International Journal of Computer Science and Applications © 2009 Technomathematics Research Foundation
Vol.6 No.1, pp.70-84

APPLICATION LAYER MULTICAST FOR EFFICIENT GRID FILE TRANSFER *

Rafael Moreno-Vozmediano
Dept. Arquitectura de Computadores y Automatica, Universidad Complutense de Madrid.
28040 Madrid, SPAIN
rmoreno@dacya.ucm.es
http://dsa − research.org

Multicast communication model can be efficiently exploited by grid systems to improve the performance of many basic operations, like file transmission, replica management or resource discovery. However, IP multicast presents important limitations to be adopted in grid environments, since it is not fully supported in the Internet and is incompatible with TCP-based applications. In the last few years, Application Layer Multicast (ALM) have emerged as an alternative technology to develop one-to-many and many-to-many applications, where multicast functionality is handled by the end hosts, instead of by the network routers. While ALM has been proved to be efficient for streaming audio/video, content distribution, or videoconferencing applications, its utilization in grid environments has not been deeply explored. In this paper we analyze the application of ALM techniques to improve the efficiency of file transfer operations in computational grids. Results will prove that ALM can obtain an important reduction of overall transmission time, in comparison with unicast transmission, and a better balance of network traffic among participants.

Keywords: application layer multicast; grid computing; overlay networks.

1. Introduction

Although today's grid technologies can be considered quite mature, however, to be fully adopted in business environments there are still several barriers to overcome, mainly related to data management, security, and licensing [Hammerle (2007)]. Regarding to data management and performance, a major concern is the pressure that many grid applications exert over the underlying communication network, specially those involving bulk data transfers to multiple destinations, which can result in a great demand of network bandwidth and long transmission latencies. In this situation, it is essential to exploit the features, protocols, and services provided by modern communication networks in an efficient way, in order to optimize the

*This research was supported by Consejería de Educación of Comunidad de Madrid, Fondo Europeo de Desarrollo Regional (FEDER) and Fondo Social Europeo (FSE) through BioGridNet Research Program S-0505/TIC/000101; by Ministerio de Educación y Ciencia through research grant TIN2006-02806; and European Union through the research project RESERVOIR Contract Number 215605.

bandwidth consumption and reduce delays. In particular, multicast communication techniques can be used to improve the efficiency of this sort of data transfers in grid environments.

While most traditional Internet applications (like Web, e-mail, FTP, etc.) are based on unicast communication paradigm (one-to-one), over the last few years there are increasing demands of streaming, interactive, and real-time applications, which can benefit from multicast communication models, since they enable the development of efficient one-to-many applications (scheduled audio/video distribution, file and content distribution, push media, etc.) and many-to-many applications (multiparty videoconferencing, multiplayer on-line games, jam sessions, chat rooms, etc.). However, although IP multicast protocols were developed more than one decade ago, the deployment of these protocols in the Internet is very limited, since most ISPs have not support for multicast. In this context, Application Layer Multicast (ALM) is emerging as an alternative to IP layer multicast [Zhang et al. (2002); Francis (2000); Banarjee et al. (2002); Mathy et al. (2001); Chu et al. (2002); Wang et al. (2002); Chawathe (2003); Pendarakis et al. (2001)]. In ALM, multicast routing functionality and packet replication is handled by the end hosts, instead of by the core network routers. The systems belonging to a given multicast group form an overlay network, where each link corresponds to an unicast path between two end systems in the underlying network. Every packet is forwarded from one system to another using unicast transmission trough the overlay network links, following some specific routing rules, until it is received by all the members of the multicast group.

Grid environments can also benefit from multicast transmission [Moreno (2007); Kouvatsos and Mkwawa (2003); Ranaldo et al. (2005); Barcellos et al. (2005)]. Resource and service discovery, file transfer, or replica management are examples of basic operation in a grid that can efficiently exploit multicast communication paradigm, to reduce bandwidth stress and transmission delays. However, the use of IP multicast in grids presents two major drawbacks. First, a grid system is composed by the aggregation of geographically distributed heterogeneous resources, owned by different individuals and/or organizations, which use the Internet as interconnection network. As we mentioned before, Internet does not fully support multicast, so even if the different organizations that conform the grid use IP multicast-compatible networks, they can be seen as multicast islands connected through the Internet unicast backbone. Although tunnelling is a solution to overcome this problem, however it is not very appropriate for dynamic environments like grids. The second problem is that basic operations in a grid, like resource/service discovery, job submission, file transference, replica management, security and authentication procedures, etc., are mostly implemented by reliable TCP-based services, which are incompatible with IP multicast communication mode. There are some works that analyze and propose reliable multicast transport protocols for grid environments [Barcellos et al. (2005); Jeacle et al. (2005)], but they involve changes in the standard TCP/IP stack, which

should be adopted by all the participant systems to guarantee full interoperability.

