VELVET: An Adaptive Hybrid Architecture for VEry Large Virtual EnvironmenTs

Jauvane C. de Oliveira
Multimedia Communications Research Laboratory
School of Information Technology and Engineering
University of Ottawa
161 Louis Pasteur Priv., Ottawa, ON, K1N 6N5
CANADA

Nicolas D. Georganas
Multimedia Communications Research Laboratory
School of Information Technology and Engineering
University of Ottawa
161 Louis Pasteur Priv., Ottawa, ON, K1N 6N5
CANADA

***Abstract*-Collaborative Virtual Environment concepts have been used in many systems in the past few years. The architectures available today provide support for a number of users but they fail if too many users are together in a small "space" in the Virtual World. This paper introduces VELVET, an Adaptive Hybrid Architecture which allows a greater number of users to interact through a CVE. VELVET introduces a novel adaptive area of interest management, which supports heterogeneity amongst the various participants.**

I. INTRODUCTION

Over the past few years, a number of interactive Virtual Reality (VR) systems have been developed. A Collaborative Virtual Environment (CVE) is a special case of a VR system where the emphasis is more on collaboration between users rather than on simulation. CVEs are used for applications such as collaborative design, training [10], telepresence and tele-robotics and many more applications are showing up daily.

Many of the applications may have potentially a very large number of users at a time and that can easily overload a fast network as well as impose huge processing requirements at the stations. In fact, if no special mechanisms are provided, one may expect a simulation to produce undesirable effects such as choppy rendering, loss of interactivity, etc. Another problem is that of heterogeneity. That also imposes some limitations to a CVE as users in very powerful systems and fast networks would need to collaborate with others with very limited hardware and networking. It is obvious that the second type of systems should not be required to deal with the same load as the first. We have designed and implemented a number of CVEs for industrial training and electronic commerce [10]. Such CVEs, while allowing rich collaboration, did not handle a large number of users very well. With all users receiving updates from every other object in the virtual world, such situation does not scale well.

In this paper, we present VELVET, an Adaptive Hybrid Architecture for VEry Large Virtual EnvironmenTs. VELVET addresses the issues mentioned above, allowing a virtually unlimited number of users to participate in a CVE while allowing users with heterogeneous hardware and available networking to collaborate in a best effort approach.

II. RELATED WORK

A number of standards and prototypes have addressed the issue of allowing a larger number of users to collaborate through a CVE. In this section we introduce several of them, namely SIMNET, DIS, NPSNET-IV, SPLINE, MASSIVE-2 and SCORE, which somewhat represent others models as well.

A. DIS and SIMNET

SIMNET (Simulator Network) [7] has been one of the very first standards developed for military simulations. SIMNET has been developed to take full advantage of Ethernet hardware, with heavy utilization of broadcasting. This reduces software selection of packets, while limiting its use for a LAN scope. Dead reckoning is used to reduce communication requirements.

DIS (IEEE 1278 standard) is a standard [5] created as an improvement of SIMNET. DIS (Distributed Interactive Simulation) uses similar Protocol Data Units (PDU) as SIMNET, as well as its terminology and some of its functionality, such as Dead Reckoning. Dead Reckoning is implemented by the idea of player and ghost [7], where the original object is represented by a ghost in other stations. Such ghost tries to predict the behaviour of the player, reducing packet exchange, since only when there is an error greater than a pre-defined threshold, the player sends an update correcting the ghost in every station, at expense of extra processing requirements. DIS requires all objects to send periodic keep-alive update messages even for the objects which do not move or are "dead".

