Resource Allocation over a GRID Military Network

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, dama, mopa}@dist.unige.it

Luciano Mancuso, Annamaria Raviola
Selenia Communications S.p.A.
via Pieragostini 80
16151 Genoa
ITALY

{Luciano.Mancuso, Annamaria.Raviola}@seleniacomms.com

ABSTRACT

A telecommunication network is composed of a number of nodes. Each node can be either a terminal host or a web cache location. Objects are downloaded throughout the network among the 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 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 means that a single node can have only a part of the file. 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 consider the mentioned issues. 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) and, after an "advanced flooding" signaling query to get informed about the object portions' position, decides from which node (or nodes) each portion needs to be downloaded; the Network Controller may change the distribution of the object portions among the nodes, 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 capacities. The order of magnitude of its intervention is weeks/months.

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.

Bisio, I.; Marchese, M.; Mongelli, M.; Mancuso, L.; Raviola, A. (2006) Resource Allocation over a GRID Military Network. In *Military Communications* (pp. 6-1 – 6-20). Meeting Proceedings RTO-MP-IST-054, Paper 6. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

I. 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 a request 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, where there is a clear distinction between the consumer and the producer of services: the client asks the server to access resources (i.e. files, machine times, distributed applications) and the server processes the workload assuring, if possible, the required resources. 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 information and functions, and different surrogate server sites, where the material is duplicated. Surrogate servers are used to find new paths to retrieve the information so avoiding possibly congested routes. 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 host, 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. If a particular file is located into two remote peers, part of this file may be downloaded from one peer and part from the other peer. 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.

Summarizing: in client-server applications, the server is the only repository of information. It may be negative because accessing the server can create bottlenecks, both in links and machine processing, but it allows companies and entities to store important information and to make it available to others on payment. CDNs follow the same philosophy. Information multiplication allows to alleviate traffic burden but increases the maintenance costs. P2P approach has many advantages but it is not safe for important information. 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 strategic network, for memory saving and for safety reasons, each object is composed of portions and not all 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 information. 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 introduction to performance improvement methods over a "grid computing" structure as state of the art for this paper. Section III contains the description of the control architecture. The formal model for the performance optimization is reported in Section IV. Section V presents a preliminary performance evaluation and Section VI shows the conclusions and possible ideas for future work.

II. THE STATE OF THE ART

Taking as reference the "grid computing" networks stated in the introduction, where information is divided into portions localized in remote nodes, the essential motivations for performance decrease may be [1]: a misconfigured or a poorly managed network, software glitches affecting the network protocol, interfering cross-traffic, unsuitable position of information. Assuming an efficient management of network and software, congestion (both of links and processing power) and position of information are the two real motivations for performance flaws.

In telecommunication networks in general there are methods to improve the performance of end-to-end communications: QoS (Quality of Service) mechanisms; QoS Management Functions; CDNs design. Additionally, many good ideas may be taken from P2P world. CDNs and P2Ps have been the object of the introduction, while QoS mechanisms and management functions are revised in the following with a constant eye on the subject of the paper.

Quality of Service mechanisms

A QoS-based service derives from reliable physical layers (including, in this case, layer 2 and layer 1) that can offer specific services to the upper layers. Fig. 1 reports a graphical model of the relation between lower (physical) and higher layers. The connections (or bundles of them) are forwarded down to a physical interface that transports the information along a channel.

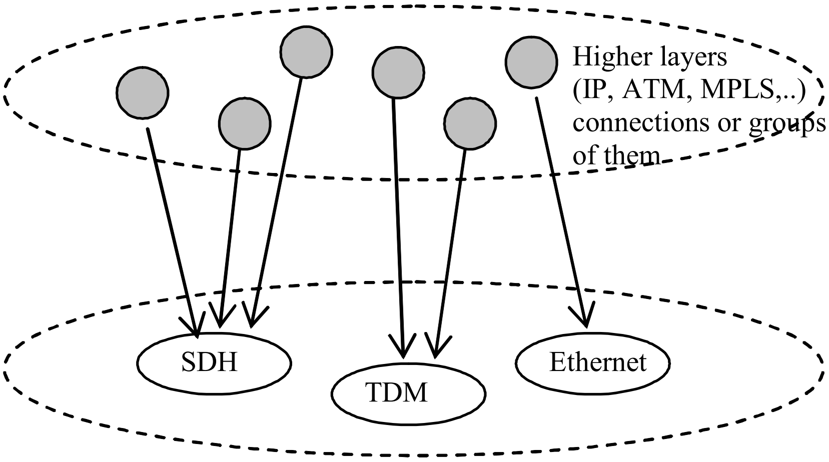

Fig. 1. Higher layers over transport technology.