Regarding the limitations of IP multicast, ALM stands out as an alternative to implement multicast in grid environments. ALM is not dependent on the multicast support of the underlying network, so it can be implemented in the Internet. With ALM, data are delivered using unicast paths between hosts through the overlay network, so it is fully compatible with any TCP-based service. Furthermore, ALM does not need any changes in the network infrastructure or protocol stack.

The goal of this paper is to analyze the use of ALM for improving the efficiency of file transfers in computational grids, using a minimum diameter overlay tree for data delivery. It is organized as follows: Section 2 summarizes the fundamentals and state of the art of ALM protocols; in Section 3 we analyze some basics operation in a grid that can improve efficiency using multicast transmission; in Section 4 we propose a method for constructing the overlay delivery tree in the grid; Section 5 shows the results of comparing unicast and ALM for different size file transfers in a grid; finally, Section 6 outlines the main conclusions and future work.

2. Application Layer Multicast

2.1. *ALM fundamentals*

Fig. 1 shows the basic idea behind application layer multicast techniques, in comparison with IP multicast. While IP multicast is implemented by the network routers, which are responsible for packet replication and forwarding, ALM is implemented by the application nodes (end systems), which form a logical overlay (tree) structure on top of the network layer infrastructure, and use the existing unicast paths of this underlying network to replicate and forward the data packets.

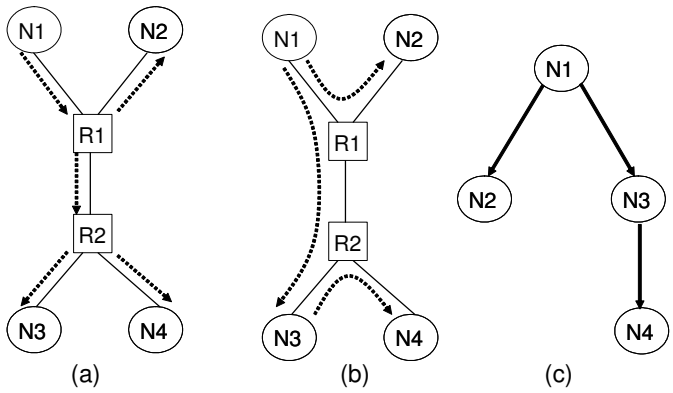

Fig. 1. a) IP layer multicast; b) application layer multicast; c) overlay tree

It is obvious that ALM is less efficient than IP multicast. While routers in IP

multicast try to avoid multiple copies of the same packet over the same link, by constructing optimal multicast trees, in ALM packets may traverse the same link several times. Furthermore, end hosts do not have detailed information about routing or network topology, so, they usually work with limited topology information, based on end-to-end measurements, like round-trip time (RTT) or end-to-end bandwidth, to construct the overlay tree.

2.2. *Related work*

Most authors classify ALM protocols in tree-first and mesh-first approaches [Hosseini et al. (2007); Janic (2005); Tan et al. (2006)]. In tree-first protocols, members of the multicast group are directly arranged in a loop-free tree structure, rooted on the source node. In this tree, data are distributed by flooding: each node receives data packets from its parent node, and replicates them to children nodes in the tree. Examples of tree-first protocols are Yoid [Francis (2000)], NICE [Banarjee et al. (2002)], HMTP [Zhang et al. (2002); Zhang et al. (2006)], and TBCP [Mathy et al. (2001)], among others. Mesh-first protocols usually take a two-step procedure to construct the delivery tree. First, members of the multicast group are organized in a richly connected mesh structure with multiple paths between node pairs. In the second step, each node executes a routing algorithm to create a source-specific delivery tree, which establishes a single path to every node in the group. Examples of mesh-first protocols are NARADA [Chu et al. (2002)], TMesh [Wang et al. (2002)], and Scattercast [Chawathe (2003)].

While tree-first protocols are very useful for single-source applications (one-to-many), mesh-first protocols are more appropriate for multi-source applications (many-to-many). In tree-first protocols, the delivery tree structure is easier to construct, however it is very sensitive to node failure, since every node pair is connected by a single path, so in case of failure the tree must be reconstructed. On the other hand, mesh-first protocols are more robust, since mesh contains multiple paths for every node pair, however tree construction is more complex, because it uses a two-step procedure and nodes must run a distributed routing algorithm.