B. NPSNET-IV

NPSNET-IV [7, 8] is a prototype developed at the Department of computer Science at the Naval Postgraduate School in Monterrey, CA, USA in 1995. NPSNET IV was designed to comply with DIS 2.0.3. NPSNET IV included the use of IP Multicast with dynamic multicast groups reflecting a hexagonal partition of the Virtual World. This guarantees a constant number (three) of hexagons to be added and deleted when an object moves from one hexagon to an adjacent one. NPSNET IV also implements the player/ghost paradigm and, as DIS, NPSNET's communication PDU includes military oriented packets, such as Fire PDU, Detonate PDU and alike, which makes it somewhat unfit for civilian applications. Other than the hexagon partitioning, NPSNET-IV achieves better results, compared with DIS, mostly due to the lack of keep-alive heartbeat messages. Each hexagon has its own multicast group and data is sent through the multicast address of the hexagon where the source of such data flow is located.

C. Open Community and SPLINE

Open Community (OC) [1] is a proposal of a standard for multiuser enabling technologies from Mitsubishi Electric Research Laboratories. SPLINE (Scalable Platform for Large Interactive Networked Environments) is an implementation compliant with OC which provides development APIs. For its communication, SPLINE uses the Interactive Sharing Transfer Protocol (ISTP). SPLINE partitions the World Model in Locales which may have any shape, going one step further from NPSNET-IV. Once a user joins a given

Locale everything which is located in that Locale, as well as in the immediate neighbourhood, is visible.

D. MASSIVE-2

MASSIVE-2 (Model, Architecture and System for Spatial Interaction in Virtual Environments) [2,3] is a prototype developed at the Computer Science Department at the University of Nottingham, U.K in 1997. The major contribution of MASSIVE-2 is the introduction of the Third-Party Objects, which allows a hierarchical dynamic space-based embodiment of multicast groups [2, 3]. The idea behind third-party objects is to allow a group of artifacts (called crowd) to be represented as a unique object which is seen by others. Only when an artifact gets into a crowd boundary, it will receive information regarding individuals within the crowd. This model allows an elaborate hierarchy of groups, as a crowd may contain other crowds, recursively. Such an approach requires that media mixing be performed in order to provide a single audio channel, for instance, representative of the whole group.

E. SCORE

SCORE [6] has recently been introduced. It was developed at INRIA – Sophia Antipolis, France in 2000. It is based on the division of the World in Cells as suggested in [4]. A user interacts with those cells, which fall, at least partially, within an area of interest. The latter is defined as a square region around a user's avatar. Each cell has its own multicast group (MG) and an avatar then subscribes for that set of MGs. SCORE allows for two policies regarding determination of cell-size: pre-calculation of a fixed cell size and dynamic re-estimation of the cell-size during the session. The dynamic estimation may be performed based on some pre-defined parameters, such as number of MGs available, density of participants, etc. Furthermore SCORE allows the partitioning of the World to have cells of different size, which allows one to have a fine grid at highly populated areas and a sparse grid of cells in unpopulated areas.

III. THE PROPOSED VELVET ARCHITECTURE

VELVET (a Hybrid Adaptive Architecture for VEry Large Virtual EnvironmenTs) is our Adaptive Hybrid Architecture. The approaches previously described, while addressing the issues of a Large-Scale Virtual Environment (LSVE), they all fail under some circumstances. We will show that VELVET clearly performs better under the circumstances discussed in section A below.

A. Limitations of Existing Models

We may assume that SPLINE represents other space-based models such as NPSNET-IV, and others based on geographical partitioning of the Virtual World. In both cases the Virtual World is partitioned and each user is supposed to receive data from objects which are located in a well-defined subset of the partition (or *Locales* for SPLINE).

Locale based models assume that users are somewhat uniformly dispersed in the Virtual World. That is, the idea of reducing the amount of data which each station must deal with is addressed by reducing the area which is "seen" by each participant. That would assume that by reducing such area, the number of visible users would equally be reduced. If, however, most (or all) users are packed together in a small area of the CVE, the number of objects a given user must deal with may still be too large. Suppose we have a Virtual Museum where some dozens of thousands of users are visiting. If all of them decide to see "The Mona Lisa", one may notice that all stations would have to deal with all the dozen of thousands of data flows, as all of them would be in the same Locale. The Locale-based approach would hence fail in the task of reducing the amount of data each host must deal with in such a case.

