes are the same for these zones and if they belong to

the same start-zone (two start-zones can never be merged).

4.1.4 Participants-to-Agent communication

There are different levels of communication in Score:

- 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 UDP unicast connection
- Agents communicate with each other on a single multicast group, the Agent multicast group (AMG).

A participant subscribes to two different kinds of multicast groups:

- Data groups associated to the cells that intersect its area of interest (but the participant sends data only to the multicast group associated to its current cell)
- Control groups associated to the start zones that intersect its area of interest. For these groups the participant is only a receiver. Agents use control groups to send mapping information relative to the start zones. This information is periodically sent for each start-zone and contains the mapping information for all the zones belonging to the start-zone (the cell-size for each zone and the associated multicast group addresses).

Each participant is connected to its nearest agent 4.2 using an UDP connection (the use of a TCP connection may rise scalability problems, due to the possible large number of participants communicating with one agent). In Score, a participant is connected to the same agent for its lifetime in the VE. Each time a participant enters a new zone-unit (every participant receives, from the agent, the mapping between the multicast groups and the VE zones in its corresponding start zone), it sends a short message to its agent. This message contains its identity, its position in the zone-unit, its current size of area of interest and its capacity. Therefore 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 Agents multicast group. However, agents do not send the exact virtual position of their associated users. Only the number of users per zone-unit is necessary to allow agents to compute periodically the density of participants per zone-unit. Also, participants have to send low rate keep-alive packets so that agents can detect a possible disconnection and have an accurate number of participants in the different zones.

When proposed, Score was the first communication architecture which dynamically partition the VE into zones, taking into account the density and the capacities of the

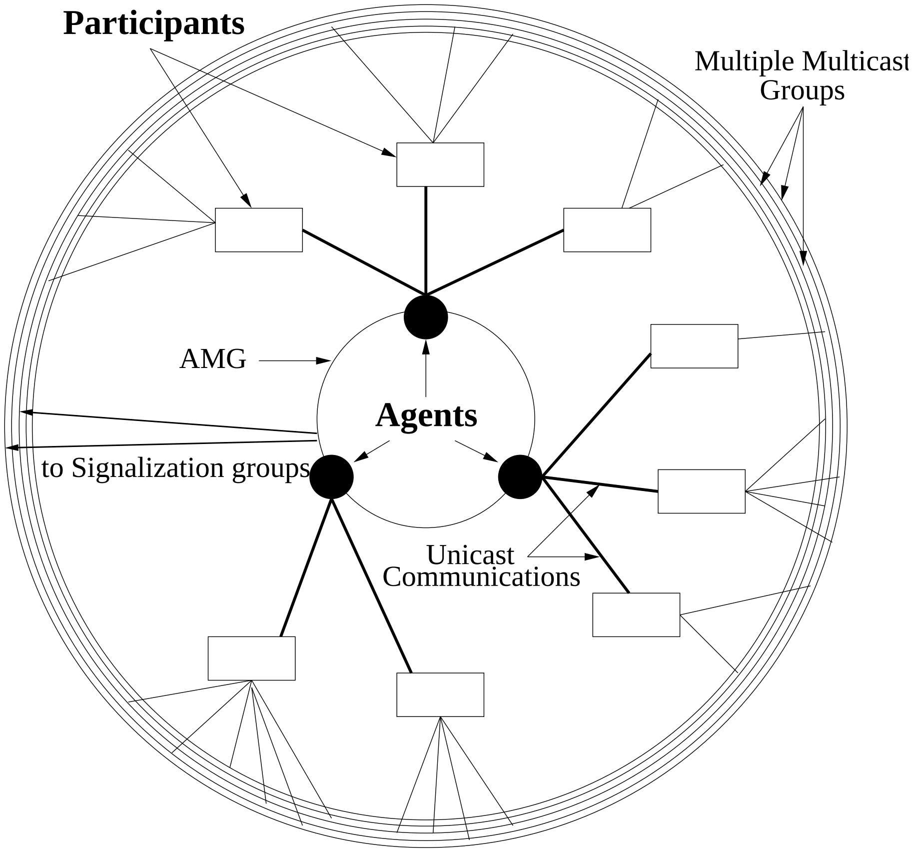


Figure 4.2: The communication architecture

participants. The purpose of this partitioning is to maximize the satisfactions of the low-capacity participants through a better usage of the multicast groups. Score takes into account the assumption that the number of multicast groups is a limited resource and the ASM model is deployed over the Internet. Since the proposition of Score, the ASM model hasn't been yet deployed in the Internet. As we have seen, other propositions, like SSM or application-level multicast, have more chances to be used for LSVEs. In addition, the limited number of multicast groups leads to the receiving of superfluous data by the participants which causes a suboptimal utilization of the network links and a diminution of the participants satisfactions. These considerations have conducted us to propose Score-SSM, which responds to the above issues, while keeping efficient mechanisms from Score, like dynamic partitioning

4.2 The Score-SSM architecture

4.2.1 Score-SSM overview

In this section we present Score-SSM, a communication architecture which allows a large number of heterogeneous participants to be simultaneous connected in a virtual environment (VE) using an SSM multicast routing. We consider a type of virtual environment where the participants exchange data only with their closest neighbors, the communication held in other parts of the VE doesn't interest them. To achieve such a communication model, each participant has to filter out the data sent by the distant participants. In the next paragraphs, we present the main characteristics of our architecture: the superfluous data filtering, the heterogeneity of the participants and the use of the SSM model. The overview of Score-SSM architecture is followed by an in-depth explanation of all the composing mechanisms.

4.2.1.1 Filtering

Score-SSM proposes a two-level filtering architecture. First, the VE is broken up into non-overlapping zones, according to the position and the density of the participants in the VE. A participant receives the approximate positions of the other participants located in the VE zones intersected by its interest. Second, the participants compute their "closest" neighbors among the participants designated by the first level of filtering and start to communicate with them. The first level of filtering of Score-SSM (we call it "agent filtering") is composed by an agent-based signaling overlay. To improve the scalability, the VE is split into zones; one agent registers all the movements of the participants in a particular zone. So, we have a pool of agents, which share together the signaling of the VE. By signaling, we understand here all the network traffic required to set-up the communication between the participants. Each agent is responsible for the signaling in one or more zones of the VE. The more participants are simultaneously connected in the VE, the more agents are deployed to manage the signaling in the VE. The agents decide the decomposition of the VE into zones. We think that a dynamic zone partitioning according to the participants distribution in VE leads to a better approximation of the interest of the participants. The second level of filtering (we call it "participant filtering") is realized through the use of the Source-Specific multicast (SSM) as the data distribution model. Once a participant had computed its closest neighbors it subscribes to their data flows. The use of SSM allows the participants to communicate only with the "closest" neighbors, without receiving undesired data flows from the "distant" participants, as in the case of using ASM. Also, SSM gives to the participants the possibility to receive only a particular flow from a neighbor (we suppose that the participants are able to send different types of data flows) because every participant uses a different SSM channel for each type of data flow it transmits. We can observe that an SSM channel corresponds to a specific data flow sent by a specific participant. This very precise distinction, between the data flows transmitted in the VE, helps a lot the filtering of incoming data at the participants. We observe that the VE participants have the possibility to precisely express their interest in receiving the others participants data flows. This is an advantage of using SSM, but which comes at the price of complex filtering mechanisms that should be implemented by the architecture in order to the precisely detect the neighbors (and the data flows types they exchange) to which a participant communicates.

4.2.1.2 Heterogeneity and multimedia

We consider the virtual environments in which the participants have very heterogeneous capabilities: different computation powers, different types of data flows sent/received, different network connections capabilities and different interests. Our architecture is designed to deal with such heterogeneous participants: Score-SSM gives to each participant connected in the VE the possibility to communicate with a number of neighbors according with its capabilities. The quantity of data sent/received by a participant is not influenced by the presence in the VE of other participants with bigger/lower capabilities than its capabilities.

We can classify the participants capabilities in terms of different communication flows transmitted or received by: *sending capability* and *reception capability.*

In the VE, each participant transmits its position and it is capable of transmitting text messages. In addition, some participants are able to transmit audio and/or video flows. These types of flows determine the sending capability of a participant.

Since participants are capable of transmitting different types of flows, their *reception capability* is defined according to the flows types a participant is able to receive. The *reception capability* of a participant is composed by several *areas of interest* of different sizes: *the sight range, the audio range and the video range.*

The *sight range* corresponds to its area of vision: the participant detects the movements of all the participants located at a distance lower than a given value which is called its sight radius. In addition, the participant is capable to receive text messages from the participants located in its sight range.

The participant's *audio range* corresponds to a circular area where the participant is capable of listening the participants located at a distance lower than a given value (audio radius).

The participant's *video range* corresponds to a circular area where the participant is capable of receiving the video flows transmitted by the participants located at a distance lower than a given value (video radius).

Each participant has two additional parameters: *the maximum number of audio* (re-

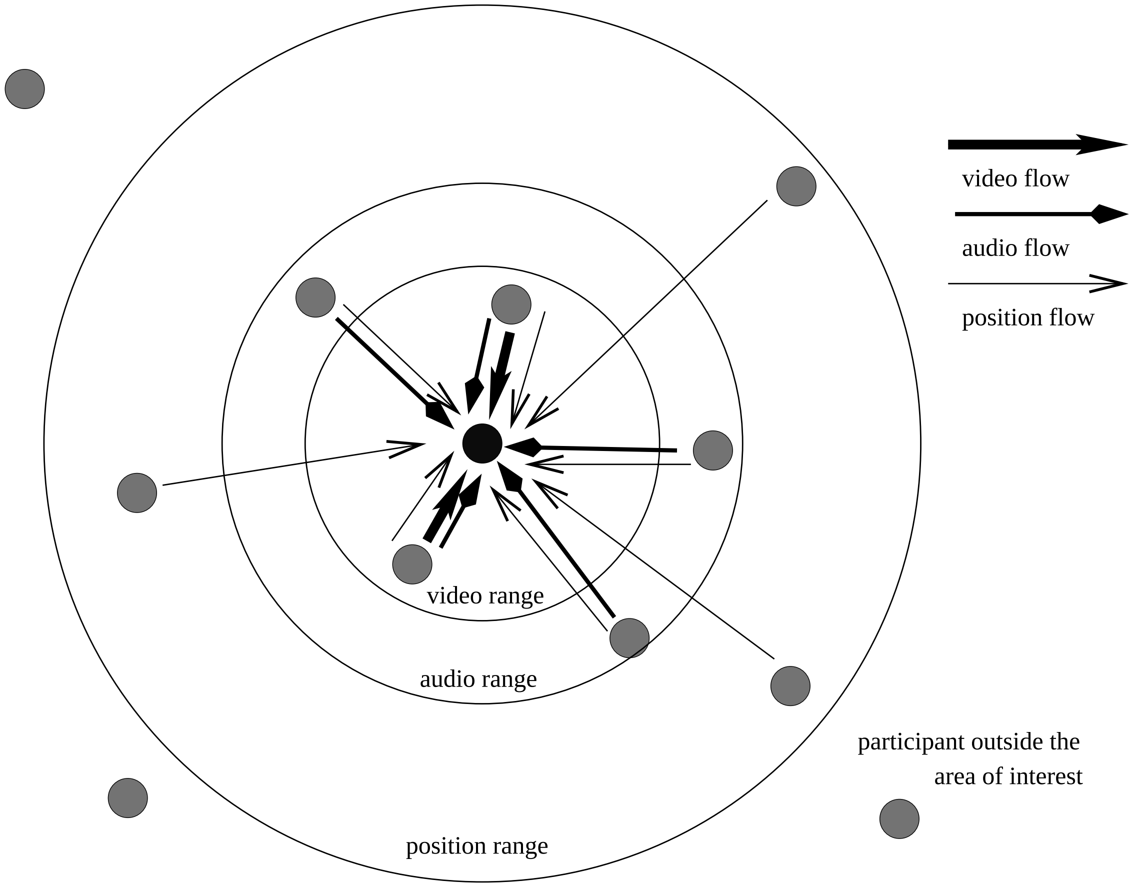


Figure 4.3: The interest circles of a participant

spectively *video*) flows it is capable to receive simultaneously. These values are two upper bounds that help to refine the capability of a participant. *The audio range and the video range* are computed starting from these values: the bigger is the number of the neighbors capable of sending audio (respectively video) flows, the smaller becomes the audio (video) range of the participant. As the participant can receive up to the *maximum number of audio flows*, it searches the audio sources among the participants located in its sight range. The audio range is small if these audio-capable participants are close to the participant. Also, these values reduce the position flows losses at the participants due to the high participant resources consumption caused by the incoming audio and video flows. As we observe on the figure 4.3, the video range is contained into the audio range, which at its turn, is contained into the sight range. In the same time the video flows rate is bigger than the audio flows rate, which is bigger than the position and text flows rates.

4.2.1.3 Multicast

For scalability reasons, audio and video flows are sent via multicast: to economize network bandwidth, the audio or video packets travel only once on a physical network link. Thus, the saved bandwidth can be used by other flows, allowing the participants to exchange data with a bigger number of neighbors. We think that shared distribution trees, as constructed by some multicast routing algorithms (CBT, PIM-SM), are not appropriate to simultaneously deliver more audio or video flows, due to the large amount of traffic which would flow through the same branches of the distribution trees (i.e. through the same network interfaces). On the other hand, the source-rooted trees allow the participants to subscribe directly to a particular video or audio flow (transmitted by a specific neighbor). In this way, the traffic concentration is avoided as well as the reception of undesired packets generated by the participants located outside the area of interest of a participant. The SSM model constructs source-rooted data distribution trees and its address space allows every participant to be the source of more multicast groups. Compared with the multicast routing protocols implementing the ASM model, it easier deployable over the Internet. The ASM model was designed to allow anybody to send data over the same multicast group. Data flows send by different sources are aggregated. Thus, a participant may receive data packets from undesired sources. On the other hand, using the SSM model, a participant subscribes explicitly to the flows it is interested in. Thus, the SSM model offers, to the participants, a "fine grained" filtering mechanism being more convenient than ASM to our architecture. However, this "network-level" simplicity requires additional complexity at the higher levels of the ISO-OSI protocol stack.

The variation of the density of the participants in different areas of the VE influences the agents computing load and it may trigger a new zone partitioning of the VE. We consider that all the participants are SSM-capable and that every participant has a different multicast channel for each type of data it transmits: one for its position, one for its text data, one for its audio data (if it transmits any audio data) and one for its video data (if it transmits any video data). To receive data from the participants located in its interest zone, a participant subscribes to the corresponding multicast channels of its closest neighbors, according to its interests in their position, text, audio, video flows. We consider that all the participants use the same IP multicast groups address "G" (within the SSM-range) for their position flow. In addition, another unique multicast group address is available for all the participant's audio flows and another one for all the participant's video flows. This saves additional signaling required to exchange group information. We consider that a participant communicates only with its closest neighbors: it can establish a communication with another participant only if it communicates at that moment with all the participants located closer than the new participant. So, in order to communicate with a given participant one has to move sufficiently close.

In the following paragraphs we enter in the details of the Score-SSM architecture by presenting the participants satisfaction metric, the two levels of filtering, the flows involved in the communication and the different communication phases.

4.2.2 Satisfaction of the participants metric

A virtual environment is a medium built to allow the participants to meet and interact with each other. The tools and the algorithms deployed in the VE must be efficient, to help at maximum the participants to use their capabilities while connected into the VE. But how can we measure the efficiency of these mechanisms from the point of view of the participants? Because the participants are the entities for which the communication architecture are built, we think that their satisfactions are a very important indicator for the efficiency of these mechanisms.

We see three approaches in defining the satisfaction of a participant. In the first approach, we can consider a participant totally satisfied if and only if all received data is useful and in addition all the useful data regarding this participant has been received and well processed. The second approach takes into account the participants capabilities: the available network bandwidth and the participants computing power. In this case a participant has maximal satisfaction if all the received data is useful, in spite of some useful data that hasn't been received due to the network and computing limitations of the participant. Here we can remark that no other algorithm or grouping scheme could allow a participant to receive and process more data than its capabilities. In the third approach, a participant is fully satisfied if all the useful information has been received and processed. In this case we don't consider any superfluous data which the participant might have received.

From the participant's point of view, the ideal situation occurs when all the traffic received is useful traffic. Even though a large number of multicast channels are available in the SSM model, it is still a major problem to realize this perfect filtering due to the difficulties to map the flow on multicast groups and to manage the subscribe/unsubscribe operations on a fine grain level. The purpose of Score-SSM is to allow the participants to filter out the received traffic in order to significantly reduce the superfluous data. Each participant has an area of interest (for each category of flow it may receive) in the VE, defined as a circular area around the participant. Using the area of interest as a mean to define the participant's filtering, we assert that the bigger are the chances that another participant enter its area of interest, the bigger the value to get the data sent by that participant. In the same time, we assert that for a participant in the VE, if the chances that some participants enter its area of interest are low, then the participant's interest in the data flows sent by these participants is low. These kinds of flows are the ideal candidates to be filtered out by a participant who tries to reduce its superfluous traffic.

We consider the satisfaction metric S for the participants defined for Score and we study its implications for Score-SSM case:

$$S = \frac{U_r}{min(U_t, C)}$$

Where:

- U_r is the useful data received and processed,
- U_t is the amount of useful data (received or not received) in which the participant is interested
- C is the participant capacity, i.e. the maximum amount of data the participant can receive and process in real-time.

Below, we explain the implications of this formula.

First, we observe that in the case $U_r = U_t$, i.e. when the participant receives all the useful data it is interested in, its satisfaction is maximal, regardless the amount of signaling received.

Second, when $U_r = C$ the participant satisfaction is again maximal. By definition $U_r \leq U_t$ so, $C \leq U_t$, so $min(U_t, C) = C = U_t => S = 1$. We observe that, in this case, the satisfaction is maximal despite that not all the useful data has been received.

We observe that there is no mention of the signaling traffic in the formula above. This doesn't mean the signaling is not taken into account: the signaling packets may use the same network connection as the data traffic and an important signaling traffic may cause useful data packets losses. The consequence of this is the diminishing of the participant's satisfaction.

Because of the signaling traffic, the situation described above, where $U_r = C$ cannot occur in practice. By no means the signaling rate reaches zero. So, a part of the network resources are spent with the signaling traffic in spite of data traffic. This means that the participant satisfaction is lower than one. If the participant wants to improve its satisfaction it must drop some of its neighbors, in such a way that $U_t <$ C. If the participant drops enough neighbors it will arrive in the situation where $U_r + signaling < C$ and $U_r = U_t$; or in other words, $S = 1$.

So, compared to Score, in Score-SSM the participants adapt their interest in order to maximize their satisfactions.

4.2.3 Agent filtering

4.2.3.1 Agents overview

Score-SSM was built for virtual environments where the participants exchange data with the entities located around them (in VE's coordinates). So, in order to communicate with its closest neighbors, a VE member has to know which participants are located close to it, in the VE coordinates. The communication architecture has to provide to the participants a way to receive the position of the other members of the virtual world located in their neighborhood. Score-SSM was meant to support a lot of participants simultaneously connected into the VE, let's say of the order of millions participants, so in this case it is clear a participant cannot receive the positions of all the other members of the VE, this is not scalable. To be scalable, the VE must be divided into smaller zones. Here we discover the role of the first level of filtering: a participant receives only the coordinates of the VE members located in the same zone of the world. If its area of interest overpasses the border of the current zone and intersects other zones of the VE, the participant subscribes to these new zones, in order to detect the participants that may be located in its area of interest. This level of filtering is realized in Score-SSM through the agents.

Score-SSM is an agent-based communication architecture which decomposes the VE into zones and uses multicast channels to transmit data and signaling. The agent are computers deployed in different places of the Internet, preferably close to the participants (we consider here the "delay" as measure of the proximity). The total number of agents managing the VE is proportional with the number of the participants connected in the VE.

The agents receive no data sent between the participants; they are control plane "agents" and not data plane "servers". They collect information about who is connected in the VE and about the approximate positions of the participants. We may compare this with a photography of the VE: the totality of the agents detain the information about the approximate positions all the connected participants. This information is updated from time to time (not in real-time), like shooting another picture of the VE. Here, there is a trade-off between the update frequency of the position of the participants and the signaling traffic required to update this information. More precise the agents know the positions of the participants, more signaling traffic is generated and flow through the network interfaces. So, while implementing the agents, a fine tuning is required to find the best compromise between the VE accuracy and the signaling rate.

To sum up, the role of the agents in Score-SSM is to make the VE communication accessible to the participants by dividing the world into smaller zones and by reducing the signaling.

4.2.3.2 Dividing the VE into zones

A VE is characterized by its physical structure and extent and by the connected participants. Regarding the first characteristic mentioned above, the VE may be composed by a number of independent, non-overlapping areas; members of one area cannot communicate with other areas members. We may consider this kind of VE as a union of autonomous VEs. In the following discussions we regard the VE as a world where everybody may communicate with everybody.

The other characteristic of the VE mentioned above, it is the number and the distribution of the participants. The agents use this information to cut the VE into zones. First, the number of the participants influences the number of the agents that should be deployed to handle the signaling generated by these participants. Each agent has the capacity to manage a limited number of participants, so we must deploy enough agents to cover all the connected participants. Second, the distribution of the participants determines the precise area where the agent cuts a VE zone into smaller zones.

Let's see more in details how the agents manage the VE. First, the VE is divided into non-overlapping areas. One agent manage an area of the VE by splitting it into smaller zones, according to the local participants density. An agent receives the approximate positions of all the participants located in its corresponding area. The more participants are present in an area, the more approximate positions the agent manages and updates. According to the update frequency of these positions, we can compute the traffic rate received and then processed by an agent. Of course, an agent doesn't have unlimited network or computing resources, so it exists a maximum number of participants which it can simultaneously manage. At the VE's start-up, one agent is sufficient to manage the entire VE, then, while more and more participants are arriving, the agent reaches its capacity. In this situation, the VE is partitioned into smaller, non-overlapping areas, and new agents are launched (from a pool of available agents) to handle the signaling of the connected participants. In the absolute values, the agent resources necessary to manage one participant are very low (about 40 bytes/second traffic rate, including the UDP/IP header, and very low computing power). For example, for 10 thousands participants, one agent's incoming traffic is 40*8*10 000 i.e. approximately 3.2 Mbps. Thus, practically, an agent can simultaneously manage tens of thousands participants, which is equivalent with the number of participants simultaneously located in the same VE area. While, from the agent's point of view, a large number of participants in one VE area rises no scalability issues, from the participants point of view, this leads to a lot of signaling traffic received if this number of participants is not reduced in smaller groups. Thus, the role of the zones composing a VE area, is to reduce the signaling to some acceptable values from the point of view of the participants. To be efficient, the decomposition of a VE area into zones is effectuated according to the local participants density. A participant receives all the signaling traffic generated in its corresponding VE zone (and possibly from some neighboring zones). The smaller is the number of participants located in a VE zone, the lower is the signaling generated in that zone. This leaded us to establish an upper limit for the number of the participants simultaneously present in a zone of the VE. All the above considerations shows in a new light the responsibility of the agents in cutting the VE. As soon as a zone reaches the established upper limit for the number of participants, the agent divides that zone. The high participant density areas of the VE are decomposed in a large number of smaller zones, while larger-size zones are constituted in the VE parts with low participants density. In the figures (4.4 and 4.5), we see a high participant density zone being cut into four smaller zones, thus reducing the signaling received by the participants located in the initial zone.

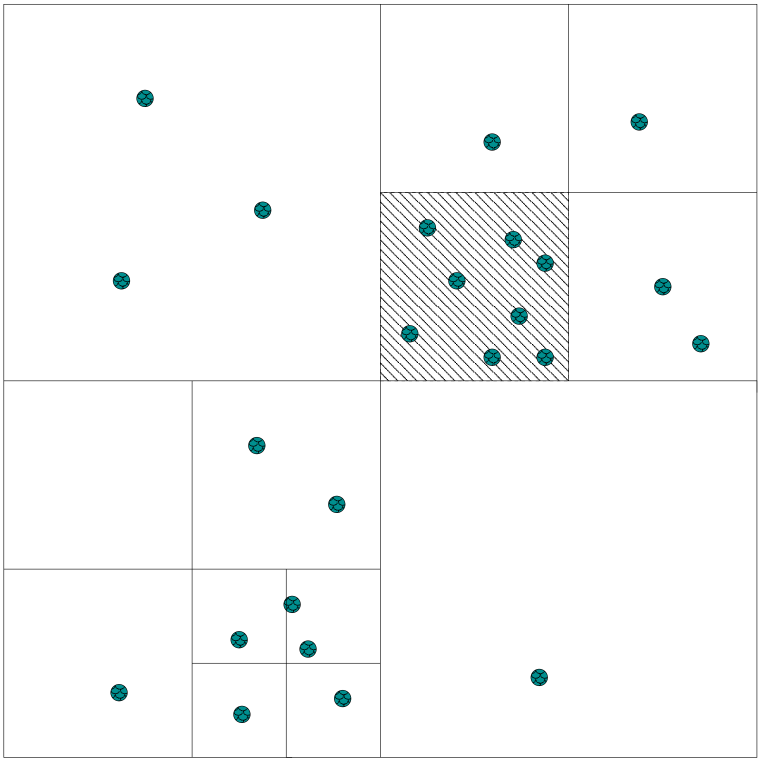


Figure 4.4: A zone of the VE which reaches the maximum number of participants

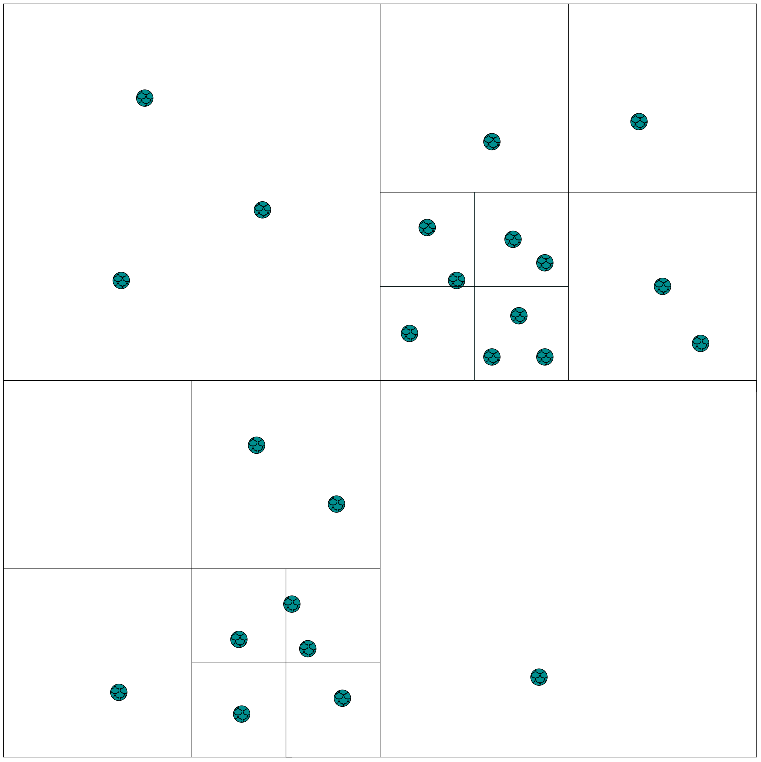


Figure 4.5: The zone is cut into smaller zones

From the agents point of view, the maximum number of participants in a zone is very small compared to the number of participants that an agent can simultaneously manage (we call it the "capacity" of the agent). This naturally leads to the situation of using one agent to manage a large number of zones in the VE. The agent can manage a number of zones at least equal to its capacity divided by the maximum number of participants in a zone (because during the runtime, the average number of participants per zone is much lower than the maximum number of participants admitted in a zone).

4.2.3.3 The agents and the multicast

The agent filtering role (dividing the VE into zones) is implemented using source-specific multicast. The role of the agents is to put in one-to-one correspondence each zone of the VE with a multicast channel. As previously mentioned, the role of zones in Score-SSM is to dynamically partition the signaling traffic containing the approximate participants positions in the VE. Each participant receives all the signaling generated in its zone and in the neighboring zones intersected by its interest. This signaling is sent by the agents toward the participants. The agents use a single multicast channel to send the signaling generated in a VE zone. An SSM channel address "(S,G)" is composed by the source address, "S", which in this case is represented by the IP address of the agent, and by a group address, "G", which is a value locally determined by the agent (it is an IP, class D, address). Contrary with the ASM case, we remark that, while using SSM, the agents are not restricted by the number of available multicast groups. In addition, different agents may use the same IP multicast group address, because the pairs which compose the SSM channels addresses are different.

An agent usually manages more zones in the VE: it designates multicast group addresses and makes "public" the mapping between them (i.e. the multicast channels) and the VE zones. The participants receive this mapping and then subscribe to the multicast channels corresponding to the VE zones intersected by their area of interest. The limitation in the number of participants per zone triggers the partitioning of the VE in a very large number of zones, when many participants are connected in the VE. This leads to a heavy signaling traffic composed of the pairs [VE zone] - [associated multicast channel]. From a participant point of view, this mapping traffic may become significant compared to its capacity. In order to reduce the mapping announcements, we decided to hierarchically partition the VE: first the VE is partitioned into areas, then, each of these areas is partitioned into zones. At this spatial organization of the VE corresponds a two-layers signaling hierarchy. The "base" signaling layer helps the participants to receive only the mapping of the zones located in their proximity. Each agent uses an unique SSM mapping channel, composed by a "well-known" group address, to transmit the correspondence between the zones in its VE area and the IP multicast group addresses of the channels used to transmit the signaling in these zones. While moving through the VE, a participant may change its corresponding agent (because the agents are linked to a VE area and not to the participants). Before entering a new VE area, it subscribes to the corresponding agent's mapping channel (using the same "well-known" IP multicast group address), in order to find out the signaling channel of the zone to which it is heading to. The "upper" signaling layer uses one SSM channel for the mapping between the agents and the VE areas they manage. This channel address is "well-known" by all the agents connected in the VE and by the "back-up" agents, as well. En entity called "the Main agent" is the source of this multicast channel. In addition, this entity controls the pool of agents and it determines which agents are running and which are kept for the back-up. More details about the functioning of the Main agent can be found in the section 4.2.5.1. The participants also listen to this "well-known" channel each time they are about to change the agent to which they communicate. Figure 4.6 shows the

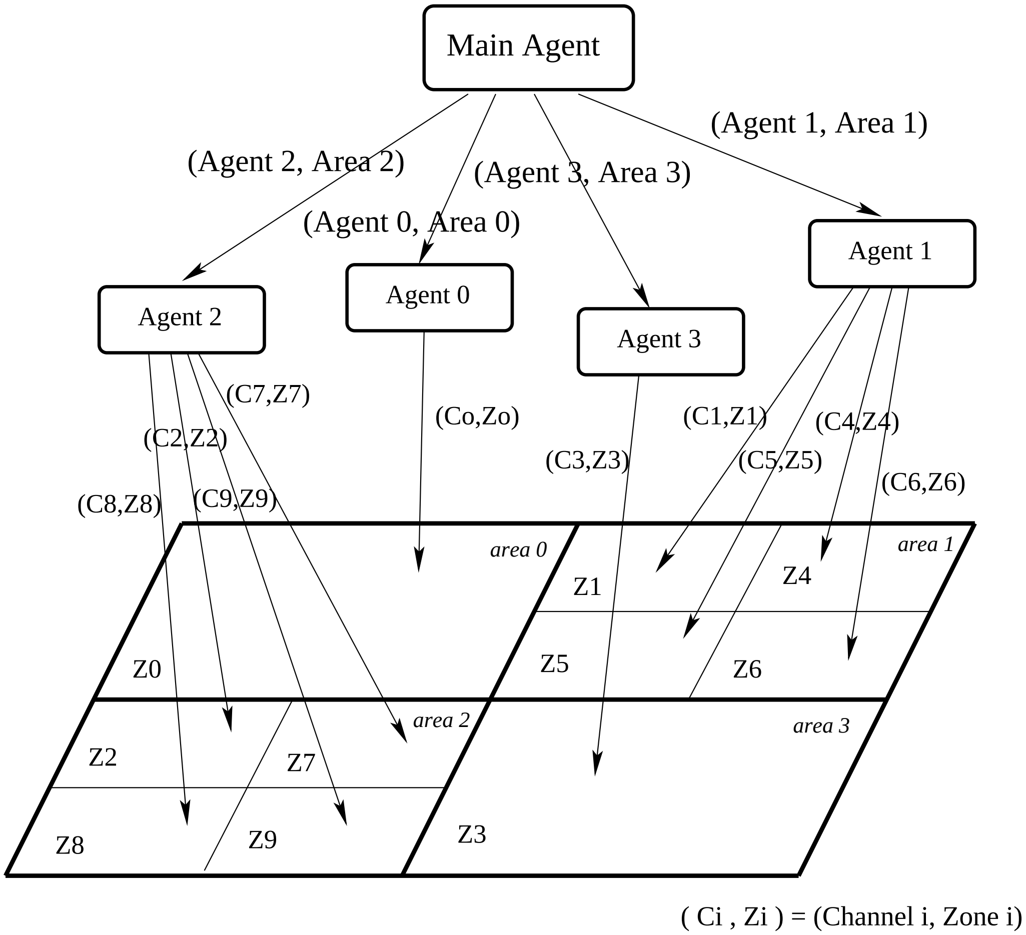


Figure 4.6: The two-layers signalling architecture

two-layers zone mapping scheme: the first layer is an [agent] - [VE area mapping], followed by the second level, the [multicast channel] - [VE zone mapping].