The construction of the overlay trees (or meshes) can be achieved in a distributed fashion [Zhang et al. (2002); Francis (2000); Banarjee et al. (2002); Mathy et al. (2001)] or by means of a centralized procedure [Pendarakis et al. (2001); Padmanabhan et al. (2002)]. In the first case, members organize themselves by choosing the most appropriate neighbors or parents (in case of a tree) based on some cost metric or distance (e.g. RTT, available bandwidth, etc.). For example, in HMTP [Zhang et al. (2002)] a newcomer member that wants to join the tree, contacts with a potential parent (initially the root node), and measures the distance to the potential parent and also the distance to the children of the potential parent. If the potential parent is the closest node, the newcomer tries to join to it, otherwise, the newcomer chooses the closest children as new potential parent, and repeats the same procedure. Distributed techniques are very appropriate for very dynamic

environments where new members join or leave the group, however they are more difficult to implement and usually lead to sub-optimal solutions, since nodes have only partial information about other nodes and distances. On the other hand, in centralized mechanisms, a central node gathers information about node distances from every node pair, and computes a near-optimal spanning tree, by using some heuristic algorithm (like Prim algorithm, Kruskal algorithm, or similar). Then, the central node submits to every node in the group the corresponding list of children for data delivery. Centralized mechanisms have obvious shortcomings in terms of scalability and reliability, and they are not very appropriate for dynamic and large environments, since the tree must be recomputed at the central node every time a node joins or leaves the group. However, they lead to more optimal trees than distributed approaches, and are appealing for single-source applications involving a reduced number of nodes where it is feasible to deploy the centralized management functionality at the source node.

3. Multicasting in grid environments

As we mentioned before, there are several basic functions in a grid which can benefit from multicast communication models, and ALM in particular. We have identified the next three operations: 1) file transfer in computational grids; 2) replica management in data grids; 3) resource and service discovery services.

3.1. *File transfer in computational grids*

One of the most common uses of a grid is the aggregation of computational resources to execute compute-intensive applications in a distributed way. In this kind of applications, once the computational grid resources have been discovered and selected, the client node (source) must submit the executable binary files and input files of the application to the selected grid resources (targets). Usually, this file transfer operation is performed by establishing multiple unicast connections between the source node and the target grid nodes, which results in great bandwidth consumption at the source, and long transmission delays.

This kind of bulk data transfers from one source to multiple destinations perfectly fits with the multicast communication model, and hence its performance can be improved using ALM techniques. In this case, the target grid nodes can be organized in an overlay tree, routed at the source node. The application files submitted by the source node are forwarded through the overlay tree, from parent to children, until they are received by all the target nodes. This technique reduces the bandwidth consumption at the source, and can also reduce the overall transmission delay.

3.2. *Replica management in data grids*

Data grids provide a platform for efficient location, access, and transmission of huge datasets (from terabytes to petabytes) in data-intensive applications. In this kind of

applications, users need to access to large data files located at remote sites, and can create local copies (replicas) of these data for local analysis or processing purposes. Usually, data grids are managed by means of a replica management service, which allows the following basic functions: 1) the registration of new files with the replica management service; 2) the creation and deletion of replicas for previously registered files; 3) the location and discovery of replicas; 4) the updating of replicas to preserve consistency when a replica is modified.

Many of these basic functions can benefit from multicasting and ALM techniques, for example, in replica creation, when several users request a replica of the same file, it can be submitted to the different users using multicast. The replica updating operation can also improve efficiency with multicasting, since the updated file can be sent from the source to all the systems that contain a copy of the replica using multicast transmission. Finally, the replica discovery service can also take advantage of multicasting, as we analyze in the next sub-section.

3.3. *Service and resource discovery*

Discovery service used in most grid systems is based on index servers, which can be organized hierarchically, that aggregate information about resources and/or services available on the network. This organization exhibits two main drawbacks: first, the index servers represent a single point of failure; second, the information provided by the index server can be obsolete, especially in very dynamic and changing environments. Using the multicast communication model, it is possible to develop a decentralized discovery system, which does no rely on a central index server, but the user can access directly to the information providers by means of a multicast enquiry. This schema presents several advantages: 1) it eliminates the central point of failure; 2) the information is obtained directly from the resources local information provider, so it is very updated.

In this work we analyze the use of ALM techniques for file transfer in computational grids. The utilization of these techniques for replica management or resource discovery is reserved for a future work.

4. Construction of the ALM overlay tree

This section presents the method for constructing an overlay tree for efficient file transfer in computational grids. In this context, the client node have to transfer one or more binary and input data files to the selected grid resources. Therefore, it is a single-source application which fits well with tree-first ALM protocols. In comparison with other ALM single source applications, like audio/video distribution or content distribution networks, which can comprise several hundreds of nodes (final users) that join and leave the distribution network in an arbitrary fashion, a typical distributed application in a computational grid usually involves a lower number of steady nodes. So, we have opted for a centralized schema, where the source node

gathers the distance metrics between every grid node pair, and computes an optimal overlay tree for data delivery.

