# A Simulation Study of Data Distribution Strategies for Large-scale Scientific Data Collaborations

Samer Al Kiswany and Matei Ripeanu
Electrical and Computer Engineering Department
The University of British Columbia
Vancouver, Canada
{samera, matei}@ece.ubc.ca

***Abstract*. Scientific instruments in fields as diverse as high-energy physics and genomics generate enormous volumes of data that need to be processed and analyzed by geographically dispersed communities. Such scientific collaborations require an efficient data dissemination technique. We analyze recent techniques proposed for peer-to-peer data distribution, select a small set of solutions representative for the various approaches currently proposed, and evaluate them, through simulation, in the context of data dissemination in scientific collaborations. This paper focuses on the performance and scalability of our simulator. Additionally, we include several preliminary recommendations for data optimal dissemination in data-intensive scientific collaborations derived form our preliminary simulation results.**

## I. INTRODUCTION

Modern science is often data-intensive: large-scale simulations, new scientific instruments, and large-scale observations generate impressive volumes of data that need to be analyzed by large, geographically dispersed user communities emerging in fields as diverse as genomics and high-energy physics. Examples include CERN's Large Hadron Collider (LHC) experiment [1], and DØ experiment [2].

These scientific collaborations often use Grid middleware to efficiently distribute data to participating sites for processing and analysis. Most existing Grid deployments, however, use tools that involve explicit data movement through batch jobs but often ignore changing network conditions and rarely exploit the collaborative nature of modern-day science.

Recently, on the other hand, data dissemination in a peer-to-peer context has generated renewed interest and a number of systems have been deployed and successfully used by communities as large as millions of users [3].

We aim to quantify the potential benefits brought by these peer-to-peer data-dissemination techniques in today's data-intensive scientific collaborations. To this end, we have selected a small set of solutions representative for the various approaches currently proposed and we evaluate them, through simulation, in the context of data dissemination in scientific collaborations. To the best of our knowledge, this is the first study to compare head-to-head a broad set of data dissemination protocols in this context.

This paper focuses on two aspects. First, we focus on the simulator we have built to comparatively evaluate data dissemination techniques. We detail our simulator design and compare its performance with similar simulators presented in other studies. Second, we briefly present simulation results for a limited number of physical topologies and derive recommendations for real-world data-centric collaborations.

The rest of this paper is organized as follows. Section 2 surveys existing work on data dissemination solutions while section 3 surveys past simulation and modeling efforts of these techniques. Section 4 presents the design of our simulator which is evaluated in Section 5. Section 6 presents simulation results for a number of scenarios. We conclude in Section 7.

## II. DATA DISSEMINATION SOLUTIONS

In addition to the primitive data dissemination technique that sets up independent transfer channels between the data source and each destination, we have identified three broad categories of techniques employed to enhance the performance of data dissemination solutions: overlays, swarming, and the use of intermediate storage capabilities. Often, deployed solutions use combinations of these three techniques.

*Overlay based techniques* employ participating end-systems to implement functionality not provided by the underlying communication layer or to improve the characteristics of services provided by lower network layers. When providing data dissemination functionality, overlays often serve as support for extracting dissemination trees used to send data from each source to all destinations. Among the solutions in this category sit: application level multicasting ALM [5], which constructs an overlay tree between participating end-nodes which is then used to disseminate data, and Spider [4], which builds multiple source-rooted trees.

*Swarming* tries to optimally exploit the 'orthogonal bandwidth' available between the nodes participating in file dissemination, that is, the physical network paths not included in a source-rooted application level tree. The file to be distributed is first divided into blocks at the source and each block is sent to a distinct set of nodes in the system. Nodes then peer dynamically to exchange blocks of interest. The challenge for these protocols is building efficient mechanisms to pair nodes and decide which blocks to exchange. Bullet [6] and BitTorrent [3] systems fit in this category. While both depend on informed delivery techniques [7] to select a set of blocks to be exchanged they differ in two important ways: BitTorrent depends on a central server to provide information on participating peers and the location of each block, while Bullet uses an overlay tree to disseminate control messages, and provide peer and block location information. Secondly,

BitTorrent uses tit-for-tat policy to prevent freeriding while Bullet assumes a cooperative environment in which all nodes are interested to spread data as fast as possible. Nodes that implement the tit-for-tat policy favor peers that reciprocate in the data exchange: that is, nodes prefer to exchange data with other peers that can serve useful data, i.e., file blocks the current node misses, as fast as they download data.