In the following, a list, which should include the main technologies [9] available in the market to provide QoS, is briefly summarised, with particular attention to the capability of marking a specific traffic flow.

ATM

It is connection-based. Information is packetized to fixed length packets, called cells, of 53 bytes (5 for header and 48 for information). ATM identifies and assures a single user flow or an aggregate of flows by using two fields contained in each cell header: Virtual Channel (VC) and Virtual Path (VP). It means that a path dedicated to specific traffic (e.g., downloading a particular file from one source) may be defined from the source to destination.

ATM has been built to guarantee QoS and uses call admission and congestion control schemes properly designed for the aim. The scientific literature of these last fifteen years is very rich concerning resource allocation and reservation schemes that use statistical multiplexing gain without penalizing users.

QoS-IP

Best Effort

Native IP is connectionless and offers best-effort services. The service received by a user depends on the network load. Queuing managing within routers is, essentially, FIFO (First In First Out). Actually it is the most used solution also in "grid computing" networks for now.

Concerning the tools to identify a traffic flow, IP needs to be differentiated between version 4 and version 6.

IPv4 offers two ways to mark traffic:

a vector composed of the following fields: "IP source address", "IP destination address", "Protocol", all contained within the IPv4 header; "TCP/UDP source port" and "TCP/UDP destination port", contained in the TCP/UDP header.

ToS field (8 bits in the IPv4 header), whose first six bits define the DSCP (DiffServ Code Point) field.
Packets with the same identifier need to be treated coherently by each router.

IPv6 may use two fields directly in the IP header:

Flow Label field (20 bits).

Traffic Class field (8 bits), functionally equivalent to IPv4 ToS field and containing the DSCP field.

The difference between IPv4 and IPv6 is that IPv6 can mark a flow through the flow label, as well as an aggregate of flows (similarly to VC/VP in ATM), while IPv4 can identify either a limited number of aggregates through the DSCP field, up to a theoretical maximum of 2^8 , or a specific flow (but not an aggregate of them) through the vector "IP source address, IP destination address, Protocol, TCP/UDP source port, TCP/UDP destination port".

Flow identification is only a starting point for QoS guarantee. Two paradigms have been proposed to match the market QoS request: Integrated Services and Differentiated Services.

The Integrated Services [9, 10]. It is based on the concept of flow defined as a packet stream that requires a specified QoS level and it is identified by the vector "IP source address, IP destination address, Protocol, TCP/UDP source port, TCP/UDP destination port". Actually, the same scheme may be used in IPv6 by using the Flow Label field but there is no standardization about it. QoS can be reached by an appropriate tuning of different blocks: resource reservation, admission control, packet scheduling, and buffer management. A status concerning the different incoming flows must be maintained in the routers. It is very different from the best-effort approach provided by the Internet. Information about flows must be periodically updated and a specific resource reservation signaling system (RSVP [10, 11]) is used for this aim. IntServ needs to detect each single flow and both packet scheduling and buffer management act on per-flow basis. Although IntServ is ATM-like, it has not the tools provided by ATM because it is derived from an intrinsically best-effort and connectionless technology. The cost and the complexity of the control system increase with the number of flows.

The Differentiated Services (DS or Diffserv) [12, 13] have been proposed to cope with the scalability problem faced by Integrated Services. The solution uses the DSCP (DiffServ Code Point) field of the IP packet header by using the first six bits either of IPv4 ToS or of IPv6 Traffic Class. The 6 bits field DSCP specifies the forwarding behaviour that the packet has to receive within the DiffServ domain of each operator. The behaviour is called PHB (Per Hop Behaviour) and it is defined locally; i.e., it is not an end-to-end specification (as for RSVP) but it is strictly related to a specific domain. The same DSCP may have two different meanings in two different domains. Negotiations between all adjacent domains are needed to assure a correct end-to-end forwarding behaviour.

The DiffServ approach does not distinguish each user flow throughout the network. The traffic is classified and aggregated in different traffic classes, each of them individuated by a label provided by setting bits in the DSCP field. The identification is performed at the network edges. Within the network core, packets are managed according to the behaviour associated to the specific identification label. Fig. 2 shows an example of aggregation: after the DiffServ router the flows are aggregated depending on their label (light or dark grey in Fig. 2) and information about each single user (A, B, C or D, in Fig. 2) is completely lost.

