RESOURCE ALLOCATION OVER GRID COMPUTING MILITARY NETWORKS

Igor Bisio, Mario Marchese, Maurizio Mongelli
DIST – Department of Communication, Computer and System Science, University of Genoa, Via Opera Pia 13 - 16145 Genoa, Italy
{Igor.Bisio, Mario.Marchese, Maurizio.Mongelli}@unige.it

and

Annamaria Raviola
Selenia Communications S.p.A., Finmeccanica Group, Via Pieragostini 80 - 16151 Genoa, Italy
Annamaria.Raviola@seleniacomms.com

ABSTRACT

In the GRID environment investigated in this work, each node of the network can be either a terminal host or a web cache location. Objects, which can represent either a file or any other resource to be shared, are downloaded throughout the network among the nodes. For memory saving and for safety reasons, each object is composed of a number of portions and not all the portions are located within the same node, because no node should have the complete knowledge of each object. It is strongly recommendable because if a single node should be either accessed without authorization, information retrieved is not sufficient to detect the overall content. Concerning networking viewpoint, the following main issues will characterize the performance of object exchange: Position of the information, Strategy to reach the information, Algorithm to download information, Capacity Planning. The paper proposes a control architecture that considers the mentioned issues. The problem is modeled through a mathematical formulation and a specific cost function, which takes into account all the necessary details, is introduced for each controller as well as a minimization procedure. A preliminary performance evaluation analyses the effect of the Local Controller.

INTRODUCTION

The problem of sharing information among different locations has been widely treated both in the literature and in practical implementations. The common action is that a user issues a request (to the server, to peer elements, to the network) and the destination site returns an answer for each request. The sites may be stand-alone servers, single gateways with similar functions and, in some cases, terminal hosts. The most common technique to access remotely located information is the client-server model. More recently, to improve the performance of the regular client-server approach, Content Distribution Networks (CDNs) are used [1]. CDNs delocalize information and functions of interest among the main server site, which contains all the information and functions, and different surrogate server sites, where the material is duplicated. A third approach is represented by peer-to-peer (P2P) overlay networks, where virtual networks of many nodes, called overlays, are built over networking infrastructures. The P2P network key point is that the hosts, considered peers, are allowed to behave as clients and servers. In practice, each host may be a server and information is exchanged among peers to provide web content and to alleviate traffic burden. The advantages of peer-to-peer communication are: scalability, knowledge sharing by aggregating information, information availability. Peer-to-peer systems are used to support several network-based applications: combining the computational power of thousands of computers [2], sharing of resources [3], distributed and decentralized searching [4]. File sharing networks [5] are perhaps the most commonly used P2P applications and, at present, compose most of Internet traffic. The widespread use of such networks can be attributed to their ease of use: [6, 7], for example, have proposed the adoption of P2P as supplementary means for providing web content in order to alleviate the traffic burden on servers. Peer-to-peer overlay networks are the best for music clips but strategic information, bank operations, trading details may be hardly delivered by using this paradigm. Moreover, strategic networks involved many security issues. A broader framework that include all the mentioned approached and refers in wide sense to information exchange is "grid computing" (see, e.g., [8]), which allows delivering distributed contents, storing information, performing remote use of time machine and sharing files over networking infrastructures [1]. This paper takes "grid computing" networks as reference and tries to propose a generic formal approach that can take the best from the three mentioned approaches: client-server, CDN and P2P. The idea is to have a "grid computing" network composed of N sites where each site contains important information. Each node can be considered a web cache location, a surrogate server and a peer host but, actually, it is part of an overall overlay network, which is itself repository of all needed information. In facts: there are a number of objects that are downloaded throughout the network; an object is a file or any other resource to be shared as well as machine time and distributed application. Being a possible application a strategic network, for memory saving and for safety reasons, each object is composed of portions and not all of them are located within the same node. It means that a single node has only a part of the file. It is strongly recommendable (even if it is not mathematically imposed in the proposed model for now) because if a single node should be accessed without authorization, information retrieved is not sufficient to detect the overall content, which needs to be composed in combination with the other nodes of the network. The paper proposes a control architecture, which is composed of three layers: Local, Network and Planning Controller. The former controls object downloading; the Network Controller checks the distribution of the object portions among the nodes; the latter changes the dimension of each single portion and increase/decrease the physical link and node capacities. The overall control structure is proposed through a mathematical formal model that considers the following performance optimization issues: capacity planning, position of the information, strategy to reach the information, algorithm to download it. At the best of authors' knowledge, it is the first time an overall formal model for an information exchange network (a "grid computing" structure) is proposed as well as a control architecture matching optimization and security issues together. The reminder of the paper is structured as follows: the next section contains the description of the control architecture. The formal model for the performance optimization is reported in Section III. Section IV presents a preliminary performance evaluation and Section V shows the conclusions and possible ideas for future work.