4.1. *Distance metrics*

Unlike IP multicast routers, ALM nodes have limited information about the underlying network topology. To infer the network metrics (e.g. delay, bandwidth), direct end-to-end measurement techniques or landmark based techniques can be used. Direct end-to-end measurement techniques require sending probe packets between every pair of end hosts for some time interval, using some monitoring tool, like *ping*, *iperf* [The Iperf Project (2005)], *ABWE* [Navratil and Cottrell (2003)], etc. These techniques can cause a significant traffic overhead, especially for large networks, and hence they do not scale well. Landmark based approaches [Sharma et al. (2006); Ng and Zhang (2002); Tang and Crovella (2003)] measure the distances between network nodes and a set of landmark nodes, and use these distances to estimate the node position (coordinates) in a Cartesian space. Although these techniques exhibit better scalability, they rely on a more complex network infrastructure (stable set of landmark nodes) and an efficient mechanism for computing network coordinates.

In this work we use direct end-to-end metrics, obtained by the *ABWE* monitoring tool, which can be used to estimate RTT and available bandwidth between host pairs. *ABWE* is a low network intrusive monitoring application, based on packet pair techniques and designed to work in continuous mode.

Each node in the grid is configured to use *ABWE* for sending continuously probe packets to all the other grid nodes, and reporting the information about delay and bandwidth to the source node (client). Therefore, the source node collects this kind of end-to-end metrics for every grid node pair, which are used subsequently to construct the overlay tree.

4.2. *Minimum diameter spanning tree with degree bound*

The set of hosts involved in the computational grid application can be represented as a fully connected graph $G = (V, E)$, where V is the set of nodes, with $|V| = n$, and E is the set of edges between nodes pairs, with $|E| = k$, where every edge represents the unicast path between two nodes through the underlying physical network. In addition, every edge $e \in E$, is endowed with a distance metric function or weight $\{w(e)\}_{\forall e \in E}$.

The problem is to find a spanning tree $T = (V', E')$, with $V' \subseteq V$ and $E' \subseteq E$, which minimizes a given cost function. Typical multicast applications are based on the construction of a minimum spanning tree, which tries to minimize the following cost function:

$$S(t) = \sum_{\forall e \in E'} w(e) \tag{1}$$

where $S(T)$ is the sum of distance metrics for all the edges in the tree.

Although this problem is NP-hard, there are several well-known heuristics algorithms, like Prim or Kruskal algorithms [Cormen et al. (2001)], which can reach a satisfactory solution with a complexity $O(n^2)$.

However, a major concern in ALM is the delay, since hosts have limited knowledge of routing information and network metrics, and then communication paths are not optimized from the point of view of the underlying physical network. To mitigate this problem, it is more important to pay attention to the overlay tree diameter, which can be defined as the maximum distance between any node pairs in the tree [Wu et al. (2006)]. In our problem, where a source node sends a file to several target nodes through the overlay network, a minimum diameter spanning tree will minimize the time taken to deliver the file to all destination nodes in the tree. Furthermore, to avoid excessive bandwidth consumption at any node, it is also important to limit the maximum node degree, i.e., the maximum number of children of any node in the tree. So the problem is to find a *minimum diameter source-specific spanning tree with degree bound*.

Let $s \in V$ the source node; let B the degree bound of any node in the tree, where $B \geq 2$; and let $path\_length(i \rightarrow j)_T$ the aggregate distance between nodes i and j in T, according to the the unique $i \rightarrow j$ path in T, $(i \rightarrow j)_T$, i.e.,

$$path\_length(i \rightarrow j)_T = \sum_{\forall e \in (i \rightarrow j)_T} w(e) \tag{2}$$

The problem is to find a spanning tree $T = (V', E')$ of G, routed at node s, which minimizes the following cost function:

$$D(T) = max_{\forall i \in V'} \{path\_length(s \rightarrow i)_T\} \tag{3}$$

where $D(T)$ is the diameter of T, subjected to the constraint that

$$d_T(i) \leq B, \quad \forall i \in V' \tag{4}$$

where $d_T(i)$ is the degree of node i in T.

Fig. 2 shows the algorithm proposed to compute the minimum diameter spanning tree, where:

- n is the number of nodes in the tree
- s is the source node
- B is the degree bound
- $parent(i)$ is the parent of node i in T
- $children\_list(i)$ is the children list of node i in T
- $d_T(i)$ is the degree of node i in T
- $e(i, j)$ is the edge between nodes i and j in the graph G
- $w(e(i, j))$ is the distance metric (weight) associated with edge between nodes i and j in the graph G

```
for each i ∈ V
   path_length(s→i)_T := 0
   d_T(i) := 0
   parent(i) := 0
   children_list(i) := ∅
end for

V' := ∅ ∪ {s}

while ( |V'| < n )
   min_path_length := MAXINT
   for each i ∈ V'
      for each j ∈ V, j ∉ V'
         if (d_T(i) < B)
            new_path_length := path_length(s→i)_T + w(e(i,j))
            if (min_path_length > new_path_length)
               min_path_length := new_path_length
               p := i   /* selected parent */
               v := j   /* selected node   */
            end if
         end if
      end for
   end for

   children_list(p) := children_list(p) ∪ {v}
   d_T(p) :=  d_T(p) + 1
   parent(v) := p
   path_length(s→v)_T := min_path_length
   V' := V' ∪ {v}

end while
```

Fig. 2. Algorithm to compute the minimum diameter spanning tree with degree bound

- $path\_length(s \rightarrow i)_T$ is the aggregate distance of the unique path from s to i in T

In each iteration of the *while* loop a new node is selected to join the tree (v), along with its corresponding parent (p). The selected node-parent pair (v, p) is that which minimizes the diameter of the partial tree, i.e., that node, v (and parent, p), with the shortest path length from the source node (s).

