

End to End Network Quality of Service in Grid environments: the QoSINUS approach

Pascale Vicat-Blanc Primet and Fabien Chanussot

INRIA-ENS LIP RESO

46, allée d'Italie

69 007 Lyon, France Telephone: +334 72 72 88 02

Fax: +334 72 72 80 80 Email: Pascale.Primet@ens-lyon.fr

Abstract—Grids rely on a complex interconnection of IP domains that may exhibit changing performance characteristics and may offer different quality of service (QoS) facilities. As lot of distributed applications require controlled communication performance, a Grid Service allowing applications to specify QoS objectives and controlling the realization of these goals in a multi IP domains context is required. This paper presents QoSINUS, a grid network service that dynamically allocates the network resources to packets belonging Grid flows in order to match their individual requirements under different network load conditions. This service aims at increasing the end to end performance of individual applications as well as optimising the grid network resource utilization. To illustrate the principles of the service, we have implemented QoSINUS with the active network technology and interfaced it with a DiffServ network. The QoSINUS proposes a simple API and a collection of independent agents, located at wide area network edges. Results of experiments conducted in the eToile French grid test bed show the validity of this approach.

I. INTRODUCTION

The purpose of Computational Grids is to aggregate a large collection of shared resources (computing, communication, storage, information) to build an efficient and high performance computing environment for data-intensive or computing-intensive applications [1]. However, the Grid network cloud is a complex aggregation of heterogeneous domains offering various performances guarantees and QoS strategies that Grid designer cannot easily control. The lack of resource control in traditional Best Effort IP networks is a real issue in this context. The end to end quality of service (QoS) the heterogeneous mix of individual flows receive affects the overall grid infrastructure throughput, utilization level and impacts each individual application performance.

Two types of solutions to control and differentiate the communications performance have been explored in the past: the first one, *end-based QoS control*, consists in masking the variability of network performance by using compensation mechanisms in applications to meet specific QoS objectives; the second approach, *network-based QoS control*, consists in adding control mechanisms inside the network in order to avoid performance variation and offer some guarantees for the transfer.

In the first approach the application monitors the experimented QoS, detects variation and reacts appropriately. For example, delay sensitive applications using the RTP/RTCP

UDP transport protocol [2], [3] integrate forward error correction (FEC) and jitter compensation mechanisms. This approach increases the network awareness of applications making their programming and debugging more complex. An other solution is to integrate generic adaptive mechanisms in the end to end transport protocol to offer a transport service with guaranteed performance. However strong delay bounds can not be guaranteed in pure IP Best Effort environments.

The second approach, named network-based QoS approach consists in providing QoS support at the network level. The IP community, has proposed several architectures:

- architectures that provide services with strict guarantees (absolutes) by resource reservation like the standardized IntServ architecture [4] and RSVP [5]. The IntServ architecture requires that control and forwarding state for all flows are maintained in routers. That presents scalability problem.
- architectures that provide services with strict guarantees without resource reservation and state management in core routers like CoreStateless model [6]. The corestateless approach requires the help of routers to compute virtual clocks with mechanisms not available in actual equipments. This approach is not deployed in existing infrastructures.
- architectures that provide services with statistical guarantees (differentiated services) to aggregates of flows through packet prioritisation like DiffServ [7]. The DiffServ architecture is the most scalable and manageable architecture. This architecture is partially deployed in most of the current IP networks comprising the grid network clouds (Geant, research networks...). It generally proposes three types of services: best effort, premium and assured services. Three types of forwarding behaviour applied to different aggregates at IP routers are specified. Expedited Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE). In the DiffServ architecture, EF is to support Premium service for hard real time applications that require bounded end-to-end delay and jitter, and AF is to support assured service for soft real-time application.

