Dynamic Grid-Based Interest Management for Distributed Virtual Environments

Ali Berrached
Department of Computer and Mathematical Sciences
University of Houston-Downtown
Houston, Texas 77002
Berracheda@uhd.edu

ABSTRACT
In large-scale distributed virtual environments, simulated entities maintain a consistent virtual world by exchanging messages about their state information in a timely fashion. The problem of Interest Management (IM) or, as it sometimes appears in the literature, Data Distribution Management (DDM), is the problem of delivering to each entity only the data updates that it absolutely requires to maintain a consistent view of the virtual world. Efficient and effective interest management is crucial to the performance and scalability of distributed virtual environments. Several approaches have appeared in the literature to address the interest management problem. In this paper we focus mainly on Grid-based interest management in the context of the High-level Architecture (HLA) standard and develop a model that dynamically adapts the grid layout to match the changing distribution of entities in the virtual space. We also discuss a hierarchical approach that exploits the structure of the underlying network to further enhance the system’s traffic filtering capability.

Keywords: Interest management, data distribution management, distributed virtual environments, High-Level Architecture.

1. INTRODUCTION
A Distributed Virtual Environment (DVE) consists of a set of virtual entities sharing a common, consistent virtual world. DVEs are employed in numerous applications including large-scale distributed systems [2, 3, 6, 12], virtual reality systems [1, 9], and multi-player networked games [5]. A key requirement of such systems is that the virtual entities share a consistent view of the simulated world. This requirement entails that the simulated entities dynamically exchange information about their state in a timely fashion. A state update occurring to a given entity is likely to have immediate significance to a whole set of other entities. Data sharing and management, across a large-scale distributed environment, requires message passing between the nodes running the simulation. A simple broadcast based scheme whereby every entity broadcasts its state updates to all other entities in the virtual world would result in an overwhelming amount of data traffic, much of it not needed by the receiving entities. For example, in a simulated war exercise, a tank is capable of “seeing” (i.e. detecting signals from) only objects that are within its radar detection range. Information received from other simulated entities in the virtual world that are outside the tank’s radar detection range are irrelevant to it and must be discarded. The process of discarding or filtering out irrelevant data traffic is at the core of the problem of Interest Management. The simple broadcast approach described above, which was used in the DIS (Distributed Simulation Systems) standard, is obviously not adequate for large-scale distributed simulation systems as it does not have any mechanisms for reducing the amount of irrelevant data traffic, and therefore is not scalable [2].

The High-Level Architecture (HLA), which is the current IEEE standard for large scale distributed simulation systems [4], provides mechanisms that facilitate the implementation of irrelevant data filtering. In effect, the HLA provides a group of services, called Data Distribution Management (DDM) services for the sole purpose of implementing interest management algorithms.

The remainder of the paper is organized as follows: the next section gives an overview of the Data Distribution Management services of the HLA. Section 3 discusses the fixed grid-based filtering approach. Section 4 describes a dynamic grid-based approach and discusses its performance merits, Section 5 introduces a hierarchical grid-based scheme and section 6 concludes the paper and identifies areas for future research.

2. INTEREST MANAGEMENT IN THE HLA
The objective of the Data Distribution Management services group in the HLA is to reduce the amount of irrelevant data updates that are exchanged among HLA federates (in HLA terminology, a federate is a simulated entity). The main concept to support data distribution is based on the so called, routing spaces [12, 6]. A routing space is a multidimensional coordinate system in which federates express their interest for either sending or receiving data. The axes of a routing space define the domains of the simulation data, such as object attributes or interactions, in which federates intend to exchange data. The need for DDM stems from the fact that, in general, a simulation object is interested in receiving a data update only if the data value is within a certain range of its domain. Federates express their interests to send or receive...
data by posting their update or subscription regions to the Run Time Infrastructure (RTI). A subscription region defines a subset of a routing space in which a federate wishes to receive data updates. An update region defines a subset of the routing space in which a federate offers to publish data updates. When an update region and a subscription region overlap, a connection is established by the RTI and data updates are routed from the publisher of the data update to the subscriber to the data update. The data distribution services, therefore, provide a mechanism for federates to send attribute and interaction updates only to those federates that have expressed interest in the updates being published. Detailed information on various aspects of DDM can be found in any of [4, 6, 11].