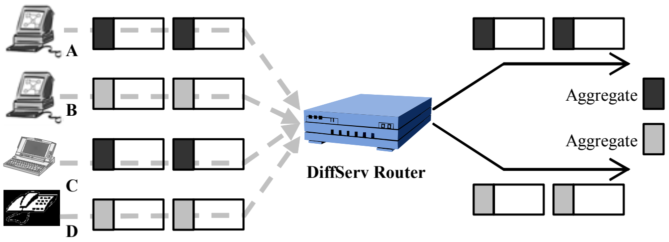

Fig. 2. DiffServ aggregation behavior.

The class selector PHB offers three forwarding priorities:

Expedited Forwarding (EF) characterized by a minimum configurable service rate, independent of the other aggregates within the router, and oriented to low delay and low loss services.

Assured Forwarding (AF) group recommended in reference [14] for 4 independent classes (AF1, AF2, AF3, AF4) although a DS domain can provide a different number of AF classes. Within each AF class, traffic is differentiated into 3 "drop precedence" categories. The packets marked with the highest drop precedence are dropped with lower probability than those characterized by the lowest drop precedence.

Best Effort (BE), which does not provide any performance guarantee and does not define any QoS level.

IPv6 can take the best of IP technology. Concerning flow identification, IPv6 offers the same capabilities of ATM but, to use all the features, it is necessary to import most of the control functions (part of the ATM approach) within IP. In practice, it would be interesting to keep IPv6 format to simplify interworking but to use control mechanisms (e.g. CAC, filtering and signalling) for each flow, as done in ATM.

MPLS

MPLS derives from the convergence between the IP world, which involves open standards and simplicity, and the ATM world, which is completely oriented to the QoS and uses traffic control techniques, as well as QoS aware routing. MPLS would like to "summarize" the best of both technologies. A label of 20 bits identifies the traffic. The label switching mechanism should be associated with control modules that allow guaranteeing a fixed level of quality. In principle, MPLS offers the same capabilities to mark flow as in ATM and IPv6. The MPLS label format is composed of 32 bits consisting of the following elements: the already mentioned Label Value; the field Exp, dedicated to future extensions; the bit S, aimed at indicating the presence of more labels; the Time to Live.

The entrance and the exit of a MPLS domain are governed by label edge routers (LERs) that generate and apply the labels when information enters the domain and remove them at the exit. The basic technology inside the domain is represented by label-switching routers (LSRs), which switch the traffic in dependence of a specific path, called label switched path (LSP), associated with the MPLS label. The label switched path defines the sequence of nodes where the traffic of a connection flows within the MPLS domain. The definition is performed through QoS-based traffic engineering techniques [15] aimed, for instance, at minimizing the number of hops, meeting bandwidth requirements, supporting precise performance requirements, bypassing potential points of congestion, directing traffic away from the default path selected or simply forcing traffic across certain links or nodes in the network.

QoS Management Functions

Meaningful QoS management functions are reported in the following.

Packet marking

The identification of packets so that they may receive a different treatment within the network is topical to guarantee QoS, ranging from a minimum priority-based service to quality assurance for a specific user. Single QoS–oriented technologies, presented before, show different methods to classify packets, as Flow Label and Traffic Class in IPv6, ToS and vector "IP source address, IP destination address, Protocol, TCP/UDP source port, TCP/UDP destination port" in IPv4, VPI/VCI in ATM, labels in MPLS.

Traffic Control (Shaping)

It is very important to guarantee performance requirements. Shaping policies limit flows to their committed rates (e.g. the flows need to be conformant with their traffic descriptors). If flows (or also single connections) exceed their bandwidth consumption specifications, the network, which has dimensioned resources in strict dependence with the declarations, cannot guarantee any specified QoS requirement.

Scheduling

Packet scheduling specifies the service policy at a queue within a node (for example, an IP router, an ATM and MPLS switch). In practice, scheduling decides the order to be used to pick the packets out of the queue and to transmit them over the channel. The main problem arises from the impossibility of assigning the committed bandwidth to a specific flow at each time instant. In general, bandwidth is allocated in average and most scheduling policies may guarantee the respect of this condition.

Flow control

In some cases, the bit rate entering the network may be ruled according to a congestion notification (ECN-Explicit Congestion Notification). Some protocols (e.g. TCP) consider packet loss as a congestion indication. Generally flow control is implemented end-to-end at the transport layer (but some mechanism are implemented at the application layer).

Resource Control

In the specific case of this paper, it is the decision from which location a particular portion should be downloaded. Alternatively, it may be the decision to use either a surrogate server (in CDNs) or a particular peer (in P2P approach) to access the needed object.