Another limitation not quite addressed by existing architectures is that of heterogeneity. Let us consider a group of 300 hosts participating in a CVE. Yet, let us assume that 290 of such systems are quite powerful systems with very good networking connections. The remaining 10 stations, however, could be assumed to be weak enough not to be able to deal with the load. Space-based solutions which tend to assign an equivalent load to all systems populating the same neighbourhood. That would work well only if all systems are able to deal with the same load, which is not true in our example. The workaround in this case is that either all systems would have to meet a minimum or all systems would have to reduce data transmitted so that the weakest of the systems could support the load. Both solutions are somewhat inadequate as the first prevents some users from joining the CVE session while the second would under-use resources.

B. VELVET's Architecture

VELVET aims at allowing each and every user to interact with the Virtual World to the maximum extent possible (or optionally as much as paid for).

We will first introduce VELVET's terminology: World – The whole set of objects; Area or *Locale* – A subdivision of the World; Object – An object which is located within the World; Avatar – Special kind of object which represents a user; Bot – Active object which is not an Avatar; Artifact – An object which is neither Avatar nor Bot; Area of Interest (AoI) – Area a user is able to view and directly interact with; Check-In – Operation which brings an object into a user's AoI; Check-Out – Operation which removes an object from a user's AoI.

VELVET is a CVE architecture which allows real time adaptation, according to the local client needs. At any point in time a given user may elect to unilaterally reduce or increase his/her own view of the World, without prejudice to any other user, reason why VELVET gracefully supports heterogeneous collaboration.

1) *Area of Interest Management*: The idea behind VELVET is that each avatar will be able to "see" whatever is located within its AoI. The AoI of a given avatar does not depend on another avatar's AoI. Such behaviour allows for each station to manage how large its own AoI is, hence how much of the World can be seen at a time. The AoI can be enlarged and reduced dynamically so that upon increase in load, by a higher density of objects around an avatar for instance, one can automatically reduce the AoI so that the load can be kept within a treatable range. Only the objects which are within the AoI are visible and only information from those objects is received. Whenever the number of objects decreases, that same avatar may have its AoI expanded so that more objects may again be visible.

If we let B be the average throughput transmitted by a participant, P_A be the number of participants a user is aware of and P_I be the number of participants a user is actually interested on, we can express incoming traffic for Space based solutions as $B \times P_A$ and for VELVET as $B \times P_I$, where $0 \le P_I \le P_A$. A more formal description of AoI Management is shown in [9].

2) *Double Layered Boundary of VELVET's AoI*: Additionally, when an object crosses the border of the AoI it will Check-In (CI) or Check-Out (CO), depending on the direction being respectively towards the AoI or leaving the AoI. In order to

avoid multiple Check-In/Check-Out operations, VELVET in fact defines two borders named Area or Interest Check-In (AICI) and Area of Interest Check-Out (AICO), so that only objects crossing AICI will Check-In and those crossing AICO will Check-Out. The "distance" between AICI and AICO is also variable and may be used to control the number of CI/CO operations. Figure 1 A, B and C shows respectively an AoI with AICI and AICO with distance zero and two different distances between AICI and AICO. One can notice the arrows displaying 13, 7 and 5 CI/CO operations for the same path.

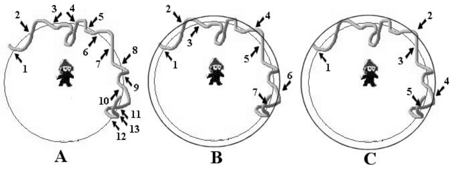

Fig. 1. VELVET's AICI/AICO.