Finally, the technique of employing *intermediate storage* capabilities placed at strategic network locations in the data dissemination system (e.g., logistical multicasting [8]), may provide better control over the data transfer paths, increase buffering capacity, and increase the ability to recover form node failure.

For our experimental study, we have built a simulator to explore the performance of representative data distribution solutions that employ the techniques described above: application-level multicast (for which we choose a particular solution, ALMI [5], which tries to optimally build an application level based on information gathered at a central location), Spider, Logistical Multicasting (LM), Bullet, and BitTorrent. Additionally, to anchor our comparison with relevant base-cases, we simulate IP-multicast and the naive approach in which data are sent through independent channels set between the source and each destination.

## III. RELATED WORK

A number of studies evaluate methods to distribute data from one source to multiple destinations in the context of file-sharing applications or data replication. These evaluation efforts lay in one of three broad categories: analytical, measurement, and simulation-based studies.

### *A. Analytical Studies:*

Analytical models have been proposed to study different file distribution protocols. Biersack et al. [9] analyze the performance of data dissemination solutions using single and multiple application-level trees. For a single tree the authors note that, as the number of children per node increases, the percentage of the nodes uploading data decreases and the interior tree nodes must serve more uploads. To balance the load, they propose setting up multiple trees concurrently such that participating nodes are interior nodes in some trees and leaf nodes in others. To optimally balance load and limit node overhead the study recommends a node fan-out between 3 and 5.

Qiu et al. [10] present a fluid model for Bittorent-like networks and use it to confirm BitTorrnet scalability (i.e., the average download time is independent of the node arrival rate) and the efficiency of the rarest-first policy in uniformly distributing file blocks.

For analytical studies however, as for any modeling exercise, the main tradeoff is between the complexity of the model and its ability to capture all system details. Consequently, it is often necessary to simplify, sometimes unrealistically, the model in order to make the analysis computationally tractable. In a large dynamic system like a data dissemination system it is unrealistic to assume a homogenous set of nodes. Without this assumption, on the other side, analytical models that accurately model heterogeneity, physical network topologies, and network contention quickly become intractable.

### *B. Measurement Studies:*

A number of measurement-based studies analyze deployed data dissemination systems [11, 12]. These studies depend on log files found at a central BitTorrent server, or on statistics collected by modified nodes participating in the dissemination network. While these studies confirm BitTorrent ability to deal with flash crowds, the necessity to decentralize the tracker component, and to add incentives for seeding, their evaluation of the efficiency of the dissemination system is limited: they cannot measure average file download time, load balancing between peers, or network stress, largely due to the limited information available at runtime.

### *C. Simulation Studies:*

While most simulation studies focus on the various design and parameter choices for a single system, few simulation studies compare different protocols. Bharambe et al. [13] study the effect of different BitTorrent mechanisms on system performance. While their findings largely agree with our simulation results, their simplifying assumptions leave some questions unanswered. This study does not estimate the impact of resource contention (mostly network contention), does not realistically model protocol overheads, , and does not address the issue of fairness to other competing traffic, all of which, we believe, are key metrics to consider when selecting a data dissemination solution for scientific collaborations.

While BitTorrent attracted researches attention due to its wide deployment, other data dissemination systems have not attracted similar attention. We believe that our simulator enables a first study to directly compare a multitude of dissemination protocols.