4.2.3.4 Dynamicity

As we have seen above, the local participant density in the VE determines the zone-partitioning performed by the agents. This density vary in time because the participants are moving through the VE and their number changes all the time, new participants may enter in the VE and other participants may leave. This confirm the necessity of continuously updating the VE zone-partitioning, which is one of the roles of the agents. Each agent periodically receives the positions of the participants located in its corresponding zones. If their number exceeds the limit of maximum participants per zone, it takes the decision to split that zone into smaller zones (for the implementation simplicity, we have chosen to divide a zone into four smaller equal-surface zones). The agent will manage additional zones instead of the initial, larger zone. If the total number of participants located into four adjacent zones is lower than the maximum number of participants admitted per zone multiplied by some sub unitary threshold the agent unifies these zones into a larger one. The surface of the new zone is equal with the total surface of the four initial zones and the participants contained in this zone are the participants located in the unified zones.

4.2.4 Participant filtering

4.2.4.1 Participant filtering explained

We have seen in the previous sub-section that the agent-level filtering reduces the signaling received by the participants. Once the filtered signaling packets arrive to the participants, each participant will determine a set of neighbors from which they intend to receive data. The signalling packets received from the agent contain the following information about the neighbors: IP address, neighbor ID, position, capacity, supported flow types (see the appendix for more details). Thus, the role of participant-level filtering is to allow every participant to select, from the signalled participants set (determined through the agent signaling), a subset of neighbors to which the participant will exchange data. For every neighbor in this subset, the participant subscribes to a number of data flows (i.e. position, audio, layered video etc...), according to its interest. We remind that each of these flows is sent on a different multicast channel.

In figure 4.7, we see a participant surrounded by its area of interest. The participant subscribes to the "neighbors" located in its area of interest. If the number of these neighbors exceeds the participant capacity, a reduction of the area of interest of the participants is required. The participant diminishes its area of interest in order to receive a number of neighbors according to its capacity. This avoids the overloading of the participant and limits the packet losses due to the network interfaces congestion. In the same time, the participant receives the signaling traffic generated in the VE zones intersected by its area of interest. The more zones the participant intersects, the more signaling it receives.

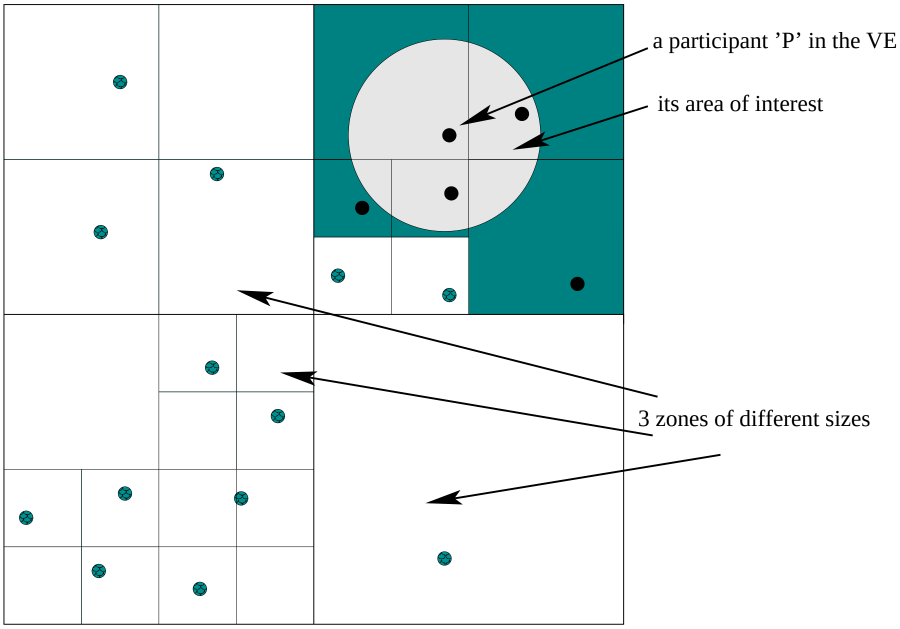


Figure 4.7: A participant 'P' in an area of the VE

When this signaling becomes important (compared with the data traffic rate) and when the participant receives a number of neighbors close to its capacity, some packets may get lost and the participant satisfaction decreases. In this case, the participant shrinks again its area of interest, in order to maximize its satisfaction.

Each zone of the VE, overlapping the participant's area of interest (figure 4.7), adds more signaling and reduces the available network and computing resources of the participant. The signaling generated in a VE zone is directly proportional to the number of participants located in that zone. In the previous section we mentioned the necessity of an upper limit in the number of participants simultaneously located in a VE zone. The smaller it is this limit, the less is the signaling generated in a VE zone. A participant whose area of interest intersects a zone of the VE receives all the signaling generated in that zone, but, in the same time, it might happen to communicate with zero (in the worst case) or very few neighbors from that zone. We call the "gain" of subscribing to a VE zone the ratio between the number of participants to which the participant communicates and the number of participants present in that zone. This gain takes values between 0 and 1. If the maximum number of participants simultaneously admitted in a zone decreases, the gain increases. In fact, a participant cannot subscribe to a number of neighbors bigger than its capacity. If the number of participants connected to a zone overpasses a participant's capacity we can be sure that the gain corresponding to that zone is less than one.

Below, we explain in details, the shrinking of a participant's area of interest. The purpose of this shrink is to lower the number of useful data flows by cancelling the communication with some neighbors. So, the area of interest radius will diminish according to the distance from the participant to its farthest neighbors. The participant knows the precise position of its neighbors, so it decides to unsubscribe from the farthest one. After dropping this neighbor, the area of interest radius is set at a value equal to the distance to the new farthest neighbor. The participant computes again its satisfaction. With the new value it may decide to continue the process of shrinking its area of interest by dropping more neighbors, or it may decide to continue the communication with the current neighbors.

4.2.4.2 Different types of data flows

Because in the Internet the network links and the computers capabilities are heterogeneous, Score-SSM was built to support different types of data flows, each of them having different data rates, like audio flows, video flows, text etc... The most important flow a participant receives is, however, the position flow because in the VE, the participants communicate with the closest (in the sense of Euclidian distance in the VE) neighbors. Every participant has two different position flows: a slow-rate position flow which represents the signalling from the participants to the agent and a fast-rate position flow which is considered as a data flow because it provides the exact location of the participant in the VE (which is essential in "Quake"-like applications). Regardless the data flows' types exchanged with the neighbors, a participant must first know their exact positions. These positions are used to compute the precise distance to the neighbors and depending upon their values, to decide which kind of data flows will receive from each neighbor. Different types of flows have different data rates. A participant should fit a number of different types of flows, with different transmission rates within its capacity. Let's simplify a little, supposing that all the video flows have the same traffic rate and all the audio flows have the same traffic rate. For example, that an audio flow rate is equivalent to three times the participant position flow rate, and that a video flow rate is equivalent to ten times the participant position flow rate. A participant with a capacity of 50 may receive simultaneously 15 position flows, 5 audio flows and 2 video flows or 8 position flows, 4 audio flows and 3 video flows. Which choice has to make the participant in this case? Score-SSM leaves this according to each participant's local interests. The things become more complicated if the participant has to shrink its area of interest: should it renounce to three position flows, or at one audio flow or at one video flow ? In the same time, each participant has an area of interest for each type of flow. These areas of interest are concentric circles having the center on the participant's position in the VE and different radius, as in the figure 4.8. The outermost circle is for the position flow, for the reasons mentioned above. Then it follows the other circles, in the order of the traffic rate each flow generates: the bigger is the traffic rate, the closer it is to the participant. There are two reasons for this choice: first, the higher is the traffic rate, the less flow fit into the participant's capacity; second, to received a high rate flow from a participant we suppose that a participant should be able to receive all

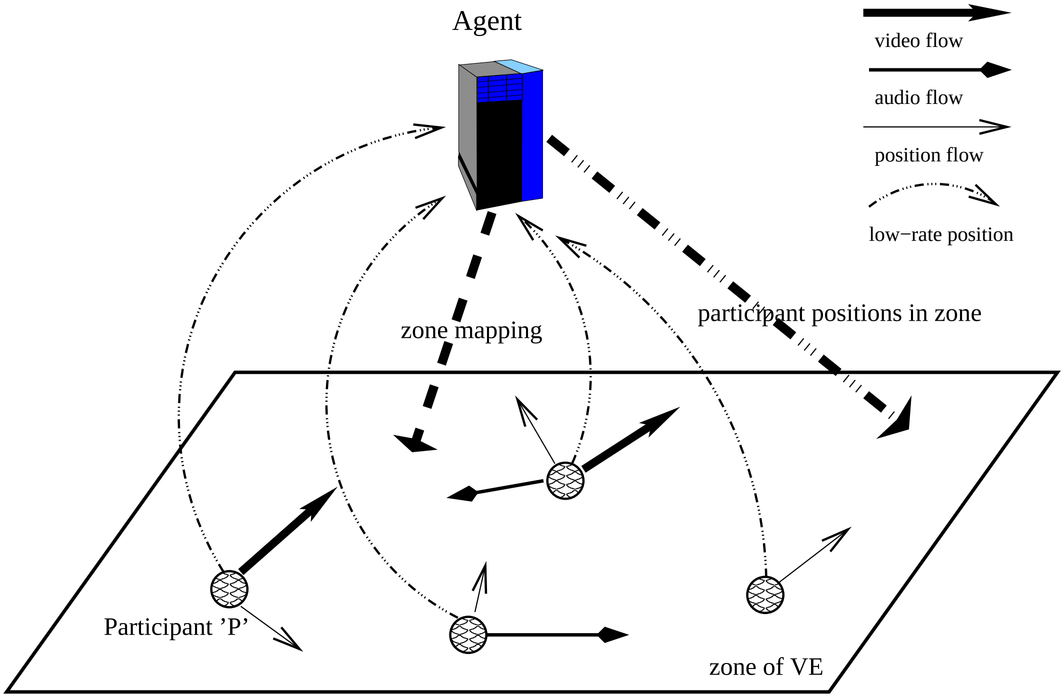


Figure 4.8: The communication flows of Score-SSM

the lower rate flows sent by this participant. For example, a participant that is able to receive a video flow from another participant it should also be able to receive the audio flow send by that participant. On the contrary, if a participant is able to receive an audio flow from another participant, it may not be able to receive the video flow sent by the same participant, due to the higher traffic rate used by the video flow compared to the audio flow.

4.2.4.3 The role of SSM in participant filtering

The use of SSM model allows to refine the filtering of the participants, at the cost of increased complexity. If a participant transmits more data flows, each of these flows is sent on a different multicast channel. The channel's source address is the participant's IP address and the channel's group address is a class D IP address. The position flow is always sent on the G_1 multicast group, while the text flow is always sent on G_2 multicast group, the audio flow is always sent on G_3 multicast group, the video flow is always sent on G_4 multicast group. So, it is enough to only know the IP address of a participant and the flows types it transmits to determine the channel address of a particular data flow sent by that participant. Of course, in order to support more participants from the same IP address we have to send, as signaling, different group addresses for each of those participants. This helps to distinguish among similar types of data flows transmitted by two different participants having the same IP address. The fact that it is sufficient to get the IP address of a participant, through the agent signaling in order to know all the channel addresses of its data flows may significantly lower the signaling. Each signaling packet has 4 bytes less, by economizing the space for advertising a channel group address. As the SSM address range is very wide, a participant may transmit a large variety of data flows without worrying to run out of multicast addresses. Also, the use of a large number of channels relieves the task of distinguishing among one participant's data flows in order to subscribe only to a specific type of data. In addition, the data coming from two different participants are well separated, and not mixed together like in some ASM based architectures. This excludes the possibility of receiving superfluous data while subscribing to a desired data flow.

In addition, the possibility to use a large number of channels for each participant data flows open more possibilities: a participant may send layered video, for example, thus each receiver may subscribe to a number of channels according to its capability and interest. This can be very useful if the participant is a cinema room in the VE, for example.

From the participant point of view, the usage of SSM channels costs almost noting: only a subscribe message at the beginning and an unsubscribe message at the end of communication. In addition the use of multicast allows the participants to send only once a data packet regardless the number of receivers willing to receive it. This is a very important gain if we think at the traffic rates of the video flows, for example. This big gain in network and computing resources can be used by the participants to improve their perception of the VE.

The visibility of the flows sent by a participant is not limited to a bounded area of the VE. Any distant participant whose area of interest contains a given participant is able to receive the data flows sent by that participant. Some participants, with big area of interest, may receive the data flows emitted by a remote participant, while, in the same time, some closer participants, but with a small areas of interest, are not capable of receiving the same data flows.

4.2.5 The different types of flows involved in Score-SSM communication

The purpose of the entities connected in the VE is to communicate with other members in the virtual world. In order to assure the exchange of data between participants and to establish the signaling required by the two levels of filtering, Score-SSM uses more communications flows.

In Score-SSM we distinguish two types of communicating entities: the agents and the participants. Different type of traffic (data and signaling) may flow between these entities. The figure 4.8 shows the different flows involved in the communication. An overview of all the flows of Score-SSM is presented in the table 4.1

Depending upon the source of these flows, we can enumerate the agent-generated flow and the participant-generated flows.

Nb.	**Name**	**Source**	**Destination**	**Type**	**SSM**
1	[agent addresses] - [VE areas mapping]	Main Agent	all	Signalling	Yes
2	participants positions flow	agent	participants	Signalling	Yes
3	inter-agents flow	agent	Main agent	Signalling	Yes
4	[multicast channel] - [zone mapping]	agent	participants	Signalling	Yes
5	fast-rate participant position	participant	participants	Data	Yes
6	slow-rate participant position	participant	agent	Signalling	No
7	data flows	participant	participants	Data	Yes

Table 4.1: The "utilization" values according to the maximum number of participants simultaneously admitted in a VE zone

4.2.5.1 The Main agent

The Main agent is the "supreme" authority of the VE. The Main agent is a machine belonging to the supervisors of the VE. The Main agent fulfils two key-roles in the VE: it provides an entry point in the VE and it supervises the signaling in the VE.

As a first role, the Main agent is the source of a "well known" multicast channel: the [agent addresses] - [VE areas mapping] flow. This channel contains the IP addresses of the agents and the areas of the VE they manage. The address of this "well known" multicast channel is hard-coded in the agent and in the participant "start-up" applications. A "new arriving" agent is in standby until it receives the invitation, over this multicast channel, to manage an area of the VE. A "new arriving" participant listens this multicast channel to get the address of an agent. Then, the participant contacts the agent in order to get the address of a signalling channel corresponding to a VE zone managed by the agent. If a participant intends to enter at a precise location in the VE, it chooses the area (i.e. the agent) respectively the zone (i.e. the signalling channel) corresponding to the desired location.

As a second role, the Main agent manages the pool of agents. If a machine is candidate to be an agent in Score-SSM, it has to register its intention to the Main agent. The Main agent has the list of the available agents, as well as the list of the connected agents and their mapping to the VE areas. The Main agent receives periodically update messages from every connected agent, containing the number of participants located in the VE area they manage. The received values are compared with the capacity of each agent (the maximum number of participants that every agent can manage, i.e. the capacity of the agent). Thus the Main agent knows at every moment the load (in terms of participants) of every agent.

Depending upon these values, the Main agent decides to introduce more agents into the VE (if there is any agent with the load close to one) or to reduce their number and put them in the list of the available agents (if two ore more agent managing neighboring areas in the VE have their loads under a certain threshold). The agents become aware of the decisions taken by the Main agent when they receive a new VE mapping over the [agent addresses]-[VE areas] mapping channel.

Due to its key role, the failure of the Main agent can bring down the whole VE, as no entry point will exist and the nobody will manage the pool of agents. In practice, to avoid the failure of the Main Agent and to improve the VE responsiveness, duplicate, synchronized machines, may exist. For example, one Main agent can be launched per continent.

4.2.5.2 Agent-generated flows

The agent generates three types of flows:

a. the participants positions flow,

b. the inter-agents flow,

c. the [multicast-channel]-[zone-mapping] flow.

a. The participants positions flows for the zones managed by the agent. This is the signaling flow containing the positions for all the participants located in the same zone of the VE. As the agent receive position updates from the participants only once per second, it has an approximate knowledge of the location of the participants in the VE. The agent may manage more zones, so it sends one position flow for each of the managed zones. For each zone managed, the agent keeps a list of the participants located in that zone. This list is updated by the participants position messages received once per second from each participant. If a new participant joins a zone, it is immediately inserted in the corresponding list; if a participant leaves or is disconnected it is erased from its corresponding zone list and the agent will not announce anymore its position. The agent sends the approximate positions of the participants with a frequency of once per second. For each zone managed by the agent, the position packets are aggregated in a bigger packet (this reduces by 50% the traffic rate, because for a position packet, the IP datagram header has the same length as the position data). From the participant point of view, it receives the approximate positions of the other participants located in the same zone, once every second. In practice, the agent may send this signaling information at different rates, considering the participants heterogeneity. Participants with lower capacities may subscribe to a slow rate signaling channel, while the participants willing more precision, than the approximate positions of the participants, will subscribe at the high rate signaling channel. Other solution to reduce signaling at the participant's level is to statically split a zone in four or nine smaller, equal-sized zones. This predefined split doesn't require the exchange of more mapping information, while in reduces the signaling (i.e. the dissemination information received by each participant through the approximate position of the participants). In fact, more multicast channels are used to send the signaling information in a VE zone, according to the predefined partitioning of the initial zone. So, this is a hybrid solution which combines the advantages of the dynamic partitioning (taking into account the participants dynamics) with the advantages of the static partitioning (less zone partitioning information). This refines more the intersection of the areas of interest of the participants with the VE zones. In addition, this requires the agent to send twice the same signaling data. If we consider that agent resources are bigger than the resources of the participants with low capacity, this may be a good solution to improve the satisfaction of the participants.

b. Inter-agents flow. This flow is used for the communication with the Main agent. The Main agent periodically transmits the load of the participants, i.e. the number of participants it currently manages divided by the maximum number of participants it can simultaneously manage. This information is used by the Main agent to decide if it is the case to introduce more agents in the VE, or to reduce their number. This flow also plays a keep-alive role. The absence of the periodical "load update" message will determine the Main agent to consider this agent as dead and to proceed to its replacement. As the other agent flows, this inter-agents flow is sent using multicast. All the possible back-ups of the Main agent subscribe to it, in order to know precisely the agent load.

c. Multicast channel - zone mapping flow. This flow advertises the mapping of the VE zones managed by the agent to the multicast channels. The packets contain the boundaries of the zones [1] and the corresponding channel addresses. Because an SSM channel address is formed by the source address and by the group address, and in this case the source address is the same for all the zones managed by the same agent (the agent IP address) the only information required to identify the channel is the group address (or more precisely the suffix of two bytes length, in the case we use the same prefix for the groups addresses). This information is used by the participants in the case they change their current zone or to determine the zones overlapped by their areas of interest. Also, this mapping information is used by the participants to detect changes in the zones boundaries after a split zone or a unify zones operation made by the agent. Note that an agent keeps

---

[1] For squared zones, we send the "x" and "y" coordinates of the lower left corner, and the size "l" of an edge.

the channel-zone mapping information only for the zones it manages; it is not interested in zones that are not under its "authority". This keeps the architecture more scalable with regard to the total number of zones composing the VE. The mapping messages can be sent more or less frequent, this indicates how fast the agent reacts to the dynamics of the participants.

4.2.5.3 Participant-generated flows

The participants are the sources of the following types of flows:

a. the fast-rate participant position flow,

b. the slow-rate participant position flow,

c. the text flow (optional),

d. the audio flow (optional),

e. the video flow (optional).

a. The fast-rate participant position flow. This flow advertises the current position of the participant. A member of the VE communicates with the participants located in its neighborhood (the other members of the VE whose positions are contained in its area of interest). This flow helps the participants to get a coherent image about their neighbors. When a participant "A" enters in the area of interest of a participant "B" [2], the participant "B" subscribes to the fast-rate position flow of "A". At its turn, the participant "A" subscribes at the fast-rate positions flows of the participants located in its area of interest. The higher is the position announcement rate of a participant, the more precisely other members of the VE know its location. Ideally, the participant casts its position after each movement in the VE (but at the price of more bandwidth used).

b. The slow-rate participant position flow. This flow is destined to the agent which manages the area of the VE where the participant is located. The participant advertises its position uniquely for its agent (using UDP, unicast) at a slow rate, for example once per second. This information is used by the agent to build an approximate view of the area of the VE it manages. The agent sends this information towards the participants on the "participants positions flow" for every zone of the VE it manages. A high frequency rate gives to the agent a better knowledge of the VE , but it costs in terms of communication bandwidth thus reducing the maximum number of participants the agent is capable to

---

[2] A participant uses the "participants position flow" of its agent to detect the new participants entering in its area of interest.

communicate with.

Another role of the position announcement flow is the "keep-alive". If the agent doesn't receive any position message from a participant for a couple of seconds, it will consider that participant as disconnected and it will cease to transmit any position announcement about it.

The participants positions flows packets contains, beside the position coordinates x and y, their corresponding IP addresses, their Ids, their capacity and the types of flows the participant is able to transmit.

c. The text, audio and video flows. These flows contain the data send by the participant. Besides moving through the VE, the participants communicate between themselves. This communication may be textual, audio or video data. A participant may transmit zero or more types of such data flow. Each of these flows has a different traffic rate. A video flow, for example, may have a data rate significantly superior to that of fast-rate participant position flow. The participant uses a different channel for each type of data it is capable to send. This separation between flows comes into support of heterogeneity: some participants may be interested (and/or capable to receive) only some types of data sent by other participants. In order to receive a specific flow of a participant it is sufficient to subscribe to the corresponding channel. The source participant doesn't need to know which the receiving participants are.

As the traffic rate of these flows may be very high, the use of multicast can significantly reduce the bandwidth required to receive this data.

We mention that all the flows of Score-SSM are multicast flows, sent over SSM channels. The only exception is the slow-rate participant position flow which has only one receiver (the agent) and in consequence is send as unicast.

4.2.6 The different communication phases

After we have mentioned the communication flows involved in Score-SSM, let's see how these flows are used by our architecture. In this section we describe the different communication phases of Score-SSM: the connection of a participant to the VE, the movement of a participant within the VE, the communication between the participants, the unification of two or more neighboring zones, the division of a zone into sub-zones and the inter-agent communication.

4.2.6.1 Participant connection to the VE

To enter in the VE the new participant needs the address of an agent. This address could already be known from a previous connection, it can be obtained from a rendezvous point, it may be taken from a web page, it may be hard-coded in the connection application etc...

It may be also possible that the participant intends to enter in the VE at a precise location. For example, this location may be the coordinates of a cinema room inside the VE, of a discussion room with a given topic, of a friend (in the "real" life) of the participant. In this case, the participant listens to the [agents addresses]-[VE areas mapping] channel in order to get the address of the agent corresponding to the desired staring location. The participant notifies the agent about its intention to join the VE. The agent responds by assigning a starting point to the participant and by communicating the positions of the participants located in the same zone of the VE. With this information, the participant computes which are the participants located in its area of interest and starts to communicate with them.

Here are the participant connection steps:

1. The participant acquires an agent address, as explained above.

2. The joining participant notifies the agent about its presence by sending its ID, IP address, its capacity and the desired staring position (or 0).

3. The agent puts the new participant in its active participants list, corresponding to the zone where the participant is located. If the participant didn't specify any desired position, the agent chooses itself a starting position for the participant in one of its subordinated zones. In addition, the agent subscribes to the slow-rate position channel of the new participant.

4. The joining participant subscribes to the participants positions announcement channel of its agent in order to receive its starting point in the VE.

5. The participant subscribes to the agent's multicast channel-zone mapping flow to get the coordinates of the nearby partitioning of the VE and the multicast addresses corresponding to these zones.

6. The participant subscribes to the participants positions announcement channels corresponding to the VE zones intersected by its area of interest.

7. If the participant area of interest overpasses the borders of the zones assigned to the current agent, the participant has to listen to the agents-VE areas channel, in order to find out the addresses of the agents responsible for these neighboring zones. Then, the participant subscribes to multicast channel-zone mapping flows of the new agents to precisely determine the channels addresses of the participants position announcement flows.

8. By listening to the participants' positions announcement channel, the joining participant finds out who are its closest participants and creates a list of neighbors to communicate with, according to its interests.

9. Once its closest neighbors are determined, the participant subscribes to the fast-rate position channels of these neighbors. In addition, it subscribes to the text, the audio or the video channels of the neighbors located in its text, audio, respectively video range and according to its capacity.

10. The participant may unsubscribe at any time from the multicast channels at which it is currently subscribed.

4.2.6.2 Typical participant behavior in the VE

The typical behavior of a VE participant consist in: communicating with other participants, moving freely in the VE, listening to the corresponding participants positions announcement flows and the multicast channel-zone mapping flows sent by the agents, periodically computing its closest neighbors, checking if its area of interest overpasses the current zone borders and if so subscribing to the corresponding channels.

We describe the participant behavior in the VE:

1. The participant determines a location (with some probability) within the VE limits and it moves toward it at a constant speed. This destination may be chosen completely random or with a given probability. Once arrived at the destination, this step will be reiterated.

2. The participant communicates with other participants: it receives the neighbors' positions and their data flows. At its turn, it sends position and different data flows (depending on its capabilities).

3. It updates, with some frequency, the closest neighbors list and it subscribes to the new neighbors flows if any, or unsubscribe from the participants flows which are no longer in its area of interest.

4. When the participant reaches the limit of its current zone it changes the subscription at the participants' position announcement channel to the address of the new zone's corresponding channel. If the new zone is managed by another agent, first, the participant will get the IP address of this new agent, according to the procedure described at the connection step, then, it subscribes to the corresponding signalling channel.

4.2.6.3 The division of a high participant density zone

Due to the dynamics of the participants within the VE, at some particular moments in time, a zone of the VE may reach the maximum number of participants admitted in a VE zone. In this case, the agent responsible for that zone splits the zone into four smaller zones of equal sizes. This procedure reflects the dynamical aspect of the Score-SSM zone-cut algorithm.

Here are the steps to be followed to divide a zone:

1. The agents detects that one of its zone has reached the maximum number of participants admitted in a zone.
2. The agent divides that zone into four smaller, equal-size zones. The first of these zones keeps the same channel address as the old zone. Three new multicast group addresses are allocated by the agent to the newly created zones.
3. The agent uses its multicast channel-zone mapping flow to advertise the new zone-cut to all the listening participants.
4. The agents start to transmit the approximate positions of the participants over the newly created multicast channels.
5. The participants directly allocated by the new zone-cut, will modify accordingly their subscriptions to the participants positions announcements flows.

4.2.6.4 The unification of under-populated zones

In order to keep the VE as compact as possible and to reduce the signaling overload, the agents are in charge of unifying some of their under populated zones. In practice, if the sum of the number of participants located in four neighboring zones (belonging to the same agent) is lower than the maximum number of participants admitted in a VE zone multiplied by a sub-unitary constant, the agent may decide to unify these zones into one larger zone, covering the surface of the smaller four zones.

Here are the steps to be followed to accomplish the zones unification:

1. Due to a periodical check, the agent detects four under populated zones, candidates to unification.
2. The agent creates a larger zone, equals to the surface of the four smaller zones. It assigns to this zone the address of the participants positions announcement flow of the zone containing the biggest number of participants among the four unified zones.[3]
3. The agent uses its multicast channel-zone mapping flow to advertise the new zone-cut to all the listening participants.

---

[3] Through this algorithm of choice, we reuse one multicast address, while minimizing the number of participants obliged to change the signalling channels to which they have subscribed.

4. The agent starts transmitting the approximate positions of the participants on the newly created multicast channel. It stops transmitting positions over the formers channels addresses. These addresses become available for further use.
5. The participants located in the former zones subscribe to the new zone's corresponding channel and leave the older channels. This update is made also for the participants whose area of interests intersects at least one of the former zones.

4.2.6.5 Adding more agents in the VE

The number of connected participants is variable during the lifetime of the VE. Each agent manages the signaling for a limited number of participants, according to the capacity of that agent. If an agent reaches its capacity, it demands to the Main agent, the launching of more agents to share its signaling traffic. The Main agent acknowledges this demand by sending the addresses of three agents ready to be launched.

Here are the steps in launching more agents:

1. An agent reaching its capacity sends a divide request to the Main agent. This message is sent on its inter-agent communication channel.
2. The Main agent acknowledges this request by sending a new mapping over the agent-VE areas channel. The area corresponding to the demanding agent is divided into four smaller areas, of equal surface. The demanding agent continue to be used for the first of these smaller zones. In addition, three new agents are launched. The Main agent has a list of the available agents, ready to be launched into the VE.
3. The demanding agent listen this announcement in order to get the addresses of these new agents.
4. The agent sends a new mapping over its multicast channel-zone mapping flow. It uses the addresses of the new agents in the channel source addresses for the zones belonging now to these agents.
5. The participants receive the new mapping and then they subscribe to the participants positions announcement channels corresponding to the new agents. In addition they send their slow-rate participants' position flows toward the new agents.
6. The new agents start to build their own list of the participants located in their zones. They use these lists to send signaling traffic over their participants positions announcement channels.