CONTROL ARCHITECTURE

The proposed approach is though with reference to the works of [9, 10, 11, 12]. The system performance is captured through a mathematical description of the users' requests and available resources [10]. The current level of congestion is exploited by real time measurements and the decision variables are tuned accordingly [9, 12]. The single node performance is highlighted [11]. The novelty of this work relies in the derivation of an optimization framework, suitable for the minimization of the downloading time "seen" by each independent user. The proposed approach is decentralized and scalable, since the control laws available for both the single user and the network manager are analyzed separately. In this way, the different time scales of action will be emphasized.

General Framework

The considered "grid computing" network is composed of N nodes. Each node can be either a terminal host or a web cache location. I objects should be downloaded throughout the network among the N nodes. An object can represent either a file or any other resource to be shared (e.g., machine time). For memory saving and for safety reasons, each object i is composed of $J^{(i)}$ portions and not all the portions are located within the same node. It means that a single node can have only a part of the file. It is strongly recommendable (even if it is not imposed for now) because, if a single node should be accessed without authorization, information retrieved is not sufficient to detect the overall content. Each single node k requires to download a specific object i and asks the other nodes of the network about the availability of portions of the object i through a specific signaling protocol. The nodes that have portions of the object i within their memory answer to node k. Node k requires to download specific portions of the object i to other nodes. The multiple choice (the node where downloading information) is performed by minimizing a cost function (local to the node) aimed at optimizing downloading time. Node k may download the same portion either from "one single node" or from "more than one node" depending on network security performance. The performance of the overall system in terms of downloading time may be monitored also in dependence of the number of requests for a specific object by a single node. After a period of time, evaluating the obtained performance, the following actions may be taken at different time scales: portions of objects may be exchanged among the nodes; link capacities may be modified. While the former may be implemented on a reduced time period whose order of magnitude may be hours/day, the latter should be applied on planning basis.

The control architecture is shown in Fig. 1. It is composed of three layers: Local, Network and Planning Controller. The Local Controller acts locally to each node at object downloading time scale (seconds/minutes); the Network Controller may change the distribution of the object portions among the nodes and acts with larger time scale (hours/day) and it is centralized; the Planning Controller may change the dimension of each single portion and increase/decrease the physical link and node capacity. The order of magnitude of its intervention is weeks/months.

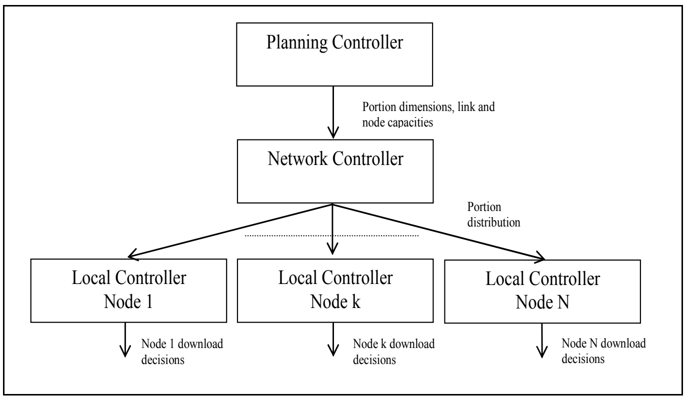

Figure 1. Control Architecture.