CAC

Call Admission Control decides whether a new connection request may be accepted or not. It is a powerful tool to guarantee quality because it allows limiting the load entering the network and verifying if enough resources are available to satisfy the requested performance requirements of a new call without penalizing the connections already in progress.

QoS Routing

Packet routing decisions are often taken with little or no awareness of network status and resource availability. This is not compatible with QoS provision. QoS routing needs to identify end-to-end paths where there are enough available resources to guarantee performance requirements in terms of metrics as loss, delay, call blocking, number of hops, reliability, as well as bandwidth optimisation.

Resource Distribution

In the framework of this paper, resource distribution means essentially “portions’ distribution”. In other words, after a measure of the performance, the position of the objects’ portions may be changed to alleviate downloading burden. In a wider sense, resource distribution may also mean bandwidth allocated to specific flows and services by the QoS technology as well as ATM, MPLS, IntServ, DiffServ. For example, downloading a specific file from a particular node may receive priority with respect to other flows.

Resource Planning

An accurate resource reservation to guarantee that traffic flows receive the correct service is strictly needed. Resource means, in this context, portions’ dimension, link and node capacity. Resource reservation acts on a larger time scale concerning network capacity planning and link dimensioning.

The intervention time scale of the mentioned functions is different. Fig. 3 shows a possible time mapping for them.

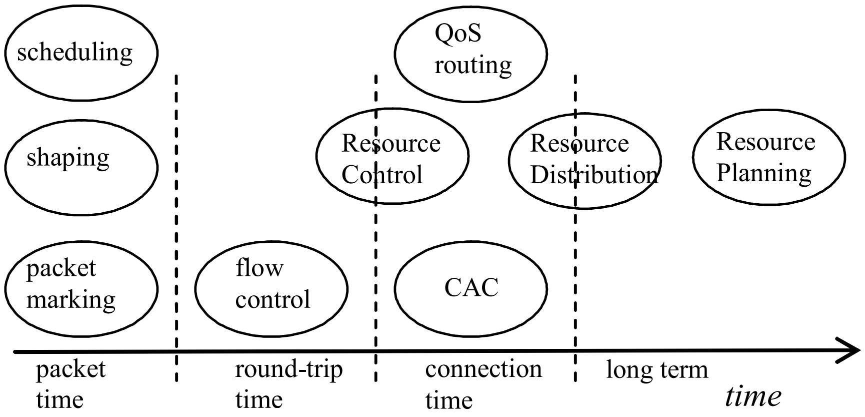

Fig. 3. QoS management functions versus time.

The control architecture (whose details will be provided in the next section), object of this paper acts through: Resource Control, Distribution and Planning, basing its operations on CAC, QoS Routing and scientific works developed in the framework of peer-to-peers networks. It is worth mentioning that early work on P2P are focused on characterizing the overall P2P system (e.g., request patterns, traffic volume, traffic categorization) and properties of shared online content as well P2P structure and dynamics (e.g., connectivity and users' behaviors). More recent research has focused on performance. Several works were developed in order to simulate and deploy such systems [16, 17, 18, 19]. The following issues are in particular hot topics of research.

The flooding-based search mechanisms used in regular P2P systems to individuate where an object is located cause a large volume of unnecessary traffic. A location aware topology matching technique is proposed in [20], aimed at alleviating this problem. On the other hand, [21] suggests a model for signaling messages emphasizing that signaling might significantly compromise performance. [22] uses measurements to optimize the selection of good downloading points, thus improving the overall system performance. However, it is important to note that few analytical models are used in this context. [23] is based on closed queuing systems. It models the system throughput on a average basis (i.e., the single user's performance is disregarded). On the other hand, [24] considers performance as a function of time at the granularity of a single P2P transfer session. In [23], an approximate method for the system throughput is studied in dependence of the chosen P2P architecture, including indexing mechanism and classes of peers. A measurement mechanism for optimizing the bandwidth on demand is studied in [22] through the analysis of real traffic traces. A novel P2P approach is proposed in [25]: it uses application and resource measurements collected in real time and provide a feedback to the self-organization of the network layers.

III. CONTROL ARCHITECTURE

Preliminary Observations

The proposed approach is though with reference to the works of [22, 23, 24, 25]. The system performance is captured through a mathematical description of the users' requests and available resources [23]. The current level of congestion is exploited by real time measurements and the decision variables are tuned accordingly [22, 25]. The single node performance is highlighted [24]. 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 nodes 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. 4. It heavily refers to the control function description reported in Section II. 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.