## IV. SIMULATOR DESIGN

To investigate the performance of different data distribution protocols we have built a high-level simulator. As in most simulators, the main tradeoff we face is between the cost of the simulation, which might limit the set of the scenarios that can be investigated, and simulation fidelity. For scalability reasons our simulator does not work at the packet-level [16]. Rather, our simulator works at the application level with file-block granularity for data transfers, a natural choice since many of the data dissemination schemes we investigate use file blocks as their data management unit. Thus, while we do not simulate physical link contention at the packet level, we do simulate physical link contention between application-flows at the block level and we explicitly simulate delays on individual physical links.

We argue that simulating these protocols at packet level would not increase significantly simulation fidelity over our choice to simulate with file-block granularity (with block sizes selected so that blocks span over a reasonable number of packets). Further, for similar reasons, block-level simulation approach is adopted in a number of other studies [13, 14].

The simulator is composed of three main modules: routing, peering, and block transfer. As their names indicate, the

routing module is responsible of running the routing protocol for all internal nodes in the topology; the peering module is responsible for constructing peering relationship between nodes according to the protocol specification; and, finally, the block transfer module uses the information provided by the two other modules to simulate block's transfer between peers using the paths provided by the routing module.

In more detail, after routing paths between nodes are selected (using a shortest path algorithm), the simulation works in rounds for dissemination solutions that do not use fixed peering between nodes, i.e., Bullet and BitTorrent: in each round, first, the peering algorithm is executed, adding or deleting new pairs of nodes that exchange data. At this stage, with these two pieces of information: the set of peers interested in exchanging blocks in the next round and the routing paths between them, network contention is simulated on each physical link and the number of blocks to be transferred between each two peers is found. Next, the set of blocks are selected to be exchanged, and finally the blocks are simulated to propagate between the peers.

For dissemination solutions that use fixed peering between nodes, i.e., application-level and logical multicast or Spider, the simulator analysis the topology in hand, determines routing paths, and uses fluid model to determine network contention at physical network level and estimate data transfer performance.

## V. SIMULATOR EVALUATION

In this section we evaluate our simulator performance. We have generated a set of Waxman topologies using BRITE [15] with different number/size of blocks and with different number of participating nodes. All simulations are executed on a system with an Intel P4 2.8 GHz processor and 1GB of memory.

The simulator for IP-multicasting, ALM, LM tree, Spider and separate transfers from the source to every node, simulates the high-level deterministic protocol behavior. Simulating one of these solutions on topologies with few thousands of end-nodes can be obtained in few minutes.

While Bullet and BitTorrent simulators use the same routing module, each has a different peering and block transfer module reflecting the protocol's characteristics. Since these are the most complex protocols we simulate they limit the size of the physical topologies we can explore. Table 1 details the complexity of each module for these two protocols.

Figures 1 and 2 present the results of simulating a set of generated topologies for Bullet and BitTorrent. As it is clear from the analysis, BittTorrent has a higher complexity. This is due to the more complex peering (tit-for-tat) and block selection (rarest-first) policies used in BitTorrent. Practically, the simulator can simulate a network of few hundreds of nodes and few thousands of blocks (a typical sitting in today's scientific collaboration systems) in few hours.

Compared to simulators described in literature, our simulator performs well. For instance, Bharambe et al. [13] present results for simulating a network with 300 simultaneously active nodes and 100 MB file of 400 blocks,

| Module | Bullet | BitTorrent |
|---|---|---|
| Routing | $O(E^3*L)$ | $O(E^3*L)$ |
| Peering | $O(E^2*B*Log(B))$ | $O(E^2*B.Log(B)+E^3)$ |
| Block Transf. | $O(E*P*B)$ | $O(E*P^2+E*P*Log(N))$ |

Table 1. The complexity of Bullet and BitTorrent protocol's modules. Notations: $E$ - the number of end nodes; $L$ - the number of links in the physical network topology; $B$ - the number of file-blocks; $P$ - the number of peers per node.