The requests are structured into X service options, characterized by a committed downloading rate ($dr^{(x)}$), where $x \in [1,X] \subset \mathbb{N}$ is the downloading option identifier. The traffic is structured into Y classes, characterized by a specific bandwidth assigned periodically within the framework of Resource Distribution by the Network Controller for each end-to-end path. The variable $ab_y^{(hk)}$ defines the bandwidth assigned to y-th traffic class with $y \in [1,Y] \subset \mathbb{N}$ for the end-to-end path from node h to k. Traffic is managed through "Complete Separation with Dynamic Partitions" [13]. It means that end-to-end path allocations (called bandwidth pipes) are not shared among the traffic classes and they are changed (if needed) at each intervention of the Network Controller. Allocations are supposed constant within the interval between two Network Controller interventions. The technology to define end-to-end bandwidth pipes for each traffic class is not specified but it may be taken from the QoS mechanisms described in the literature. The bandwidth pipe of a specific class may be obtained through ATM VPs, IntServ and, most probably, through DiffServ.
In dependence of their priority a set of service options will be conveyed through a specific traffic class. For example, if there are: six service options and their associated downloading rate ($X = 6$, $dr^{(1)} = 1024\ kbit/s$, $dr^{(2)} = 512\ kbit/s$, $dr^{(3)} = 256\ kbit/s$, $dr^{(4)} = 128\ kbit/s$, $dr^{(5)} = 64\ kbit/s$, $dr^{(6)} = 32\ kbit/s$) and three traffic classes ($X = 3$) with associated allocations each end-to-end path (e.g., $b_1^{(hk)} = 50\ Mbit/s$, $b_2^{(hk)} = 30\ Mbit/s$ and $b_3^{(hk)} = 20\ Mbit/s,\ \forall hk$), service options 1 and 2 can be transported through class 1 bandwidth pipe, service options 3 and 4 through class 2 bandwidth pipe and service option 5 and 6 through class 3 bandwidth pipe.

Operative Details

The implementation of the control structure is based on a signaling mechanism, used at the beginning, when a file request is issued by a node, to reveal the position of the different portions composing the desired file. The most popular search mechanism (in use in P2P networks) blindly floods a query to the network. To avoid useless multiplication of signals, this paper uses an "advanced flooding" algorithm, where, differently from "blind flooding", a node does not forward all the requests already received but performs this operation having some knowledge about the "cost" (measured in terms of residual bandwidth) of the paths traversed. The steps of the object request algorithm are briefly summarized in the following including also some observation about CAC and QoS routing schemes.

1. When a file request is issued by generic node k, exploratory signaling packets are generated by node k and forwarded to each node of the network through the advanced flooding scheme.
2. Exploratory signaling packets:
   a. traverse the network node by node;
   b. check the bandwidth availability of each link along the back route (or, more precisely, of the proper bandwidth pipe of each link); the minimum bandwidth availability (b_{min}^{hk}) over the path from h to k "defines" the cost of the path as $\left(b_{min}^{hk}\right)^{-1}$;
   c. each node forwards exploratory signaling packets by using the minimum bandwidth availability cost defined above; each exploratory signaling packet memorizes the Shortest Path Route from h to k, followed to get to node h in the reverse direction.
3. Generic node h of the network receives a number exploratory signaling packets containing: the file request from node k, the bandwidth availability and routing information. It sends back another set of packets (called location_info), which, traversing the network back, reports information about routing, bandwidth availability and location of the portions to node k.
4. Node k: receives all the location_info packets that contain all information about: location of the portions, best route and bandwidth availability; if there is not residual bandwidth availability also for just one portion, CAC acts and the object request is aborted, otherwise, node k: selects the portions to download, the nodes where the selected portions are located and the downloading rates (i.e. either the committed ones, if possible, or assigning the residual bandwidth on the path) by minimizing the cost function proposed in the next section within the Local Controller. The node k selects the best route to get to the selected locations and forwards a resource_confirmation packet, which reserves the resources over the selected shortest path, informs the nodes about portions to download and rates and authorizes file downloading. If the resources should not be available any longer over the path, CAC acts again and the file request is blocked.

PERFORMANCE OPTIMIZATION MODEL

The following definition should help formalize the problem. Definitions reported are aimed at focusing on the main content of the paper (file downloading and distribution), so service option and traffic class indexes are neglected for the sake of clarity. The formal description, in practice, assumes just one service option defined by its downloading rate and just one traffic class, whose bandwidth is the same of the physical link capacity. The same assumption is kept in the performance analysis.