5. Results

Although GridFTP is the main protocol for high-performance and secure data transfers in Globus-based grids, there are other alternative file transfer protocols like *https*, *sftp*, *scp*, *bbcp*, *bbftp*, etc., which are more familiar to many users and are also common and in use in grid environments. For example, some grid infrastructures, like TeraGrid, allow the transference of small and medium size files (until 1Gb) to grid nodes using *scp* [The Teragrid Project (2001)]. Some recent initiatives like SCE [Dai et al. (2007)] and Mesh [Kolano (2007)] propose a lightweight grid middleware based on SSH, which can use *scp* and *sftp* applications for file transfer. Even the Globus Toolkit has developed a version of OpenSSH that supports Grid authentication based on GSI (Grid Security Infrastructure), called GSI-OpenSSH [The Globus Alliance (2008)], which includes services for remote login (*gsi-ssh*), remote copy (*gsi-scp*), and secure FTP (*gsi-sftp*). Our experimental setup uses the OpenSSH *scp* application for file transfer between grid nodes.

It is important to remark that the GridFTP protocol has incorporated very recently, and in parallel with the development of the present work, a driver for multicast transmission using also an overlay network. However, the current implementation of GridFTP Multicast does not include any particular method or algorithm for computing the overlay tree, so the user is responsible for manually specifying the list of children nodes for file delivery at command line. For a forthcoming work, we plan to apply our minimum diameter spanning tree with degree bound algorithm to automatically construct the overlay tree in GridFTP Multicast.

The experiments have been achieved using a real testbed consisting of 21 nodes distributed in 9 sites across four different countries (Spain, France, UK and USA), as shown in Fig. 3 and Table 1. The source node is *node0*, which is located at *site0* (Spain).

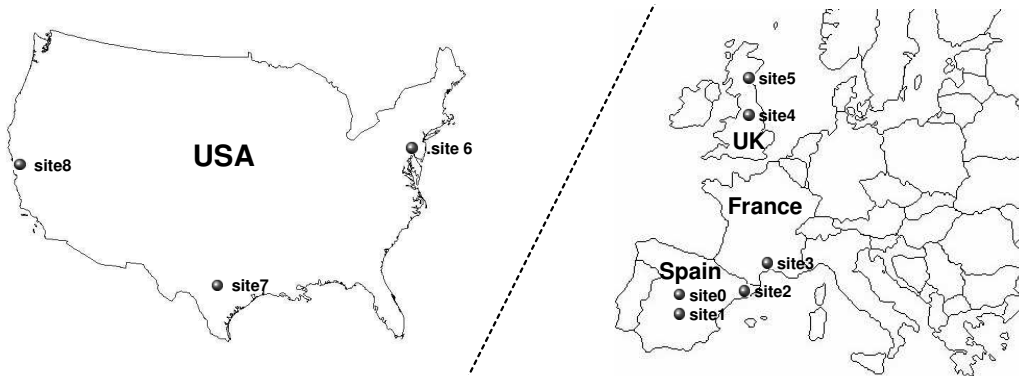

Fig. 3. Experimental Testbed

Table 1. Nodes at the experimental testbed

Site	No. Nodes	Nodes
Site0 (Spain)	1	node0 (source)
Site1 (Spain)	2	node1, node2
Site2 (Spain)	2	node3, node4
Site3 (France)	3	node5, node6, node7
Site4 (UK)	2	node8, node9
Site5 (UK)	2	node10, node11
Site6 (USA)	4	node12, node13, node14, node15
Site7 (USA)	3	node16, node17, node18
Site8 (USA)	2	node19, node20

The end-to-end metrics used for ALM tree construction are collected by running the *ABWE* monitoring tool in all the grid nodes. Each node is configured for sending continuously probe packets to all the other grid nodes using *ABWE*, and reporting the information about delay and bandwidth to the source node. In this work, we use the end-to-end delay as distance metric for the ALM tree.

In order to avoid high bandwidth consumption at any of the nodes, we have limited the maximum node degree (B) to different values. In particular, we have constructed three different ALM trees with different values of degree bound: *B=2*, *B=3*, and *B=4*. Figures 4, 5, and 6 depict, respectively, the three resulting minimum diameter spanning trees rooted at *node0*.