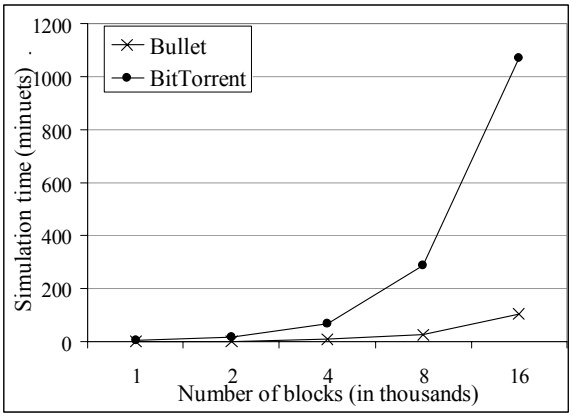


Figure 1: Simulation time for a 25 nodes topology and 1GB file.

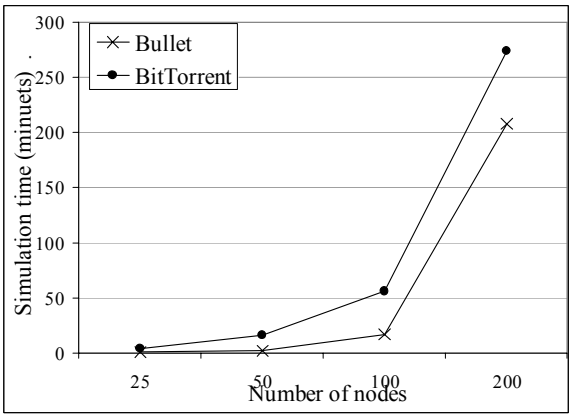


Figure 2: Simulation time for disseminating a 1 GB file.

without incorporating physical topologies and consequently not simulating network contention. Similarly, Gkantsidis et al. [14] present a simulation results for a topology of 200 nodes and a file of 100 blocks.

## VI. SIMULATION RESULTS

Limited space prevents us from presenting detailed results of our preliminary simulation experiments. We have generated a number of Waxman topologies using BRITE [15]. These topologies differ in the number of nodes and the density of the network core. All simulations explore the performance of distributing a 1GB file over the different topologies; with the file being divided into 2000 blocks.

Figures 3 and 4 present the result of the simulation for one of the generated topologies (Spider is not presented here as it does not build more than one dissemination tree and thus it is, for this topology, equivalent to IP-multicast). We note that generally, the other topologies present a similar performance

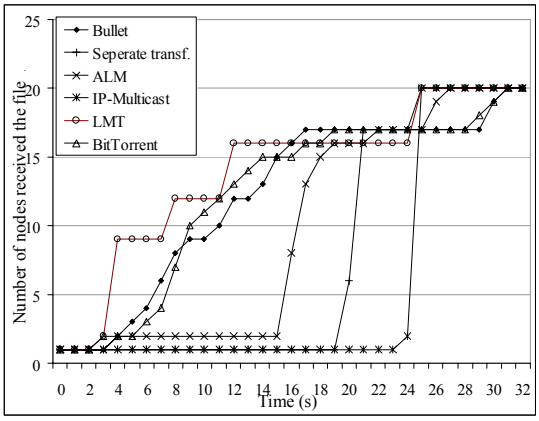


Figure 3: Number of destinations that have completed the file transfer for a topology of 30 nodes (20 end nodes and 10 routers).

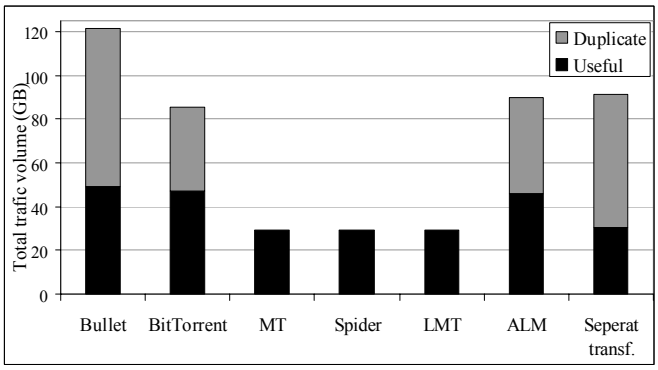


Figure 4: Overhead of each protocol on 30-node topology.

distribution of the data dissemination solutions we analyze as the one presented in Figures 3 and 4. The following observations are common:

- While IP-multicast and Spider are the optimal solutions in terms of generated overhead and in delivering the file to the slowest node, as they optimally exploit the bandwidth on bottleneck links, their intermediate progress is poor. This is because these multicasting schemes do not include buffering at intermediate points, nor exploit the perpendicular bandwidth found between the nodes in the network; consequently, they limit their data distribution rate to the rate of the bottleneck link.