4.2.6.6 Reducing the number of agents in the VE

At some moments during the runtime of the VE, the number of participants connected in the VE may decrease and some agents may experience very little load. In order to simplify the communication infrastructure and to reduce the signaling traffic, some agents may be put off-line, in the list of stand-by agents. It is the Main agent who has a global view of the agents load and who initiates the unification procedure for some neighboring agents.

Here are the steps to accomplish for reducing the number of connected agents:

1. Each agent transmits periodically a message toward the super agent containing its load percentage: the rapport between the participants located in its subordinated area and the capacity of the agent.

2. The Main agent surveys the agents load. When sum of the loads of four agents managing neighboring VE areas is under a certain threshold, the Main agent takes the decision to unify these agents, by replacing them with only one agent.

3. The Main agent sends a new mapping over its agent-VE areas channel. In this mapping the four selected agents are replaced by only one agent (chosen among these four) who will manage a VE area equal to the unified areas managed by the selected agents.

4. When the concerned agents receive the new agent-VE areas mapping, they will send, over their [multicast channel]-[zone mapping] flows, the new agent coordinates.

5. The participants located in the concerned VE area receive the address of their new agent. They will send their slow-rate participants' position flows toward the new agent.

6. The new agent starts to build the list of the participants located in its managed area. This list is used to determine a new zone-cut in that VE area.

So, in this chapter we have presented two communication architectures for LSVE, one of them based on the ASM multicast model and one of them based on the SSM multicast model. Score-SSM has a more complex design compared to Score: the former requires the transmission of more information to establish the communication, but, on the other hand it provides a "finer grain" filtering compared to the later approach. In the next chapter, we compare the efficiency of the two communication architectures through experiments.

Chapter 5

Score-SSM experiments

This chapter is composed of four parts: in the first part we present the experiments set-up, in the second part we compare the performances of Score and Score-SSM. In the third part of this chapter, we present some tests meant to illustrate the behavior of Score-SSM with different parameters values. With these tests results we improve the performances of our architecture, by using the appropriate parameters settings and by introducing some specific mechanisms in our architecture implementation. In the four part of this chapter we

5.1 Experiments set-up

In order to proof the feasibility and the efficiency of our communication architecture we have realized an implementation of Score-SSM. This implementation consists in two independent programs: the *"Agent"* and the *"Participant"*. Their code was written in C++, and they run over NetBSD, Linux and Windows. The code have been built to use the Source Specific Multicast model. So, IGMPv3 must be executed on the participants machines and PIM-SM, supporting SSM address range, should be deployed on the network's routers. The Score-SSM implementation was made with the purpose of having simultaneously connected in the VE hundreds (thousands) of participants. After running a large number of tests, the improved version of software has been integrated into a virtual environment application (see chapter 6).

5.1.1 Implementation details

In order to test Score-SSM as a communication architecture for VE, we have defined a specific virtual world, without using any graphical engine. We have defined a square world with the size of 600*600 units. During the runtime, the agents divide the VE into square zones, according to the local density of participants. There is no limit for the number of zones in which the VE is divided. When the number of participants in a zone overpasses a certain threshold [1] (in most of our experiments we have considered this threshold equal to 30 participants), that zone is divided into four square zones with equal surfaces. On the contrary, if the total number of participants located into four neighboring zones of equal surface is lower than a certain threshold (0.8 multiplied with 30, in our experiments) those zones are unified into a single, larger zone. So, at any moment, the number of zones composing the VE is directly proportional to the number of participants connected in the VE. In addition, the size of the zones from an area of the VE is inversely proportional to the density of the participants in that area. In order to evaluate the efficiency of the dynamic zone-cut mechanism, we have considered a static division of the VE. The VE was divided into 6*6 square zones, each of these zones having the side length equal to 100 units. The static zone-cut doesn't take into account the number of participants connected in the VE or their distribution. On the other hand, a dynamic zone-cut limits the signalling in the areas of the VE with high participants density by partitioning these areas into a number of small zones. The dynamic partitioning is effectuated with a given frequency. In our tests we have chosen two different frequencies: each 1 second in the fist case, each 6 seconds in the second case. The agents transmit the VE partitioning to all the connected participants.

5.1.2 The experimental platform

In our experiments, we have used a single agent to manage the signaling of the entire VE. The experiments have shown that the processor "load" for this unique agent is low (3% load for a Pentium 500MHz with 512MB of RAM) even for a large number (700) of participants connected in the VE. The use of a single agent does not affect the quality of our experiments because the mechanisms evaluated are not influenced by the presence of one or more agents.

The tests were realized over the Planete research team's experimental network at INRIA Sophia Antipolis. The experimental environment (figure 5.1) was composed of 5 NetBSD computers, connected at a 100Mbits/sec Ethernet network (3 of these computers were Pentium 500MHz with 512MB of RAM, one computer was Pentium 2GHz with 1GB of RAM and one computer Pentium 250MHz with 256MB of RAM). One of those computers was executing the agent; the other 4 computers hosted the participants: one participant in reception mode launched per computer and, in addition, 82 participants launched in no-reception mode ("automata") on three of these computes. So, in total 250 participants were launched. The participants launched in no-reception mode were meant to generate data traffic and signaling (through subscribe/unsubscribe messages) in the zones of the VE. The four participants launched in reception mode have subscribed to the data flows

---

[1] This threshold represents the maximum number of participants simultaneously admitted in a zone of the VE.

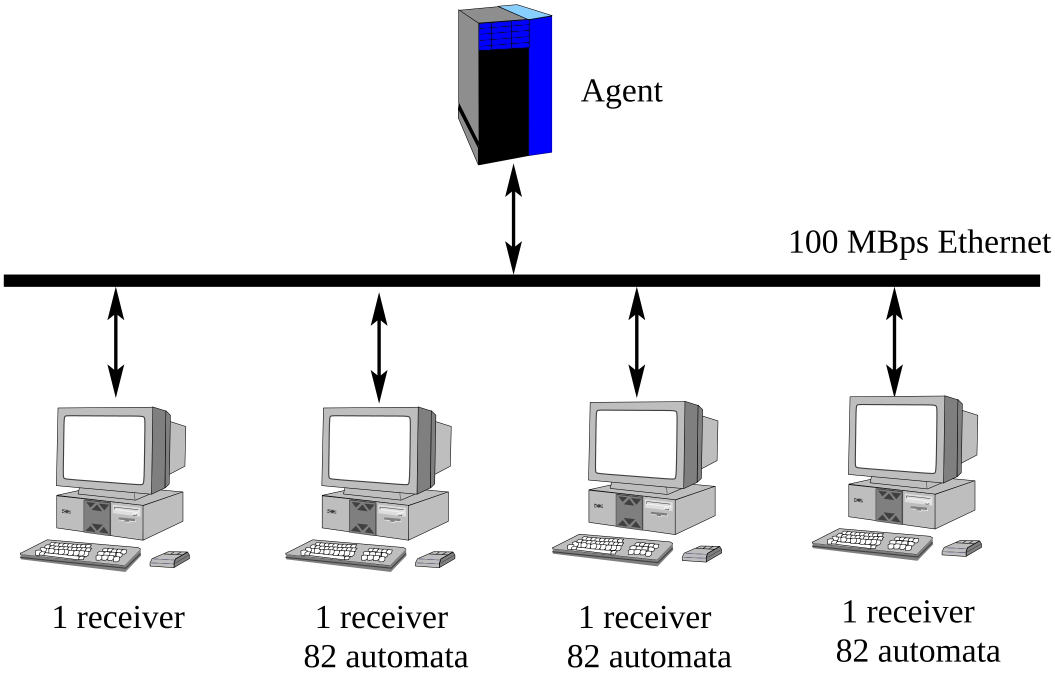


Figure 5.1: The experimental testbed used in our experiments

generated by their neighbors and they have used the data packets received to compute their satisfactions.

5.1.3 Trace generation

At the end of the execution of any experiment a trace file is generated. A large number of parameters were traced in order to evaluate the performances of the participants. The useful information was obtained from the traces generated by the participants launched in reception mode and also from the agent's traces.

Here is the list of information contained in the trace files of the *"participant"*:

- the number of data packets received from the neighbors
- the number of data packets sent
- the number of subscribe/unsubscribe operations to the data multicast channels of the neighbors
- the number of subscribe/unsubscribe operations to the multicast channels corresponding to the zones of the VE overlapped by the area of interest of the participant
- the number of signalling packets received from the agent containing the position of the neighbors
- the number of packets send to the agent as slow-rate position signaling

- the average number of packets received from the neighbors as fast rate position advertisement
- the average participant satisfaction in the VE
- the number of data packets lost by the participant - the evolution of number of multicast channels subscribed by the participant

And the traces of the *"agent"*:

- the evolution in the number of zones in which the VE is partitioned
- the evolution of the number of participants connected in the VE and their repartition over the zones composing the VE

A program have been written (in C++ language) in order to analyze the trace files and to provide an output in a format which allows easily the generation of the graphs (using "gnuplot"), over Linux operating system.

5.1.4 Parameters used in tests

1. In order to express the **heterogeneity** of the participants during the experiments, the capacity "C" of each participant is randomly chosen at the launching time. The participants have the same emission rate (using the fast-rate position advertisement flow), so we can express the capacity of a participant as the maximum number of sources (neighbors) that it is able to receive and process in the same time. Different distributions of capacity have been used in our experiments:

    - we use an uniformly distributed capacity between 20 and 40 sources/second,
    - between 10 and 50 sources/second
    - 99% of the participants have their capacities uniformly distributed between 30 and 50 sources/second and 1% of the participants between 10 and 20 sources/second.

    Note that in Score-SSM architecture, every participant subscribes to a number of neighbors less or equal to its maximum capacity. There is no superfluous data traffic. The possible losses are due to the network congestion or to the peaks in the signaling traffic received by the participant at a given time.

2. We use a static or a dynamic **partitioning (or zone-cut)** of the virtual world, in order to evaluate the interest of using a dynamic zone-cut. A VE is "statically" partitioned if the virtual world is divided at the launch time into smaller areas, and this partitioning doesn't change during the existence of the VE, regardless the number of participants connected in the VE and their distribution. On the other hand, a VE

is "dynamically " partitioned if the partitioning changes according to the number and the local density of the participants connected. In figure 5.2 we have applied a static, respectively a dynamic partitioning of the VE, to a given configuration of the participants.

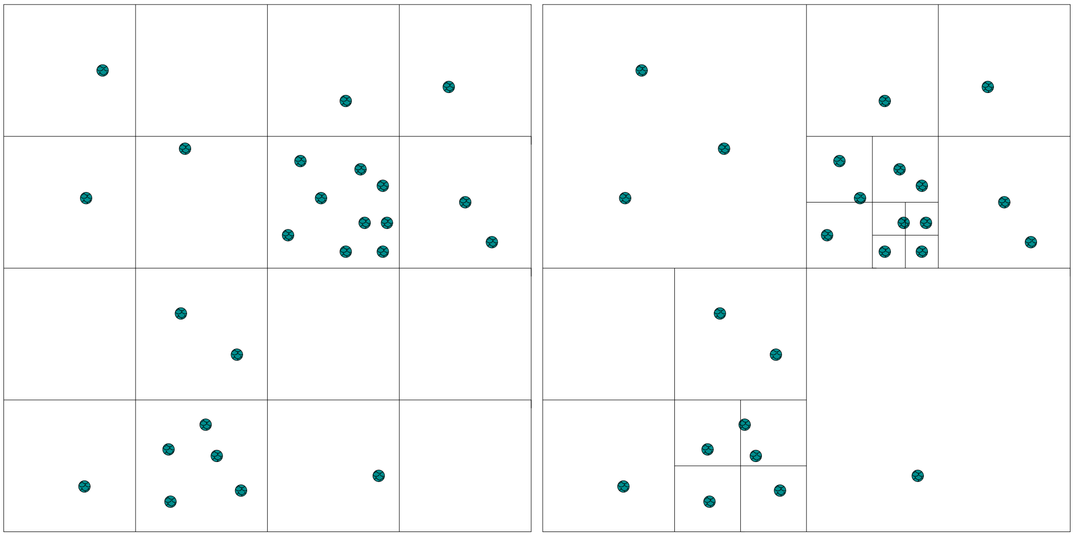


Figure 5.2: Static (left) and dynamic (right) partitioning of the VE

The **zone-cut** computation **frequency**, it is the interval between two agent computations of the VE partitioning, in the dynamic case. We have considered two frequencies: 1 second and 6 seconds. In the experiments it is noted with **"PR"**.

3. We use different sizes of **area of interest** (we note it **IArea**) of the participants, as we can see in the figure 5.3. The area of interest is a square with the participants centered in the middle of this square. In most of the following experiments we have studied 10 different area of the interest: the smaller area has the side of 10 units while the bigger area has the side equal with 100 units. 100 units is also the side of the zone in the static partitioning case (we note it ***CellArea***). In the experiments we express the area of interest of a participant as a percentage of the area of a zone in the static case. For example, a square area with the side of 50 units may be expressed as IArea = 0.25 CellArea. The participant subscribes to the data flows of the neighbors located in its area of interest, in the limit of its capacity.

4. The degree of dynamics of the participants in the VE is given by their moving **speed**. We have used two values for the speed of participants: 5 units/second and 50 units/second, where 50 is the segment size of a square cell in the static zone-cut case. In the experiments the speed is noted with **"V"**. The participants are moving in the VE according to following rule:

   - first a destination is computed (as explained below)

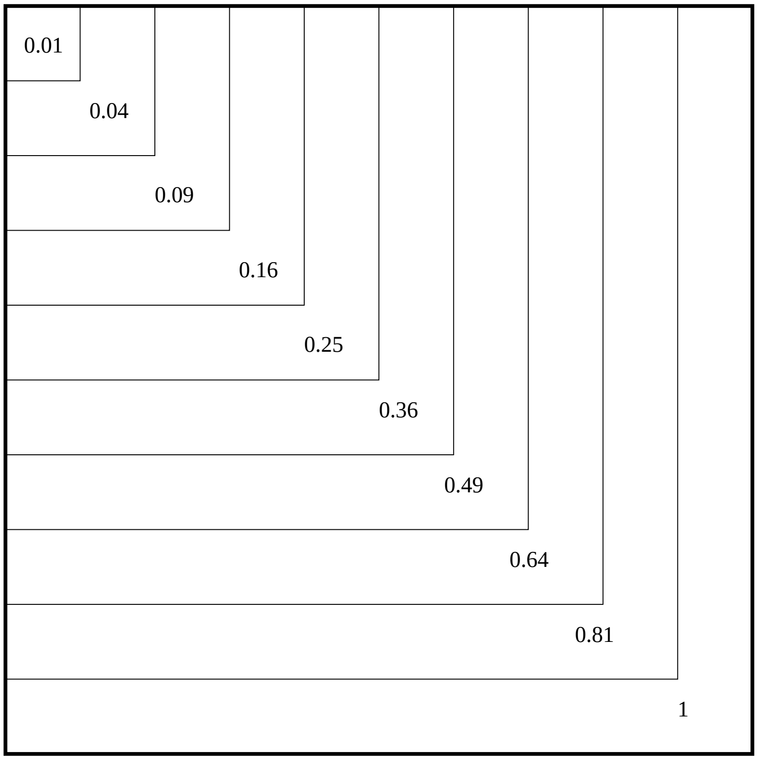


Figure 5.3: Different sizes of the area of interest of a participant used in our experiments

- then the participants are moving toward the destination with its constant speed (chosen at the launching of the participant)
- when the destination is reached, a new destination is computed, as in the first step

Concerning the dynamics of the VE, the choice of the destination is important to compute the trajectory of the participants in the VE. While entering the VE, the participant is randomly placed by the agent, according to a given distribution law (a second possibility is that the participant specify an entry point in the VE). Then, its destination is computed using the following steps. First, the VE is statically divided into 3*3 start zones of equal surfaces. The destination start-zone is chosen according to a probability matrix having the following values: $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}, \frac{1}{256}\}$. The mapping between the start zones and the above probabilities is updated periodically by the agent, in a random manner, all the possible mappings having the same probability. These probabilities are meant to create "hot spots" in the VE, to be more realistic in simulating a virtual world in which the participants are walking and communicating with each other. After a destination zone is chosen for a participant, the destination point is randomly chosen using a uniform probability. When a participant reaches its destination point, a new destination is computed, using the above algorithm.

In order to facilitate the execution of different experiments, the software have been built to accept a large number of parameters, which may be easily changed from the command line.

For the "*Participant*" application we can specify the following parameters:

- the size of the Virtual Environment (VE)
- the IP address of the Main agent, in order to connect to the VE
- the lifetime of the participant in the VE
- the reception capacity of the participant
- the outgoing throughput of the participant
- the moving speed of the participant in the VE
- the size of the area of interest of the participant
- the possibility to run in reception mode, in order to trace participant interaction in the VE or to run in "automata" mode i.e. only to send data traffic and generate signalling
- the frequency of sending slow and fast rate position message
- the maximum number of neighbors from which the participant may simultaneously receive data

For the "*Agent*" application we can specify the following parameters:

- the size of the VE
- the agent lifetime in the VE
- the possibility to run in "static" or in "dynamic" partitioning [2] mode of the VE
- the maximum number of participants simultaneously admitted in a zone of the VE
- the timer value: for transmitting agent-zone mapping, for participants' positions announcement rate, etc...
- the maximum number of participants which the agent is able to simultaneously manage in the VE

---

[2] Alternatively, the term "zone-cut" is used as a synonym for "partitioning" in order to designate the decomposition of the VE into zones.

5.2 Score vs. Score-SSM comparison

In the first set of experiments, we have compared the performances of Score-SSM with the performances of Score. We have chosen Score as representing LSVE communication architecture due to its capability to dynamically adapt to the participants distribution in the VE and due to the availability of a large set of tests results.

In all of the following figures we emphasize the cumulative distributions of the satisfactions of the participants. These satisfactions are computed using the participant satisfaction formula described in the section 4.2.2.

$$S = \frac{U_r}{min(U_t, C)}$$

These satisfactions have been collected in the trace files, during the simulations runtime. For a given satisfaction value "N", measured on the "x" axis, we can observe, for every curve, the percent of satisfactions less or equal with "N" (the projection over "y" axis).

5.2.1 Static and dynamic zone-cut in the case of heterogeneous participants

In this experiment we compare the satisfactions of the participants in the case where the virtual world is statically partitioned into smaller zones with the case where the virtual world is dynamically partitioned, according to the local density of the participants.

Two different heterogeneity levels have been considered for each partitioning mode: the participants capacities are uniformly distributed between 20 and 40 sources/second, respectively the participants capacities are uniformly distributed between 10 and 50 sources/second.

The tests have been run for 10 different area of interest of the participants, the side of the IArea have been taken equal to {10,20,...,100} units (as in figure 5.3). The speed of the participants was equal with 100 units/second and the zone-cut frequency equal with one second.

Score case

From the 10 area of interest considered in the experiments we present here only two graphs, in figure 5.4 left it is shown the graph for IArea = 0.16 CellArea while in figure 5.4 right it is shown the graph for IArea = 0.81 CellArea. For every curve, for a given value "S" of the satisfaction, represented on the "X" axis, on the "Y" axis is represented the cumulative percentage of participants having their satisfactions less or equal with the value "S".

For all the graphs and regardless the participants heterogeneity level, the use of Score architecture leads to an important increase of the participants satisfactions (compared with a static partitioning of the VE). On the Score graphs (figure 5.4) we can observe that for both heterogeneity levels of the participants, the curve representing the cumulative satisfactions of the participants in dynamic zone-cut case (Score) is below the curve of static zone-cut case.

On the other hand, the Score graphs (figure 5.4) also show that the larger the participants area of interest becomes (for ex. IArea = 0.81 CellArea), the smaller is the performance increase due to the dynamic zone-cut mechanism. The overall participants satisfaction is also lower compared to the case of small area of interest (for ex. IArea = 0.16 CellArea). Similarly , for small area of interest, the traffic received by the participants is lower than for larger area of interest and the participants satisfactions are close to one. In this situation the use of Score is less necessary, due to the additional signaling cost compared to the static case.

Figure 5.4 also shows that for the dynamic zone-cut, the participants having their capacities distributed between 20 and 40 are more satisfied than the participants having their capacities distributed between 10 and 50. This is because Score's zone-cut mechanism diminishes the size of the cells containing the less satisfied participants ("max-min fairness"). So, the algorithm maximizes the minimal participant satisfaction, than to increase the average participant satisfaction.

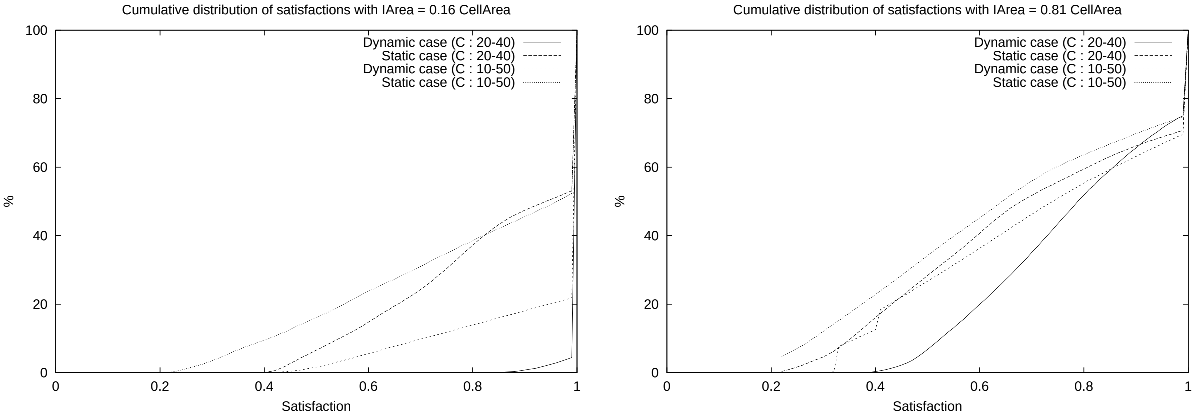


Figure 5.4: The satisfactions of the participants in Score

Score-SSM case

In the figure 5.5 we have on the "X" axis the side of the area of the interest and on the "Y" axis the mean satisfaction of the participants.

Running the previous experiment in the Score-SSM case, we have obtained the participants satisfactions located in the interval [0.99, 1] (figure 5.5). So, all the participants have maximal satisfaction regardless the heterogeneity of the participants and regardless the VE administration mode (dynamic or static zone-cut).

This proves the efficacy of the filtering mechanism of Score-SSM, we will display only for this experiment the graph of the satisfactions, in the experiments described in the following sections we will not show it anymore, but we will enter in more details presenting figures describing the behavior of the other parameters.

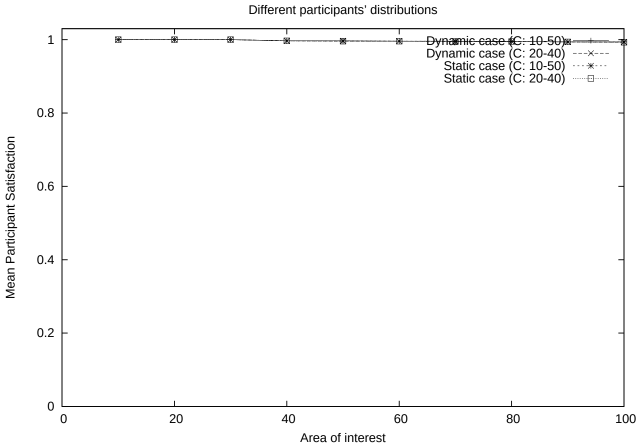


Figure 5.5: The satisfactions of the participants, in Score-SSM

The participant-level filtering mechanism of Score-SSM ensures that every participant receives data only from its designated neighbors and the number of these neighbors does not exceed the participant's capacity. In Score-SSM, a participant communicates with the neighbors located in its area of interest. The capacity parameter of a participant influences only the maximum number of neighbors this participant communicates with. When the neighbors' number exceeds the participant's capacity, a selection is made in order to communicate with the closest (with respect to the Euclidian distance) neighbors in the VE. It is clear that for a large area of interest (ex. for IArea>0.64CellArea), the participant encounters more neighbors and a if its capacity is big enough (ex. C>30 participants/second) it will communicate with more participants than for a small area of interest (ex. for IArea<0.25CellArea). For a small area of interest, the participant's capacity has little influence over the number of neighbors a participant communicates with, because during the lifetime of a participant having a small area of interest, few participants will be simultaneously located in its area of interest. A bigger capacity would be useful only if the participant is present in an area with high density of the participants.

The static or dynamic zone-cut mode has no influence, in Score-SSM, on the number of neighbors a participant communicates with, as the role of the zone-cut is to filter out the superfluous signaling. Different zone-cut modes mean different signaling overload received by the participants, but the set of participants to communicate with remains the same. Compared with the static zone-cut, the dynamic zone-cut is meant to ensure the scalability of the signaling in the case when a big number of participants are located in a small area of the VE. In such a case, the signaling traffic generated by the crowd of participants may become even bigger than the useful data traffic of the participant neighbors.

Score vs. Score-SSM

After we have seen the experimental results of the two communication architectures, we can compare their efficiency: for smaller areas of interest, Score has good performances, closer to Score-SSM. For large area of interest, Score has low performances (close to the static partitioning), the interest of using Score is to improve the satisfactions of low capacity participants, while Score-SSM provide maximum satisfaction to the participants.

5.2.2 Two different distributions of the capacities of the participants

In this second experiment, we compare the satisfaction of the participants for two distribution of capacities:

- In the first distribution, 1% of the participants have their capacities uniformly distributed between 10 and 20 sources/second, while the remaining 99% participants have their capacities uniformly distributed between 30 and 50 sources/second. We consider this as "non-uniform" capacity distribution.
- In the second distribution the participants have their capacities uniformly distributed between 10 and 20 sources/second. We consider this as "uniform" capacity distribution.

  The experiments have been run for 10 different sizes of the area of interest of the participants. The speed of the participants was 10 units/second. The VE was dynamically partitioned (PR=6).

Score case

From the 10 area of interest considered in the experiments we present here only two graphs, in figure 5.6 left it is shown the graph for IArea = 0.49 CellArea while in figure 5.6 right it is shown the graph for IArea = CellArea. For every curve, for a given value "S" of the satisfaction, represented on the "X" axis, on the "Y" axis is represented the cumulative percentage of participants having their satisfactions less or equal with the value "S".

Score experiments (figure 5.6 right) have shown that for large area of interest (approaching to the surface of a cell in the static zone-cut case) the cumulative distribution of the satisfactions of the participants is almost identical for the two capacity distribution studied. For small area of interest (for ex. IArea = 0.49 CellArea), more participants with capacities distributed between [30, 50] have maximal satisfaction (as in figure 5.6 left).

After a first zone-cut, according to the density of the participants (a part of the total available multicast groups are mapped to the zones of the VE), these participants have a satisfaction close to one, the remaining multicast groups (if any) are used to improve the satisfactions of the ten low capacity participants. So, in this situation Score mechanisms have dealt efficiently with heterogeneous participants. For example, for an area of interest equal with half the size of a cell in static zone-cut mode, less than 40% of the satisfactions of the participants are lower than 0.5 in the non-uniform capacity distribution case, while this percentage is 80% in the uniform capacity distribution situation.

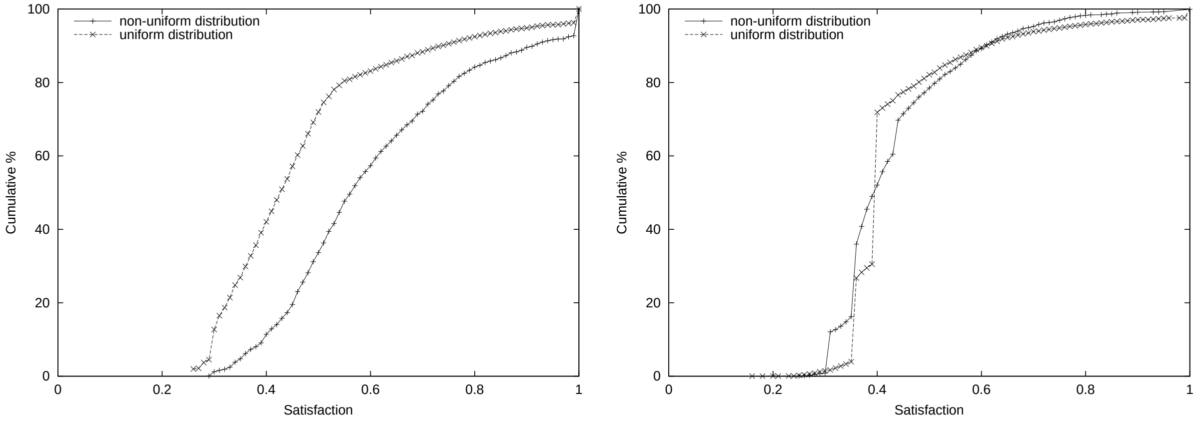


Figure 5.6: The traffic received according to the capacities of the participants, in Score, where: IArea = 0.49 CellArea (left) and IArea = CellArea (right)

Score-SSM case

This experiment, run for Score-SSM architecture, has shown that the satisfactions of the participants are placed in the interval [0.99, 1]. Thus, all the participants are satisfied regardless the uniform or non-uniform capacities distribution. This is due to the participant-level filtering of Score-SSM architecture, which allows any participant to subscribe to a number of neighbors less or equal with its capacity. The capacity of a participant does not influence in any way the data received by other participants because the agent doesn't partition the VE according to the local satisfaction of the participants, as in Score (where the role of the agent is to increase the satisfaction of the least satisfied participants). In Score-SSM, every participant adjusts its satisfaction by subscribing to an appropriate number of neighbors. In a VE zone, a participant with low satisfaction subscribes to a number of neighbors less or equal with its capacity. In order to do this, it selects the closest neighbors among the participants located in its area of interest. A participant with high capacity often is able to subscribe to all the participants present in its area of interest, so the global useful data traffic is more important than for the participants with lower capacities. The signaling data received by a participant is independent of its capacity, but directly dependent on the size of its area of interest. So, a participant with low capacity receives the same signaling data traffic as a participant with high capacity, but on the other hand, the useful data traffic is much lower in the case of a participant with small capacity than for a participant with big capacity. For a large area of interest the signaling traffic becomes important, and for a participant with lower capacities, this traffic rate may be equivalent to the data traffic rate of one or more participants. So, because of the high signaling, the participant's capacity, already low, diminishes further more. In our tests the participants have enough bandwidth to receive all the required signaling. In real case this may not be true, so each participant should adapt its capacity taking into account the amount of the signaling (by unsubscribing from a number of neighbors, in order to reduce the incoming traffic's rate).