As per the AoI's rule for expansion/shrinking we shall add that it is not necessarily based on virtual space (distance) from the avatar but rather based on a pre-defined metric in use by that user's VELVET management subsystem. That is, one can see what is more important to him/her rather than what is geometrically closer. Of course the metric itself could be that of virtual distance.

3) *Parallel Virtual World of VELVET*: The metric defines a Parallel Virtual World (PVW) for a given user, in which objects are placed according to the metric chosen by that participant. Figure 2 shows the PVW.

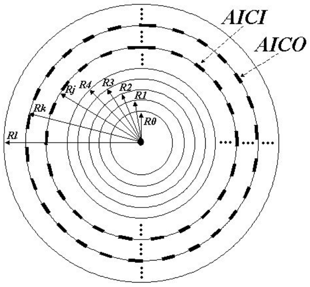

Fig. 2. VELVET's Parallel Virtual World.

The rings define levels in the metric oriented PVW. Each avatar has its own PVW and the management subsystem decides how many of the rings shown in Figure 2 will be subscribed for. Note that as each Avatar has its own PVW, the rings can be particularly arranged based on each participant's interest, for instance each ring may have a single object or a collection of those.

Let MS be a set of metrics, $MS = \{M_0, M_1, \ldots, M_m\}$, where,

M_0 = Metric 1, e.g. Number of Users;
M_1 = Metric 2, e.g. Network Traffic;
M_2 = Metric 4, e.g. Distance in Number of *Locale* Hops;
M_3 = Metric 7, e.g. A mix of the above, such as $M_2 \times 10000 + M_1$;
$\vdots$
M_m = Metric m+1, e.g. Others.

Assume that in VELVET, an avatar A_i has its own parallel virtual world (PVW$_i$) with a metric M_γ at a given time t.

$PVW_i\ (M_\gamma) = \{R_0, R_1, \ldots, R_{l-1}\}$, where $M_\gamma \in MS$, R_k is the (k+1)th level of the metric M_γ, $0 \le \gamma \le m$, and l is the number of levels in the current PVW_i, where R_{l-1} is the maximum level of M_γ.

The AoI will be such that it will include R_k, $0 < k < l$, so that $\sum_{n=0}^{j} \xi(R_n) \le T \le \sum_{n=0}^{k} \xi(R_n)$; where $\xi(R_n)$ is a function which gives the cost associated with the level R_n, e.g. the number of participants in level n for a metric considering the number of users. T is a value which will be optimized according to a pre-defined target value for a given metric. For instance, if one is behind a 56K modem connection and the metric considered is Total Network Traffic, the Target could be something like 48Kbps. T would be maximized considering that it must remain below such target.

4) *Degree of Blindness and Support for Heterogeneous Systems of VELVET*: Since participants unilaterally decide which objects to subscribe for, based on the PVW, one can notice that such behaviour can lead to inconsistencies. Such inconsistency shows up when a given avatar A "sees" an avatar B even though B can't "see" A. That would happen if A's AoI is expanded enough so that B is enclosed while B's AoI is shrunk enough not to enclose A. This is called in VELVET's terminology "degree of blindness", which limits one's vision. That is perfectly legal in VELVET. In fact it is the very reason why VELVET graciously supports collaboration among users in heterogeneous systems in a best effort approach. The smaller the AoI the greater is the Degree of Blindness. The Figure 3 shows users A and B adopting similar metrics, the area around each avatar is that based on the PVW rather than the Euclidean space.

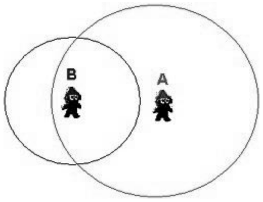

Fig. 3. VELVET and Degree of Blindness.