Local Controller

Definitions: I, number of objects to share; i, object identifier $1 \le i \le I,\ i \in \mathbb{N}$; N, number of nodes of the network; L^i, dimension of the $i-th$ object; J^i, number of portions that compose the $i-th$ object; D^{ij}, dimension of the $j-th$ portion of the $i-th$ object; R^{ki}, number of requests from node k regarding $i-th$ object (traffic matrix); C^{lm}, physical capacity of link (lm); b^{lm}, residual bandwidth available over link (lm); pd^{lm}, propagation delay over link (lm); dr, committed downloaded rate; $Path(hk)$, sequence of links (defined as couple of nodes) composing the path from h to k; $C_{hk}^{min} = \min_{(lm) \in Path(hk)} C^{lm}$, physical capacity bottleneck for $Path(hk)$; $b_{min}^{hk} = \min_{(lm) \in Path(hk)} b^{lm}$, minimum residual bandwidth available for $Path(hk)$; $\tau^{hk} = \sum_{(lm) \in Path(hk)} pd^{lm}$, propagation delay for $Path(hk)$; M^k, storage capacity of node k;

$$\phi_k^{ij} = \begin{cases} 1, \text{ if } j\text{-th portion of } i\text{-th object is present at node } k \\ 0, \text{ otherwise} \end{cases};$$

$$\boldsymbol{\varphi}^i = \begin{pmatrix} \phi_1^{i1} & \cdots & \phi_1^{iJ^i} \\ \vdots & \ddots & \vdots \\ \phi_N^{i1} & \cdots & \phi_N^{iJ^i} \end{pmatrix},$$ distribution matrix of $i-th$ object;

$$\mathbf{A}_k^i = \begin{pmatrix} A_{1k}^{i1} & \cdots & A_{1k}^{iJ^i} \\ \vdots & \ddots & \vdots \\ A_{Nk}^{i1} & \cdots & A_{Nk}^{iJ^i} \end{pmatrix},$$ matrix defining the decisions of node k concerning the portions of $i-th$ object;

$$A_{hk}^{ij} = \begin{cases} 0, \text{ if } \phi_k^{ij} = 0 \\ \{0,1\}, \text{otherwise} \end{cases}.$$

The estimation of the capacity b_{min}^{hk} is topical to get an estimation of the downloading time of each single portion. In practice, b_{min}^{hk} is verified through the signaling protocol at the beginning of the operation. Being the approach followed within a bandwidth reservation framework and knowing the downloading rate that is considered fixed for the overall downloading operation, when the signaling packets flow through the network, they can verify exactly which is the residual bandwidth on each link and take the minimum value. The approach might be used also in best effort networks with self-regulating TCP connections. In this case signaling should perform a measure of the average bandwidth still available. Details of that should include the format of signaling packets, the presence of time stamps and any other information that can help estimate the bandwidth available. Being:

$$f^{hk} = \begin{cases} dr, \text{ if } b_{min}^{hk} \ge dr \\ b_{min}^{hk}, \text{ otherwise} \end{cases} \quad (1)$$

the bandwidth assignable to the file download request on the $Path(hk)$.

The contribution of node h to downloading time of the $i-th$ object "seen" by node k is defined by the function $T_{hk}^i(\mathbf{A}_k^i)$ in (2).

$$T_{hk}^i(\mathbf{A}_k^i) = \sum_{j=1}^{J^i} \left[\frac{D^{ij}}{f^{hk}} \cdot A_{hk}^{ij} \right] + \tau^{hk} \cdot \Psi(\mathbf{A}_k^i) \quad (2)$$

where

$$\Psi(\mathbf{A}_k^i, h) = \begin{cases} 1, \text{ if } \sum_{j=1}^{J^i} A_{hk}^{ij} \ge 1 \\ 0, \text{ otherwise} \end{cases} \quad (3)$$

It means there is at least one portion of the $i-th$ object downloaded from node h.