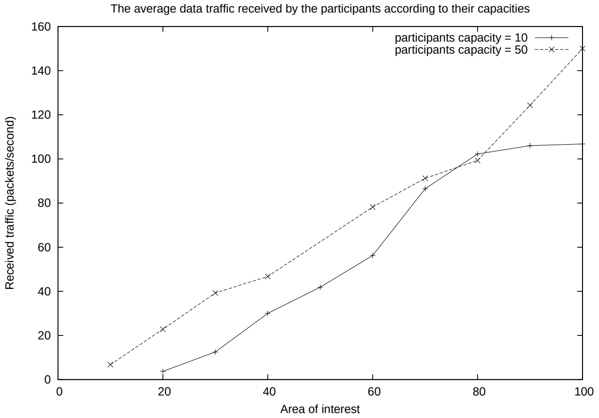


Figure 5.7: The traffic received according to the capacities of the participants, in Score-SSM

The experiments have shown that the capacities of the participants don't influence their satisfactions. On the other hand, a lower capacity participant receives less data traffic (figure 5.7). For small area of interest, the participants are not receiving simultaneously the data traffic generated by many neighbors, so a lower capacity participant receives an amount of data close to the data received by a high capacity participant. On the contrary, for large area of interest, the participants with low capacities receive less data compared to the high capacity participants. However, on the graph we can see that the data received by the small capacity participants (for ex. C=10) it is significantly lower than the amount of data received by the higher capacity participants, even for small area of interest. We explain this by the participants distribution in the VE: as a participant is moving through the VE, it can arrive to an area of the VE with higher participants density, this means a lot of participants are located in its area of interest, so it must limit the number of entering data flows (the maximum number of neighbors simultaneously accepted) to its capacity. On the figure 5.7, we observe that for the areas of interest with the side bigger than 80, the amount of traffic received by the participants with low capacities tends to be constant.

In this situation, a participant has most of the time enough neighbors to fill its capacity, the only thing which stops the participant to get full data traffic is the signaling and the continuous selection of the closest neighbors among the participants located in its area of interest. In the same time, the high capacities participants receive 50% more traffic for the side of their area of interest grater than 80 units. In very few cases, they reach their capacities, most of the time they communicate with all the neighbors located in their area of interest.

Score vs. Score-SSM

Score mechanisms improve the satisfactions of the low capacity participants but they don't improve the mean satisfaction of all the participants. For large areas of interest, the cumulative satisfactions of the participants are similar regardless the capacities distribution. The satisfactions of the participants using Score-SSM are not influenced by the distributions of the capacities of the participants: each participant selects the neighbors to which it communicates, independent of others participants capacities. Score-SSM outperforms Score, regardless the distribution, by assuring maximal satisfaction to the participants of the VEs using this architecture.

5.2.3 Different zone-cut frequencies and participants speeds

The role of this experiment is to study the effect of the participants speeds and the effect of the VE partitioning frequency over the cumulative distribution of the participants satisfaction.

In order to realize the experiment, we have considered two different speeds of the participants: 5 and 50 units per second.

For each of these two speeds we have been considered three partitioning mode of the VE: 1 second and 6 seconds partitioning frequency and the static partitioning case.

This experiment have been executed for 8 different areas of interest.

Score case

Among the 8 area of interest considered, we present in the figure 5.8 the case where IArea = 0.25 CellArea (in the left graph the participants speed is V = 10 and in the right graph the participants speed is V = 100) and in the figure 5.9 the case where IArea = 0.64 CellArea. In these experiments, the dynamic zone-cut leads to higher satisfaction of the participants compared to the static case. This is true regardless the speed of the participants and regardless the partitioning frequency of the VE.

In addition, Score tests have shown that using 1 second zone-cut leads to higher satisfaction of the participants than using 6 seconds zone-cut. This difference between the satisfaction of the participants becomes more important while the participants are moving with a speed of 100 than with a speed of 10.

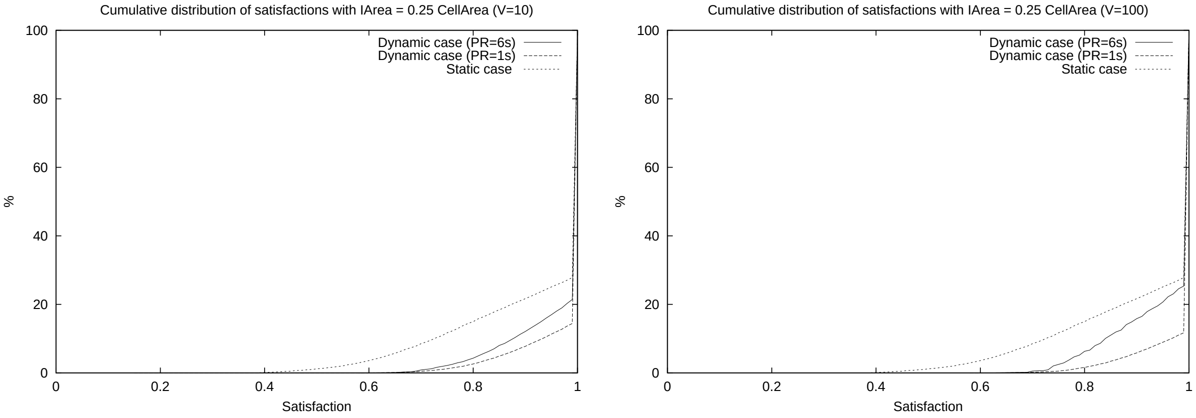


Figure 5.8: The cumulative distribution of the participants satisfaction with IArea = 0.25 CellArea in Score

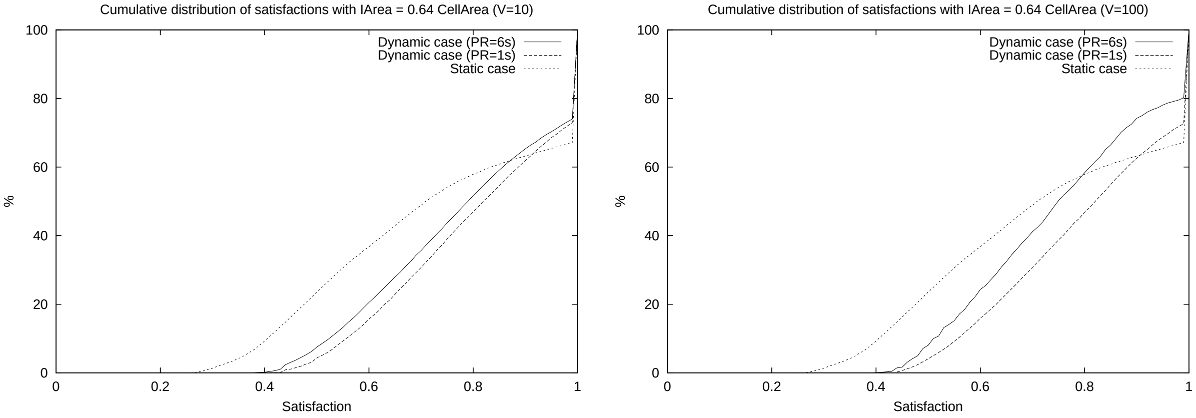


Figure 5.9: The cumulative distribution of the participants satisfaction with IArea = 0.64 CellArea in Score

Score-SSM case

In the 8 areas of interest considered, the satisfaction of all the participants was situated in the interval [0.99, 1]. This shows again the efficiency of the participant-level filtering mechanism. The influence of the zone-cut frequency over the satisfactions of the participants is less important in Score-SSM compared to Score, because it has direct impact only over the signaling traffic send by the agent to the participants, and not over the data flows received by the participants. Score-SSM design ensures that the neighbors to which a participant communicates are always the same, regardless the VE zone-cut frequency. Of course, a high signaling rate might reduce the capacity of a participant (i.e. the maximum number of neighbors to which it communicates simultaneously).

The participants speed directly influence the amount of received data: moving faster allows a participant to communicate with more neighbors, but for shorter time. The experiments have shown that the amount of useful data received by the participants is significantly lower for highly dynamic participants (speed = 50) than for less dynamic participants (speed = 5).

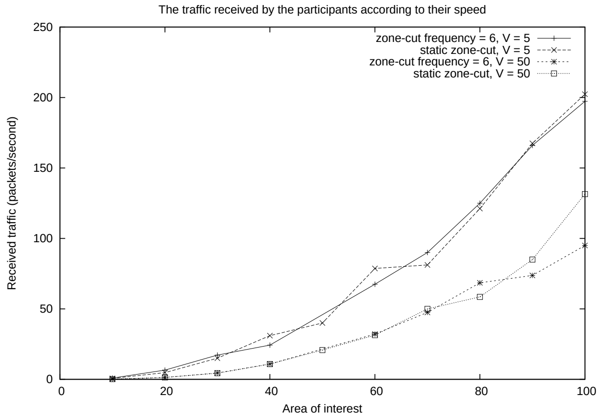


Figure 5.10: The traffic received according to the speed of the participants in Score-SSM

Figure 5.10 shows that the mean data traffic received by the participants vary a little between the static and the dynamic zone-cut cases, regardless the moving speed of the participants. The received traffic increases proportionally with the participant's area of interest: we observe that for an area of interest with side of 70, so about 0.49 of the area of interest with side of 100, the received traffic is half of the traffic received for an area of interest with side 100. If we compare the amount of data received by the participants for the two speed considered, we observe a decrease in the data traffic received when the speed of the participants increases. The participants moving with the speed of 5 units/second receive approximately twice more traffic than the participants moving with the speed 50 units/second, for all the area of interest considered in our experiments.

In order to explain the decrease of the data traffic received, we look at the number of "Subscribe to participant" (figure 5.11) operations effectuated: we observe a very important difference, between [0.08, 1.38] operations/second for a speed of 5 units/second respectively between [0.07, 5.31] operations/second for a speed of 50 units/second. The difference between the two curves increases with the of area of interest size. Figure 5.11 shows, for example, that a participant moving with a speed of 50 units/second meets about 5 times more neighbors than a participant moving with a speed of 5 units/second, for the area of interest with side of 80 (i.e. IArea = 0.64 * CellArea).

Figure 5.12 shows the mean number of data packets received by a participant. On both

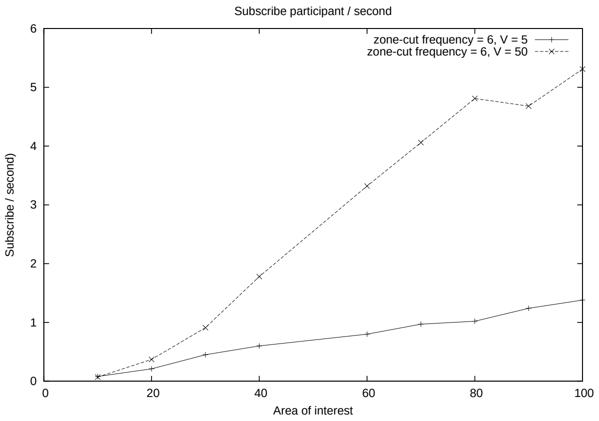


Figure 5.11: The number of subscribe operations to the data flows of the participants, in Score-SSM

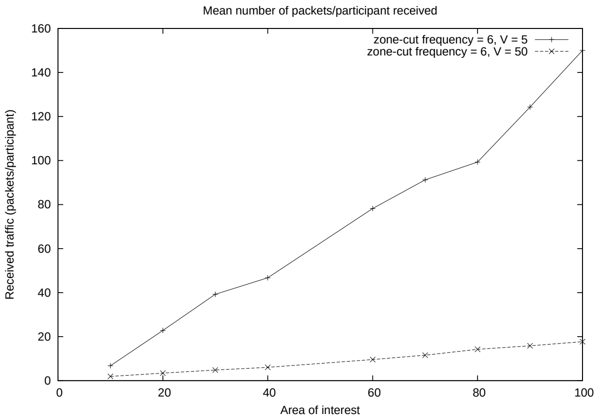


Figure 5.12: The mean number of packets received from the participants, in Score-SSM

curves we observe that the number of the data packets received increases with the area of interest size: for the participants moving with a speed of 50 units/second this increase is more important compared to the participants moving with a speed of 50 units/second. This is explained by the fact that a large area of interest side means that a participant detects its neighbors from farther away, thus they communicate for a longer time, compared to a small area of interest side. The figure 5.12 shows that the differences between the curves corresponding to speeds of 5 units/second, respectively 50 units/second is very significant; for the area of interest side of 100, a participant moving with speed of 5 units/second, receives, in the average, 7 times more data packets, compared to a participant moving with a speed of 50 units/second. Because we have considered participants which send data at the same rate, we can conclude that the communication time between a participant and its neighbors is 7 times longer, for a speed of 5 units/second.

Score vs. Score-SSM

To conclude, this experiment clearly proves that the participants using Score-SSM have maximal satisfaction regardless their speed or the partitioning frequency of the VE, while Score architecture is affected by the partitioning frequency and by the dynamics of the participants.

5.2.4 The amount of data traffic received

The purpose of this experiment is to study the effect of VE partitioning mode over the mean data traffic rate received by the participants. In other words, this experiment shows the efficiency of the filtering mechanisms proposed by the two communication architectures (based on the use of multiple multicast groups and on the dynamic partitioning of the VE).

Two different cases were studied: static partitioning mode of the VE and the dynamic partitioning mode with 1 second zone-cut frequency.

The traffic is measured as the number of sources/second received by a participant. A source is a participant sending "fast-rate" position data. The participants speed have been taken equal to 5 units/second. The participants are randomly distributed in the VE and they are choosing their destinations as described at the beginning of this chapter.

Score case

This experiment have been realized on a local network where the bandwidth is superior to the total traffic exchanged between the participants. Thus, the traffic received by a participant can be considered as the total input traffic for that participant, the packet losses being negligible.

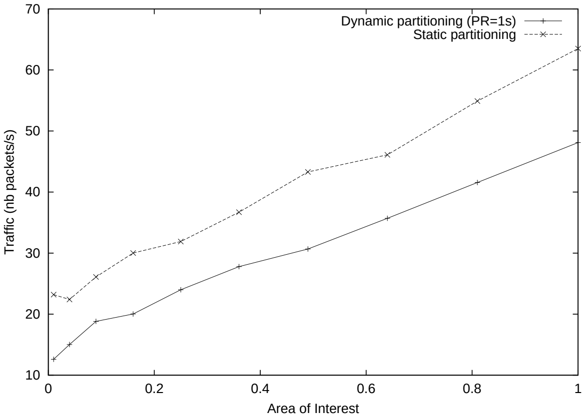


Figure 5.13: The traffic received by the participants in Score

Figure 5.13 shows that, in Score, the difference between the two curves remains approximately constant: the data traffic received by the participants using 1 second zone-cut frequency is lower than the traffic received when the VE is statically partitioned. For example when IArea = 0.8*CellArea a participant receives, in the average, 41 sources/second when the static partitioning is used and it receives, in the average, 55 sources/second when the dynamic partitioning is used.

We remind here that, in Score, a participant subscribes to all the multicast groups corresponding to the zones intersected by its area of interest, so it receives the data traffic sent by all participants located in these zones. So, on this figure, the curve corresponding to 1 second zone-cut, is already an upper bound for the useful traffic received by a participant. So, the Score mechanisms allow the participants to better approximate their area of interest, due to the creation of smaller cells in the areas of the virtual world where the participant density is important. In these areas, even a smaller reduction of the superfluous area leads to an important decrease of the undesired traffic received by the participants.

Score-SSM case

We remind how a participant receives traffic in the Score-SSM architecture. Imagine the participant describing a random trajectory in the virtual world, surrounded by its area of interest. As soon as another participant trajectory intersects its area of interest, it subscribes to the "fast-rate" position flow [3] of that participant (if its neighbors list is not already full), for the interval of time in which the new participant remains in its area of interest. So, the subscription period to another participant data flow is directly influenced by the size of the area of interest (larger area of interest means longer connection time, so more data packets received), by the moving speeds of the two participants in the VE (faster they move, less time to exchange packets they have), and by the angle formed by the participants trajectories. A static or a dynamic zone-cut model of the VE is less important here, as long as the participant knows which are its neighbors (and both VE partitioning models provide this), the participant communicates with the neighbors located within its area of interest, according to its capacity. The role of the zone-cut model is to minimize the signaling received by the participants, in order to use available bandwidth for the communication between the participant and its neighbors. A zone-cut model is more efficient than another zone-cut model if the proportion between the useful data and the signaling is bigger in the former model compared to the later model.

Figure 5.14 shows that the average number of sources received by the participants is almost the same for both VE partitioning models considered. This number increases with the size of the area of interest of the participants. For large area of interest (IArea >

---

[3] In our experiments, when we talk about data flows, we refer to the "fast-rate" position flows.

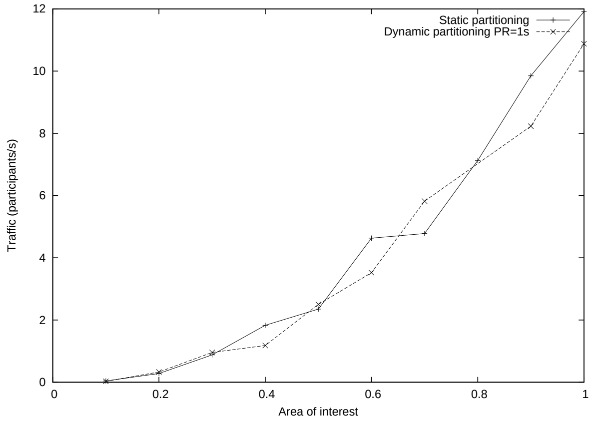


Figure 5.14: The data traffic received by the participants in Score-SSM

0.8*CellArea) we observe that a frequent zone-cut reduces the traffic, compared to the static case. We explain this by the fact that a less accurate zone-cut combined with the moving speeds of the participants in the VE leads to more encounters between the participants, at the price of higher signaling traffic (as shown in the next section of this chapter).

Score vs. Score-SSM

The mean number of sources received by the participants using Score is higher than the mean number of sources received by the participants using Score-SSM. If we compare the curves for the dynamic partitioning case in figures 5.13 and 5.14, we observe that in Score, the participants receive about 10 times more traffic for small areas of interest (IArea = 0.2*CellArea) and about 4 times more traffic for large areas of interest (IArea = CellArea) than in Score-SSM. This is due to the usage of a common multicast group for all the data sent by the participants located in the same zone of the VE, regardless if they are located in the participant's area of interest or not, according to Score architecture. So, most of the time, a participant receives data from more sources than it desires. This extra data can be seen as additional overhead of Score architecture, besides the signaling. The number of data flows received by the participants is given by the Score-SSM communication architecture. Here, for every participant, as soon as another participant enters in its area of interest, it is detected via agent's announcements and its data flow starts to be received. When a participant detects that a neighbor has left its area of interest, the corresponding entering data flow is stopped. So, there is no superfluous data traffic in Score-SSM, but some extra packets may be received due to the leave (from the multicast group) latency [49].

So, Score-SSM clearly outperforms Score with respect to the amount of data received by the participants: in Score a part of the bandwidth is wasted by the traffic send by superfluous sources, while in Score-SSM the participant-level filtering reduces this traffic to zero.

5.2.5 Measure and estimation of the signalling overhead

Two experiments are described in this section. In the first experiment, we compare the signalling exchanged between the agents and the participants (in order to establish the partitioning of the VE) in the two communication architectures. In the second experiment, we measure the signalling generated by the participants in order to join the multicast data groups of their neighbors.

5.2.5.1 Received signaling traffic and emitted control traffic

In this experiment, we evaluate the overhead of the signalling traffic generated by the two communication architectures. This signalling traffic is measured in number of packets per second emitted by the participants and by the agents.

Score case

The signaling in Score is composed by three types of flows: signaling flows (sent by the agents to the participants using multicast), control flows and keep-alive flows (both are sent, using UDP unicast, by the participants toward their agents).

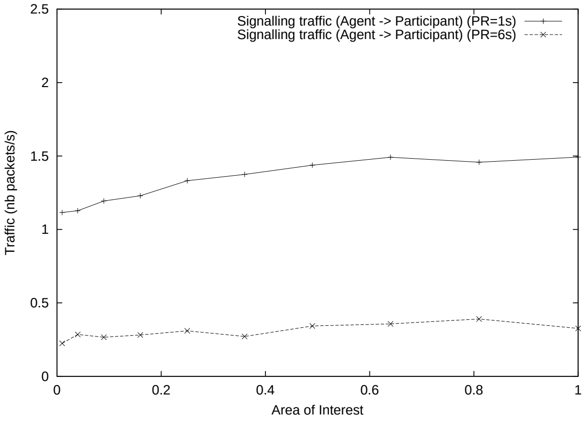


Figure 5.15: The signalling traffic in Score

Looking at the Score graphs (figure 5.15), we remark the low signaling overhead between the agents and the participants. For the signaling traffic emitted by the agents, we observe that the participants receive at maximum 1.5 packets per second, for an area of interest equal with 100 (equivalent with the cell size in the static zone-cut), or about 48 bytes/second [4].

Figure 5.16 shows that the control traffic and the keep-alive traffic (sent by the participants to the agents), have very low rates, situated in the interval [0.05, 0.1] packets per second.

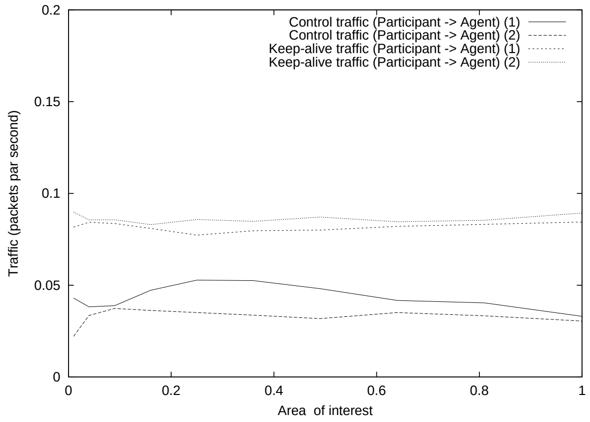


Figure 5.16: The control traffic, and the keep-alive traffic in Score

Score-SSM case

The signaling in Score-SSM is composed by two different types of flows: the slow-rate position flows sent by the participants to the agents (using unicast) and the positions flows of the participants sent by the agents towards their corresponding participants (using multicast).

The first type of flow is the slow-rate participant position flow, which is emitted by every participant toward their corresponding agent, on the slow-rate position flow. Each participant in the VE transmits one UDP packet per second to a well-known port of the agent, containing the position of the participant in the VE, its ID, its channel address, the participant capacity and its area of interest. The role of these packets is twofold: they inform to agent of the current position of the participant in the VE, and they have also the role of keep-alive packets. If the agent doesn't receive any packet from a participant in the last 5 seconds, that participant is considered as being "disconnected" and its entry in the agent's participants list is deleted. The size of a position a packet is: 8 (position) + 4 (ID) + 2 (capacity + interest) = 14 bytes. We add to this the 20 bytes of IP header and another 4 bytes of UDP packet. So, we get 40 bytes for a participant position packet. Every participant sends the slow rate position packets at a fixed rate, for the participant lifetime. We have chosen to send one packet per second in our experiments. This frequency

---

[4] The size of the signalling packets is 8 bytes of data plus the UDP/IP header which is 24 bytes long. Thus, a signalling packet has 32 bytes in Score.

(together with the speeds of the participants) gives the precision at which the agent knows the positions of the participants located in its corresponding zones. We observe that the more frequently the participants send position update packets, the better the agents know their positions, although the signaling traffic exchanged became more important. So, the control traffic sent by a participant in Score-SSM remains constant, regardless the participant speed, capacity or area of interest size; in our case it is equal with 40 bytes/second.

The second flow type is the participants positions flow which is emitted by each agent toward the participants located in the VE zones they manage. One different signaling flow is transmitted for each managed zone. In our experiment, we have used only one agent, which sends one signaling flow for each zone composing the VE. Every participant subscribes to the flow corresponding to its zone, and, in addition, to the flows corresponding to the zones intersected by its area of interest. The signaling packets contain the approximate positions (i.e. the position of every participant is updated once per second) of the participants located in a zone of VE. We have aggregated the positions of more participants in one single data packet, in order to lower the headers overhead. The position information in a packet is about the same size with the IP header, so an aggregated position packet reduces the signaling overhead to one half, due to the exclusion of many packet headers. The size of an aggregated packet is always lower than 1500 bytes, in order to avoid the packets segmentation, thus additional signaling overhead.

In our experiment, the position of each participant located in a VE zone is advertised once every second. Since the participants advertise their position to the agent once per second, the agent knows their position with the precision of one second multiplied by the participant speed. An agent announcement rate bigger than one packet/second increases the signaling traffic because it advertise some redundant (old) positions of the participants. An agent announcement rate lower than one packet/second reduces the accuracy of the VE (the participants detect later their neighbors) while diminishing the signaling traffic. As the position packets are sent at a constant rate, we observe that the agent signaling traffic rate is directly proportional to the number of the participants located in that zone. From the participant point of view, the more participants are located in its zone of the VE, the bigger becomes the signaling traffic received.

The total signaling traffic emitted by an agent on a zone's corresponding channel is: 18 (8 location + 4 ID + 1 capacity + 1 interest + 4 IP address) * (number of the participants in zone) + (20 + 4)*number of aggregated packets (i.e. the headers size). In fact, if there are less than 80 participants in a zone, one aggregated packet is sufficient to advertise their positions.

Figure 5.17 shows that the amount of signaling traffic is important. In fact, when the side of the area of interest is lower than 25 units the signaling traffic it is more important

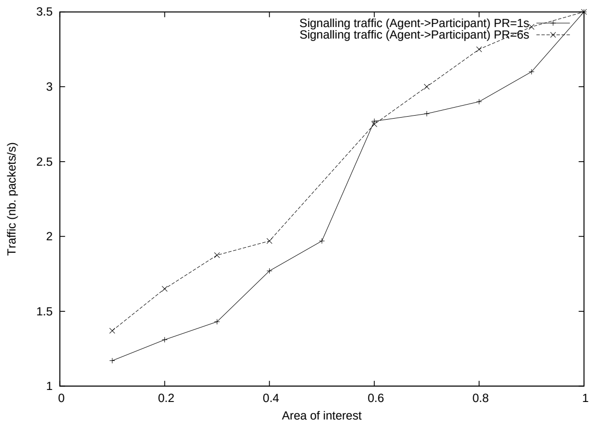


Figure 5.17: The signalling traffic in Score-SSM

than the data traffic (as it is shown in the figure 5.14)! For example, when the participant's area of interest has the side equal to 60 units, the signaling traffic is equivalent to the fast-rate participant position traffic sent by one VE's participant; in the same time, the participant receives, in average, the data traffic send by four different sources. This leads to a proportion of 1/4 between the signaling and the data traffic.

We observe that, for all area of interest sizes, the signaling traffic for 1 second partitioning frequency is lower than the signaling traffic for 6 seconds partitioning frequency; thus a better partitioning of the VE according to participant's density.

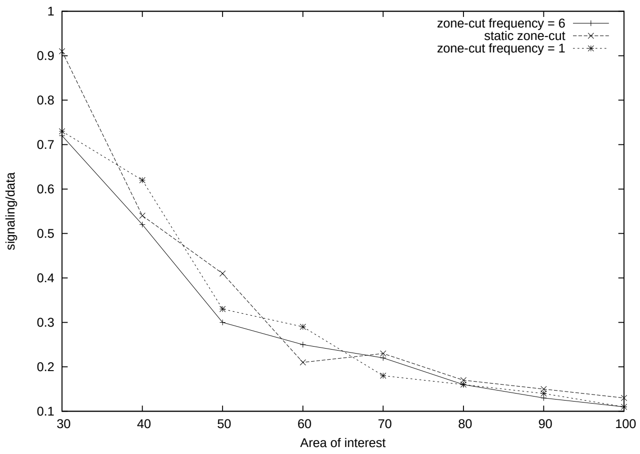


Figure 5.18: The proportion between signalling and data traffic in Score-SSM

Figure 5.18, which illustrates the proportion between the signaling and the data traffic received by the participants using Score-SSM, shows that for small area of interest sides, the signaling rate is more important than the data rate received form the neighbors, regardless the VE partitioning mode. This is explained by the fact that a small area of interest leads to few and short contacts (represented by the exchanged data flows) between a participant and its neighbors. In the same time, the amount of signaling received remains important because it is composed by (at least) all the participants located in the same VE zone with the participant considered.

The figure 5.18 also shows that for large areas of interest (IArea>0.81*CellArea) of the participants, the proportion between signaling and data goes down toward 10%, regardless the VE partitioning mode. We observe that, while the differences between the three zone-cut curves diminishes with the increase of the area of interest side, the static zone-cut case generates more signaling, for the same amount of data received.

Score vs. Score-SSM

Considering the figures 5.15 and 5.17, we compare the signalling traffic received by the participants for the two communication architectures. The size of the signalling packets are respectively 8 bytes for Score and 16 bytes in Score-SSM, while the UDP/IP header is 24 bytes long. Given these figures we chose to implement aggregation, i.e. to send the signalling packets representing all the packets from a zone using a unique UDP/IP packet. All our experiments have shown that when the agents partition the VE according to a maximum of 30 participants/zone, we get an average of 12 participants in a zone with our scenario. So, on average, the Score-SSM signalling packets size is 16*12+24 = 216 bytes, which corresponds approximatively to 7 Score signalling packets. For Score, we observe that when the zone-cut frequency is 1 second, the signalling is about 4 times larger than when the zone-cut frequency is 6 seconds. In addition, the signalling traffic increases very slowly with the size of the area of interest. In Score-SSM, the signalling traffic linearly increases with respect to the area of interest. If we compare the number of signalling packets received by the participants we observe that in Score-SSM this number is much more important than in the Score case. The aggregate of the signalling information received for each zone, every second, significantly reduces the signalling traffic rate in the Score-SSM case. For example, when IArea=CellArea, and the zone-cut frequency is 1 second (i.e., when the signalling traffic reaches it maximum in Score) we observe 1.5 packets/s for Score and 3.5 packets/s for Score-SSM. As the size of 3.5 Score-SSM packets is approximately equal to 7*3.5= 24.5 Score packets, this means that 16,3 times less bandwidth is used.

5.2.5.2 The subscribe frequency according to the zone-cut frequency

In this experiment, we trace the average number of "Subscribe" operations per second to a zone of the VE (i.e. to its corresponding multicast channel) effectuated by the participants. We have studied the participants moving with the speed of 10 units/second and with the speed of 100 units/second.

The static partitioning mode and the dynamic partitioning mode of the VE were considered in this experiment.

Score case

Figure 5.19 shows that, using the dynamic partitioning mode, the join frequency [5] is higher than using the static partitioning mode, both in the cases when the participants moves with a speed of 10 units/second or with a speed of 100 units/second. For example, in the case where IArea = 0.49*CellArea there are twice more joins in the dynamic case compared to the static case, for a speed of the participants equal to 10 units/second. For the same area of interest surface, but for the speed of the participants equal with 100 units/second, the number of joins is about 2 reports/second in the static partitioning case and it is about 3.5 reports/second in the dynamic partitioning case.

In the same time, comparing the number of join operations per second for the two participants speeds chosen, we observe that in the static partitioning mode, the participants moving with a speed of 100 units/second effectuate 10 times more join operations compared to the participants moving with a speed of 10 units/second (for IArea $>$ 0.49*CellArea). For the dynamic partitioning mode, this proportion arrives up to 7 times more join operations for a speed of 100 units/second compared to a speed of 10 units/second.