Define $\partial(A_i, A_j) \equiv \partial_{ij}$ as the distance between participant A_i and A_j in the PVW of participant A_i and ρ_i the radius of the AoI in participant A_i's PVW, as shown in Figure 4. We can define that $A_i \subset A_j \Leftrightarrow \partial_{ji} \le \rho_j$, i.e. if A_i is within A_j's Area of Interest. Similarly we can define that $A_i \supset A_j \Leftrightarrow \partial_{ij} \le \rho_i$. In other words $A_i \supset A_j \Leftrightarrow A_j$ is within the A_i's AoI, i.e. if A_i can "see" A_j. Considering space based solutions, for two avatars A_i and A_j within the same Local L_n, $A_i \supset A_j \Leftrightarrow A_j \supset A_i$ holds, because $\rho_i = \rho_j; \forall i, j$ and $\partial_{ij} = \partial_{ji}; \forall i, j$. In VELVET $\partial_{ij} = \partial_{ji}$ may not necessarily hold, as after all A_i and A_j may be using completely different metrics. Furthermore in VELVET $A_i \supset A_j$ does not lead to $A_j \supset A_i$ because the metrics can be different, and, in the event that both participants use the same metric, $\rho_i \ne \rho_j$ may hold as well, which is the reason for the existence of different Degree of Blindness for each user, which allows heterogeneous systems to collaborate in a best effort approach, i.e. if one is participating in a VELVET session with a processor weak system or behind a dial-up 56k modem, it would still be possible to interact with a limited number of participants (high degree of blindness), while other user in a supercomputer with a very fast networking connection would be able to interact with a comprehensive view of the World (if desired).

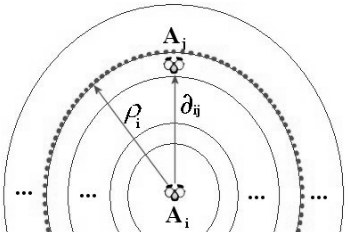

Fig. 4. Parameters ∂ and ρ.

C. Internal Structures and Functionality of VELVET

In VELVET, the World is partitioned into Areas and each Area has a multicast address like in SPLINE and NPSNET-IV. VELVET accomplishes the flexible functionality described in the previous section by assigning a multicast group for each object which generates a flow of data (such as avatars and bots). That defines an Object Transmission Channel (OTC). Each client has three threads running in parallel. The first thread is the one responsible for sending data through the network, the second receives and acts upon arriving packets while a third thread performs management of AoI, joining and leaving Areas and OTCs as appropriated. Each *Locale* has similarly its own *Locale* Transmission Channel (LTC).

1) *Data Transmission Control*: Each object is supposed to send data only on its own OTC and only those who have explicitly signed for such channel will receive such data. That ensures that no host will ever receive unsolicited user data. Moreover each VELVET system can pinpoint exactly which users should be receiving data from, based on a given metric. Table I shows the amount of superfluous data received by participants in various architectures.

TABLE I
SUPERFLUOUS DATA

MASSIVE-2	From objects in the appropriated areas, which are of no interest.
SPLINE NPSNET-IV	From objects in the appropriated areas, which are of no interest.
SCORE	From objects in the appropriated areas, which are of no interest, as well as objects which, even tough not in the Area of Interest, are located in cells which fall partially within the AoI.
VELVET	Minimum superfluous data, based on metric.

VELVET is adaptive because each system may choose to sign for a larger or smaller number of groups on the fly. For instance, if a given system is connected to a VELVET World and experiences network overload, it may simply unilaterally shrink its own AoI (reducing its parameter *T*), which immediately reduces the flow of data arriving at that end.

IV. MODELING AND SIMULATION

VELVET has been modeled using OPNET Modeler 6.0 PL12. In such modeling, we created a multicast enabled router which allowed VELVET to perform exactly as described here and detailed on [9]. Several simulations were run and different profiles were chosen.

In order to make the setup of the simulation more convenient, we have introduced a DHCP server in the simulation. This allows easy reconfiguration since it is possible to add many stations to the simulation with no need for manual configuration. This DHCP setup leads to unnaturally high packet traffic at startup, but such high traffic only occurs only at startup. In the simulations we had a station with Space Based AoI Management, which was further compared with those using VELVET.