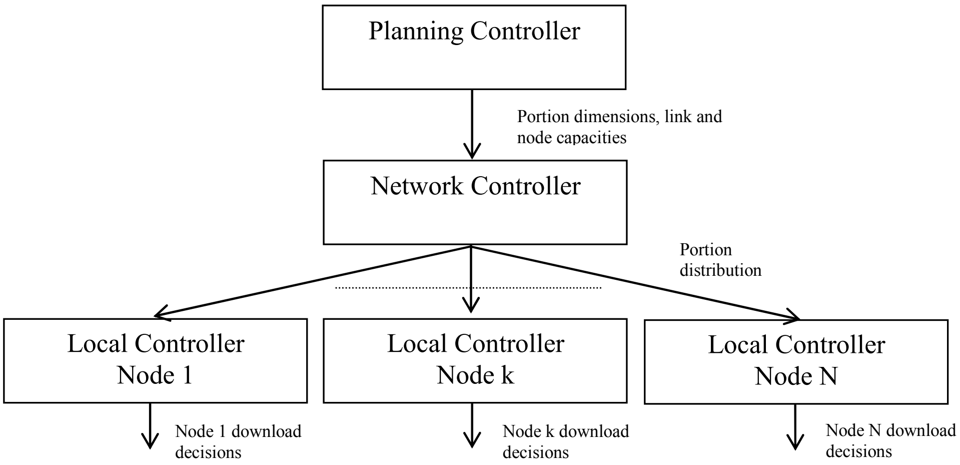

Fig. 4. Control Architecture

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" [26]. 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 previous section. 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 by itself but performs this operation having some knowledge about the "cost" (measured in terms of residual bandwidth) of the paths traversed. Signaling scheme is topical to get good performance. Future work will be dedicated to its study but, for now, it is out of the scope of this proposal. 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); minimum bandwidth availability (b_{min}^{hk}) over the path from h to k defines" the cost of the path as $\left(\frac{1}{b_{min}^{hk}}\right)$;

   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; bandwidth availability and routing information and 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 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 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.

IV. 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 $1 \leq i \leq I,\ i \in \mathbb{N}$;

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;

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

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

$A_k^i = \begin{pmatrix} A_{1k}^{i1} & \cdots & A_{1k}^{iJ^i} \\ \vdots & \ddots & \vdots \\ A_{Nk}^{i1} & \cdots & A_{Nk}^{iJ^i} \end{pmatrix}$ 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, but 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. The topic is very interesting and it will be the object of future research.

Being:

$$f^{hk} = \begin{cases} dr, \text{ if } b_{min}^{hk} \geq dr \\ b_{min}^{hk}, \text{ otherwise} \end{cases} \tag{1}$$

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 $T_{hk}^i(A_k^i)$ in (2).

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

where

$$\Psi(A_k^i, h) = \begin{cases} 1, \; if \sum_{j=1}^{J^i} A_{hk}^{ij} \geq 1 \\ 0, \; otherwise \end{cases} \tag{3}$$

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).

$$T_k^i(A_k^i) = \max_h \left[T_{1k}^i(A_k^i), ..., T_{hk}^i(A_k^i), ..., T_{Nk}^i(A_k^i) \right] \tag{4}$$

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

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

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 portion 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:

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

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

$$\overline{T}^i_{hk}(A^i_k(\varphi^i)) = \sum_{j=1}^{J^i}\left[\frac{D^{ij}}{\overline{\beta}^{hk}}\cdot A^{ij}_{hk}(\phi^{ij}_k)\right] + \tau^{hk}\cdot\Psi(A^i_k) \qquad (7)$$

$$\overline{T}^i_k(A^i_k(\varphi^i)) = \max_h\left[\overline{T}^i_{1k}(A^i_k(\varphi^i)),\ldots,\overline{T}^i_{hk}(A^i_k(\varphi^i)),\ldots,\overline{T}^i_{Nk}(A^i_k(\varphi^i))\right] \qquad (8)$$

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

$$\overline{T} = \sum_{i=1}^{I}\sum_{k=1}^{N}\overline{T}^i_k(A^i_k(\varphi^i)) \qquad (9)$$

with the constraints (5) and

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

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 ξ^{hk} is the computed bandwidth value under the physical constraint $\xi^{hk} \le C^{min}_{hk}$, the object portion relocation can proceed as envisaged above from (7) on, but substituting $\overline{\beta}^{hk}$ with ξ^{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^{min}_{hk} and $M^{(k)}$.

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

V. 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 is found with the advanced flooding signaling as well as the objects portions position, is reported in Fig. 5.