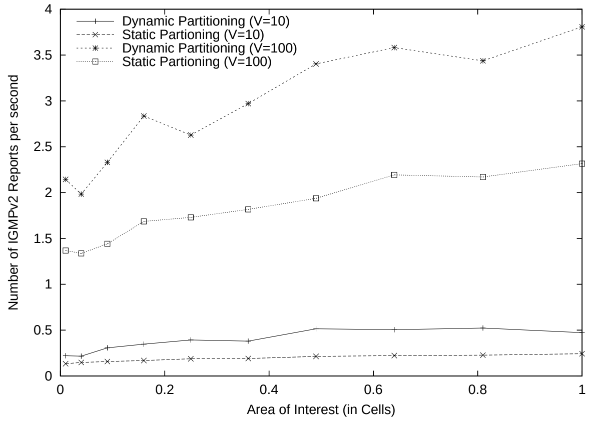


Figure 5.19: IGMPv2 join operations per second, in Score

Score-SSM case

In the figure 5.20, we compare the number of joins to the VE's zones for two different speeds of the participants: 10 respectively 100 units/second. For each of these two speeds

---

[5] Here, we employ the term 'join' to a multicast group for Score, which uses IGMPv2 with the ASM model; which is equivalent with the term 'subscribe' to a multicast channel for Score-SSM, which uses IGMPv3 with the SSM model.

of the participants there are two curves: for 6 seconds zone-cut (dynamic partitioning), and the static partitioning.

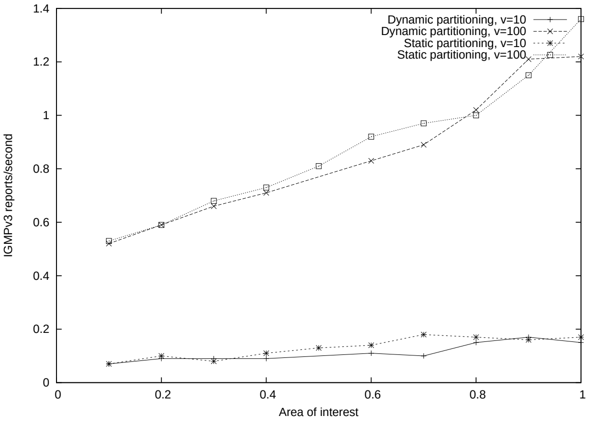


Figure 5.20: IGMPv3 operation per second, in Score-SSM

In the Score-SSM case, we observe that the number of joins is 6-7 times bigger for the participants moving with a speed of 100 units/second than for the participants moving speed of 10 units/second, for every area of interest surface. The join frequency values are close in the static and in the dynamic partitioning case, because the numbers of zones in which the VE is decomposed, in the two VE partitioning cases, is similar. In addition, the participants are moving towards some "hot-spot" locations of the VE with a given probability (as presented in the introduction of this chapter), where the density of the participants is particularly high. So, according to the dynamic zone-cut algorithm, the VE is divided into a large number of small zones, such that no zone contains more than "max-Participants/zone" participants. The participants are moving, with a given probability, toward these "hot-spots", so they are crossing an important number of zones in a small interval of time.

Figure 5.20 also shows that the join frequency doubles from the IArea = 0.01 * CellArea to IArea = 0.64 * CellArea (for both VE partitioning modes). This corresponds to an increase of 64 times in the surface of the area of interest of a participant. For the areas of interest with the side bigger than 80 units, the join frequency increases even faster.

Score vs. Score-SSM

In all the curves presented above, a participant using Score-SSM effectuates fewer join operations than a participant using Score.

Comparing Score and Score-SSM curves, we note that for a small area of interest (IArea < 0.16 * CellArea), there are three times less "join" messages in the static partitioning case and four times less "join" messages in the dynamic partitioning case. For larger area of interest, (IArea > 0.64 * CellArea), there are two times fewer "join" messages in the static partitioning case and three times fewer "join" messages in the dynamic partitioning case. The above differences are similar both for V=10 units/s and for V=100 units/s. The explanation of the above result is that Score sends more "join" messages to the agents than Score, because the VE is cut into a larger number of smaller zones.

5.2.6 The "number" of multicast groups used

In this section, we study the number of multicast groups used by the Score architecture, as well as the number of multicast channels required to deploy the Score-SSM signaling infrastructure. From the experiments, we have extracted two categories of results:

- the number of multicast groups used by the agents,
- the distribution of the participants in the VE's zones (i.e. in the multicast groups).

5.2.6.1 The number of multicast groups used by the agents

We remind that our experiments have been executed using only one agent for the whole VE. So, all the multicast groups used by the two architectures were managed by a single agent. On the graphs, we show the mean number of multicast groups used by the agent for a dynamic zone-cut of the VE. This number is computed for different area of interest of the participants. The number of multicast groups used corresponds to the number of zones composing the entire VE.

Score case

A maximum of 144 multicast groups have been allocated for the VE decomposition in zones. In the first phase of the zone-cut, each Score zone is divided into cells, according to the local density of the participants. In the second phase, some cell may be furthermore divided, if the satisfaction of the participants in these zones is reduced. In figure 5.21 we observe that, as the area of interest increases, all the multicast groups allocated for the application are utilized.

Score-SSM case

As expected, the experiments have shown that, the average number of multicast channels used, it does not depend on the sides of area of interest of the participants. The zone-cut depends only of the total number of participants simultaneously connected in the VE. Their local density doesn't influence the average number of zones composing the VE. For 250 participants connected, the number of zones composing the VE is 19 or 22 most of the time.

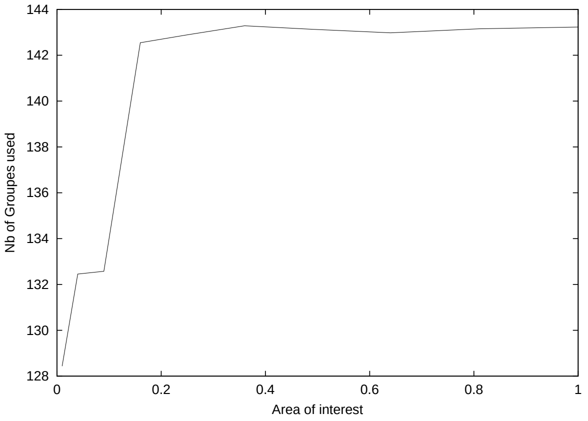


Figure 5.21: The number of multicast groups used by the agent in Score

We remind that, in our experiments, we have considered a maximum of 30 participants simultaneously located in the same zone of the VE.

In addition, we have studied the average number of multicast channels used for different values of participants per zone limit. For a maximum of 15 participants per zone, there are 43 or 46 channels used, most of the time. For a maximum of 50 participants per zone, there are 13 or 16 channels used, most of the time. With these values, we can compute the average "utilization" of a multicast channel, i.e. the average number of participants in a multicast channel divided by the maximum number of participants admitted in a multicast channel. The table 5.1 shows the "utilization" values for different limits of the maximum number of participants simultaneously admitted in a VE zone:

15	**30**	**50**
0.35	0.4	0.32

Table 5.1: The "utilization" values according to the maximum number of participants simultaneously admitted in a VE zone

We observe that the best utilization is obtained for 30 participants per zone, but we should mention that this value depends on the participants distribution in the VE and on the number of participants connected. For small limits of the number of participants admitted per zone (15 participants/zone), the VE zone-cut contains a lot of empty or under-utilized zones, which reduces the mean "utilization" value of the zones composing the VE.

High limits (50 participants/zone) means that the VE is composed by a small number of zones. If the distribution of the participants in the VE is not uniform, this may lead to zones with very few participants. Because the number of zones is low, the weight of a under-utilized zones in computing the average utilization becomes more important in this case, than for small limits. Thus, the participants density in the VE directly influence the utilization of the VE zones.

Score vs. Score-SSM

The above utilization values of the VE zones shows that Score uses a significantly higher number of multicast groups, compared to Score-SSM. This is due of the filtering, which is effectuated through the grid in Score, while in Score-SSM is effectuated by a combination of grid-based and entity-based mechanisms. In Score, the more multicast groups are available, the better the filtering is. On the contrary, Score-SSM accepts a large number of participants in one zone of the VE, hence few multicast groups are required to cover all the VE. Score-SSM has the purpose to find the best filtering in order to reduce the signaling. In the same time, we should remind that Score uses those multicast groups to send data between the participants, while Score-SSM uses additional multicast channels to distribute data. The above multicast channels are used only for signaling. But, while the ASM model strictly limits the number of multicast groups used by an application, the SSM model imposes no restrictions with respect to the number of multicast channels available.

5.2.6.2 The distribution of the participants in the signaling multicast groups

In this experiment we have studied the distribution of the participants in the multicast groups (i.e. in the VE's zones). We have compared the dynamic and the static partitioning cases.

In Score, eight different figures have been drawn, for eight different areas of interest of the participants. In Score-SSM, we have considered only one area of interest size, as the zone-cut doesn't dependent on surface of the area of interest of the participants, as shown in the previous experiments.

In the figures 5.22 and 5.23, for a given number of participants 'N', on the 'X' axis, we can observe for each curve, on the 'Y' axis, the percentage of multicast groups containing between 'N' and 'N-2' participants.

Score case

From the 8 area of interest studied we have chosen to present in this section only one case, when IArea = 0.25 CellArea. All the curves corresponding to the dynamic zone-cut of the VE have a peak value around 7 in all the graphs. This value is explained by the number of participants connected (1000) and the number of available multicast groups (144). The rapport of these values gives 6.94 which is the average number of participants per multicast

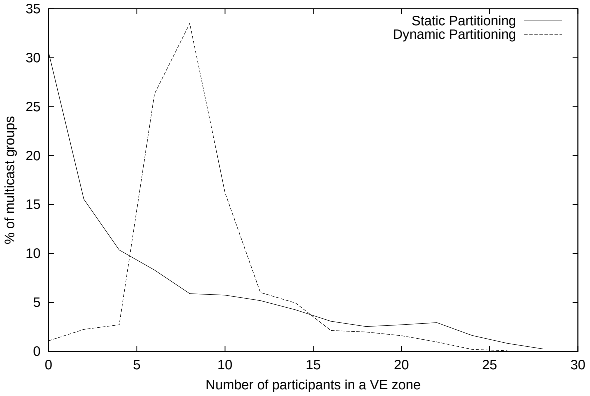


Figure 5.22: The participants distribution in the multicast groups for IArea = 0.25 CellArea, in Score

group. This shows the capability of Score to deal with non-uniform distributions and with dynamic participants.

The figure 5.22 shows that, in the static case, almost 30% of the multicast groups contain 0, 1 or 2 participants. This signify that a big number of multicast groups are under-exploited while these groups are a very valuable resource for the data filtering. In addition, the number of groups containing a big number of participants is more important in the static case than in the dynamic case.

Score-SSM case

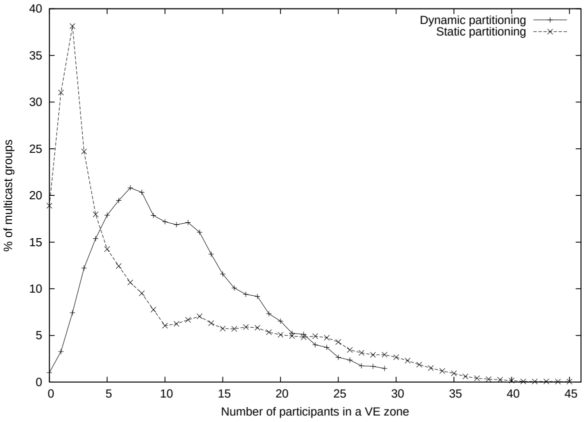


Figure 5.23: The participants distribution in the multicast groups, for IArea = 0.25 CellArea, in Score-SSM

On the figure 5.23, we observe that in the static case almost 40% of the multicast channels have 0, 1 or 2 participants. In the same time, the number of zones containing a big number of participants is more important in the static case than in the dynamic case. Some of the zones reach even 45 participants.

In the dynamic case, the participants distribution in the VE's zones approaches the normal distribution. Only 10% of the groups contain 3 or less participants and only 10% of the zones contain over 20 participants, a peak value is registered around 7 participants/zone. This number remains almost constant between 8 and 13 participants.

Score vs. Score-SSM

Both architectures show the better utilization of the multicast groups when using a dynamic zone-cut, in the opposition with the static zone-cut which leads to a large number of empty zones.

5.2.7 Conclusion of the experimental comparison of the two communication architectures

The bottom line of these experiments can be stated as follows. Score generates less signalling traffic than Score-SSM, but in exchange, the participants receive superfluous data which happens to be quite substantial in some cases.

In Score-SSM, participants do not receive any superfluous data traffic and they can decide with finer granularity which data flows they receive. Although Score-SSMgenerates more signalling traffic than Score, the amount of extra signalling traffic does not depend on the data flows rate exchanged between the participants. The same amount signalling is required to put in contact the participants which exchange position data between them (as in our experiments) or which exchange high rate data flows (such as audio or video flows). Thus, Score-SSM is an appropriate communication architecture for LSVE which contains multimedia flows. [?]

5.3 Enhancing Score-SSM performances

A series of experiments have been conducted, in order to improve the performances of the Score-SSM architecture. By "improving the performances" we mean: give the maximum possible satisfaction to the participants, reduce the amount of network resources used to deliver a given quantity of data to the participants, allow and support a big number of different data flows and heterogeneous participants in the VE.

In order to enhance the Score-SSM performance, we have run the following scenarios:

- reduce the agent to participant signaling
- study the low participant speed case
- propose an hierarchical zone-cut, ex. multicast sub-zones
- find the optimum limit of maximum participants in a zone (according to which the agents made the partitioning of the VE), in order to minimize the signaling costs

5.3.1 Reducing the agent to participant signaling

In this test we try to reduce the signaling received by the participants. We are aware that a less accurate knowledge of the VE participants positions leads to a later detection of the neighbors and finally to less data packets received. For participants with low capacities (C<10) an important reduction in signaling permits, in the end, to receive more neighbors; less signaling avoids, maybe, to drop some neighbors due to the participant's low bandwidth. In order to improve the proportion between the data and the signaling, we can increase the amount of data received for a given amount of signaling or we can decrease the signaling while trying to receive the same amount of data. In order to increase the amount of data received, a solution is to maintain the connection with the neighbors for a longer time, i.e. even after the neighbor quits a participant's area of interest. Of course, we may do this as long as the participant's neighbors list is not full.

In order to decrease the signaling received by a participant we see three solutions: to reduce the signaling packets frequency, to reduce the size of the signaling packets or to provide a more "fine-grained" partitioning in order to reduce the superfluous participants positions announcements received by the participants.

In this experiment, we have studied the effect of reducing the signaling packets frequency sent by the agent to the participants located in a zone of the VE. In all the experiments presented in the previous section, this frequency was taken equal to one announcement/second. This value was chosen because the participants send to the agent their position in the VE once per second. So, when a participant receives an agent signaling packet, it can be sure that the signaling data contained has been updated in the last second, so it knows the approximate position of the participants with an error of maximum one second multiplied by the participants speed. The tests effectuated have shown a high signaling traffic, so we experimented with a lower signaling packets frequency: the approximate position of each participant in a zone is advertised once every five seconds. In addition, as soon as a new participants joins the VE, this frequency will be increased, for a short time, at one second, in order to speed up the connection of the new participants to the VE.

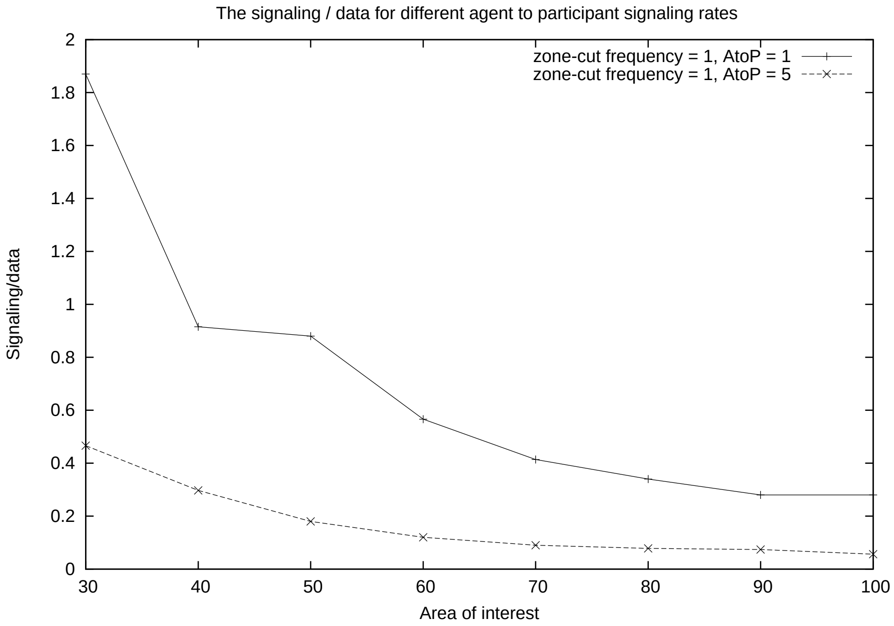


Figure 5.24: The signalling over data proportion

The curves in the figure 5.24 have been computed by dividing the number of participants advertised by the agent (remark that this is not the number of packets) to the number of data packets received. The two curves show the signalling/data proportion for two signalling (from the agent towards the participants, noted "AtoP", over the "participants position flow") packets frequency of once per second (AtoP = 1) and once each 5 seconds (AtoP = 5). This signalling/data ratio represents, at a logical level, the number of approximate position advertised by the agent in order to get that amount of data traffic. In practice, the participants advertised by the agent are encapsulated in one aggregated packet, reducing in this way the signaling bandwidth to approximately a half (by reducing the number of UDP/IP packet headers). Thus, in order to get the real amount of signalling, we should divide by two the values of the above curves. The figure 5.24 shows that the signaling proportion compared to the data received is greatly reduced: for participants having the area of interest side bigger than 70 units the signaling is reduced about 5 times. For example, considering an area of interest with the side of 100 units, the signaling represents about 5.6% of data; taking into account the packet aggregation we arrive at 3% of the data traffic received by a participant.

Also from the above curves we observe that for both signaling rates, the signaling/data ratio is decreasing: faster for small area of interest sides (IArea side $<$ 50 units), slower, almost getting constant for larger area of interest sides (IArea side $>$ 80 units). In the absolute values, the signaling traffic is increasing with the area of interest, because the participant intersects more zones of the VE. In the same time, there is an important increase of the data traffic for large area of interest sides, as we can see of the figure 5.25.

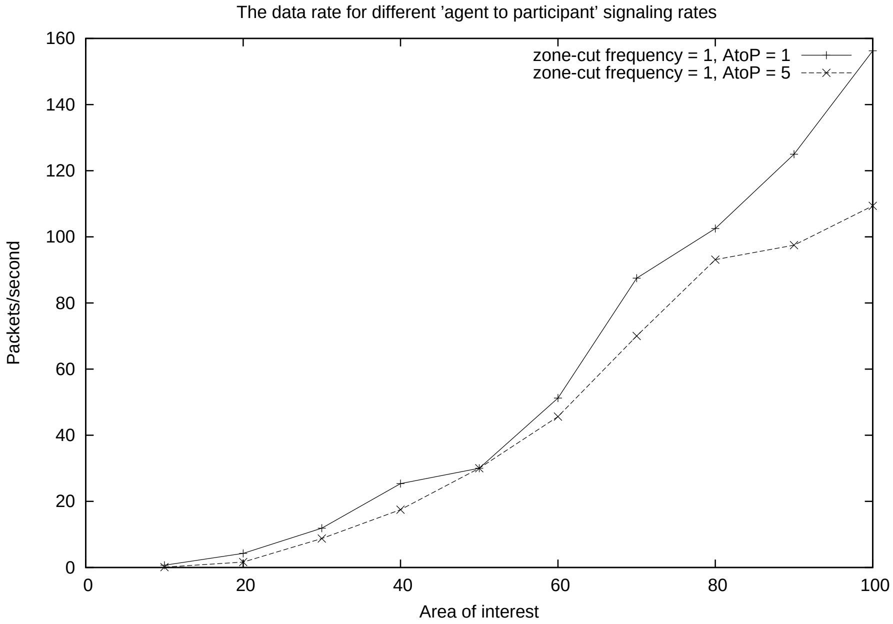


Figure 5.25: The influence of the signalling rate over the received data rate

In the figure 5.25, we compare the mean traffic received by the participants when the agent-to-participant packets are sent once each 5 seconds with the case when these packets are sent once per second. We observe that the price to pay for the low signaling is a reduced data traffic received by the participants. Low frequency agent messages mean a later detection of the neighbors, so a later connection thus a reduced communication time. This explains the differences in the received data packets traffic. For larger area of interest (IArea $>$ CellArea) the difference of the received traffic in two cases is bigger because the participant crosses more neighbors in the VE than for smaller IArea, but each of these neighbors are detected with a certain delay compared with once/second frequency, which explains the reduced number of data packets received.

5.3.2 Reducing the zone-mapping announcement traffic

The zone-mapping announcement is the signaling traffic send by the agent toward all the participants it manages, over the "multicast-channel-zone-mapping" flow. As it was explained in the previous chapter, the role of this flow is to allow every participant to detect its closest neighbors, in order to communicate with them. In Score-SSM, the agent uses one multicast channel to transmit the complete mapping of the VE zones it manages. As each VE zone has a limited number of participants, and this value is very small compared to the total number of participants that an agent can manage, we observe that an agent is able to manage a large number of VE zones. This implies a big zone-mapping announcement traffic. For example, in Sony's "EverQuest" an agent is capable of managing up to 1500 users. [6] If we consider some thousands of users managed by one Score-SSM agent this leads to hundreds of VE zones managed by that agent. One mapping information consists of at least 18 bytes, two zone corners coordinates of 4 bytes each coordinate and the multicast channel group offset of two bytes.

We send all the mapping entries aggregated in one packet in order to economize the IP packet headers size (of course, if there are a lot of entries, the packets are fragmented). For 100 entries the signaling is: 100*18 + IP headers, which is about 1.8 KB. If this mapping is sent once per second, this means that every participants receive about 2KB of signaling only for mapping. Hence, the reduction of the zone-mapping announcement traffic is an important concern from the scalability and the heterogeneity support.

Below, we present a proposition for reducing the zone-mapping announcement signaling traffic. Of course, we can envision solutions which send the mapping over different multicast channels, but this will introduce more multicast subscription overload and it will require the use of an increasing number of multicast channels.

Instead of sending, to every participant, all the mapping of the VE's area managed by one agent, we propose to send to each participant only the mapping information of a reduced surface of the VE, located around the participant's position in the VE. To be more specific, we propose to send, for each zone of the VE, only the mapping information corresponding to its neighboring zones, i.e. the zones of the VE which have common points with the current zone. So, each time the agent computes a new mapping, it will also compute the mapping information for each zone. Instead of using new multicast channels to send this specific zone-mapping information, we propose the use of the existent "participants position flow" channels. These channels are the signaling channels specific to each VE's. For each zone, this channel is used by the agent to send the positions of participants located in that zone. Now, a new type of packet will be sent on this channel:

---

[6] $http://www.mudconnect.com/mud_graphical.pc.html$

the mapping of the neighboring zones. In practice, in order to keep the data packets small, we differentiate these two types of packets using their first field: if the value of this field is positive, this means that the current packet is an aggregation of participants positions. If it is negative, this means that the current packet is an aggregation of zone-mapping information. The module value of this field contains the number of items (positions or zone-mappings) aggregated in one signaling packet.

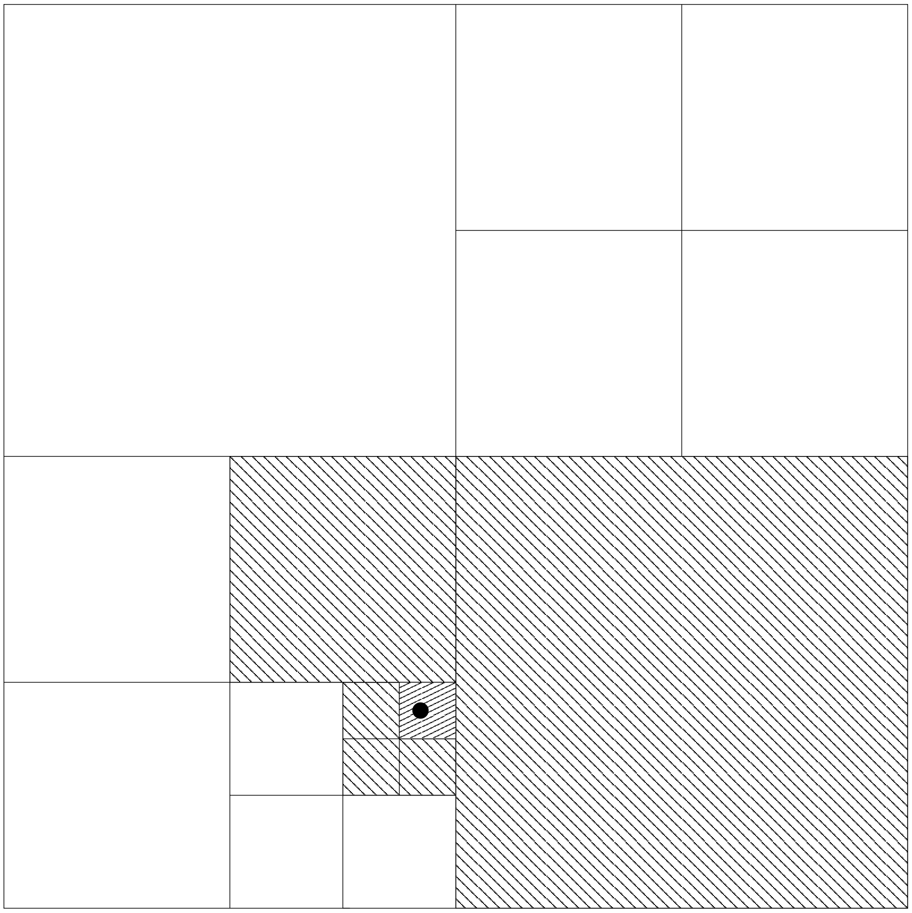


Figure 5.26: The mapping information of the neighboring zones

On the figure 5.26, we can see the mapping information received by one particular participant: it receives the mapping of all the VE's zones having at least one point in common with its current zone.

From the participants point of view, it is clear that this new mapping information reduces the participant's signaling overload because it is restricted only to a neighborhood around the participants. In addition, no subscription to additional multicast channels is required.

The above approach doesn't require the existence of the "multicast-channel-zone-mapping" flow. However, this approach rises some problems when the participants connects in the

VE and for the handovers (i.e. when the participants change their agent). To enter in the VE, the participants need to get the zones-mapping from the agent. The agent can send this mapping in two ways: unicast, by responding to each individual request of by multicast, by periodically sending the complete zone-mapping. Comparing the unicast and the multicast solution, we observe a tradeoff between the fast connection time and low network overload but higher computing overhead at the agent, using unicast and less agents overhead but longer connection delays and higher mapping signaling overhead when the multicast is used. The chosen solution is different according to the participants number, their dynamics and their arrival time.

The handover takes places every time when a participant's area of interest intersect a VE zone which is managed by a different agent. In this situation, the participant need the mapping of that zone: it obtains the agent's address from the "agents addresses - VE area" mapping flow. Having the agent address, it needs to obtain the mapping effectuated by this agent. The existence of a general mapping flow for all the VE area managed by the agent would have helped, but we have chosen to eliminate this channel in the favor of a local mapping. To solve the mapping problem in the handover case, we decided that every agent sends periodically the mapping information for all the zones located at the border of the VE area it manages. In fact, it is a limited use of the "multicast-channel-zone-mapping" flow. The proposed solution makes sense when the agents manage VE areas which are decomposed in a large number of zones (which it is the case when a large number of users are connected in the VE).

5.3.3 Different speeds of the participants

In this experiment, we have compared the data traffic received by the participants moving at different speeds: 1 unit/second, 5 units/second and 50 units/second. We have considered 10 different areas of interest sizes, with the sides between 10 and 100 units. The purpose of this experiment is to study the impact of the speeds of the participants on the data traffic received. In figure 5.27, we have traced the average communication time between two participants. A curve was traced for each participant speed. In fact, for speed=1 unit/second we have drawn two curves, because the participants travel a shorter trajectory in the VE compared to the participants moving with a speed of 5 unit/second or 50 unit/second, thus the experimental results are influenced by the local participants density in the VE areas traversed by the participants. As we have traced a low number of participants (four participants were launched in "reception" mode) and for a limited amount of time (15 minutes), the trace files have shown significant differences in the values of the parameters traced. So, we have taken the decision to trace a curve for the shortest connection times observed in the trace files and one for the longest connection times observed in the trace files, for the participants moving at a speed of 1 unit/second. Also, this low number of traces explains the big fluctuations observed on the curves belonging to figures drawn for this experiment.

In order to understand better the results, we remark that the relative speed between two participants may reach the double of one participant speed. This relative speed depends directly on the angle made by the trajectories and the direction of the movements of the participants. For example, if the participants are moving at 50 units/second, their relative speed may reach 100 units/second when the participants are moving over the same trajectory, but in opposite directions.

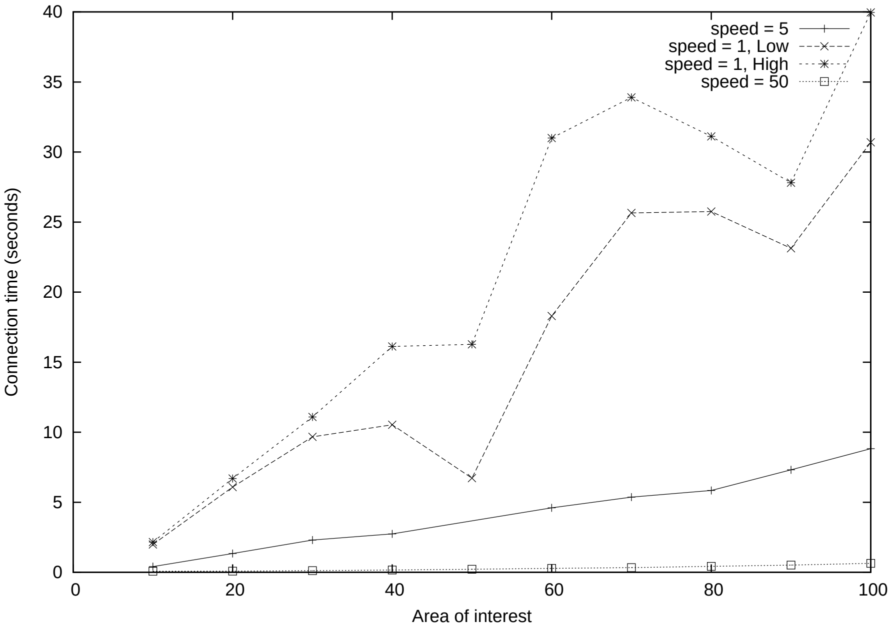


Figure 5.27: Mean connection time to a neighbor

On all the curves, we observe a linear growth of the mean connection time between two communicating neighbors. This growth is proportional to the increase in the area of interest of the participants. If the mean of the inter-participants relative speeds is the same, a large area of interest leads to a longer communication time between two participants, because the communicating neighbors are detected earlier and they exit from the area of interest of the participant later. While comparing these curves, we observe that for each size of the area of interest, the mean values are inversely proportional to the participants moving speeds. So, the participants moving with the speed of 5 units/second communicate in general 5 times less time the participants moving with a speed of 1 and they communicate 10 times longer than the participants moving with a speed of 50.