The downloading time of the $i-th$ object "seen" by node k is defined as in (4).

$$T_k^i(\mathbf{A}_k^i) = \max_h \left[T_{1k}^i(\mathbf{A}_k^i), \ldots, T_{hk}^i(\mathbf{A}_k^i), \ldots, T_{Nk}^i(\mathbf{A}_k^i) \right] \quad (4)$$

Node k takes decisions about which portions to download from where by minimizing $T_k^i(\mathbf{A}_k^i)$ under the variable $\mathbf{A}_k^i$. In other words, it defines the matrix $\mathbf{A}_k^i$, which minimizes the cost $T_k^i(\mathbf{A}_k^i)$ with the constraint in (5).

$$\sum_{h=1}^{N} A_{hk}^{(ij)} \ge \Lambda, \forall j \quad (5)$$

where Λ defines the redundancy, i.e. the minimum number of nodes from which each portion needs to be downloaded. If $\Lambda = 1$ (as done in the performance analysis), it means that that each portion of the $i-th$ object must be downloaded at least from one node. The Local Controller layer acts at "object request" time scale.

Network Controller

Supposing a time C of consecutive network operation, it is feasible to have an overall centralized network optimization, acting with period C, that "decides" file portion exchanges on the basis of the performance obtained in the period C and also the new bandwidth portions for traffic class. It important to give some detail about portion exchange in this context. In more detail, defining:

$P^{k,ji}(C)$, number of requests from node k regarding $j-th$ portion of $i-th$ object within the observation interval C (traffic matrix in the period C);

$dl_r^{hk,ji}$, identifier of the $r-th$ download from node h to node k regarding $j-th$ portion of $i-th$ object within the observation interval C;

$\beta^{hk}(d_r^{hk,ji})$, bandwidth allocated by the Local Controller (or average capacity measured in case of best effort) for $r-th$ download from node h to node k regarding $j-th$ portion of $i-th$ object within the observation interval C; in case of best effort network, the measure is performed by the ratio among the portion dimension and the real downloading time.

The average bandwidth availability from node h to node k for the observation interval C is:

$$\overline{\beta}^{hk} = \frac{1}{I}\sum_{i=1}^{I}\frac{1}{J^i}\sum_{j=1}^{J^i}\frac{1}{P^{k,ji}(C)}\sum_{r=1}^{P^{k,ji}(C)}\beta^{hk}(d_r^{hk,ji}) \quad (6)$$

Similarly as the previous case, the average downloading time for node k and object i derives from (7) and (8):

$$\overline{T}_{hk}^{i}(\mathbf{A}_k^i(\boldsymbol{\varphi}^i)) = \sum_{j=1}^{J^i}\left[\frac{D^{ij}}{\overline{\beta}^{hk}}\cdot A_{hk}^{ij}(\phi_k^{ij})\right] + \tau^{hk}\cdot\Psi(\mathbf{A}_k^i) \quad (7)$$

$$\overline{T}_{k}^{i}(\mathbf{A}_k^i(\boldsymbol{\varphi}^i)) = \max_h\left[\overline{T}_{1k}^{i}(\mathbf{A}_k^i(\boldsymbol{\varphi}^i)),\ldots,\overline{T}_{hk}^{i}(\mathbf{A}_k^i(\boldsymbol{\varphi}^i)),\ldots,\overline{T}_{Nk}^{i}(\mathbf{A}_k^i(\boldsymbol{\varphi}^i))\right] \quad (8)$$

The average downloading time for the overall network may be written as:

$$\overline{T} = \sum_{i=1}^{I}\sum_{k=1}^{N}\overline{T}_k^i(\mathbf{A}_k^i(\boldsymbol{\varphi}^i)) \quad (9)$$

with the constraints (5) and

$$\sum_{i=1}^{I}\sum_{j=1}^{J^i}D^{ij}\cdot\phi_k^{ij} \le M^k,\ \forall k,\ 1\le k\le N \quad (10)$$

stating the limitation on the memorization capacity of each node.

If the Network Controller acts also on the bandwidth pipes, a possible choice is to assign for each $Path(hk)$ a bandwidth that considers both the number of object requests and the number of rejected calls. The computation is similar to (6) but also not accepted connections should be included as well as the constraint . If $\overline{\xi}^{hk}$ is the computed bandwidth value under the physical constraint $\overline{\xi}^{hk} \le C_{hk}^{min}$, the object portion relocation can proceed as envisaged above from (7) on, but substituting $\overline{\beta}^{hk}$ with $\overline{\xi}^{hk}$.