DiffServ present several issues. It is a *pure in network* solution that care on aggregates of packets (level 3 building block)while end to end users pay attention to the performances

of individual flows (level 4 abstract unit). For example, it appears that end to end flow using the assured service are not assured to receive bounded guarantees [8]. Moreover, users applications have no way to specify directly and in advance the service they expect at flow level. An other issue is that DiffServ is based on domain concept and presents some limitation in multi domains environment. The Differentiated service semantic (ie. bounded end-to-end delay and jitter) is not provided in most multi-domains contexts.

This paper explores an alternate hybrid approach that proposes a solution for end to end service differentiation based on an adaptive packet marking strategy. The goal is to enable Grid users and applications to transparently access the available QoS capabilities offered by the underlying network interconnection to best fit their requirements. However, the marking strategy does not rely on the pre-defined network service semantic. The flow-level semantic is assured by a distributed grid network service. As the QoS resources are scarce and/or costly, this service also finely manages the QoS resource allocation at the edge the wide area network (WAN) to best fit all the various demands coming from heterogeneous grid flows.

This paper is organized as follows: section 2 presents the QoSINUS service design and details its core adaptive algorithms. The results of experiments conducted in the context of the French grid testbed eToile are detailed in section 3. Related works are presented in section 4.

II. THE QOSINUS APPROACH

A. Goals and usage scenario

The approach we propose aims at enabling Grid applications, that may strongly benefit from QoS guarantees, to simply access the various QoS forwarding behaviours, integrated in routers and configured in IP domains. This model uses classical packet differentiation at IP level merged with performance adaptation at intermediate level (edge to edge adaptation). It avoids classical IP QoS approach drawbacks as it does not impose strict end to end performance objectives to network services and does not care on their semantics in terms of absolute guarantees. It also transparently improves the end to end service, without requiring any modification of the transport protocol. The goal is to provide an end to end *Best effort QoS service*. The aims of the QoSINUS service are to:

- 1) provide heterogeneous Grid flows with a mean to specify their QoS objectives;
- 2) dynamically map these objectives with the IP QoS services provided in the network, according to the state of the link, the QoS mechanisms configured and the experienced performances.

A Grid oriented QoS API and a programmable QoS service have been designed to introduce flexibility and dynamic in the management, control and realization of end to end, flow level QoS in Grid context. This approach combines application aware and infrastructure aware components activated within

the network, at the interface of the Grid computing domain and the Grid long distance networks, as shown in figure 1. Such an approach increases slightly the complexity at the frontier points, but maintains the core network and the grid applications simplicity.

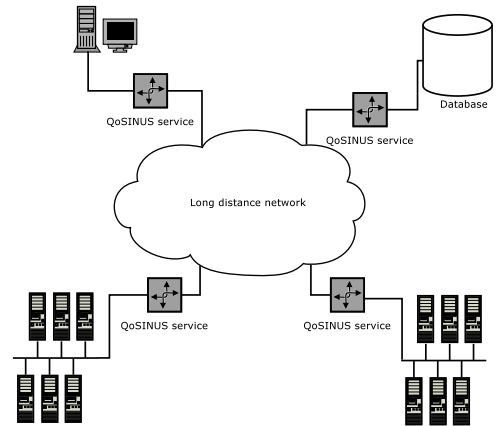


Fig. 1. System topology.

From the user point of view the QoSINUS service is invoked in two phases, as presented in figure 2. During a first programming phase, a QoS request is transmitted to the QoSINUS service. Then the sender transmits data packets: the QoSINUS decision component chooses dynamically the appropriated class of service to cross the domain, trying to simultaneously optimise the QoS resource usage and the flow performance.

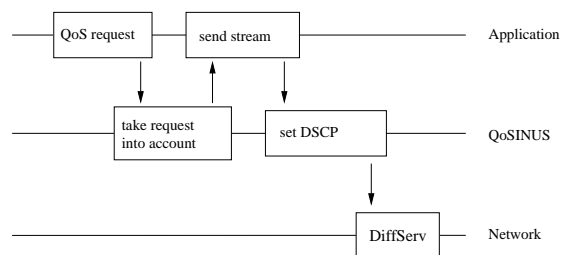


Fig. 2. QoSINUS Usage scenario.