So, this figure shows us that the faster a participants moves, the lower is the interval of time it communicates with its neighbors, thus it exchanges less data. On the other hand, the faster a participants moves through the VE, the more participants it encounters, so it exchanges data with more participants. We can observe here an interesting trade-off: communicate for a longer time with less participants or communicate for a shorter time with more participants. In which of these cases the participants receive mode data?

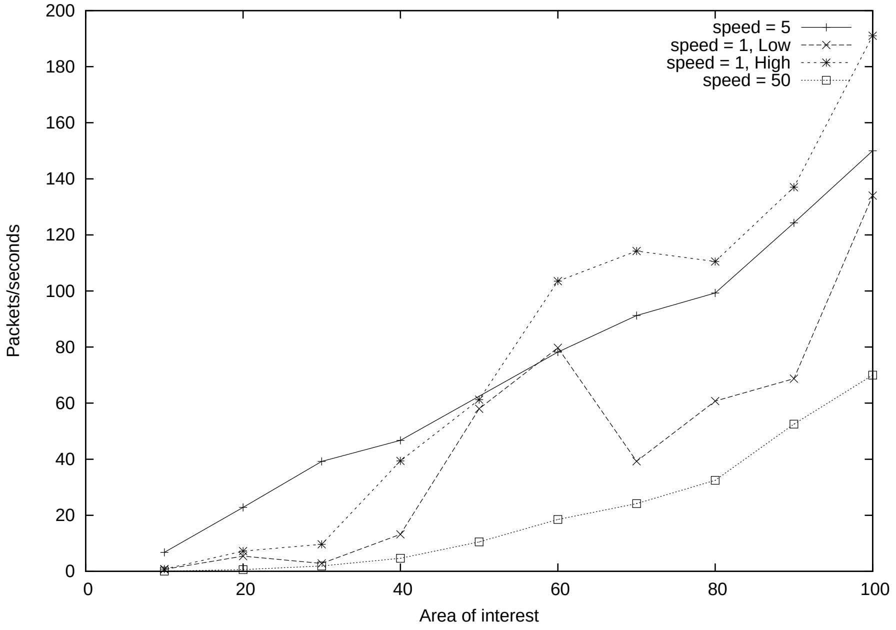


Figure 5.28: Data traffic rate received by a participant

The figure 5.28 gives us the experimental answer for this trade-off. We observe that for the areas of interest with the side less than 50, the participants moving with the speed of 5 units/second receive the biggest amount of data. We explain this by the fact that the areas of interest of the slow moving participants intersect the trajectory of very few other participants (we made the observation that an area of interest with the side of 20 units represents only 4% of an area of interest with the side of 100 units, so there are 25 times less chances to cross other participants). On the contrary, the fast moving participants receive almost no packets from the neighbors they intersect, because the areas of interest are too small compared with their speed. On the other hand, for large areas of interests, the participants moving with one unit/second may receive the highest amount of data. Looking at the two curves representing the participants moving with a speed of one, we can appreciate that in the average, a participant moving with a speed of 5 units/second receives the same amount of data as a participant advancing with 1 unit/second. Also, we observe that a participant moving with a speed of 50 units/second, receive less data compared to the participants moving at a lower speed, regardless its "area of interest side". In addition, for small areas of interests, the amount of data received is very small, but it exponentially increases for large areas of interest. For the area of interest side equal to 100 units, the data traffic received reaches the half of the data traffic received by the participants moving with a speed of 1 unit/second. We explain this by the fact that their areas of interest become significantly larger compared to their moving speed.

5.3.4 Maximum number of participants admitted in a VE zone

In this experiment, we study the effect of establishing a maximum number of participants admitted in a VE zone on the signaling and the data traffic received by the participants.

The virtual world is decomposed into non-overlapping zones such that any point of the virtual world belongs to at least one VE zone, (or more for the points located on the zones boundaries). The agents are responsible for this decomposition: the only criterion used is to fix an upper limit for the number of the participants simultaneously located in the same zone of the VE. We remark a trade-off in establishing this upper limit: a low value requires the use of a big number of multicast channels (this implies additional costs for managing and advertising these multicast channels), while a large value means that a participant will receive the signaling corresponding to a large number of participants located in its zone. Also, a large value means that the agents have to use more network and computing resources to manage a zone of the VE. According to the Score-SSM model, the participant/zone limit is the key of the first level of filtering. In the same time, the amount of data received by the participants is independent of the first level of filtering. So, in the following paragraphs we study the proportion of the signaling over the data received according to the variation of the participant/zone limit. We have chosen to study four cases: the maximum number of the participants in a VE zone has been considered equal to 15, 30, 50, respectively 100. All the participants are moving with a speed of 5 units/second and their capacities are equal with 50 participants/second.

The smaller is the participants/zone limit, the more zones are composing the VE. The more zones are composing the VE, the more subscribe operations (to the VE zones) the participants have to perform. These operations are part of the Score-SSM signaling. Figure 5.29 confirms the assumptions that the number of subscribes/second is inversely proportional to the participants/zone limit. In addition, we observe that for small areas of interest of the participants (IArea $<$ 0.49*CellArea), these curves are almost constant. For large areas of interest (IArea $>$ 0.64*CellArea) these curves grow linearly. An exception is the 100 participants/zone curve which remains constant even for larger areas of interest. This is explained by the fact that the VE is split in a very low number of (larger) zones, for example about 7 zones when 250 participants are simultaneously connected in the VE. We also observe that for the curve of 15 participants/zone limit, the increase in the subscription frequency is significant (starting from (IArea $>$ 0.16*CellArea)).

Figure 5.30 shows the proportion signaling/data obtained in our experiments. We remark that the curves have similar shapes regardless the participants/zone chosen limit: for smaller areas of interest the curves show a significant decrease and for larger areas of

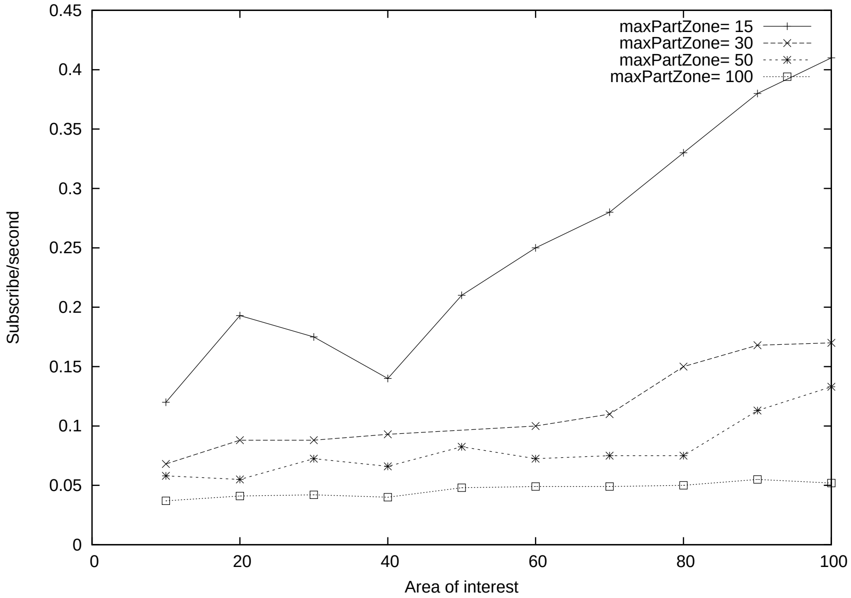


Figure 5.29: Number of Subscribe to VE zones per second

interest the curves tend to become constant. In fact, on figure 5.30 we didn't show the curves for the area of interest side equal with 10, respectively 20 units because these values are very large, the signaling is tens of times bigger than the received data. Displaying them on the same graph would reduce the results readability for larger areas of interest. We can explain this by the fact that every participant receives as signaling the positions of the other participants located in the same VE zone. Having a small area of interest means to communicate with few neighbors for a very short time. So, this amount of data is lower than the signaling, (but here we should remind that we consider as data only the position flow). If the participants receive, in addition, some audio or video flows, it may considerably change the signaling/data ratio.

For large areas of interest (IArea $>$ 0.81*CellArea) the test shows that, for a participant's point of view, we arrive at a fixed proportion between the number of participants subscribed and the number of participants located outside its area of interest, but located in zones intersected by its area of the interest (participants whose positions are received as signaling on the agent's flows). In the absolute values, we observe that in order to receive the fast-rate position of the subscribed participants, the signaling required represents between 11% of the traffic for 30 participants/zone and 44% of the traffic for 100 participants/zone. Comparing the absolute values of the four curves, we observe that the lowest signaling/data proportion corresponds to 30 participants/zone curve followed by 15 participants/zone curve and by 50 participants/zone curve. The tests have shown that breaking the VE in too many zones increases the signaling as much as having many participants in the same VE zone.

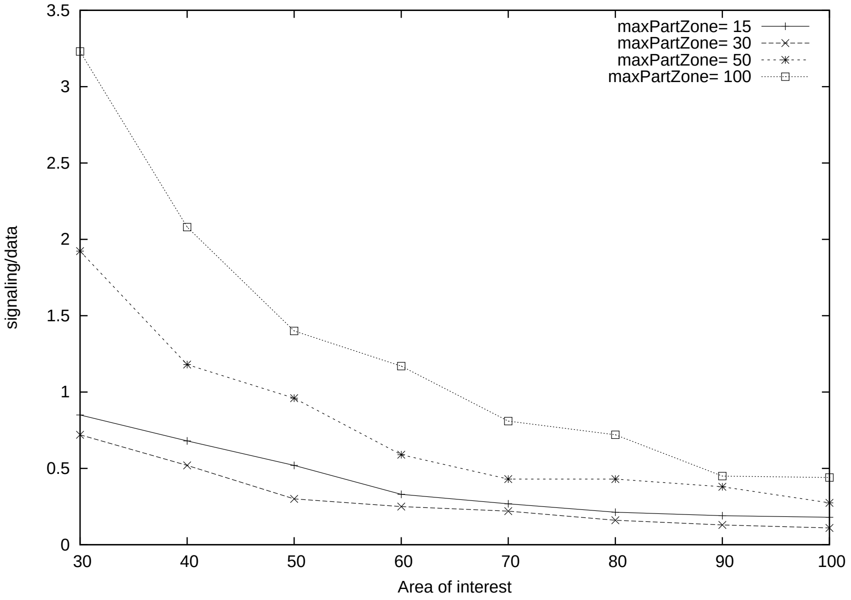


Figure 5.30: Signalling over data proportion

5.3.5 Conclusion of the experimental improvements of Score-SSM performances

In this section we have explored different solutions to improve the performances of Score-SSM by reducing the ratio signalling/data.

In the first experiment, we have studied the effect of reducing the agent to participants signalling rate. The experimental results have shown that for small areas of interest (IArea < 0.5*CellArea) there is an important reduction in the signalling/data ratio while the participants receive the same amount of data. On the other hand, for larger areas of interest (IArea > 0.5*CellArea), this reduction in the agent signalling rate affects the amount of data traffic received by the participants.

The second experiment reduces the signalling traffic from the agent to participants, due to the mapping information between the multicast channels and the zones managed by an agent. Each participant will receive only the mapping of its neighboring zones.

The third experiment studies the traffic received by the participants according to their behavior in the virtual world. We show that the faster the participants move, the less time they communicate with their neighbors and the less data they received.

In the last experiment, we study the signalling/data ratio according to the maximum number of participants simultaneously accepted in a zone of the VE. It shows that an intermediate value (of 30 participants) in a zone had the lowest signalling/data ratio. In addition, the experiment had shown that for larger area of interest (IArea > 0.8*CellArea) the signalling/data ratio are close to the same value regardless the maximum number of participants simultaneously accepted in a zone of the VE.

5.4 Implementing Score-SSM in V-Eye

In this section we briefly present the "V-Eye" application, and the modifications required in order to use the Score-SSM communication module.

5.4.1 V-Eye

Overview V-Eye (or Virtual-Eye) is a 3D virtual environment application over the Internet in which a large number of participants move and interact with each other by sending and receiving multimedia flows. The aim of this tool is to experiment on a real application, the scalability of multicast transmission protocols and the optimization of multimedia transmission over very heterogeneous environments. V-Eye handles text messages, audio and video flows (by interfacing with "Rat" [50] audio tool and "Vic" [51] videoconferencing using the "MBus" [52] protocol).

V-Eye currently uses the SCORE architecture developed in the Planete research team [7]. Each participant in V-Eye interacts with a limited number of participants and it is interested in receiving a sub-part of the overall flows. V-Eye filtering approach is done at the transport-layer, using multiple multicast groups and multiple agents. This approach involves the dynamic partitioning of the virtual environment into spatial areas and the association of these areas with multicast.

Architecture implementation details V-Eye is written in C and C++. 3D rendering is based on "OpenGL" and the portable toolkits "Glut" and "Glui". Participants can also send and receive audio and video flows with others within its neighborhood. This is performed by a modified version of the Rat and Vic application from UCL. V-eye has been tested on Linux machines (Redhat 8 or 9, Mandrake) of the VTHD++ backbone. The binary code of V-Eye is available for download at the following web address:

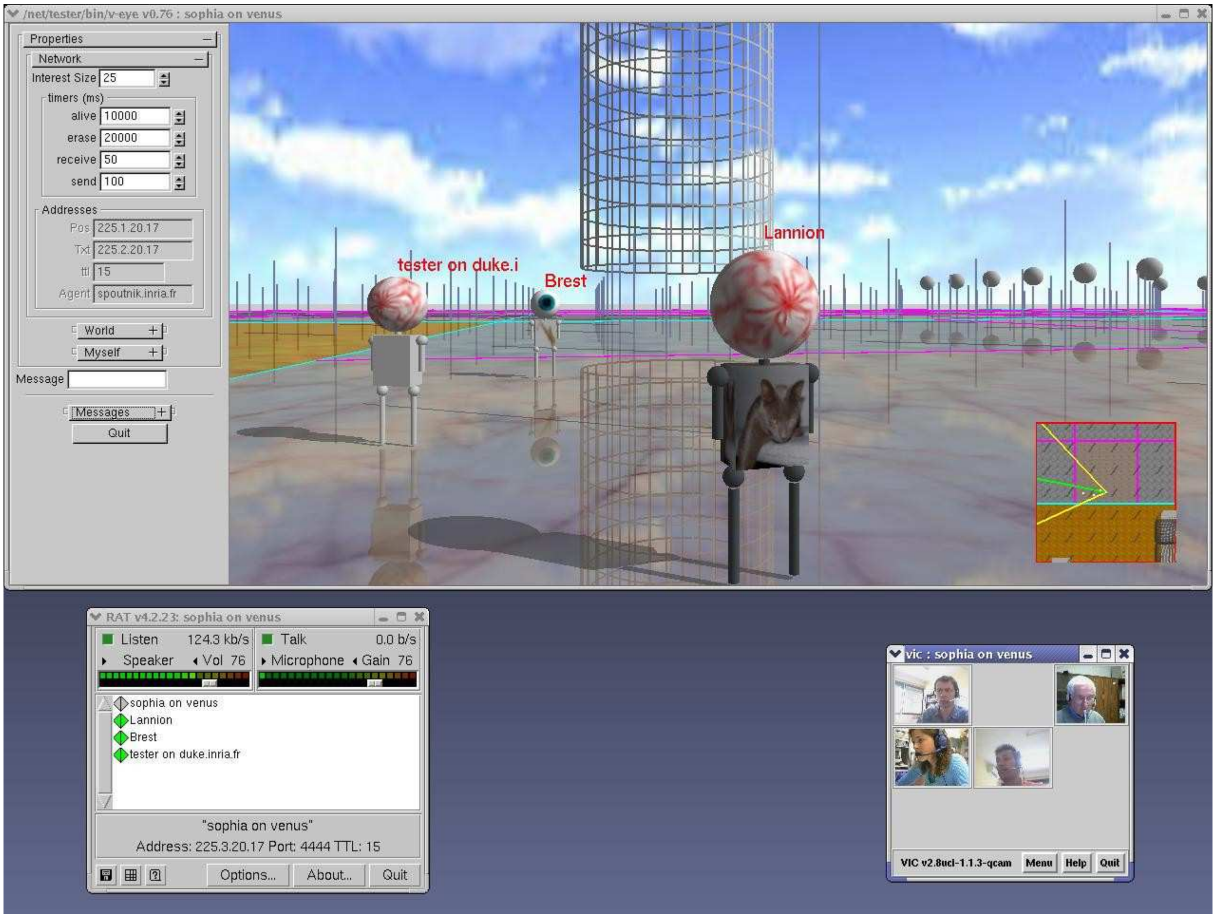
ftp://ftp-sop.INRIA.fr/rodeo/veye

System Requirements

- Linux with Redhat 8 or 9, Mandrake 8
- For faster 3d rendering a graphic card is recommended (nvidia)
- Phillips Webcam for vic
- Alsa or Oss audio card for rat

---

[7] http://www-sop.inria.fr/planete/



Interest Size 25

timers (ms)
alive 10000
erase 20000
receive 50
send 100

Addresses
Pos 225.1.20.17
Txt 225.2.20.17
ttl 15
Agent spoutnik.inria.fr

Listen 124.3 kb/s
Talk 0.0 b/s
Vol 76
Gain 76

sophia on venus
Lannion
Brest
tester on duke.inria.fr

"sophia on venus"
Address: 225.3.20.17 Port: 4444 TTL: 15

Figure 5.31: V-Eye components

5.4.2 Integrating Score-SSM in V-Eye

In this section, we explain the steps in modifying V-Eye code in order to include the Score-SSM architecture implementation. We present the Score-SSM implementation impact on V-Eye and the new functional characteristics of V-Eye using the new communication module.

5.4.2.1 Implementation impact

Score-SSM communication architecture is implemented in two software components: the agent and the participant. The code of the participants is composed by a functional library and by the participant engine. The participant engine was used in the experiments and has replaced the V-Eye previous engine. On the other hand, the participant library remains unchanged regardless the engine. We describe below the modifications required in the agent and in the participant code to include Score-SSM in V-Eye.

The Score-SSM agent

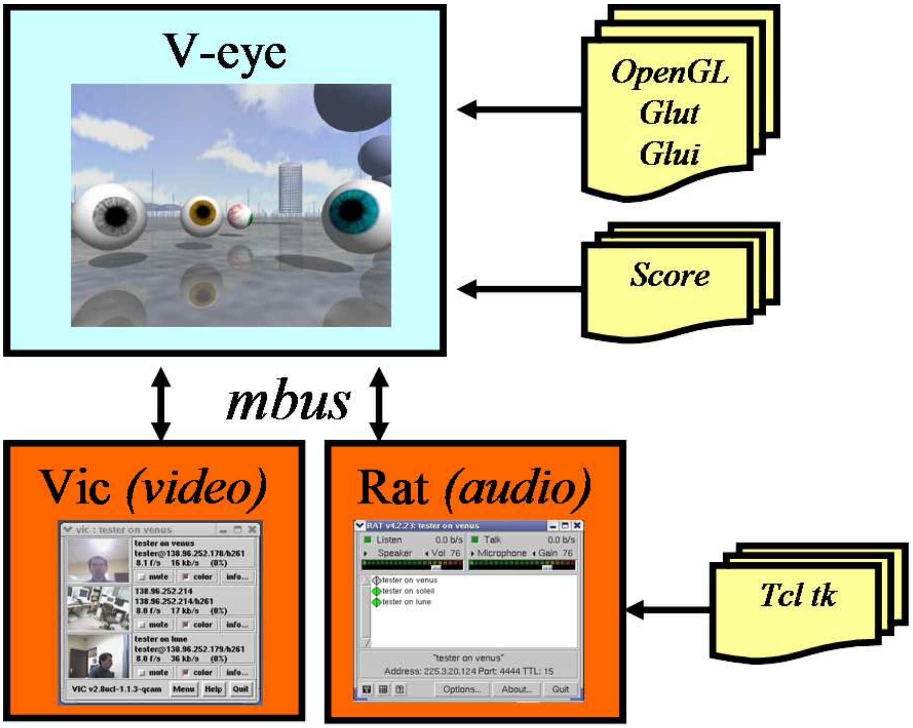
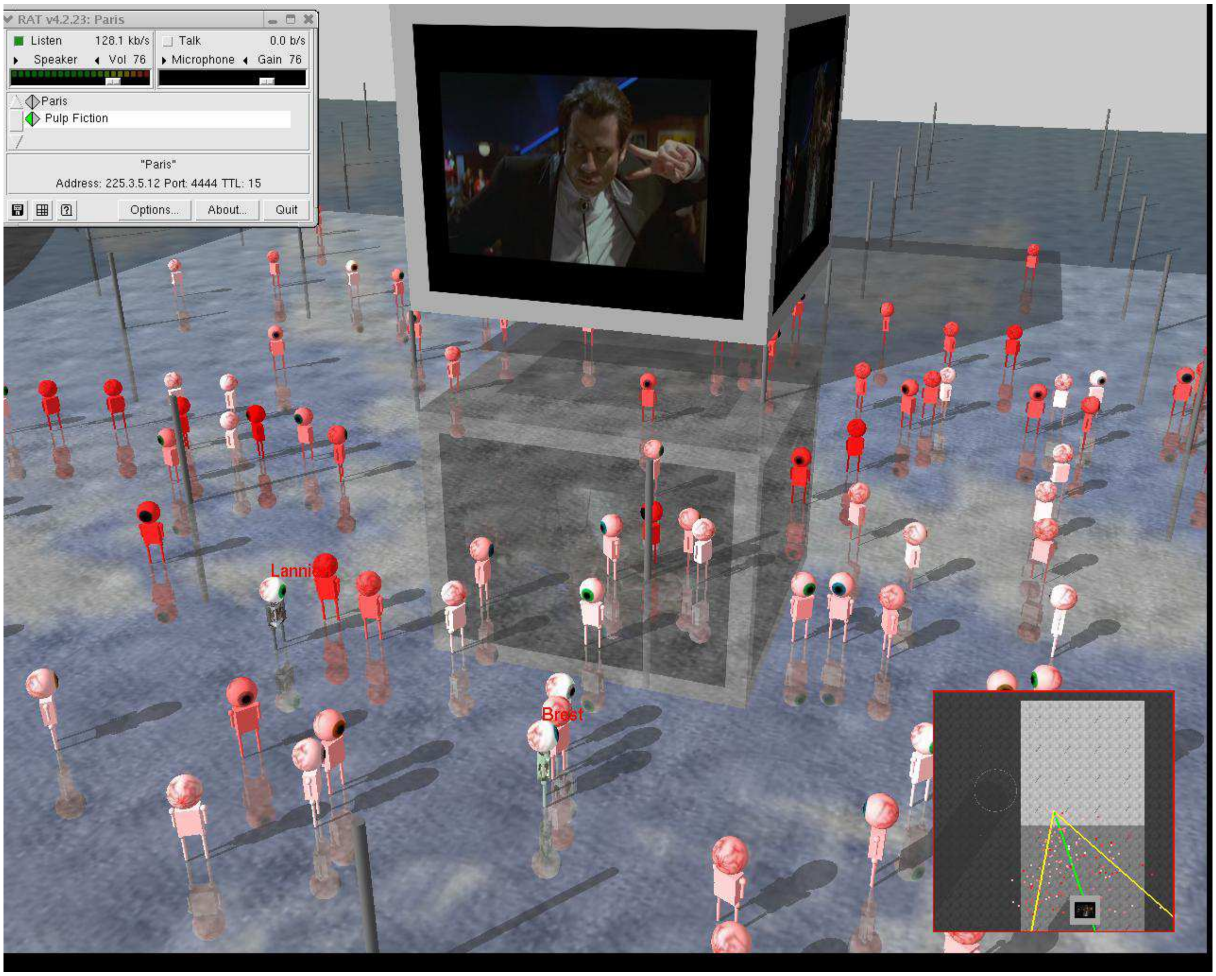


Figure 5.32: V-Eye components

At this moment, only one agent is responsible for all the zone-cut made in V-Eye. No inter-agent communication module is implemented, but when the signaling scalability becomes a problem (due to a large number of simultaneously connected participants), new SSM agents should be launched. The Score-SSM agent has other functionalities in addition to the responsibility to manage the VE dynamic zone-cut. It has a rendezvous functionality, which provides to each entering participant the address of the signaling channel associated with the VE zone containing the entering point of the participant. The agent has a neighbors advertising part too, which transmits over an SSM channel (different for each VE zone) the approximate positions of the participants.

Compared to the Score agent who effectuates the VE zone-cut according to the satisfaction of the participants, the Score-SSM agent effectuates the VE zone-cut according the local participants density.

The Score-SSM agent code is independent from the V-Eye application implementation, or from the Score agent. So, in order to use Score-SSM, the Score agent should be completely replaced by the new agent. This agent may be enhanced with a graphical interface which shows the current VE zone-cut. Modification over Score-SSM agent is mainly concerning the initialization phase, like: the channel addresses used in V-Eye, the application ports, the timers used by the agent engine and the VE parameters.



Address: 225.3.5.12 Port: 4444 TTL: 15

Figure 5.33: V-Eye components

The Score-SSM participant

The Score-SSM participant behavior is implemented within a library. This library was used to realize the tests described in the Experimentations chapter. In order to allow the V-Eye engine to use this communication module an interface was implemented. This interface translates the data and the functions-calls from V-Eye application format to Score-SSM communication module format. The interface is composed by the following functions:

- **Init()** which initializes the data structures used by the communication module
- **SendData()**: the V-Eye engine constructs data structures to be send over the net. These structures are passed to the interface and transformed in network messages and delivered.
- **ReceiveData()**: the input sockets are listened and according to the message type arrived (data, signalling, mapping etc...) the corresponding functional modules are

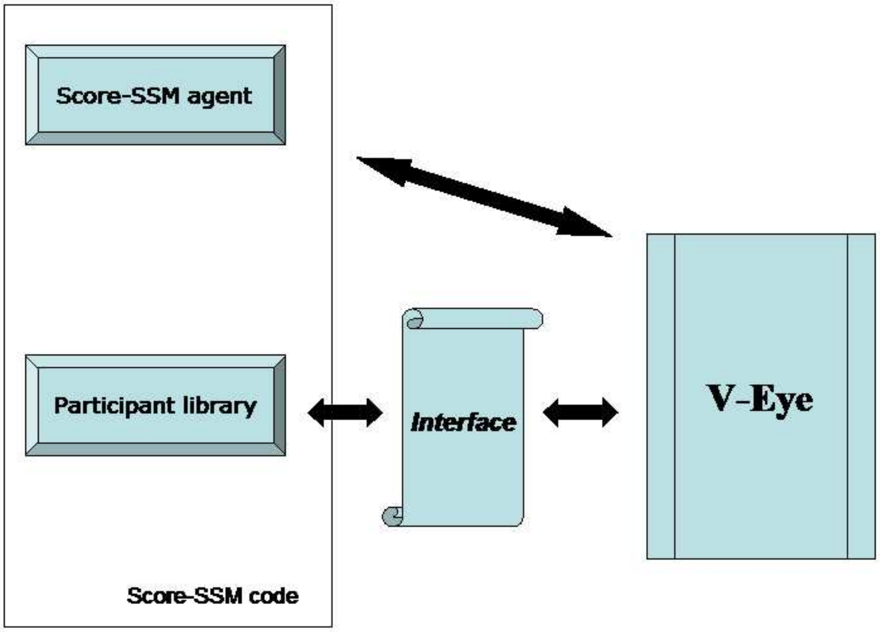


Figure 5.34: Integrating Score-SSM in V-Eye

called. Also, this function passes the information received in the Score-SSM format to V-Eye engine.

- **KeepAlive()**: messages sent to maintain/check the network connections.
- **UpdatePosition()**: updates the current participant position after an input from the user interface or after a move over a predefined movement over a given trajectory
- **EndParticipant()**: write down in the trace file the statistics over the participant behavior in the VE.

The Score-SSM participant library implements the following capabilities:

- It initializes the multicast channels used for data and signalling,
- it connects to the agent and listens to the signalling multicast channel for its current VE zone,
- it determines the participants to communicate with among its neighbors and according to its interest,
- it receives the current zone-mapping and updates the intersection between its area of interest and the VE's zones,
- it checks if the agent and the participants to which it communicates are still alive and disconnect from them if not,

- its sends data and signalling over its corresponding multicast channels.

At the data level, Score-SSM supports different types of flows. V-Eye supports audio flows sent by "Rat" and video flows sent by "Vic". "Vic" and "Rat" have been built for the ASM model. In order to use Score-SSM, these applications should be modified to support multicast channels (they have to subscribe at a sources address in addition to the group address).

Due to the graphical part of V-Eye, the Score-SSM inter-participants messages are extended with new data fields containing information which helps the graphical representation of the neighbors within the VE. Among this information we cite the angle theta, the azimuth, type and state.

5.4.2.2 Functional impact

In this section we describe the new functionalities of V-Eye implementing Score-SSM communication module.

As it was shown in the tests presented in the previous chapter, Score-SSM offers a more precise filtering than Score, thus limiting the superfluous traffic received by the participants. In addition, Score-SSM gives to the participants the possibility to transmit and to receive different types of data flows. Each type of data flow may be send using different quality levels. So, using Score-SSM, V-Eye supports heterogeneous participants and the satisfactions of these participants are maximal. Let's see more in details the new capabilities of V-Eye due to the Score-SSM communication architecture.

a. High satisfaction of the participants. As each participant is capable of adjusting its area of interest such that the incoming traffic will not overpass its own capabilities, it has most of the time a satisfaction close to one. On the other hand, Score was meant to improve the satisfactions of the less satisfied participants and not the overall satisfaction. In practice, this maximal satisfaction means very good communication quality, which is an important issue for video or audio flows.

b. Precise area of interest filtering. The effect of the two-level filtering used by Score-SSM is very little superfluous data traffic received by the participants. Thus, the participants have the possibility to use their capacity saved by the filtering to connect and communicate to additional VE members.

c. Explicit control over the received flows. A participant has the possibility to explicitly express its interest in the reception of a given flow. In Score, a participant receives all the data flows sent in the zones intersected by its area of interest. In addition, a participant controls the precise subscribe and unsubscribe instances to a given flow. If it prefers, it stays connected for ever to a given data flow (for example the data flow send by a friend connected in the VE). In addition, each participant has the possibility to choose which flow to receive from which participant according to its temporary interest. For example, at a given moment it might have to choose between simultaneously receiving 15 text flows, 5 audio flows and 2 video flows or 8 text flows, 4 audio flows and 3 video flows.

d. Different quality levels for the transmitted data flows. We think, for example, at the possibility to transmit layered video. Each layer composing the data flow is transmitted over a multicast channel. The interested receivers subscribe to a number of layers according to their capacities and their interests. Thus, the low capacity receivers which are not capable the get full-quality video flows are still capable to receive low quality video from the interested receivers.

e. Improved zone-cut. In Score the zone-cut is decisive for the amount of data received by the participants. Score's zone-cut is made to improve the less satisfied participants; this may have as a secondary effect the satisfaction diminishing for other participants. In addition, as a consequence of a zone-cut operation, Score may disconnect two participants communicating with each other, even if they are located very close in the VE space. On the other hand, tests have shown that Score-SSM zone-cut doesn't influence the communication between participants. Situations as described above will not occur with Score-SSM communication architecture.

f. Improved scalability. Score scalability is reduced by the limited number of multicast group addresses available in the ASM model. We remind that in Score, each VE zone has an associated multicast address. Score-SSM doesn't suffer any address space limitation; the use of a large number of multicast addresses leads to a better data flows dissemination. In addition, unlike Score, the Score-SSM zone-cut is dynamically made according to the local density of the participants. This avoids a large number of participants to be located in the same VE zone, which clearly will degrade the connections quality of the participants located in such kind of zone. Such situation is opposed to the scalability requirement.

g. Easier deployment over the Internet. The use of Score requires the deployment of an ASM multicast routing algorithm over the Internet. In the state of the art chapter we explain the multicast deployment issues which prevent the ASM model to become widespread in the Internet. The current ASM solution, PIM-SM using MSDP for inter-domain multicast, it is not scalable. The SSM model requires only a modified version PIM-SM which works in the SSM address range (and, of course, IGMPv3 between the host and its corresponding router).