Planning Controller

A complete optimization of the network may be reached at Planning Layer where also the dimensions of each portion may be modified. In this case the evaluation period Q should be much larger than at network Control Layer and Q may have the scale of weeks or months. The cost function to be used is similar as in (9) but also the variables D^{ij} are object of the optimization process under the additional constraint $\sum_{j=1}^{J^i}D^{ij} = L^i$. To complete the analysis, it is also possible to extend the optimization process at the channel and capacity acting on the constraints C_{hk}^{min} and $M^{(k)}$.

The authors will focus on the Local Control leaving the Network Controller and the Planning Controller to future work.

PERFORMANCE EVALUATION

The performance evaluation is limited to check the behavior of the Local Controller. In practice, the aim is showing the performance of the object download decision process. The considered overlay network, which implements the advanced flooding signaling to get information about position of the object portions, is reported in Fig. 2.

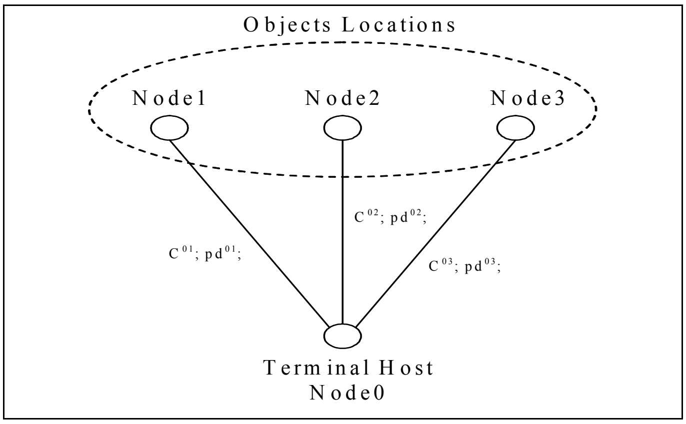

Figure 2. Overlay Network Considered.

Only Node0 is considered as terminal host. The objects are located in Nodes from 1 to 3. The following data have been used in the tests: number of objects to share: I=1; number of nodes in the network N=4; dimension of the object: L^1=5 Mbit; number of portions that compose the object: J^1=5; dimension of the object portions: D^{1j}=1

Mbit $\forall j \in \left[1,...,J^1\right]$; number of requests from node 0 regarding object 1 (traffic matrix): R^{01}=1; the physical capacity of links (10) and (30), equivalent to the physical capacity bottleneck of $Path(10)$ and $Path(30)$, is $C_{10} = C_{30}$=(1+Bandwidth Increase) Mbit/s, where the parameter "Bandwidth Increase" is varied in the tests performed; the physical capacity of link (20), C_{20}= 1 Mbit/s. The capacities of the links are always considered fully available and coincident with the residual bandwidth available over the network links. The committed downloading rate is supposed to be equal to the overall bandwidth available over the paths. The propagation delay over each link is fixed and equal to 10 ms.

Three local control strategies are compared in the tests: the algorithm proposed in Section III and called "Opt" in the following figures, a "Blind" method where all the portions of the object, if present, are downloaded from all the nodes, and a "Heuristic" method where downloading of portions, when present, is performed from the node reporting the largest value of the expression:

$$\left(D_h \cdot C_{min}^{hk\,-1} + \tau^{hk}\right)^{-1} \quad (11)$$

Where D_h is the overall dimension of requested object's portions available at node h. It means that the downloading is performed considering "intelligent" distinction of the impact of the single portions on the average download time. In practice, the decision is taken as if the different portions contained in a node were just one single unit. In the tests, the object, composed of 5 portions, is distributed as reported in the following table:

Type	Node1	Node2	Node3
Full Distribution	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5
Random	Random assignment		
Case 1	1,2,3,4	5	5
Case 2	1,2,3,4	4,5	4,5
Case 3	1,2,3,4	3,4,5	3,4,5
Case 4	1,2,3,4	2,3,4,5	2,3,4,5

Table 1. Object distribution considered in the tests.

All the figures show the Downloading Time versus the "Bandwidth Increase" parameter, acting over links (10) and (30). It is expressed in Mbit/s.