Current implementations of DDM are based on using the multicast services provided by the underlying network to route data from a publisher to the group of federates that have expressed interest in receiving the data. Multicast addressing makes it easier for a sender process to send a message to multiple recipients. It does this by shifting the burden from the sender application process to the lower layers of the underlying network architecture which take the responsibility of creating and maintaining the multicast groups, determining the various multicast routes, updating the routing tables across the network as group memberships change etc. The overhead associated with multicasting is effected by four main factors: (1) the number of multicast groups in use, (2) the number of active members, (3) the rate at which members join and leave multicast groups and (4) the size and structure of the network (e.g. LAN or WAN).

A straightforward approach to using multicast addressing in the context of DDM is to associate a multicast group with each update region and join all the federates whose subscription regions overlap the update region. This straightforward approach, however, is only adequate for small simulations of few hundred entities (at best). For larger applications, say of the order of hundred of thousands entities, the number of multicast groups required would be very large and the overhead associated with membership changes and other multicast maintenance operations would be tremendous. In the context of large scale distributed systems, the number of multicast groups is a scarce resource that needs to be utilized efficiently.

3. FIXED GRID-BASED SCHEME

An implemented approach to the multicast allocation problem for data distribution is the fixed grid-based approach [3, 11]. In the fixed grid-based approach, each routing space is subdivided into a predefined array of fixed size cells. Each grid cell is assigned a multicast group. A federate joins those multicast groups whose associated grid cells overlap the federate subscription regions. When a federate wishes to send a data update, the update is delivered only to those multicast groups whose associated grid cells overlap the sender’s update region. The idea here is that if an update region and a subscription region intersect, they must both overlap at least one grid cell in common. The multicast group(s) associated with the overlapped cell(s) serve to route the data updates from publishing to subscribing federates.

The fixed grid approach is relatively simple to implement and incurs little overhead. Its main feature is that no interaction between federates is needed to determine the required connectivity between senders and receivers. This is due to the fact that the grid is predefined and the regions occupied by the grid cells are regular (i.e. d-dimensional rectangles) and remain unchanged at run time. Matching of subscription and update regions to grid cells is therefore done by each federate locally.

Despite those features, the fixed grid-based approach suffers from a number of shortcomings. First, it does not filter out all irrelevant data. Irrelevant data can be delivered if a subscription and update regions overlap the same grid cell but do not themselves intersect. This situation is illustrated in Figure 1. The figure depicts a 2-dimensional routing space subdivided into a 4 by 4 grid and two regions, S and U, which represent the subscription and update regions of federates F1 and F2, respectively. Note that S and U do not intersect and therefore data updates from F2 should not be delivered to F1. However, because both regions overlap cell 9, any data update from F2 will be delivered to any subscriber to the multicast group associated with cell 9, including F1. Irrelevant data delivery can be reduced by reducing the size of the grid cells. However, this would increase the number of multicast group usage and its associated overhead.

![Figure 1. Irrelevant data is delivered from F2 to F1 even though S and U do not intersect.](image-url)
regions of the routing space and insufficient multicast
groups in the more heavily populated regions.

4. DYNAMIC GRID BASED SCHEME

4.1 Description

The fixed-grid approach is based on uniformly allocating
multicast groups across a routing space. In typical
simulations, however, objects attribute values tend to be
“clustered” in small regions of a routing space. This is
because objects, in general, do not act independently of
one another but tend to operate in a predefined structured
fashion. In a military setting, for instance, troops are
organized in companies, platoons, etc. and they tend to
behave, with their associated equipment (tanks, helicopters, etc), in a collective fashion.