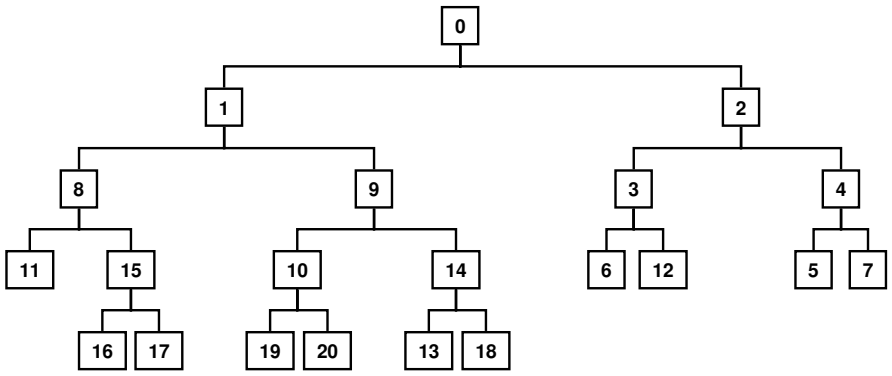

Fig. 4. Minimum diameter spanning tree with B=2

To evaluate the efficiency of application layer multicast in comparison with traditional unicast, we have transmitted several files of different sizes (from 100 KB to 100 MB) using the *scp* program from source node (*node0*) to the other 20 des-

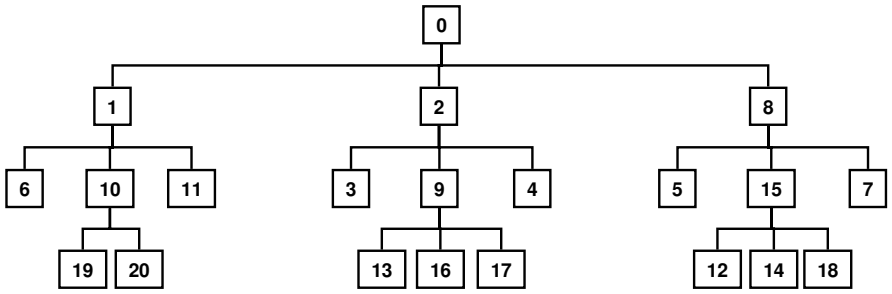

Fig. 5. Minimum diameter spanning tree with B=3

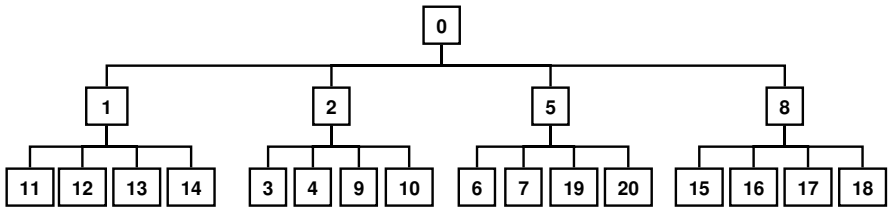

Fig. 6. Minimum diameter spanning tree with B=4

tination nodes in the grid. Fig. 7 shows the overall transmission time using unicast and using ALM with *B=2*, *B=3*, and *B=4*. As we can observe, the time consumed for transmitting medium-to-large files (size over 10 MB) is significantly lower when using ALM. For example, for a file of 100 MB, the time consumed to send the file from the source to 20 destinations using unicast is 1,564 sec., while using ALM with *B=4*, the overall transmission time is 884 sec. (56% lower than unicast).

We can observe also that the transmission time with ALM using *B=4* is lower than using *B=3*, and *B=2*. This is because of the lower number of levels of the tree obtained with *B=4* (3-level tree), in comparison with *B=3* (4-level tree), and *B=2* (5-level tree). A higher number of levels in the tree can result in a longer transmission time, since the transmission delays at every level are accumulative.

From the point of view of bandwidth consumption, ALM also provides important benefits, since the network traffic is distributed among the different nodes involved in the transmission. This fact is reflected in Fig. 8, which shows the inbound and outbound traffic observed on every site when the source node (*node0*) transfers a file of 1 MB to the 20 destinations in the testbed grid, using unicast and ALM (with *B=2*, *B=3*, and *B=4*). With unicast transmission, *site0* (where the source node is located) exhibits much high bandwidth consumption in comparison with other sites, because all the outbound traffic is concentrated on this site. However,