Figure 5 shows results from 13 stations in a 2 *Locales* World, where SPLINE would lead to rendering of all users located within the World, while VELVET would allow filtering as described above. For this simulation all stations were sending packets according to an exponential distribution with a mean outcome of 0.125 seconds (average of 8 update packets per second). The probability of an avatar changing *Locale* was set to 15%.

The station behaving like SPLINE matches with that of the VELVET station receiving information from all hosts, averaging 115 packets per second. VELVET allows for filtering even within a single *Locale*, since the World is seen through the PVW for the AoI Management protocol of VELVET. If a given avatar chooses to expand both AICI and AICO so that they would coincide with the last level known, then all objects would check into the AoI of the avatar. More formally, if J=K=L so that $\partial_{JK} = 0 \Rightarrow \partial_{i\alpha} \leq \rho_J, \forall\alpha$, for an avatar A_i. In this special case, the avatar A_i in VELVET would receive packets from every object as well. For the other avatars different values of ρ_J were chosen, leading to the various levels shown.

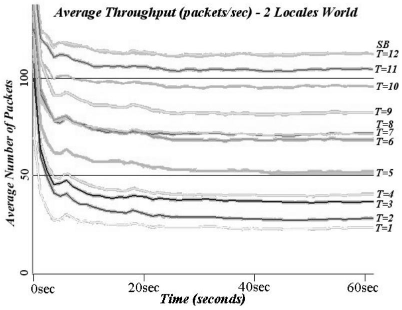

Fig. 5. Average Incoming Traffic – Packets/Second.
VELVET (12 lines) vs. Space Based Solutions (upper line)

Figure 6 shows a better comparison of VELVET vs. Space Based Solutions, using data gathered from simulation results. In this graph we selected the VELVET station with the highest Incoming Throughput (i.e. that with the AoI more expanded amongst the various VELVET stations, in Figure 3 for instance). We then run the simulation increasing steadily the number of users populating the World. We can see that this VELVET station keeps a relatively stable Average Incoming packet count while Space Based Solutions grow steadily.

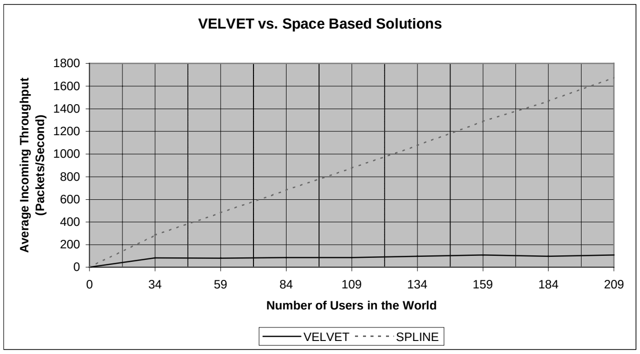

Fig. 6. Comparison of VELVET vs. Space Based Solutions.

These simulation results show how VELVET behaves when compared with Space Based Architectures. Such results are somewhat obvious since VELVET can behave exactly like SPLINE as well as allow a more aggressive filtering of incoming data and hence reducing incoming traffic. It is important to mention that a VELVET station could have its AoI expanded enough so that it could achieve the same packet count shown above for Space Based Solutions. One advantage of VELVET is exactly this unilateral flexibility.

V. CONCLUSION

We have presented VELVET: an Adaptive Hybrid Architecture for VEry Large Virtual EnvironmenTs. VELVET has shown to be flexible, allowing a broad range of LSVE which would otherwise fail, to work gracefully.