The objective of the dynamic grid approach is to capture
this clustering characteristic. It is based on recursively
subdividing a d-dimensional routing space into subspaces
(grid cells), with each subspace being divided in half along
each of its d dimensions. The subdivision continues
recursively until the maximum number of subspaces
(cells) is reached. The result is a grid with variable size
cells which can be represented with a 2^d-tree structure
[11]. As for the uniform grid, each grid cell is assigned a
multicast address through which data updates are routed
from publishers to subscribers whose update/subscription
regions intersect with the grid cell. Figure 2 shows a
dynamic grid layout for a two dimensional routing space.
A number of approaches can be used to guide the
decomposition process of the grid. The grid layout in
Figure 2 was obtained by recursively subdividing the
quadrant with the largest “population” (i.e. largest number
of update regions) in four sub-quadrants. The result is a
grid layout with small grid cells in highly “populated”
areas and larger cells in sparsely populated ones.

As objects attribute values change during the course of a
simulation, the grid layout needs to be reconfigured
dynamically and redistributed to the federates. Dynamic
grid re-configuration incurs extra overhead and should,
therefore, be done sporadically, either periodically or on
demand, based on some measure of attribute values
distribution that can be tracked by the RTI.

This approach inherits some of the simplicity and the
scalability of the fixed grid approach: matching
subscription to update regions is done implicitly through
the grid. By providing each federate with a copy of the
grid layout (i.e. its tree representation), matching of
subscription regions to update regions can be done locally
by each federate without any interaction with other
federates. However, as mentioned above, dynamic grid
reconfiguration incurs extra overhead as the new grid
layout needs to be distributed to the federates each time it
is reconfigured.

Like all grid-based approaches, the dynamic-grid based
scheme does not guarantee total elimination of irrelevant
traffic. However, since the grid layout is configured to
match the attribute values distribution in the routing space,
it has the potential of achieving more accurate filtering. In
the following section, we present a quantitative comparison
between the fixed-grid and the dynamic grid approaches.

4.2 Evaluation of the Dynamic-Grid

A detailed simulation program was developed to evaluate
the performance of various HLA based filtering
approaches. The program simulates a location-based two-
dimensional routing space (x and y coordinates) and allows
for a number of simulations parameters to be varied;
including the extents of the routing space, the number of
simulated objects, the number of simulation sites, the size
of subscription regions, the number of grid cells, and
various other parameters associated with objects
movements (such as maximum speed and trajectories).

The results presented in this paper were obtained for a
simulation with 10000 objects (ground vehicles) on a
100x100 kilometers terrain. Subscription regions were set
to 2x2 kilometers rectangular regions and objects
maximum speed was set to 150 m/s. Certain simplifying
assumptions were made. We assumed that each federate
simulates exactly one entity and that each entity has exactly
one subscription region and one update region. The update
regions were assumed to be point update regions
represented by the entity’s current location.

Figure 3 compares the performance of the dynamic grid
approach to that of the fixed-grid approach. To simulated
the clustering effects, we assumed 9000 of the 10000
objects to be organized in 100 clusters, with each cluster
spread over a 2km x 2km area. Objects in each cluster
move along the same trajectory and at the same speed. The
remaining 1000 objects were assumed to be spread
randomly over the whole terrain. Both simulators were run
for the same set of scenarios for a duration of 180 time
units. The dynamic grid was reconfigured every 60 time
units.

Figure 3 plots the total number of irrelevant data updates as
a function of the number of multicast groups allocated to
the filter. As expected, the effectiveness of the filter, under both approaches, increases as the number of multicast groups is increased. However, with equal number of multicast groups, the dynamic grid generates much less irrelevant data updates than does the fixed grid. This is especially the case when the number of allocated multicast groups is relatively small (25 to 400). This occurs because the dynamic grid is able to allocate its multicast groups more efficiently across the routing space, which results in smaller cell sizes in highly populated areas where most of the traffic occurs. For the simulation studied, the dynamic grid generated, on average, 59.7% less irrelevant traffic than does the fixed grid. These results were obtained with the grid reconfigured every third of the simulation time (every 60 time units of a total of 180 time units).