- Logistical Multicast is among the first to finish the file dissemination process and also offers one of the best intermediate progress performance with no overhead traffic. This is expected since LM assumes an optimal dissemination infrastructure in terms of nodes storage capability.

- Application-level multicast, Bullet and BitTorrent are worse but close to Logistical Multicast both in terms of finishing time as well as intermediate progress. However, these protocols generate more overhead traffic (the overhead is defined: as the aggregate of the volume of duplicate traffic that traverses all physical links).

## VII. CONCLUSION

We have studied, analyzed, and categorized different data dissemination protocols found in the literature and built a simulator to simulate these protocols in the context of data intensive scientific collaboration systems. Compared to simulators described in literature, our simulator scales well: it requires a matter of hours to simulate a physical topology of few hundreds of nodes in complex data dissemination scenario of thousands of file blocks.

Our preliminary simulation results indicate that, for some existing Grid deployments, sophisticated peer-to-peer data dissemination techniques bring limited benefits due to over provisioned network cores. However, simulations indicate that this situation changes dramatically as the core networks supporting these deployments become bandwidth constrained. In these cases, adaptive, peer-to-peer techniques are able to sustain an equivalent un-degraded performance.

## REFERENCES

[1] LHC, The Large Hardon Collider Experiment website: http://lhc-new-homepage.web.cern.ch/lhc-new-homepage/.

[2] The D0 Experiment, Fermi National Laboratory, http://www-d0.fnal.gov.

[3] BitTorrent web site: http://www.bittorrent.com, 2005.

[4] S. Ganguly, A. Saxena, S. Bhatnagar, S. Banerjee, and R. Izmailov, "Fast replication in content distribution overlays," IEEE INFOCOM, Miami. 2005.

[5] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure," USITS'01, 2001.

[6] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: high bandwidth data dissemination using an overlay mesh," SOSP'03, Lake George, NY, 2003.

[7] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, "Informed content delivery across adaptive overlay networks," SIGCOMM2002, Pittsburg, PA, 2002.

[8] M. Beck, T. Moore, J. Plank, and M. Swany, "Logistical networking: sharing more than the Wires," Active Middleware Services Workshop, Norwell, MA, 2000.

[9] E. Biersack, P. Rodriguez, and P. Felber, "Performance analysis of peer-to-peer networks for file distribution," QofIS'04, Fifth International Workshop on Quality of Future Internet Services, 2004.

[10] D. Qiu and R. Srikant," Modeling and performance analysis of BitTorrent-Like peer-to-peer networks," SIGCOMM 2004.

[11] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips,"The Bittorrent P2P file-sharing system: measurements and analysis," IPTPS 2005

[12] M. Izal, G. Urvoy-Keller, E. Biersack, P. Felber, A. Al Hamra, and L. Garc´es-Erice, "Dissecting BitTorrent: five months in a torrent's lifetime," 5th Passive and Active Measurement Workshop, PAM, 2004.

[13] A. Bharambe, C. Herley, and V. Padmanabhan, "Analyzing and improving a BitTorrent network's performance mechanisms", IEEE INFOCOM 2006.

[14] C. Gkantsidis, and P. Rodriguez, "Network coding for large scale content distribution," IEEE INFOCOM, Miami. 2005.

[15] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems- MASCOTS '01, Cincinnati, Ohio, 2001.

[16] P. Huang, D. Estrin, and J. Heidemann, " Enabling large-scale simulations: selective abstraction approach to the study of multicast protocols". In Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 241-248. Montreal, Canada, IEEE. July, 1998.