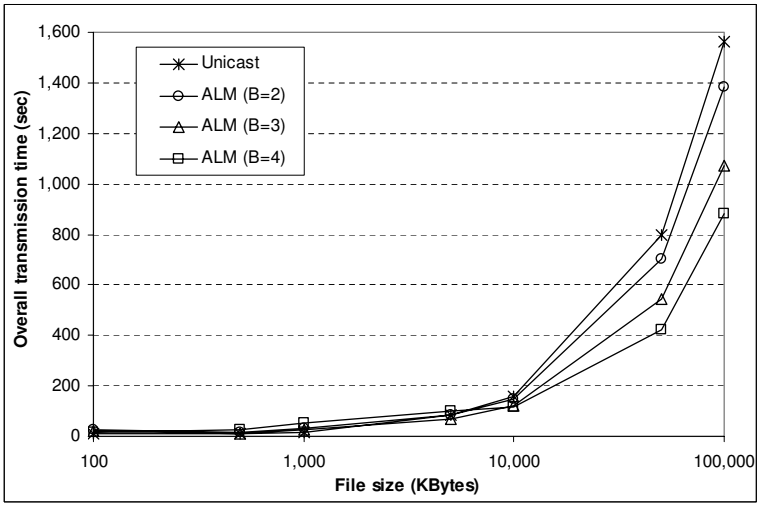

Fig. 7. Overall transmission time for different file sizes

using ALM, the outbound traffic is shared out among several nodes, so that the level of bandwidth consumption at the different sites is better balanced.

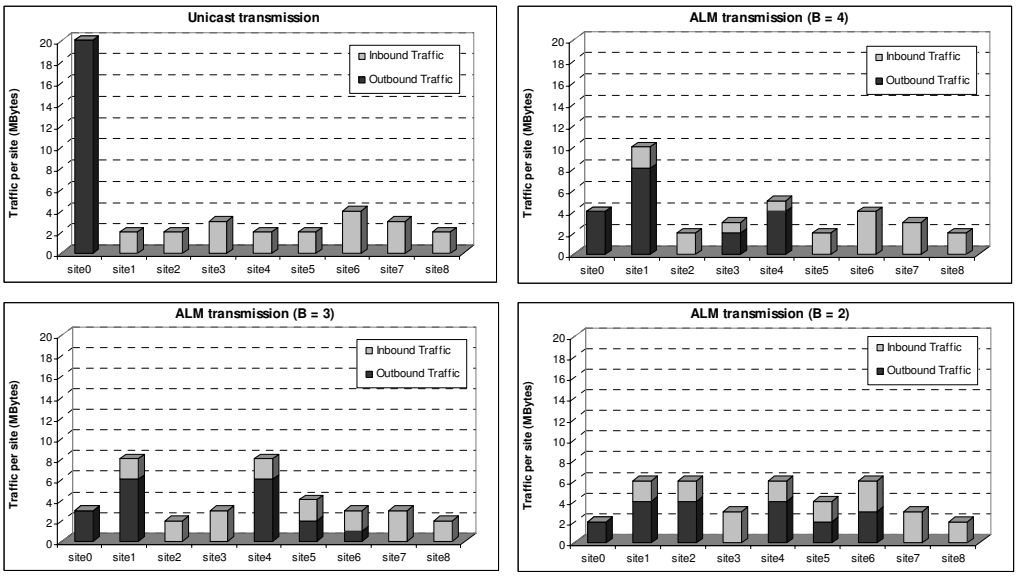

Fig. 8. Observed traffic per site (sent file = 1MB in size)

6. Conclusions and future work

In this work we have analyzed the utilization of application layer multicast techniques for improving the performance of one-to-many file transfers in computational grids. ALM does not depend on the multicast support of the underlying network, so it can be implemented in any grid environment. Furthermore ALM uses unicast paths for data delivery, so it is fully compatible with TCP-based services. For constructing the overlay tree, we have presented a minimum diameter spanning tree algorithm with degree bound, which uses end-to-end measurements (RTT) gathered from a monitoring tool (*ABWE*) as distance metrics. The results, which have been obtained using a real testbed, show that ALM can reach a significant reduction of the overall transmission time and also a better balance of bandwidth consumption. We can identify several research lines opened for future work: the application of ALM techniques to other grid operations (replica management, resource discovery, etc.); the comparison of different metrics for constructing the overlay tree (RTT, bandwidth, landmark based metrics, etc.); the utilization of decentralized/distributed techniques for overlay tree construction in dynamic environments; and the application of different overlay tree construction techniques to GridFTP Multicast.

References

Banarjee, S., Bhattacharjee, B. and Kommareddy, C. (2002). 'Scalable application layer multicast. *Proc. of ACM SIGCOMM 2002*: 205–217.

Barcellos, M.P., Nekovee, M., Koyabe, M., Daw, M. and Brooke, J. (2005). Evaluating high-throughput reliable multicast for grid applications in production networks. *Proc. of the Fifth IEEE International Symposium on Cluster Computing and the Grid (CCGrid'05)*: 442–449.

Chawathe, Y. (2003). Scattercast: an adaptable broadcast distribution framework. *Multimedia Systems*, **9-1**: 104–118.

Chu, Y., Rao, S.G., Seshan, s. and Zhang, H. (2002). A Case for End System Multicast. *IEEE Journal on Selected Areas in Communication (JSAC), Special Issue on Networking Support for Multicast*, **20-8**: 1456–1471.

Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2001). Introduction to Algorithms (Section 23.2: The algorithms of Kruskal and Prim). *MIT Press and McGraw-Hill, second edition*: 567–574.

Dai, Z., Wu, L., Xiao, H., Wu, H. and Chi, X. (2007). A Lightweight Grid Middleware Based on OPENSSH-SCE. *Sixth Int. Conference on Grid and Cooperative Computing (GCC'07)*: 387–394.

Francis, P. (2000). Yoid: Extending the Internet multicast architecture. *http://www.icir.org/yoid/docs*: 1–38.

Globus Alliance (2007). GSI-OpenSSH, GSI-SCP, and GSI-SFTP. *http://www.globus.org/grid_software/data/scp.php*

Hammerle, H. (2007). Grid Challenges for Business. *Reports From the EGEE User Forum, GridToday (www.gridtoday.com).*

Hosseini, M., Ahmed, D.T, Shirmohammadi, S. and Georganas, N.D. (2007). A Survey of Application-Layer Multicast Protocols. *IEEE Communications Surveys and Tutorials*, **9-3**: 58–74.

Iperf Project (2005). *http://dast.nlanr.net/Projects/Iperf/*

Janic, M.(2005). Multicast in Network and Application Layer. *PhD Thesis, Delft University of Technology.*

Jeacle, K., Crowcroft, J., Barcellos, M.P. and Pettini, S. (2005). Hybrid reliable multicast with TCP-XM. *Proc. of the 2005 ACM conference on Emerging network experiment and technology table of contents*: 177–187.

Kolano, P.Z. (2007). Mesh: Secure, Lightweight Grid Middleware Using Existing SSH Infrastructure. *Proceedings of the 12th ACM symposium on Access control models and technologies*: 111–120.

Kouvatsos, D.D. and Mkwawa, I.H. (2003). Multicast communication in grid computing networks with background traffic. *IEE Proceedings Software*, Vol. 150, Issue 4 (2003), pp. 257–264.

Mathy, L., Canonico, R. and Hutchison, D. (2001). An Overlay Tree Building Control Protocol. *Lecture Notes In Computer Science*, **2233**: 76–87

Moreno-Vozmediano, R. (2007). Application Latyer Multicast Techniques in Grid Environments. *Proc. of Euro American Conference on Telematics and Information Systems (EATIS'07).*

Navratil J. and Cottrell R.L. (2003). ABwE: A practical Approach to Available Bandwidth Estimation. *Passive and Active Measurement Workshop (PAM'03).*

Ng, E. and Zhang, H. (2002). Predicting Intemet network distance with coordinates-based approaches. *Proc. of IEEE INFOCOM 2002.*

Padmanabhan, V.N., Wang, H.J., Chou, P.A., and Sripanidkulchai, K. (2002). Distributing streaming media content using cooperative networking. *Proc. of the 12th int. workshop on Network and operating systems support for digital audio and video*: 177–186.

Pendarakis, D., Shi, S., Verma, D. and Waldvogel, M. (2001). ALMI: An application level multicast infrastructure. *Proc. of 3rd Usenix Symp. on Internet Technologies and Systems (USITS'01)*: 49–60.

Ranaldo, N., Tretola, G. and Zimeo, E. (2005) Hierarchical and Reliable Multicast Communication for Grid Systems. *John von Neumann Institute for Computing, NIC Series*, **33**: 137–144.

Sharma, P., Xu, Z., Banerjee, S. and Lee, S.J. (2006). Estimating network proximity and latency. *ACM SIGCOMM Computer Communication Review*, **36-3**: 39–50.

Tan, S., Waters, G. and Crawford, J. (2006). A performance comparison of self-organising application layer multicast overlay construction techniques. *Computer Communications*, **29-12**: 2322–2347.

Tang, L. and Crovella, M. (2003). Virtual landmarks for the internet. *Proc. of the 3rd ACM SIGCOMM conference on Internet measurement.*

The Teragrid Project (2001). User Support and Documentation. Data: Moving Local Files to the TeraGrid. *http://www.teragrid.org/userinfo/data/basics.php*

Wang, W., Helder, D., Jamin, S. and Zhang, L. (2002) Overlay Optimizations for End-host Multicast. *Proc. Networked Group Communication*: 154–161.

Wu, Y., Cai, Y., Huang, J. and Xu, X. (2006). Minimum Diameter Application Layer Multicast Tree Algorithm. *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*: 546–551.

Zhang, B., Jamin, S, and Zhang, L. (2002). Host Multicast: a framework for delivering multicast to end users. *Proc. of IEEE INFOCOM 2002*: 1366–1375.

Zhang, B., Wangb, W., Jaminb, S., Masseyc, D. and Zhangd, L. (2006). Universal IP multicast delivery. *Computer Networks*, **50-6**: 781–806