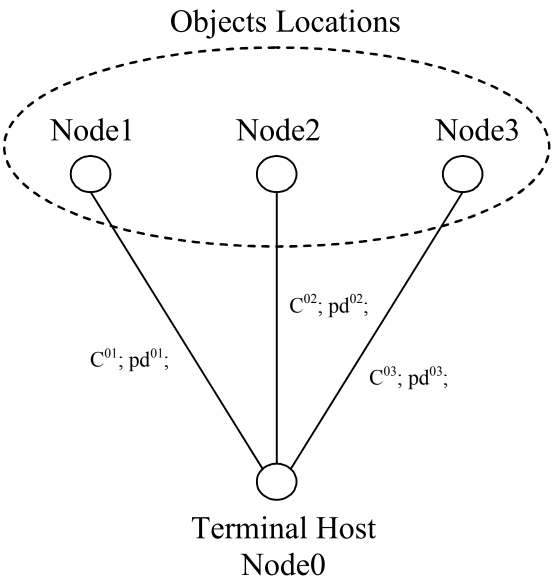

Fig. 5. Overlay Network Considered.

Only the Node0 is considered as terminal host. The objects are located in Nodes from 1 to 3 and the following data have been assumed in the tests:

number of objects to share: $I=1$;

number of nodes of 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 portion of the object $D^{1j}=1\ Mbit \quad \forall j \in \lfloor 1,...,J^1 \rfloor$;

number of requests from node 0 regarding the object 1 (traffic matrix): $R^{01}=1$;

the physical capacity of link (10), which is equal to the physical capacity of the bottleneck for the $Path(10)$, is C_{10} *=(1+Bandwidth Increase) Mbit/s*, where the "*Bandwidth Increase*" parameter is varied in the tests performed;

the physical capacity of link (20), which is equal to the physical capacity of the bottleneck for the $Path(20)$, is $C_{20} = 1\ Mbit/s$;

the physical capacity of link (30), which is equal to the physical capacity of the bottleneck for the $Path(30)$, is C_{30} *=(1+Bandwidth Increase) Mbit/s*.

The capacities of the link are considered always fully available thus they are 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 of the overlay network, taken as reference, is fixed and equal to *10 ms*.

Three kind of strategies are compared in the tests: the algorithm, related to the local controller, proposed in previous section 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 an "Heuristic" method where downloading of portions, if present, is done from the node reporting the largest value of the expression:

$$\frac{1}{\frac{D_h}{C_{\min}^{hk}} + \tau^{hk}} \tag{11}$$

Where D_h is the overall portion of the object available in node $h-th$. It means that the downloading of portions is done from the node with the minimum total delay for the download of the object portions given the available bandwidth in the path.

In the tests, the object, composed of 5 portions, is distributed as reported in the following table:

Table 1. Object distribution considered in the tests.

Type	Node1	Node2	Node3
Full Distribution	1,2,3,4,5	1,2,3,4,5	1,2,3,4,5
Random Distribution	Defined with a 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

All the figures show the Downloading Time versus the "*Bandwidth Increase*" parameter, different by zero in the links *(10)* and *(30)*, expressed in *Mbit/s.*

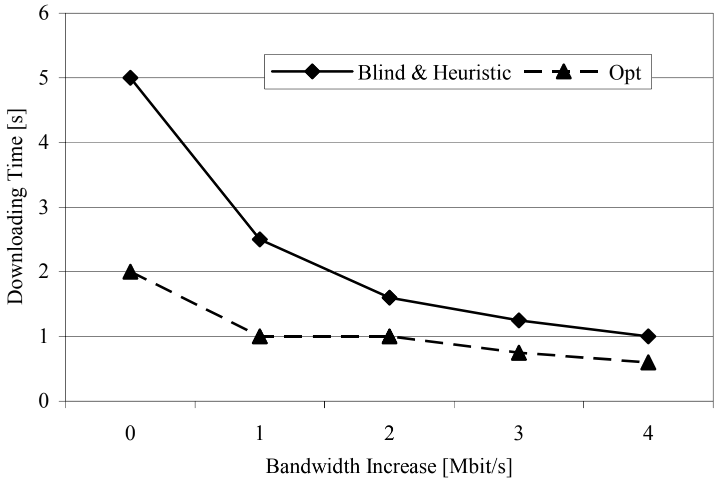

Fig. 6. Downloading Time versus *Bandwidth Increase* [Full Distribution case].

The results of Full Distribution (Fig. 6) case highlight the main advantage of the performance optimization model proposed in this work. The procedure defined allows the simultaneous download of different portions of the object without portions download duplications. The Downloading Time versus the "*Bandwidth Increase*", when the "Opt" strategy is used is, at least, the half of the Downloading Time performed by the others methods.