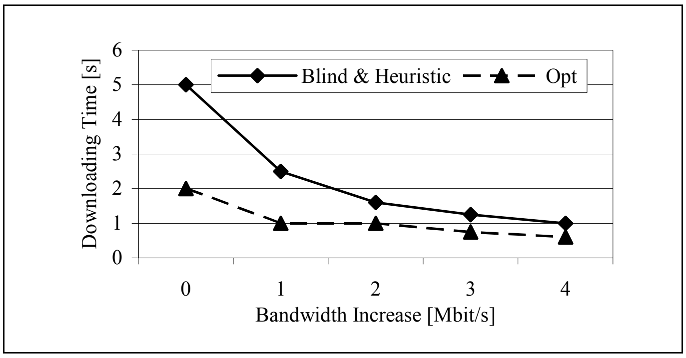

Figure 3. Downloading Time versus *Bandwidth Increase* [Full Distribution case].

The results of the Full Distribution (Fig. 3) case highlight the main advantage of the performance optimization model proposed in this paper. The procedure defined allows the simultaneous download of different portions of the object without duplications and, in particular, uses the fragmentation to optimize the downloading time. Actually, the Downloading Time versus the "Bandwidth Increase", when "Opt" is used, is much lower than the other two cases, which offer exactly the same performance in this case. Also in the Random Distribution case (Fig. 4) the "Opt" solution experiences the lowest Downloading Time.

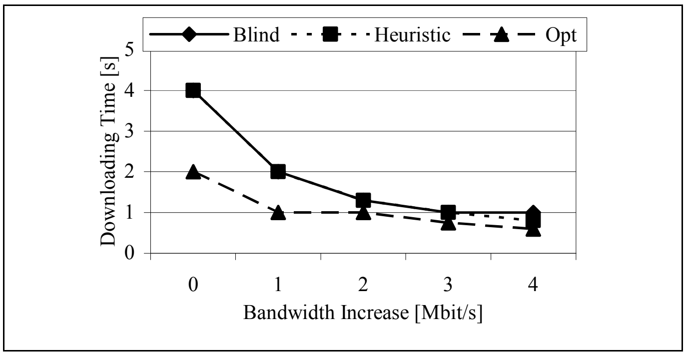

Figure 4. Downloading Time versus *Bandwidth Increase* [Random Distribution case].

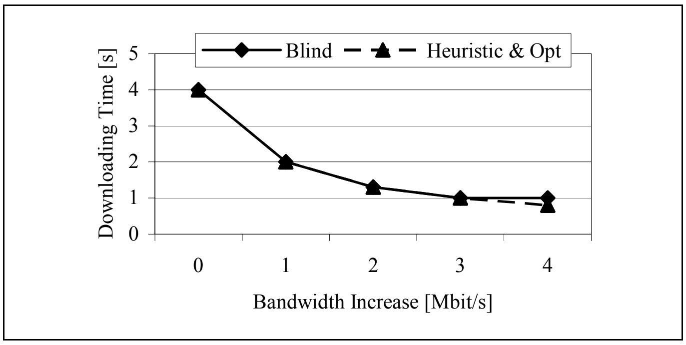

Figure 5. Downloading Time versus Bandwidth Increase [Case 1].

In the Case 1 (Fig. 5), the Downloading Time is practically the same, even if, when the "Bandwidth Increase" is high, "Heuristic" and "Opt" (undistinguished in this case) have better performance than the "Blind" method. This result is due to the distribution of the portions (see Table 1 – Case 1) that are concentrated in node 1, so removing the advantage introduced by the new method. Actually, there is only one choice and it is followed by all the schemes except for "Blind". When the information is more distributed through the network all the features of "Opt" are used and it offers a very satisfying performance (Figures 6 and 7).

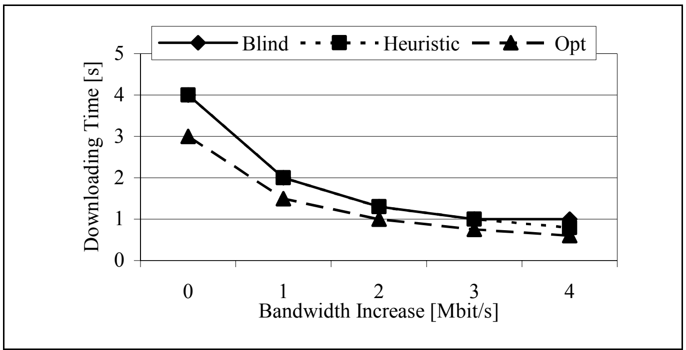

Figure 6. Downloading Time versus *Bandwidth Increase* [Full Distribution case].