One disadvantage over other architectures is that VELVET makes use of a potentially larger number of Multicast Addresses which leads to a potentially large number of entries in routing tables. More specifically, VELVET uses a Multicast Group (MG) for each Area as well as for each participant, hence the cost regarding MGs in VELVET has an upper bound O(M+P) with M being the number of Areas in the Word (*Locales*) and P the number of participants. SPLINE's lower bound MG is O(M) since each *Locale* has a multicast group. MASSIVE-2 has an upper bound at O(M+P) as well when considering the case where every couple of objects creates a third party object. SCORE has a lower bound MG usage at O(C), where C is the number of cells, which is much larger than the number of Areas (*Locales*) M.

It is worth mentioning that even though VELVET has an O(M+ P) number of MGs, each router only needs to deal with those subscribed for stations whose traffic goes through it. Part of the metric of a given station could also include some cost measurements of the load in routers in the neighbourhood, in which case VELVET would drop MGs if some routers would be on their limit. The adaptability of VELVET and asymmetric presentation of the World for the various participants allows such features without much overhead, since all changes can be performed unilaterally by each participant.

It is also worthy of mention that as technology evolves routers get faster and more powerful and such limitations tends to be diminished as time goes by. Regarding the number of multicast addresses available, IPv6 is increasing their number to the same range of the total IPv4 unicast addresses available today.

ACKNOWLEDGMENTS

We acknowledge the financial assistance of the Brazilian Ministry of Education Agency's CAPES scholarship and the Ontario Research and Development Challenge Fund. We also thank Dr. Seok Jong Yu, for all the input and helpful discussions about this paper and Mojtaba Hosseini, for reviewing this text.

REFERENCES

[1] J. W. Barrus, R. C. Waters & D. B. Anderson, "Locales: Supporting Large Multi-User Virtual Environments," *IEEE Computer Graphics and Applications*, 16(6), 50-57, 1996.

[2] C. Greenhalgh, "Large Scale Collaborative Virtual Environments," *Ph.D. Thesis, Computer Science Department, University of Nottingham, UK*, 1997

[3] C. Greenhalgh, & S. Benford, "A Multicast Network Architecture for Large Scale Collaborative Virtual Environments," *Multimedia Aplications, Services and Techniques – ECMAST'97, Lecture Notes in Computer Science*, Vol. 1202, Springer-Verlag, 1997.

[4] D. J. Hook, S. J. Rak, & J. O. Calvin, "Approaches to Relevance Filtering". *Proceedings of 11th DIS Workshop*, 1994.

[5] Institute of Electrical and Electronics Engineers, International Standard, ANSI/IEEE Standard 1278-1993, Standard for Information Technology, Protocols for Distributed Interactive Simulation, March 1993.

[6] E. Lety, "Une Architecture de Comunication pour Environnements Virtuels Distribues à Grande-Echelle sur l'Internet," *Ph.D. Thesis, L'Universite de Nice-Sophia Antipolis, France*, December 2000.

[7] M. Macedonia, "A Network Software Architecture for Large-Scale Virtual Environments". *Ph.D. Thesis, Computer Science Department, Naval Postgraduate School, Monterey, CA, USA*, 1995.

[8] M. Macedonia, M. Zyda, D. Pratt, P. Barham & S. Zeswitz, "NPSNET: A Network Software Architecture for Large Scale Virtual Environments". *Presence*, 3(4), 265-287, 1994.

[9] J. C. Oliveira, & N. D. Georganas, "VELVET – An Adaptive Hybrid Architecture for Large Scale Virtual Environments", *Submitted for publication*.

[10] J. C. Oliveira, X. Shen, & N. D. Georganas, "Collaborative Virtual Environment for Industrial Training and e-Commerce," *IEEE VRTS'2000 (Globecom'2000 Workshop), San Francisco, CA, USA.*

[11] S. Singhal, "Effective Remote Modeling in Large-Scale Distributed Simulation and Visualization Environments," *Ph.D. Thesis, Computer Science Department, Stanford University, CA, USA*, 1996.

[12] S. Singhal, & M. Zyda, "Networked Virtual Environments, Design and Implementation," *ACM Press/Addison Wesley*, 1999.