Chapter 6

Conclusion and Outlook

6.1 Summary and discussions

We are still at a time where the multicast retards to become widespread over the Internet and where the large-scale virtual environments are rather small-scale with respect to the number of users participating in these worlds and with respect to their occurrence in the Internet. Most of the existent virtual worlds are proprietary VEs and they were built in a precise purpose (military training, online games, etc...). In our work, we had in mind a general purpose virtual environment, where people move, meet other participants and communicate. It is more like the virtual environments' version of the well-known IRC (Internet Relay Chat) service.

We have described in this thesis the design, the implementation, the analysis and the integration of a communication architecture for large-scale virtual environments over the Internet. In order to effectuate this work, we have previously identified the design issues specific to this type of applications, specially the scalability and heterogeneity issues.

We have presented Score-SSM, the first LSVE communication architectures based on the SSM model. This architecture uses agents and participants to implement a dynamic, combined filtering scheme, composed by two filtering mechanisms. First, a grid-based filtering is implemented through a hierarchy of agents and it is used to reduce the signalling; second, an entity-based filtering is implemented at the participants and it is used to define precisely the interest of the participants in receiving specific data flows. This filtering incorporates two mechanism taking into account the participants dynamics: first, the virtual environment is divided into smaller zones according to the local participants density (more participants are in precise area of the VE, more smaller zones will compose this area); second, each participants constantly determines the data flows it intends to receive by taking into account who are its closest neighbors and which kind of flows each of them transmits (then it subscribes/unsubscribes to the multicast channels corresponding to this flows).

Although LSVE communication architectures implementing entity-based filtering have been already proposed, our solution implements this filtering at the routing level, using networks mechanisms which become widespread deployed (we think at IGMPv3 which is delivered at this time with the latest versions of Windows, Linux and *BSD operating systems). The routing-level solutions are more efficient from the network point of view than the application-level solutions due to their network bandwidth savings realized through the filtering of packets before sending them over the network and due to the use of shortest routing paths.

We have implemented Score-SSM in C++ over NetBSD, Linux and Windows operation systems. We have built a network communication module (library), an agent and a client application which use this library.

Using this implementation, we have performed extensive tests. First, we have compared our proposition with Score, an ASM multicast-based communication architecture, which implements a dynamic grid-based filtering, while Score-SSM combines the dynamic entity-based filtering with the dynamic grid-based filtering. Second, we have conducted experiments meant to find the most appropriate parameters and settings of Score-SSM in order to improve the architecture's performances. The experiments have proven the efficiency and the feasibility of our proposition and they have argued the use of the SSM model as a data distribution solution for the VEs in which the participants exchange multimedia flows.

Taking into account the experiments results, we have integrated Score-SSM in a VE application called V-Eye, developed in the *Planete* research team at INRIA Sophia-Antipolis.

We present below some issues and an outlook for future work to do for the Score-SSM communication architecture:

- The development of an IPv6 version, which requires the use of a different application interface. This will allow the users connected in the Internet over IPv6 to join the VEs which deploy the Score-SSM communication architecture.

- The complete implementation of the proposed multi-agent filtering architecture. Currently Score-SSM uses only one agent. This is a main issue with respect to the scalability of our proposition. So, the implementation of the inter-agents communication module is required in order to deploy Score-SSM at the scale of the Internet.

- We should add mechanisms for security, authentication (users' database) and reliability (avoid agent's crashes). These mechanisms will be added through the implementation of the Main agent (in fact a number of replicated Main agents). In addition, the Main agent role is also to manage the pool of available agents and of the available IP multicast addresses. Actually, a phd. student works on "security in the VE" in Planete Research Team.

- Any LSVE which becomes popular over the Internet should responds at the requirements of "persistence" and "extensibility", i.e. the VE should be "on and working" at any time and it should be possible to enlarged it by adding new world areas, new participants or new agents, without stopping the current execution of the virtual world or disconnecting the online users.

- At this time, for the tests simplicity, the VE's zone-cut is made using squares; in practice the VE may have different shapes which require different zones profiles. Moreover, currently a Score-SSM zone is divided into four smaller squares, this may not be efficient in some cases, dividing only in two smaller zones, for example, may reduce the signalling.

- The agent-level filtering, (we mean here the grid filtering represented by the VE's zone-cut) can be enhanced by providing to each agent as feed-back the participants satisfactions. At this time, the agents only take into account the participants distribution in the VE when they decide a new zone-cut. This additional information permits to have zones with different numbers of participants, realizing a better utilization of the multicast addresses while reducing the zone-mapping signalling traffic.

6.2 Research perspectives

Based on our works, we can envision different research perspectives, such as:

- Building LSVE communication architectures which combine the multicast and the peer-to-peer networks. We envision the use of peer-to-peer to find out who are the neighbors of a participant in the VE and then to use the multicast to send the data flows on distinct multicast channels. We think about the necessity of deploying an agent layer with the role of the super-nodes in peer-to-peer networks (authentication, fast discovery, robustness etc...).

- We also think at the LSVE communication architectures which use multicast routing in combination with the application-level multicast. This allows users which don't have multicast at the routing level, to benefit by some of the advantages of the multicast. Of course, it is not the same thing to use multicast at the application level or at the routing level, but this may contribute at the incremental deployment of the multicast: users which initially run the VE over the application-level multicast, later they may switch to the routing-level multicast.

- Better heuristics and algorithms can be designed for the filtering of the participants between different types of incoming data flows, sent by different participants.

- The use of the SSM model leads to large multicast routing forwarding tables in the routers over the Internet. The (S,G) forwarding tables' entries aggregation problem is still under research. Large routing tables raise scalability concerns and packet forwarding delays, due to longer search times. Using an addressing scheme easy to aggregate, it may limit the scalability issues raised by the large-scale virtual environments. For example, here we can use an unique multicast group address for all the audio channels of an application, only the source address being different.

- The use of the SSM model as data distribution model assumes packets loss due to the use of UDP as the transport-level protocol for multicast. Some reliable mechanism may be integrated in order to transmit the data flows which don't admit any losses.

Our opinion is that in the future years the large-scale virtual environments will be common applications over the Internet. The network and the processing power availability for a large number of Internet users, will lead at the development of new types of applications over the Internet. New uses of the virtual environments will complete the current efforts conducted specially in the network games and military training domains. We can easily imagine a 3D virtual shop distributing music and movies, where the users can walk like in a real store, they can check the shelves and they have the possibility to preview a "videoclip" or to listen a song and they can ask for advice from a vending person or other clients before buying the desired items.

Regarding the communication architecture used, we believe that each VE application will have the possibility to choose, according to the types of flows it supports, the desired data distribution model: peer-to-peer, multicast, application-level multicast etc...

Appendix A

Data structures and message types

In this part we described the data structures constructed by the agents and by the participant. We also describe the messages exchanged in Score-SSM.

Data structures

As a consequence of the messages exchanged, each entity of the VE constructs its own structure representing the knowledge it has of the physical details of the VE and of the other entities connected. We distinguish three types of entities constructing their own structures: the Main agent, the participants and the ordinary agents.