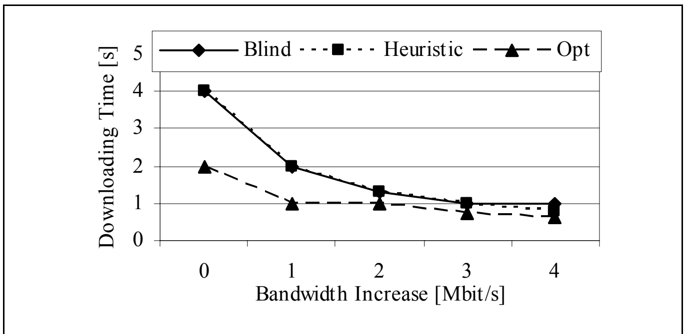

Figure 7. Downloading Time versus Bandwidth Increase [Case 3 and Case 4].

CONCLUSIONS

The paper has proposed a control architecture composed of three layers within the framework of "grid computing" networks: Local, Network and Planning Controller. It is based on an "advanced flooding" signaling algorithm that transmits information about the object portions' position and bandwidth availability along the paths. The performance evaluation has highlighted the basic mechanisms used by the Local Controller and has given a first idea about the advantages and drawbacks of the proposal. Both the Network Controller and the Planning Controller is put off for future research.

REFERENCES

[1] D. C. Verma, "Content Distribution Networks, An Engineering Approach," John Wiley & Sons, Inc., New York 2002.

[2] SETI@Home, The Search for Extraterrestrial Intelligence. Available from: http://setiathome.ssl.berkeley.edu.

[3] Freenet. Available from: http://freenet.sourceforge.org.

[4] NeuroGrid P2P Search. Available from: http://www.neurogrid.net/.

[5] M. Parameswaran, A. Susarla, A.B. Whinston. 2001. "P2P Networking: An Information Sharing Alternative," Computer Journal, IEEE Computer Society, July, vol. 7, no. 34, pp. 31-38.

[6] A. Stavrou, D. Rubenstein, S. Sahu, "A Lightweight, Robust P2P System to Handle Flash Crowds," in Proceedings of IEEE ICNP 2002, Paris, France, November, 2002.

[7] T. Stading, P. Maniatis, M. Baker, "Peer-to-Peer Caching Schemes to Address Flash Crowds," in Proceedings of IPTPS'02, Cambridge, MA, March 2002.

[8] M. Ripeanu, I. Foster, A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-scale Peer-to-Peer Systems and Implications for System Design," IEEE Internet Computing, vol. 6, no. 1, pp. 50-57, Jan.-Feb. 2002.

[9] T. S. E. Ng, Y. Chu, S. G. Rao, K. Sripanidkulchai, H. Zhang, "Measurement-Based Optimization Techniques for Bandwidth-Demanding Peer-to-Peer Systems," In Proceedings of the Conference on Computer Communications 2003 (IEEE INFOCOM 2003), March 2003, pp. 2199-2209.

[10] Z. Ge, Daniel R. Figueiredo, S. Jaiswal, J. Kurose, D. Towsley, "Modeling Peer-Peer File Sharing Systems," In Proceedings of the Conference on Computer Communications 2003 (IEEE INFOCOM 2003), March 2003, pp. 2188-2198.

[11] X. Yang, G. de Veciana, "Service Capacity of Peer-to-Peer Networks," In Proceedings of the Conference on Computer Communications 2004 (IEEE INFOCOM 2004), March 2004, pp. 2242-2255.

[12] V. Kalogeraki, F. Chen, "Managing Distributed Objects in Peer-to-Peer systems," IEEE Network, vol. 18, no. 1, Jan 2004, pp. 22-29.

[13] K. Ross, "Multiservice Loss Models for Broadband Telecommunication Networks," Springer Verlag, Berlin, 1995.