![Figure 3. Performance comparison of fixed and dynamic grids](image-url)

5. A HIERARCHICAL GRID APPROACH

The limit on the number of multicast groups and their associated maintenance overhead is most critical at the wide area network level. At the local area network level, multicasting is much more efficient. In effect, multi-access local area networks (e.g. Ethernet or token bus based LANs) provide hardware support for multicasting. And though multicast group membership maintenance incurs some overhead, it is not nearly as significant as it is in the case of wide area networks. This property can be used to provide a hierarchy of grid-based filters with increasing levels of granularity at successive levels of the network.

Figure 4 shows a two-level filter in a network configuration with three local sites. A finer grid is used at the local network level and a coarse grid at the wide area network level. Filtering agents are used at the gateways to implement filtering at the wide area network level. Each gateway filtering agent keeps a record of the subscription regions of all the federates in its site. The subscription region of a gateway is the union of the subscription regions of the federates in its site. At the wide area network level, each higher-level grid cell is assigned a wide-area network multicast group. A gateway joins those WAN multicast groups whose associated higher-level grid cells overlap its subscription region. At the local level, federates are only aware of the lower-level (finer) grid. An attribute/interaction update is sent to the local multicast groups whose associated lower-level grid cells overlap with the update region of the publishing federate. In addition, a copy of the update and the update region is also forwarded to the local site gateway filtering agent. Using the high-level grid, the gateway agent sends the update to the WAN multicast groups whose associated higher-level cells overlap with the update region of the publishing federate.

When an update is received from a remote site, the local gateway filtering agent matches the update with the lower-level grid to find out those local federates that are interested in the update. The update is delivered to those federates that are members of the multicast groups associated with the lower-level cells that overlap the update region.

In the example shown in Figure 4, attribute and interaction updates from federate F1 are routed within the local network of site 1 only to federate F2 since the subscription region of F2 and the update region of F1 overlap the same cell of the low-level grid. A copy of the update is also sent to the gateway agent of site 1 which takes the responsibility...
of delivering it to interested gateways on the WAN using the high-level grid. In the example of Figure 4, the update is sent to multicast group 6 of which only site 3 is a member. Finally, the gateway agent at site 3 uses its low-level grid to route the update to the interested federates in its local network.

It is clear from the above description that the lower-level grid provides a mechanism to filter out irrelevant data that pass through the higher-level grid. More effective filtering is therefore achieved (from the subscriber federate perspective) without increasing the number of multicast groups at the wide area network level (where they are most costly).

The two-level scheme described above can be extended to any number of levels depending on the network structure. This allows filtering to be done in a sequence of stages with increasing levels of “accuracy” as data moves in the network hierarchy from the sender to the receiver. Different grid granularities and grid layouts can also be used at different levels of the network hierarchy and within each sub-network, depending on their specific needs and capabilities. The gateway filtering agents act as interfaces between the different levels of the hierarchy and have the responsibility of ‘ironing out’ their differences.

6. CONCLUSION

The selection of an appropriate data distribution scheme is crucial to the scalability of large scale distributed virtual environments. Focusing on grid-based data distribution management with the HLA context, we presented a dynamic-grid scheme that allows grid layout to be configured to match the distribution of objects attribute values in the routing space. Our simulation studies have shown that this approach achieves better traffic filtering and therefore better system scalability. We also presented a hierarchical approach that allows filtering to be done in a sequence of stages with increasing levels of accuracy as data moves from sender to receiver. This is especially useful for large scale distributed simulations where it is very difficult to design one filter that can respond to the needs and characteristics of the numerous entities being modeled in the virtual world. The hierarchical approach provides a framework for partitioning a large federation into a hierarchy of smaller “sub-federations”. Different filtering schemes can then be used in each of the “sub-federations”, depending on their needs and capabilities. Though the paper presented a filter hierarchy that is based solely on the underlying network hierarchy, other factors may also be taken into consideration in the partition process.

Throughout this study, we made the assumption that message delivery is done through the IP multicast system provided by the underlying network. Though most HLA implementations have traditionally relied on IP multicast, some studies have shown that its deployment has not been as efficient as expected [10]. Our plans for future research include investigating other mechanism for message delivery, including the use of range queries and associative memory [7, 8] and how it can be effectively used with grid-based interest management.

7. REFERENCES