Main agent structures

The Main agent constructs two types of data structures: the first structure is the agents' pool, containing the agents' machines and the second structure is the mapping agents-VE areas. The agents' pool contains information about the all the VE agents:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         agent ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       agent IP address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|               |                                             |
|F| Capacity field|                                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
where AF = active flag ( 0 = stand-by; 1 = active )
```

The second structure is the mapping between the active agents and the VE areas.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Agent ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                X coordinate down-left corner                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Y coordinate down-left corner                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                X coordinate upper-right corner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Y coordinate upper-right corner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Agents structures:

Each agent constructs two different data structures: the *list of the participants* located in its corresponding area of the VE and the *list of zones* composing this area.

The agents keep a list with all the participants located in the part of VE it manages. The information collected about the participants is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         participant ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  IP address of the participant                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          X coordinate                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Y coordinate                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|   Capacity    |   Flow Types  |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The agents keep a structure with the zone-cut they made in the VE area they manage. For each zone they allocate a multicast channel to send signaling over. As the source address is the IP address of the agent, they keep in the structure only the group address.

The information recorded is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     group IP address offset   |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                X coordinate down-left corner                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Y coordinate down-left corner                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                X coordinate upper-right corner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Y coordinate upper-right corner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Participants structures:

Each participant constructs two different data structures: a structure keeping the VE zones intersected by its area of interest and a structure containing the list of the subscribed neighbors.

The subscribed zones structure contains the following information:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Agent IP                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      group IP address offset  |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
|                X coordinate down-left corner                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Y coordinate down-left corner                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                X coordinate upper-right corner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Y coordinate upper-right corner                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In addition, each participant keeps a structure concerning the neighbors to which it subscribed:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Neighbor ID                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    neighbor IP address offset                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            X coordinate                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Y coordinate                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Capacity    |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Messages types

We have seen that Score-SSM communication is composed of a large number of flows. Over each of these flows one or more types of messages are sent. In the following paragraphs we present the format of each type of message present in Score-SSM. The messages used by Score-SSM are:

- **Hello** (participant->agent)
- **Notify position** (participant->agent)

- **Participant position** (agent -> participant)
- **Zone Mapping** (agent -> participant)
- **Area Mapping** (Main agent -> agent)
- **Agent Hello** (agent -> Main agent)
- **Agent Load** (agent -> Main agent)
- **Fast-rate participant position** (participant -> participant)
- **Data packets** (participant -> participant)

When first enters in the VE, the participant sends a **'Hello'** message. It is a unicast, UDP message sent over the slow-rate participant position flow. Its role is to inform the agent about its presence in the VE, and to obtain from the agent the list of the VE members situated around its location. The participant may enter in the VE at a precise location (the X and Y coordinates) or it may get from the agent a start location (in this case, in the packet, the X and Y coordinates are equal to 0). The information contained in a packet is the same with the information registered by the agents in the 'participants list'. When it receives a 'Hello' message, the agent creates a new entry in its current participants list.

The the length of the message is 20 bytes (IP header) + 8 bytes (UDP header) + 8 bytes (location) + 4 bytes (User ID) + 1 byte (Capacity) + 1 byte (Flow types) = 42 bytes

The **'Notify position'** message is sent periodically (typically once per second) over the slow-rate participant position flow. Its role is to keep informed the agent about the approximate position of the participant. The information contained in a packet is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         participant ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          X coordinate                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Y coordinate                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Using this information, the agent updates the entry in its current participants list corresponding to the above ID.

Thus the length of the message is 20 bytes (IP header) + 8 bytes (UDP header) + 4 bytes (User ID) + 8 bytes (location) = 40 bytes

The **'Participant position'** message is sent by the agent to advertise the position of the active participants in a zone. The message contains the participant entry in the agent data base, so its format is the same with the entry format presented above.

So, the message length is: 20 bytes (IP header) + 8 bytes (UDP header) + 4 bytes (IP address) + 4 bytes (participant ID) + 8 bytes (X coordinate + Y coordinate) + 2 bytes (Capacity and supported Flow Types) = 46 bytes are sent at each participant announcement.

The **'Zone mapping'** message is sent by an agent to announce to the participants and to the neighboring agents the current zone mapping, or possible changes in zone size. The message format is similar with the entry in the list of zones hosted by the agent.

The multicast channel address associated to that zone is formed by the IP address of the agent (it is already known by the participant) and by the multicast group address which is transmitted in the above structure.

The **'Area mapping'** message is sent by the Main agent over the "well known" multicast channel, to announce to the current partition of the VE into area, and the mapping of these areas to the agents. The packet format is the same with the 'Zone mapping' message, only the 'group IP address' filed was replaced with the IP address of the agent.

The message length is: 20 bytes (IP header) + 8 bytes (UDP header) + 4 bytes (IP address) + 4*4 bytes (X coordinate + Y coordinate) = 48 bytes

The **'Agent hello'** message is sent by the agent toward the Main agent when a new agent enters in the VE. The Main agent adds a new entry in its agents list, and put the new agent in stand-by. The agent capacity is the maximum number of participants that the agent can simultaneously manage.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            agent ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        agent capacity                         |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The agent IP address is included in the IP header, so there is not necessary to include it in the data filed.

The **'Agent load'** message is sent periodically by every active agent (a reasonable value is once/second) toward the Main agent. This message give the percentage of load of the agent, i.e. the proportion between the number of participants currently managed by the agent and its capacity. The Main agent adds a new entry in its agents list, and put the new agent in stand-by.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           agent ID                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  agent load   |                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Every participant send the **'Fast-rate participant position'** messages toward other participants interested to receive its precise position. Its format is similar with the 'Notify position' packet.

The message length is: 20 bytes (IP header) + 8 bytes (UDP header) + 4 bytes (participant ID) + 8 bytes (location) = 40 bytes

Bibliography

[1] Björn Knutsson, Honghui Lu, Wei Xu, and Bryan Hopkins. Peer-to-peer support for massively multiplayer games. In *Proceedings IEEE INFOCOM*, 2004.

[2] S.E. Deering. Multicast routing in a datagram internetwork. *PhD Thesis, Stanford University*, 1991.

[3] S.Deering D. Waitzman, B.Partridge. Distance vector multicast routing protocol. *RFC 1075*, November 1988.

[4] S. Deering and D. Cheriton. Multicast routing in datagram networks and extended lans. In *ACM Transactions on Computer Systems,8(2)*. ACM Press, May 1990.

[5] J. Moy. Multicast extensions to ospf. *RFC 1584*, March 1994.

[6] William Siadak Andrew Adams, Jonathan Nicholas. Protocol independent multicast - dense mode (pim-dm): Protocol specification. *Work in Progress.*, September 2003.

[7] A. Ballardie. Core-based trees (cbt) multicast routing architecture. *RFC 2201*, 1997.

[8] A. Ballardie. Core-based trees (cbt version 2) multicast routing – protocol specification. *RFC 2189*, September 2001.

[9] A. Helmy D. Thaler S. Deering M. Handley V. Jacobson C. Liu P. Sharma L. Wei. D. Estrin, D. Farinacci. Protocol independent multicast - sparse mode (pim-sm): Protocol specification. *RFC 2362*, June 1998.

[10] Hugh Holbrook Isidor Kouvelas Bill Fenner, Mark Handley. Protocol independent multicast - sparse mode (pim-sm): Protocol specification. *Work in Progress.*, February 2004.

[11] B. Lyles H. Kassem D. Balensiefen C. Diot, B. Levine. Deployment issues for the ip multicast service and architecture. In *Proceedings IEEE Network*, 2000.

[12] D. Meyer Ed. B. Fenner, Ed. Multicast source discovery protocol (msdp). *RFC 3618*, October 2003.

[13] Rekhter Y. Chandra R. Bates, T. and D. Katz. Multiprotocol extensions for bgp-4. *RFC 2858*, June 2000.

[14] D. Thaler. Border gateway multicast protocol (bgmp): Protocol specification. *Work in Progress.*, January 2004.

[15] Handley M. Thaler, D. and D. Estrin. The internet multicast address allocation architecture. *RFC 2908*, September 2000.

[16] Patel B. Hanna, S. and M. Shah. Multicast address dynamic client allocation protocol (madcap). *RFC 2730*, December 1999.

[17] M. Handley and S. Hanna. Multicast address allocation protocol (aap). *Work in Progress.*

[18] Govindan R. Handley M. Kumar S. Radoslavov P. Estrin, D. and D. Thaler. The multicast address-set claim (masc) protocol. *RFC 2909*, September 2000.

[19] K. C. Almeroth, S. Bhattacharyya, and C. Diot. Challenges of integrating ASM and SSM IP multicast protocol architectures. *Lecture Notes in Computer Science*, 2170:343, 2001.

[20] D. Meyer and P. Lothberg. Glop addressing in 233/8. *RFC 3180*, September 2001.

[21] Hitoshi Asaeda. Protocol analysis of any-source multicast and source-specific multicast. Technical Report RR-5080, INRIA Research Report, 2004.

[22] Deering S. Kouvelas I. Cain, B. and A. Thyagarajan. Internet group management protocol, version 3. *RFC 3376*, October 2002.

[23] B. Haberman. S. Deering, W. Fenner. Multicast listener discovery (mld) for ipv6. *RFC 2710*, October 1999.

[24] W. Fenner. Internet group management protocol, version 2. *RFC 2236*, November 1997.

[25] D. McGregor, A. Kapolka, Michael Zyda, and D. Brutzman. Requirements for large-scale networked virtual environments. In *Contel*, Zagreb, Croatia, 2003.

[26] M. Zyda S. Singhal. *Networked Virtual Environments, Design and implementation.* Adisson-Wesley, ACM Press, 1999.

[27] Michael R. Macedonia, Donald P. Brutzman, Michael Zyda, David R. Pratt, Paul T. Barham, John Falby, and John Locke. NPSNET: A multi-player 3d virtual environment over the internet. In *Symposium on Interactive 3D Graphics*, pages 93–94, 210, 1995.

[28] J. W. Barrus, R. C. Waters, and D. B. Anderson. Locales: Supporting large multiuser virtual environments. *IEEE Computer Graphics and Applications*, pages 16(6):50–57, November 1996.

[29] E. Lety. Une architecture de communication pour environnements virtueles distribues a grande-echelle sur l'internet. Technical Report PhD Thesis, Universite de Nice-Sophia Antipolis, 2000.

[30] Chris Greenhalgh. Spatial scope and multicast in large virtual environments. Technical Report TTCS-TR-96-7 1, September 1996.

[31] Greenhalgh C. Awareness management in the massive systems. *Distributed Systems Engineering, Vol 5, No 3*, September 1998.

[32] J.C.Oliveira and N.D. Georganas. Velvet: An adaptive hybrid architecture for very large virtual environments. In *In Proc. Of Virtual Reality Systems and Technology (VRST) 1998, Multimedia and VoIP, Services and Technologies.* New York, NY, USA, April 2002.

[33] J. Dahman, J. R. Weatherly, and F. Kuhl. *Creating Computer Simulation Systems: An Introduction to The High Level Architecture.* Prenctice Hall, 1999.

[34] K. Watsen M. Zyda, H. Abrams. Three tiered interest management for large-scale virtual environments. In *In Proc. Of Virtual Reality Systems and Technology (VRST) 1998.* ACM, Taipei, Taiwan, November 1998.

[35] Daniel J. Van Hook, Steven J. Rak, , and James O. Calvin. Approaches to relevance filtering. In *Proceedings 11th DIS Workshop*, September 1994.

[36] Steven J. Rak and Daniel J. Van Hook. Evaluation of grid-based relevance filtering for multicast group assignment. In *Proceedings 14th DIS workshop*, 1996.

[37] Li Zou, Mostafa H. Ammar, and Christophe Diot. An evaluation of grouping techniques for state dissemination in networked multi-user games. In *ICMP'99*, 1999.

[38] Emmanuel Léty, Thierry Turletti, and Francois Baccelli. Cell-based multicast grouping in large-scale virtual environments. Technical Report RR-3729, INRIA, July 1999.

[39] B. N. Levine, J. Crowcroft, C. Diot, J. J. Garcia-Luna-Aceves, and J. F. Kurose. Consideration of receiver interest in content for ip delivery. In *Proceedings IEEE INFOCOM*, 2000.

[40] James Kurose Don Towsley Micah Adler, Zihui Ge and Stephen Zabele. Channelization problem in large scale data dissemination. In *Proceedings of the 9th IEEE International Conference on Network Protocols (ICNP)*, 2001.

[41] Zihui Ge Ping Ji Mark Keaton James F. Kurose Jonathan K. Shapiro Donald F. Towsley Steve Zabele, M. Dorsch. Sands: Specialized active networking for distributed simulation. *DANCE*, September 2002.

[42] Christophe Diot Manuel Oliveira, Jon Crowcroft. Router level filtering for receiver interest delivery. In *Proceedings of NGC 2000 on Networked group communication*, pages 141 – 150. ACM Press, Palo Alto, California, United States, 2000.

[43] C. Greenhalgh and S. Benford. Massive: A distributed virtual reality system incorporating spatial trading. In *Proceedings 15th International Conference on Distributed Computing Systems (DCS'95)*, June 1995.

[44] C. Greenhalgh. Dynamic, embodied multicast groups in massive-2. Technical Report NOTTCS-TR-96-8 1, University of Nottingham, 1996.

[45] Chris Greenhalgh Jim Purbrick. An extensible event-based infrastructure for networked virtual worlds. In *Presence: Teleoperators and Virtual Environments, v.12 n.1*, February 2003.

[46] C. Carlsson and O. Hagsand. Dive - a multi user virtual reality system. In *Proceedings IEEE VRAIS*, Seattle, Washington, September 1993.

[47] E. Lety and T. Turletti. Issues in designing a communication architecture for large-scale virtual environments. In *Proc. of NGC 1999*. ACM Press, November 1999.

[48] J.C.Oliveira. Issues in large scale collaborative virtual environments. Technical Report PhD Thesis, School of Information Technology and Engineering, University of Ottawa, December 2001.

[49] L Vicisano, L. Rizzo, and J.Crowcroft. Tcp-like congestion control for layered multicast data transfer. In *Proceedings IEEE INFOCOM*, San Francisco, CA, March 1998.

[50] V. Hardman and M. Iken. Enhanced reality audio in interactive networked environments. In *Framework for Interactive Virtual Environments (FIVE)*, 1996.

[51] Steven McCanne and Van Jacobson. vic : A flexible framework for packet video. In *ACM Multimedia*, pages 511–522, 1995.

[52] D. Kutsher J. Ott and C. Perkins. The message bus : A platform for component-based conferencing applications. In *The CSCW2000 workshop on Component-based Groupware*, 2000.

Présentation des Travaux

Le sujet de cette thèse est la conception d'un module de communication reseau pour les environnements virtuels. Il est au sujet de l'existence possible au-dessus de l'Internet, dans un proche avenir, des environnements virtuels hétérogènes, à grande échelle, distribués, avec des millions d'entités.

Context et motivations

Ci-dessous, nous présentons le contexte et les motivations de notre travail, du point de vue de la gestion de réseau et du point de vue des environnements virtuels. Du point de vue de la gestion des réseaux, nous assistons à une évolution et à une expansion continues de l'Internet. Nous sommes dans le contexte où de plus en plus des foyers deviennent connectés à l'Internet et dans le même temps le haut debit est arrivé plus près des utilisateurs finals. Ici nous pensons aux connexions ADSL de plus en plus rapides qui remplacent les liens bas debit.[1] Ainsi, nous pouvons prévoir que la vitesse des connexions ADSL continuera à augmenter dans les années à venir. Cette évolution a été accompagnée, dans les deux ou trois dernières années, de la propagation à large échelle d'une nouvelle classe d'applications: les applications de pair à pair qui fournissent une nouvelle manière pour les transferts des fichiers (tels que Napster, Gnutella, Kazaa, Freenet etc.....). Ces deux évolutions sont en fait liées: les applications de pair-à-pair ont convaincu plus de personnes de souscrire aux connexions ADSL, alors que l'existence d'un grand nombre de connexions plus rapides à l'Internet a mené à un plus grand nombre de réseaux pair-à-pair.[2] L'existence des connexions à grande vitesse arrivant jusqu'aux ordinateurs d'utilisateurs leur facilite l'accès à plus de ressources d'Internet. Un contenu de meilleure qualité, des jeux en ligne, des téléconférences, de la musique en ligne ou de la vidéo à la demande constituent seulement quelques types des applications et de services rendus accessibles par les connexions haut debit à l'Internet. Si nous pensons aux flux video, par exemple, plus la vitesse du connex-

---

[1] En mars 2004, 512kbps est la vitesse commune des raccordements ADSL en France, alors que, dans le même temps, la vitesse est de 50Mbps au Japon.

[2] Le réseau "Fast Track" (Kazaa) a eu environ 3,5 millions d'utilisateurs connectés en même temps, en Mars 2004.

ion de l'utilisateur est élevée, plus la qualité visuelle est meilleure. Pendant que les jours s'écoulent, de plus en plus des personnes ont la possibilité d'échanger des flux multimedia en utilisant l'Internet. Le multipoint IP est un outil idéal pour échanger de données qui nécessite une bande passante large entre les groupes d'utilisateurs. Il économise la bande passante en envoyant les paquets de données seulement une fois sur la même interface de réseau sans se soucier du nombre des récepteurs. Malgré le fait que des connexions réseau à haut débit sont aujourd'hui disponibles, l'utilisation du multipoint est toujours très importante, si nous pensons, par exemple, à une station de TV envoyant des données vers des milliers de récepteurs. Malheureusement, un grand nombre de problèmes de déploiement empêchent le modèle "standard" de multipoint sur l'Internet, Any-Source Multicast (ASM), de devenir répandu à l'échelle de l'Internet. Au moment où nous avons commencé ce travail, un modèle simplifié du multipoint, le multipoint Source-Spécifique (SSM), a été proposé dans l'objectif d'alléger les problèmes de déploiement du modèle ASM. Sachant que le multipoint est l'outil nécessaire pour la communication multimedia en groupe, nous avons choisi d'effectuer notre travail autour du modèle de SSM, parce qu'il permet d'effectuer un filtrage plus élaboré, et qu'il a de meilleures chances de déploiement sur l'Internet. Nous croyons que notre travail aide à illustrer l'intérêt d'utiliser le multipoint et il contribue à un déploiement plus large du multipoint au-dessus de l'Internet. Les environnements virtuels (EV) permettent aux entités d'agir les unes sur les autres en temps réel, sans se soucier de leur emplacement dans le monde réel. Une grande variété d'applications tombent dans la catégorie des environnements virtuels, certains d'entre eux contient des graphiques 3D réalistes et du son. Nous pouvons mentionner ici les simulations militaire, les jeux multijoueurs, les centres commerciaux et les salles d'exposition virtuelles, les conférences en ligne, l'école à la distance etc..... Parmi eux, deux marchés des milliards de dollars sont représentés par l'industrie du jeux et par les simulations militaires. La croissance d'industrie des jeux pour les ordinateurs est si importante que son revenu a dépassé récemment le revenu de l'industrie des films. Un exemple classique de jeu est "Quake arena", dans laquelle chaque utilisateur commande un avatar et il essaye de tirer sur les avatars des autres utilisateurs. Cet jeu a une graphiques 3D et un son réaliste ; d'autre part, le nombre maximum des utilisateurs accepté est réduit. D'autres jeux en ligne, comme EverQuest[3] ou le Sims Online[4], contient des milliers des utilisateurs agissant l'un sur l'autre simultanément, dans un environnement virtuel simple. Ils contient des graphiques moins sophistiqué, autres aspects du jeu ont été augmentés ici. Dans ces jeux, selon le comportement des utilisateurs de l'EV, on peu identifiés des profils différents d'utilisateurs: les accomplisseurs, les explorateurs, les tueurs, les sociables etc... Pour les applications militaires, des classes d'environnements virtuelles ont été définies, comme: High Level Architecure(HLA) ou simulations interactifs distribuées (DIS). Les environnements virtuels sont utilisé pour former des soldats, pour des simulateurs de vol, pour les champs de bataille virtuels qui peuvent reproduire le territoire ennemi, ou pour l'usage des armes simulées au lieu de vrais missiles (qui coûte des milliers de dollars et qui peut accidentellement blesser quelqu'un pendant les essais). La complexité des applications réseau il est en train d'augmenter: ils doivent contrôler des ensembles plus grands et dispersés d'utilisateurs hétérogènes, ils doivent soutenir des types de flux des données multiples envoyés au-dessus des flux unicast et multipoint, ils doivent répondre aux contraintes des temps réel, ils doivent fournir le contrôle d'accès, il doive être conformé et fiable... Ce nombre croissant de participants connecté dans les mondes virtuels souligne l'importance de la scalabilité des applications. Les utilisateurs du EVs sont situés dans une partie précise du monde, ils ne sont pas intéressés par chaque événement produit dans le EV et ils ne peuvent pas traiter tout. De leur point de vue, un bon filtrage augmente leur connectivité dans le EV (et dans le même temps la scalabilité des applications). L'évolution permanente des environnements virtuels exige une adaptation de leurs architectures de communication. Les évolutions récentes des réseaux IP (concernant en particulier le multipoint et le pair-à-pair) peuvent fournir des réponses appropriées. Dans cette thèse, nous répondons aux issues mentionnées ci-dessus par la proposition du SCORE-SSM, une architecture pour les environnements virtuelle à grande échelle. SCORE-SSM soutienne un grand nombre des récepteurs hétérogènes connecte en utilisant une infrastructure basée sur le multipoint (SSM), il fournisse des différents types des flux multimedia et il effectue un partition dynamique de l'environnement virtuel selon la densité des participants.

---

[3] http://www.everquest.com/

[4] http://www.eagames.com/official/thesims/thesimsonline

Les contributions de la These

Les contributions principales de cette thèse sont:

- L'élaboration de l'architecture de communication SCORE-SSM. Nous avons conçu la première architecture de communication pour les EV basée sur le multipoint SSM. Cette architecture emploie une filtrage dynamique à deux niveaux qui permet la diffusion efficace de données. Le premier niveau du filtrage (filtrage basé sur grille) assure la scalabilité de la signalisation et il est mis en application par une hiérarchie des agents. Le deuxième niveau de filtrage (filtrage oriente vers l'entité) assure la scalabilité des donnée reçu par les participants et il est réalise à travers un mécanisme de filtrage déployé chez les participants.

- Avoir conduit un grand nombre des experimentations qui prouvent la faisabilité et évaluent les performances du SCORE-SSM compare a SCORE, une architecture de communication (pour les EV) existante.
  Même si il est largement accepte que le modèle ASM est la meilleure approche pour des applications multipoint, nos résultats prouvent q'une architecture de communication basée sur SSM peut réaliser des meilleures performances q'une architecture de communication basée sur ASM, au coût des taux des signalisation réduite, et peuvent même fournir des avantages additionnels (comme une plus "fine" granularité pour le filtrage).

- L'implementation du SCORE-SSM, qui a été pratiquement réalisés parallèlement avec la phase de correction de l'implementation du protocole IGMPv3, au-dessus de NetBSD, fait par le Kame project[5]. En outre, cette implementation a été intégré dans V-Eye, une application du monde virtuelle développe dans l'équipe de recherche Planete à l'INRIA Sophia-Antipolis. Ceci nous a donné la possibilité de vérifié dans des vraies conditions, l'efficacité de l'architecture SCORE-SSM.

---

[5] http://www.kame.net/

Le plan de la Dissertation

Cette thèse est organisée en cinq chapitres. Elle a une structure monolithique, la lecture d'un chapitre étant basé sur le contenu des chapitres précédents. Le chapitre 2 présente l'etat de l'art du multipoint IP et de la communication réseaux pour les environnements virtuels, les deux briques principales utilise pour construire notre architecture de communication. Tenant compte des caractéristiques du multipoint et des environnements virtuels, nous expliquons en chapitre 3 les choix que nous avons fait afin de concevoir notre architecture de communication. Une fois que nous avons indiqué les choix de conception, en chapitre 4 nous présentons en détail SCORE-SSM, l'architecture de communication multipoint que nous proposons pour la communication de groupe dans les EVGE. Le chapitre 5 présente un grand nombre des experimentations, effectués afin de prouver la faisabilité et d'évaluer l'efficacité du SCORE-SSM. Dans la première partie de ce chapitre, nous présentons les comparaisons faites avec l'architecture de communication SCORE; dans la deuxième partie, le rôle des expériences était de trouver les meilleurs mécanismes et valeurs des paramètres pour SCORE-SSM. Afin de réaliser les expériences décrites dans le chapitre 5, nous avons réalisé une implementation du SCORE-SSM. La dernière partie du chapitre 5 présente l'intégration de l'implementation du SCORE-SSM dans une application pour les environnements virtuelles appelée "V-Eye". Cette application a été développée dans l'équipe de recherche Planete à l'INRIA Sophia-Antipolis, et elle est basée à l'origine sur l'architecture SCORE. Nous présentons l'interface cree afin de permettre à l'application V-Eye d'employer l'architecture de communication SCORE-SSM. Le chapitre 6 présente le sommaire du travail réalisé et les conclusions des résultats obtenus pendant cette thèse. Quelques perspectives sont indique pour les travaux futurs dans les EVGE, du point de vue de la communication.

Problématique

En ce chapitre nous expliquons les hypothèses que nous avons pris en considération, afin de concevoir une architecture de communication pour les LSVE qui, dans le même temps, est scalable, soutient les participants très hétérogènes et répond aux demandes de temps-réel des applications multimedia.

Vue d'ensemble

Nous présentons les hypothèses que nous avons assumée afin de construire notre architecture de communication et une application de LSVE qui utilise cette architecture:

- **Le Multipoint.** La première option devait profiter de SSM comme modèle de la distribution de données et comme mécanisme de filtrage de réseau-niveau. En fait, un des rôles de notre application de LSVE doit montrer que l'utilisation du multicast pour la livraison de données mène aux architectures scalable (c.-à-d. un grand nombre d'utilisateurs peuvent être simultanément reliés, les utilisateurs reçoivent très peu des données superflues, la signalisation est réduite etc...). Afin de fournir les mêmes services et fonctionnalités aux utilisateurs finals, un module de communication pour les LSVE basée sur SSM sera plus complexe q'un module de communication basée sur ASM.

- **Hétérogénéité.** Selon les caractéristiques des participants reliés à l'Internet, l'application LSVE devrait soutenir les différents contraintes de qualité selon les possibilités des participants (largeur de bande du réseau et puissance de calcul). En d'autres termes, l'architecture de communication doit fournir un appui fort pour l'hétérogénéité des participants.

- **Filtrage et scalabilité.** L'architecture de communication devrait efficacement intégrer les demandes de filtrage spécifiques des participants avec les conditions globales de scalabilité. Les participants ont besoin d'une manière de montrer ou changer leur propre intérêt momentané dans une zone du VE ou au niveau de qualité des communication avec un voisin particulier. Dans le même temps, le but de l'architecture réseau est d'accepter simultanément autant de participants possible sans dégrader la qualité de communication des participants déjà reliés dans le VE.

- **Les contraintes de temps-réel.** Un des principaux rôle d'une application pour les LSVE est de soutenir l'échange de l'information entre les participants. Les flux audio et vidéo, par exemple, sont parmi les flux de données les plus utilisés pour la communications entre participants. Ce sont des trafic de communication en temps réel qui exigent des délai des communication réduites et une certaine largeur de bande passante disponible. Ceci impose la demandé d'une architecture de communication qui construit des chemins plus courts entre participants.

Notre choix

Dans les paragraphes précédentes, nous avons présenté les hypothèses que nous pensons très importantes pour concevoir une architecture de communication pour les LSVE. Après nous avons analyse ces hypothèses, nous proposons notre solution: l'architecture Score-SSM.

- Du point de vue du multicast, nous proposons l'utilisation du modèle SSM.
- Dans notre architecture chaque participant est capable d'envoyer des flux des données des types différentes et chaque type de flux des données est envoyé sur un canal multicast SSM différent. En outre, chaque participant peut avoir un centre d'intérêt pour chaque type de flux des données q'il reçoit. Il peut modifier dynamiquement les diamètres de ses centres d'intérêt en concordance avec les préférences momentanées.
- Pour fournir le filtrage et la scalabilité, nous proposons une solution hybride de filtrage basée sur:
  - un filtrage basée sur l'entité, qui donne aux participants la possibilité de exprimer leur intérêt particulier,
  - un filtrage basée sur grille, qui distribué l'information globale du VE dans des zones plus petites.

  Le filtrage basée sur l'entité détermine d'abord les voisins candidats pour communiquer (situé dans le centre d'intérêt du participant) et alors il choisit quelle type de flux il va échanger avec chaque voisin. Le filtrage basée sur grille est hiérarchique. D'abord, le monde virtuel est coupé en secteurs plus petits, chacun de ces secteurs est contrôlé par une machine appelée l'*agent*. En second lieu, chaque agent divise son secteur en zones plus petites, selon la densité du participant local. Ce filtrage hiérarchique est censé pour augmenter la scalabilité du modèle de communication en tenant compte des caractéristiques des participants (la taille d'une zone de VE est déterminée selon les capacités des participants) et des caractéristiques des agents (chaque agent contrôle une région du VE selon ses possibilités).

- Notre architecture de communication est conçue pour soutenir des flux multimedia comme les flux audio et video. Elle emploie le modèle SSM, qui construit des arbres enracinés au source de données. Ceci réduira les paquets retardé, parce que les participants reçoivent les données des sources préférées via le chemin de le plus court.

L'architecture Score-SSM

Dans cette section nous présentons Score-SSM, une architecture de communication qui permet à un grand nombre de participants hétérogènes d'être reliés simultanément dans un environnement virtuel (VE) qui utilise un cheminement multicast SSM. Nous considérons le type d'environnement virtuel où les participants échangent des données seulement avec leurs voisins plus proches, la communication tenue dans d'autres parties du VE ne les intéresse pas. Pour réaliser un tel modèle de communication, chaque participant doit filtrer les données envoyées par les participants éloignés. Dans les prochains paragraphes, nous présentons les caractéristiques principales de notre architecture: le filtrage des données superflues, l'hétérogénéité des participants et l'utilisation du modèle SSM.

Le Filtrage

Score-SSM proposent une architecture de filtrage à deux niveaux. D'abord, le VE est coupe en zones non-recouvertes, selon la position et la densité des participants dans le VE. Un participant reçoit les positions approximatives des autres participants situés dans les zones du VE coupées par son intérêt. En second lieu, les participants calculent leurs voisins "les plus proches" parmi les participants indiqués par le premier niveau du filtrage et il commence à communiquer avec eux. Le premier niveau du filtrage du Score-SSM (nous l'appelons "filtrage basse sur les agents") se compose d'un couche de signalisation entre agents. Pour améliorer la scalabilité, le VE est coupé en zones: un agent enregistre tous les mouvements des participants dans une zone particulière. Ainsi, nous avons un group des agents, qui partagent ensemble la signalisation du VE. Par la signalisation, nous comprenons ici tout le trafic réseau nécessaire pour établir la communication entre participants. Chaque agent est responsable de la signalisation dans une ou plusieurs zones du VE. Plus de participants sont simultanément reliés dans le VE, plus d'agents sont déployés pour contrôler la signalisation dans le VE. Les agents décident la décomposition du VE en zones. Nous pensons qu'une decoupage dynamique en zones selon la distribution des participants dans le VE mène à une meilleure approximation d'intérêt des participants. Le deuxième niveau du filtrage (nous l'appelons "filtrage basse sur les participants") est réalisé par l'utilisation du multicast Source-Spécifique (SSM) comme le modèle de distribution de données. Une fois qu'un participant avait calculé ses plus proches voisins il souscrit à leurs flux de données. L'utilisation du SSM permet aux participants de communiquer seulement avec leur voisins "les plus proches", sans recevoir des flux de données peu désirés à part des participants "éloignés", comme dans le cas d'utilization du ASM. SSM donne aussi, aux participants, la possibilité de recevoir uniquement le flux d'un voisin particulier (nous supposons que les participants peuvent envoyer différents types de flux de données) parce que chaque participant utilise un canal SSM différent pour chaque type de flux des données qu'il transmet. Nous pouvons observer qu'un canal SSM correspond à des flux des données spécifiques envoyé par un participant spécifique. Cette distinction très précise, entre les flux de données transmis dans le VE, aide beaucoup le filtrage des données reçu par les participants. Nous observons que les participants de VE ont la possibilité de montrer avec précision leur intérêt de recevoir les flux de données des autres participants. Donc c'est un avantage d'employer SSM, mais qui vient au prix des mécanismes de filtrage complexes qui devraient être mises en application par l'architecture, pour détecter les voisins avec précision (et les types de flux de données qu'ils échangent) auquel un participant communique.

L'Hétérogénéité et les flux multimedia

Nous considérons les environnements virtuels dans lesquels les participants sont très hétérogènes: différentes puissances de calcul, différents types de flux de données envoyé/reçu, différentes connexions réseau et différents intérêts. Notre architecture est conçue pour prendre en compte tels participants hétérogènes: Score-SSM donnent à chaque participant relié dans le VE la possibilité de communiquer avec un certain nombre de voisins en concordance avec ses possibilités. La quantité de données envoyé/reçu d'un participant n'est pas influencée par la présence dans le VE d'autres participants avec des capacités différente.

Nous pouvons classifier les participants selon les différentes possibilités en termes de flux des communication transmis ou reçu en: *capacités de transmission* et *capacités de reception*

. Dans le VE, chaque participant transmet sa position et il est capable de transmettre des messages textes. En outre, quelques participants peuvent transmettre des flux audio et/ou des flux video. Ces types flux déterminent les capacités de transmission d'un participant. Puisque les participants sont capables de transmettre des différents types des flux, leurs *capacités de reception* sont définies selon les types des flux que un participant peut recevoir. Les *capacités de reception* d'un participant se composent de plusieurs *areas d'intérêt* de différentes tailles: *le cercle de presence, le cercle audio et le cercle visuel.*

Le cercle de presence correspond à son secteur de la vision: le participant détecte les mouvements de tous les participants situés à une distance plus petite qu'une valeur donnée qui s'appelle son rayon de vue. En outre, le participant est capable de recevoir des messages texte envoyé par les participants situés dans sa gamme de vue.

Le cercle audio du participant correspond à un secteur circulaire où le participant est capable d'écouter les participants situés à une distance plus petite qu'une valeur indiquée (rayon audio).

Le cercle visuel du participant correspond à un secteur circulaire où le participant est capable de recevoir les flux visuels transmis par les participants situés à une distance plus petite qu'une valeur indiquée (rayon visuel).

Chaque participant a deux paramètres additionnels: nombre maximum des flux d'audio (respectivement video) qu'il est capable de recevoir simultanément. Ces valeurs sont deux limites supérieures qui aident à raffiner les capacités d'un participant. *Le cercle audio* et *le cercle video* sont calculées à partir de ces valeurs: plus grand est le nombre de voisins capables d'envoyer des flux audio respectivement video, plus grand devient *le cercle audio (video)* du participant. Comme le participant peut recevoir jusqu'au "nombre maximum d'écoulements audio", il recherche les sources audio parmi les participants situés dans son *le cercle video. Le cercle audio* est réduit si ces participants audio-capables sont près du participant. En outre, ces valeurs réduisent les pertes des flux de position des participants du à la utilisation excessive des ressources reseau de participant, provoquée par les flux audio et de vidéo reçu. *Le cercle video* est contenue dans *le cercle audio*, qui à son tour, est contenue dans *le cercle visuel.* Dans le même temps le débit des flux video est plus grand que le débit des flux audio, qui est plus grand que les débits des flux position et de textes.

Le Multicast

Pour des raisons de scalabilité, les flux audio et video sont envoyés en utilisant le multicast: pour économiser la largeur de bande passante, les paquets audio ou video traversent seulement une fois un lien physique du réseau. Ainsi, la largeur de bande sauvée peut être employée par d'autres flux, permettant aux participants d'échanger des données avec un nombre plus grand de voisins. Nous pensons que les arbres de distribution partagés, comme ils sont construits par quelques algorithmes de routage multicast (CBT, Pim-SM), ne sont pas appropriés pour fournir simultanément des flux audio et video, en raison de la grande quantité du trafic qui traverserait les mêmes branches des arbres de distribution (c.-à-d. par les mêmes liens du réseau). D'autre part, les arbres enracinés aux sources permettent aux participants de souscrire directement à un flux vidéo particulier ou à un flux audio particulier (transmis par un voisin spécifique). De cette façon, la concentration de trafic est évitée, comme la réception des paquets peu désirés envoyé par les participants situés en dehors du centre d'intérêt d'un participant. Le modèle SSM construit les arbres de distribution de données enracinés aux sources et son espace des adresses permet à chaque participant d'être la source de plusieurs groupes multicast. Comparé au routage multicast fournit par le modèle ASM, il est deployable plus facilement au-dessus de l'Internet. Le modèle ASM a été conçu pour permettre à toutes les entités d'envoyer des données sur le même groupe de multicast. Les flux de données envoient par différentes sources sont agrégés. Ainsi, un participant peut recevoir des paquets de données envoyé par des sources non désirées. D'autre part, en utilisant le modèle SSM, un participant souscrit explicitement aux flux qu'il l'intéressé. Ainsi, le modèle SSM offre aux participants, "un mécanisme de filtrage de granularité fine" étant plus adapté que l'ASM à notre architecture. Cependant, cette simplicité à "couche réseau" exige une complexité additionnelle aux niveaux plus élevés du hiérarchie des protocols ISO-OSI.

La variation de la densité des participants en régions différentes du VE influence les agents calculant la charge et elle peut déclencher une nouvelle partition en zones du VE. Nous considérons que tous les participants sont SSM-capables et que chaque participant a un canal multicast différent pour chaque type de données qu'il transmet: un canal pour sa position, un canal pour ses données des textes, un canal pour ses données audio (s'il transmet des données audio) et un canal pour ses flux vidéo (s'il transmet des flux vidéo).

Pour recevoir des données des participants situés dans sa zone d'intérêt, un participant souscrit aux canaux multicast correspondants à ses voisins les plus proches, selon son intérêts pour leur position, texte, acoustique, flux vidéo. Nous considérons que tous les participants emploient la même adresse IP, "G", de groupes multicast (dans le modelé SSM) pour leur flux de position. En outre, une autre adresse unique de groupe de multicast est disponible pour les flux audio des tout les participants et des encore une adresse pour les flux vidéo de tout les participants. Ceci réduit la signalisation additionnelle exigée pour échanger l'information de groupe. Nous considérons qu'un participant communique seulement avec ses voisins les plus proches: il peut établir une communication avec un autre participant seulement s'il communique à ce moment avec tous les participants localisés plus proche que le nouveau participant. Ainsi, afin de communiquer avec un participant donné il doit se déplacer suffisamment proche.

La partie expérimentale

Afin de prouver les performances du Score-SSM nous avons réalisé un certain nombre d'expériences. Dans ces expériences, nous comparons le trafic de communication d'un LSVE utilisant Score-SSM avec le trafic de communication d'un LSVE déployé sur Score.

Dans une première expérience, nous examinons le trafic moyen reçu par les avatars. Score-SSM mène à une meilleure utilisation de la bande passante du réseau, parce que les avatars ne reçoivent aucune donnée superflue. Dans Score, la grande partie des paquets de données reçus par les avatars sont généré par des sources non-désirées.

Dans une deuxième expérience, nous étudions la distribution des avatars dans les groupes multipoint associe aux zones composant l'EV. Ceci représente un indicateur de l'efficacité de l'algorithme de decoupage. Les expériences montrent que dans les deux cas, le decoupage dynamique du EV mène à une meilleure utilisation des groupes multipoint.

Dans une troisième expérience nous examinons la fréquence d'abonnement des avatars aux zones de l'EV. Les expériences montrent que pour toutes les types de decoupage de l'EV et les vitesses de déplacement des avatars considère, les avatars croisent moins de zones tout en se déplaçant dans l'EV, dans le cas du Score-SSM. Les zones composant l'EV sont plus grandes en Score-SSM comparées au Score c.-à-d. il y a une meilleure utilisation des groupes multipoint destine au filtrage basé sur la grille.

Dans la quatrième expérience nous comparons le trafic de signalisation reçu par les avatars. Cette signalization est sensiblement plus grande dans le cas du Score-SSM comparé avec Score. Dans Score-SSM, les agents fournissent aux avatars des informations sur leurs voisins, ainsi la taille de paquets est plus grande et leur fréquence est plus élevé comparée avec Score.

La conclusion des expériences peut être énoncé comme suit: Score produit moins de trafic de signalisation que Score-SSM, mais, d'autre part, les avatars reçoivent des données superflues dans des quantités qui peuvent devenir très importantes dans certains cas. En Score-SSM, les avatars ne reçoivent aucun trafic de données superflu et ils peuvent décider avec un precision plus grande aux quelle flux de données ils vont souscrire. Bien que Score-SSM produise plus de trafic de signalisation compare à Score, la quantité de trafic de signalisation supplémentaire est négligeable quand on considère du trafic de données tel qu'audio ou video.

Conclusion et perspectives

Sommaire et discutions

Nous sommes toujours dans une contexte où le multipoint tarde a devenir répandu dans l'Internet et où les environnements virtuels à grande échelle sont plutôt de petite taille en ce qui concerne le nombre d'utilisateurs participant à ces mondes et en ce qui concerne leur occurrence dans l'Internet. La plupart des mondes virtuels existants sont des EV propriétaire et elles ont été construites dans un but très précis (formation militaire, jeux en ligne, etc...). Dans notre travail, nous avons considère un environnement virtuel tout usage, où les gens se déplacent, rencontrent d'autres participants et communiquent. Il resemble à la version pour les environnements virtuels du service bien connu IRC (Internet Relay Chat).

Nous avons décrit dans cette thèse la conception, l'exécution, l'analyse et l'intégration d'une architecture de communication pour les environnements virtuels à grande échelle au-dessus de l'Internet. Afin d'effectuer ce travail, nous avons précédemment identifié les issues de conception spécifiques à ce type d'applications, particulièrement les issues de scalabilité et d'hétérogénéité.

Nous avons présenté SCORE-SSM, la première architecture de communication pour les EVGE basées sur le modèle SSM. Cette architecture emploie des agents et des participants pour mettre en application un filtrage dynamique et il est composé de deux mécanismes différente. D'abord, un filtrage *basé sur grille* est mis en application par une hiérarchie des agents et il est employé pour réduire la signalisation; en second lieu, un filtrage *basé sur entités* est mis en application aux participants et il est employé pour définir avec précision l'intérêt des participants pour recevoir des flux de données spécifiques. Cet filtrage incorpore deux mécanismes qui tient compte de la dynamique des participants: d'abord, l'environnement virtuel est divisé en plusieurs petites zones selon la densité locale de participants (plus de participants il y a dans une région précise du EV, plus des petites zones composeront ce secteur); en second lieu, chaque participant détermine constamment les flux de données qu'il prévoit de recevoir en tenant compte de qui sont ses voisins les plus proches et quel genre des flux des données transmet chacun d'eux.

Bien qu'on ait déjà proposé des architectures de communication pour les EVGE mettant en application le filtrage basé sur les entités, notre solution met en application cet filtrage au niveau du routage, en utilisant les mécanismes de réseaux qui sont en train de devenir répandus (nous pensons à IGMPv3 qui est livré en cet moment avec les dernières versions des logiciels d'exploitation Windows, Linux et *BSD). Les solutions au niveau du routage sont plus efficaces du point de vue réseau compare avec les solutions au niveau application parce'que ils économise la bande passante du réseau en réalisant le filtrage des paquets avant de les envoyer en réseau et parce'que ils utilise les chemins de routage les plus courts.

Nous avons implémenté SCORE-SSM en C++ au-dessus des systèmes d'exploitation NetBSD, Linux et de Windows. Nous avons établi un module de communication réseau (bibliothèque), un "agent" et une application "client" qui utilize cette bibliothèque.

En utilisant cette implementation, nous avons réalisé plusieurs experiments. D'abord, nous avons comparé notre proposition aux SCORE, une architecture de communication multipoint basée sur ASM, qui met en application un filtrage dynamique basé sur grille, alors que SCORE-SSM combinent le filtrage dynamique basé sur les entités avec le filtrage dynamique basé sur grille. En second lieu, nous avons conduit des expériences censées de trouver les paramètres les plus appropriés pour SCORE-SSM afin d'améliorer les performances de l'architecture. Les expérimentés ont prouvé l'efficacité et la faisabilité de notre proposition et ils proposé l'utilisation du modèle SSM comme solution de distribution de données pour les EV dans lequel les participants échangent des flux multimedia entre eux. Après avoir pris en compte les résultats des expériences, nous avons intégré SCORE-SSM dans une application de EV appelée V-Eye, développé dans la équipe de recherche *Planete* à INRIA Sophia-Antipolis.

Nous présentons ici quelques perspectives pour des travaux futurs autour de l'architecture de communication de SCORE-SSM:

- Le développement d'une version IPv6, qui exige l'utilisation d'une interface d'application différente. Ceci permettra les utilisateurs reliés dans l'Internet au-dessus d'IPv6 de joindre le EVs qui déploient l'architecture de communication SCORE-SSM.
- L'implementation complète de l'architecture de filtrage multi-agent proposée. Actuellement SCORE-SSM emploient seulement un agent. C'est une problème principale en ce qui concerne la scalabilité de notre proposition. Ainsi, l'exécution du module de communication inter-agents est exigée afin de déployer SCORE-SSM à l'échelle de l'Internet.
- Nous devrions ajouter des mécanismes pour la sécurité, l'authentification (une base de données des utilisateurs) et la fiabilité. Ces mécanismes seront ajoutés par l'exécution de l'Agent Principal (en fait un certain nombre d'agents principaux repliés). En outre, le rôle de l'Agent Principal est également de contrôler le nombre des agents disponibles et la repartition des adresses IP disponibles en multipoint.
- Toutes les EVGE qui devient populaire au-dessus de l'Internet doivent réponde aux conditions de la "persistence" et de l' "extensibilité", c.-à-d. l'EV est "en marche" à tout moment et il est possible de l'agrandir en lui ajoutant des nouveaux secteurs du monde, des nouveaux participants ou des nouveaux agents, sans arrêter l'exécution courante du monde virtuel ou débrancher les utilisateurs en ligne.
- Pour la simplicité des essais, à cette heure, l'EV est découpe en zones carre; dans la pratique le EV peut avoir de différentes formes qui exigent des différents profils de

zones. D'ailleurs, actuellement une zone de SCORE-SSM est divisée en quatre plus petites carres, ceci ne peut pas être efficace dans certains cas, en divisant seulement dans deux plus petites zones, par exemple, on peut réduire la signalisation.

- Le niveau des filtrage des agents peut être augmenté en fournissant à chaque agent comme donnes les satisfactions des participants. Actuellement, les agents tiennent compte seulement de la distribution des participants dans l'EV quand ils décident un nouveau decoupage. Cette information additionnelle permette d'avoir des zones avec différents nombres des participants, en réalisant une meilleure utilisation des adresses multipoint tout en réduisant le traffic de signalisation

Les perspectives de recherche

A partir de nos travaux, nous pouvons envisager différentes perspectives de recherches comme:

- Des architectures de communication pour les EVGE qui combinent le multipoint et les réseaux pair-à-pair. Nous envisageons l'utilisation du pair-à-pair pour découvrir qui sont les voisins d'un participant dans un EV et d'employer le multipoint pour envoyer les flux de données sur des différentes canaux multipoint. Nous pensons à la nécessité de déployer une couche des agents avec le rôle des super-nIJuds dans des réseaux pair-à-pair (authentification, découverte rapide, robustesse etc...).
- Nous pensons également aux architectures de communication pour les EVGE qui emploient le routage multipoint en combination avec le multipoint au niveau applicative. Ceci permet aux utilisateurs qui n'ont pas le multipoint au niveau reseau de bénéficier de certains avantages du multipoint. Naturellement, ce n'est pas la même chose d'employer le multipoint au niveau d'application ou au niveau du routage, mais ceci peut contribuer à la déploiement successive du multipoint: les utilisateurs utilise au debout le multipoint au niveau application, plus tard ils peuvent commuter au multipoint au niveau routage.
- Des meilleurs heuristique et algorithmes peuvent être conçus pour le filtrage des participants entre différents types de flux de données entrants, envoyés par des différents

participants.

- L'utilisation du modèle SSM mène à de grandes tables de routage pour l'acheminement des paquets multipoint dans l'Internet. Le problème d'agrégation des entrées (S, G) dans les tables de routage est toujours un sujet de recherche. Les tables de routage des grandes taille augmentent les problèmes de scalabilité et les délais d'expédition des paquets, en raison des temps de recherche plus longs. En utilisant un système d'adressage facile à agréger, ont peut limiter les issues de scalabilité des environnements virtuels à grande échelle. Par exemple, ici nous pouvons utiliser une adresse unique de groupe multipoint pour tous les canaux audio d'une application, seulement l'adresse de source étant différente.
- L'utilisation du modèle SSM comme le modèle de distribution de données assume des pertes des paquets due à l'utilisation de l'UDP comme protocole de niveau transport pour le multipoint. Un mécanisme de fiabilité peu être intégré afin d'assurer la transmission des flux de données qui n'admettent pas des pertes.

Notre avis est que prochainement les environnements virtuels à grande échelle seront des applications communes au-dessus de l'Internet. Les réseaux et la disponibilité des capacités de traitement pour un grand nombre des utilisateurs dans l'Internet, mèneront au développement des nouveaux types d'applications au-dessus de l'Internet. Les nouvelles utilisations des environnements virtuels accompliront les efforts courants conduits particulièrement dans les jeux en réseau et dans le domaine de la formation militaire. Nous pouvons facilement imaginer un magasin virtuel de distribution de la musique et des films, où les utilisateurs peuvent marcher dedans comme dans un vrai magasin, où ils peuvent vérifier les étagères et ils ont la possibilité de visionner préalablement un "clip video" ou écouter une chanson et peut demander le conseil d'une vendeur ou des autres clients avant d'acheter les articles désirés. Concernant l'architecture de communication utilisée, nous croyons que chaque application de EV aura la possibilité de choisir, selon les types des flux des données utilise, le modèle de distribution de données désiré: pair-à-pair, multipoint, multipoint au niveau applicative etc...

Abstract

The subject of this thesis is situated at the confluence of the virtual environments (VE) and the IP networking. It is about the possible existence over the Internet, in the near future, of heterogeneous, large-scale, distributed virtual environments, inhabited by millions of entities.

The thesis presents the development, the implementation and the experimentation of a communication architecture for a large-scale VE application. The goal of this architecture is to allow a large number of participants, located in a VE, to communicate between them, using various data flows (video, audio, text). In the design of this architecture we have considered the following objectives: to minimize the traffic in the network, to allow a large number of participants to be simultaneously connected in the VE, to accept participants with very different capabilities, to give to each participant the possibility to communicate according to its capacities and to take into account the real-time constraints specific to the multimedia flows.

We have chosen to use the multicast, as the group communication model, in particular for the aspects of congestion control, in order to reduce the traffic in the network. A simplified model of multicast have been proposed by the IETF, the SSM model (or Specific Source multicast). We have chosen to use it in our architecture, because it it better suited to express the various preferences of the participants. Our architecture is composed by two filtering levels. On the first level of filtering, the VE is dynamically partitioned into zones, according to the density of participants in the world. On the second level of filtering, the participant determines its neighbors in the VE and begins to communicates with them, according to their capacities. A participant may transmit different types of data flows. Each flow is sent over a different SSM channel. The experiments have proved the feasibility and the performances of our architecture compared with other architectures suggested in the literature. We have integrated our communication module in V-Eye, a VE application developed in the Planete Research Team.

Keywords: Virtual Environments, Multicast, SSM, Filtering, Scalability, Multimedia, Heterogeneity, Architecture, Networks.

Résumé

Le sujet de la thèse est situe à la confluence du IP multicast et des environnements virtuelles. La thèse présente l'élaboration, l'implémentation et les expérimentation d'une architecture de communication pour une application d'environnement virtuel à grande échelle. Le but de cette architecture est de permettre à un grand nombre de participants, situés dans un monde virtuel, de communiquer entre eux (chaque participant communique seulement avec ses voisins), à travers différents flux de données (vidéo, audio, texte). Dans la conception de cette architecture nous avons considéré les objectifs suivants: minimiser le trafic dans le réseau, permettre à un très grand nombre de participants d'être connectés simultanément dans le monde virtuel, accepter des participants avec des capacités très différentes, donner à chacun d'entre eux la possibilité de communiquer selon ses capacités et des prendre en compte les contraintes des temps réel spécifique a des flux multimédia.

Nous utilisons le multipoint comme moyen de communication de groupe et en particulier les aspects contrôle de congestion afin de réduire le trafic dans le réseau. Un modèle simplifié de multipoint ayant été proposé à l'IETF (le modèle SSM ou Source Spécifique Multicast), nous avons choisi de l'utiliser dans notre architecture, à cause du fait qu'il était mieux adapté pour exprimer les différentes préférences des participants. Pour réalise la communication dans des mondes virtuels, nous avons défini une architecture compose de deux nivelles de filtrage des données. Au première niveau de filtrage, le monde virtuel est dynamiquement découpé en zones selon la densité de participants dans le monde. Un participant connaît tous les autres participants présents dans sa zone. Au deuxième niveau de filtrage, le participant calcule ses voisins dans le monde virtuel et commence a communique avec eux selon leur capacités. Un participant peut transmettre des différentes flux des données. Chaque flux est envoyé sur un canal SSM différent. Les expérimentes réalisé ont démontre la faisabilité et les performances de notre architecture compare aux autres architectures proposées dans la littérature. On a intègre le module de communication propose dans V-Eye, une application de monde virtuel développée par le projet Planète.

Mots-clés: Environnements Virtuels, Multipoint, SSM, Filtrage, Passage à l'echelle, Multimédia, Hétérogénéité, Architecture, Reseaux.