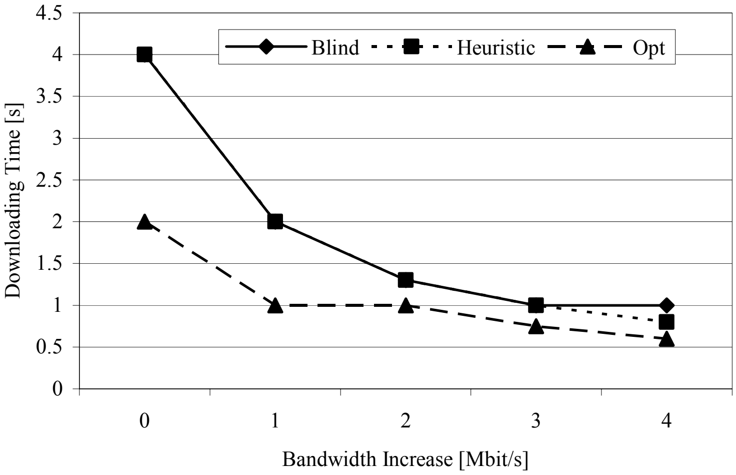

Fig. 7. Downloading Time versus *Bandwidth Increase* [Random Distribution case].

In the Random Distribution case (Fig. 7) the "Heuristic" method has better performance than the "Blind" when the "*Bandwidth Increase*" (thus bandwidth capacities available in the network) is high. The "Opt" solution experiences always the lowest Downloading Time.

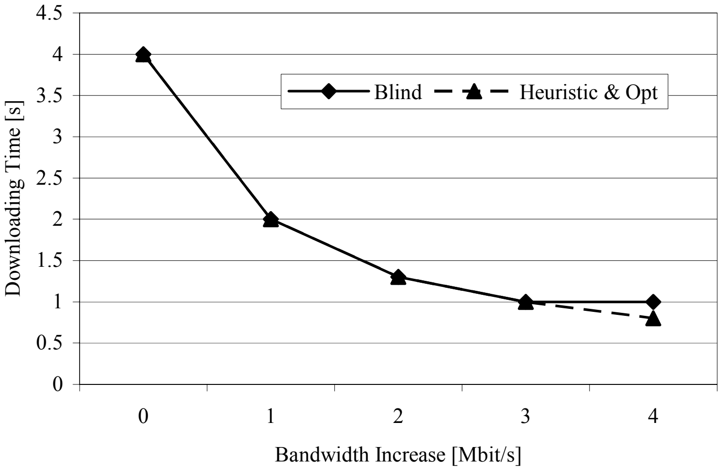

Fig. 8. Downloading Time versus *Bandwidth Increase* [Case 1].

In the Case 1 (Fig. 8), the Downloading Time is often the same but when the "*Bandwidth Increase*" is very high "Heuristic" and "Opt" (undistinguished in this case) have better performance than the "Blind" method. This result is due to the distribution of the object (see Table 1 – Case 1) that is concentrated in the node 1 thus the simultaneous download is not allowed because no portion of the object is replicated in the others nodes. When the distribution of the object in the network is more uniform (case 2, 3 and 4), the simultaneous download is allowed and the "Opt" strategy has always better performance (Figures 9 and 10).

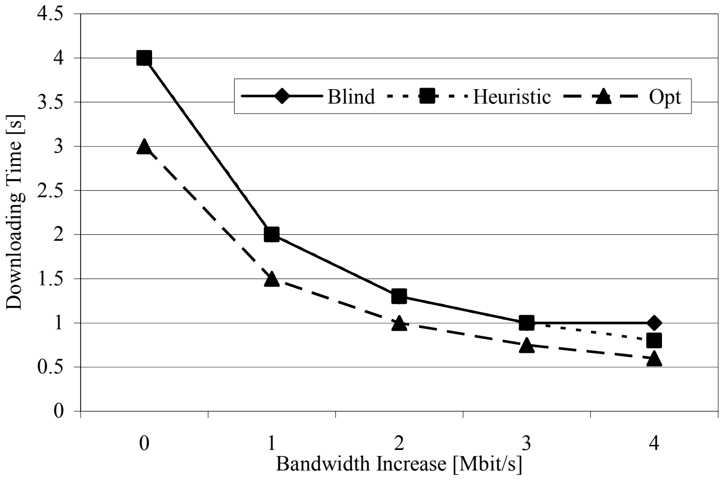

Fig. 9. Downloading Time versus *Bandwidth Increase* [Case 2].

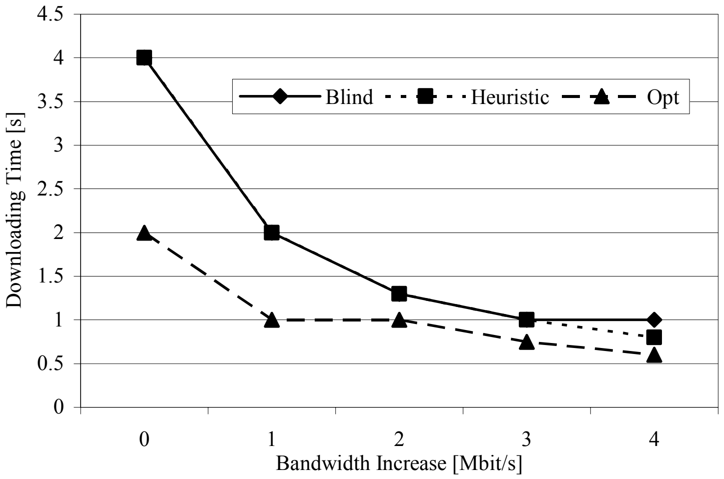

Fig. 10. Downloading Time versus *Bandwidth Increase* [Case 3 and Case 4].

All the figures show that the "Blind" method represents an upper bound of the performance and the "Heuristic" method is often overlapped in the cases considered in the tests. The performance increase obtained by the optimization algorithm ("Opt") is clear in all the figures reported because the functional cost (reported in (4)) allows simultaneous download from different node of the network.

VI. CONCLUSIONS

The paper has proposed a control architecture composed of three layers within the framework of "grid computing" networks: Local, Network and Planning Controller. The Local Controller acts locally to each node and decides which portions to download from which nodes. It is based on an "advanced flooding" signaling algorithm that transmits information about the object portions' position and bandwidth availability along the paths. The problem is modeled through a mathematical formulation and a specific cost function, which takes into account all the necessary details is proposed for each controller as well as a minimization procedure. 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. Network Controller and Planning Controller may change the dimension of each single portion and increase/decrease the physical links and node capacities. Its detailed description 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] H. J. Chao, X. Guo, "Quality of Service Control in High-Speed Networks," John Wiley & Sons, LTD: Chichester, England, 2002.

[10] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReserVation Protocol (RSVP) - Version 1 Functional Specification," *IETF, RFC 2205*, September 1997.

[11] J. Wroclawski, "The use of the Resource Reservation Protocol with the Integrated Services," *IETF, RFC 2210*, September 1997.

[12] IETF Differentiated Services Working Group, http://www.ietf.org/html.charters/diffserv-charter.html, last access June 2004.

[13] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services," *IETF RFC 2475*, December 1998.

[14] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group," *IETF RFC 2597*, June 1999.

[15] E. Crawley, R. Nair, B. Jajagopalan, H. Sandick, "A Framework for QoS-based Routing in the Internet," *IETF RFC 2386*, August 1998.

[16] O. Babaoglu, H. Meling, A. Montresor, “Anthill: A Framework for the Development of Agent-Based Peer to Peer Systems,” In Proceedings of 22nd International Conference on Distributed Computing Systems (ICDCS'02), Wien, Austria, July 2002, pp. 15-22.

[17] N. Minar, R. Burkhart, C. Langton, M. Askenazi, “The Swarm Simulation System, A Toolkit for Building Multi Agent Simulations,” Technical report, Swarm Development Group, June 1996. Available from: http://www.swarm.org.

[18] B. Horling, V. Lesser, R. Regis, “Multi-Agent System Simulation Framework,” In Proceedings of the 16th IMACS World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation. EPFL. Lausanne, Switzerland. August 2000.

[19] Q. He, M. Ammar, G. Riley, H. Raj, R. Fujimoto, “Mapping Peer Behavior to Packet-level Details: A Framework for Packet-level Simulation of Peer-to-Peer Systems,” In Proceedings of the 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS 2003). October 2003. pp. 71-78.

[20] L. Yunhao, X. Liu, X. Li, L. M. Ni, X. Zhang, “Location-Aware Topology Matching in P2P Systems,” In Proceedings of the Conference on Computer Communications 2004 (IEEE INFOCOM 2004), March 2004, pp. 2220-2230.

[21] J. Ritter, “Why Gnutella can’t Scale. No, Really,” in URL http://www.tch.org/gnutella.html, 2001.

[22] 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.

[23] 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.

[24] 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.

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

[26] K. Ross, “Multiservice Loss Models for Broadband Telecommunication Networks,” Springer Verlag, Berlin, 1995.