B. Design model

1) *API*: The first issue to solve is to allow grid applications to specify and control their QoS. This issue is addressed by a dedicated API with 3 functions. *QoS_Set* lets an application specify its QoS needs in terms of delay, rate and loss, or transmission schedule and rate. *QoS_Invoke* and *QoS_Release* are called by the client application to let QoSINUS know that the data stream for which QoS was requested is about to start or has ended. This API permits to define end to end service level specifications (SLS) in XML. The only modification of grid application or grid middleware components required is the integration of provided QoS calls in the initialisation phase of a communication session.

2) *Service architecture*: The dynamic mapping of the flows QoS specification to the existing IP QoS facilities is addressed by a flexible service architecture that combines flow aware and infrastructure aware components.

We have identified four types of component for flexible QoS programming and control: programming component, performance measurement component, adaptive control component, and enforcement component.

The QoS programming component is invoked for initiating, propagating and storing the flow QoS goals in the programmable nodes of the path.

The QoS performance measurement is responsible for the characterisation of the specific paths. In a DiffServ context, this component can directly measure the performances of each DS class on each grid path with active out of band probes or filter and gather flow performances in band. However, as several Grid infrastructures provide network measurement architecture, this QoSINUS component may also simply invoke a such general Grid network service.

The adaptive control components is responsible for class mapping and for adapting the packet marking influencing the forwarding performances regarding the QoS goals and the current state of network classes. It monitors and updates the allocated bandwidth of each class and acts as a decision component that decides which class has to be attributed to the packets when the performance measurement component indicates some change. The adaptive control component is flexible, extensible, replaceable and it is simultaneously dependant on the infrastructure QoS properties and on the flow properties. It represents the kernel of the QoSINUS service. It decides the adaptation according to the performance experienced by the packets of each particular flow. With DiffServ, this adaptation means dynamic rate shaping or packet remarking.

The QoS enforcement service, intercepts and adapts the data flows when it is necessary. It realizes the packet marking and conditioning functions. Various tables and soft states are associated to this service: Class allocation table that stores the status of the allocated class bandwidth, *QoS goal* structure that maintains the flows objectives and *QoS flow status* that monitors the flow. The number of states managed in each programmable node is limited to the number of active flows going out the local grid site and depends on the size of the cluster. In the worst case, it will be in the order of a few hundreds.

Other components, that are not designed in this architecture, may be added for status report exchange between programmable QoSINUS nodes for example. A AAA service may also be interfaced with the QoSINUS service to insure policies and security control.

The QoSINUS service has been implemented with the Active network technology that allow to integrate new functionalities in IP equipments. The service class diagram is presented in figure 3. In this figure, QaSEngine and QaSXMLIf correspond to the programming component, QaSMonitoringIf corresponds to the performance measurement component interface, QaMapper is the adaptive control component that

can be a static mapper, an adaptive mapper or an Ack-React mapper, QaSKernelIf represents the interface to the enforcement component.

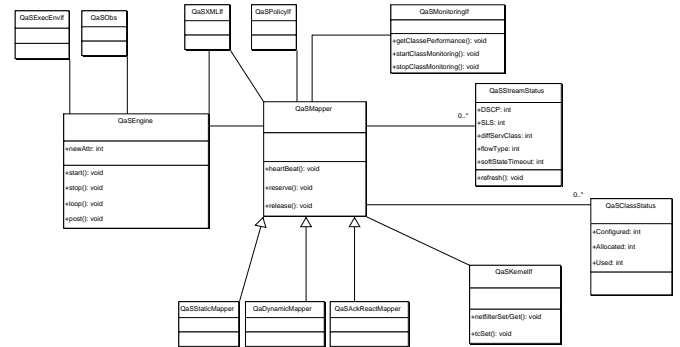


Fig. 3. The QoSINUS class diagram.

C. Adaptive algorithms

Different mappings and adaptive algorithms have been designed. The simplest one, QaSStaticMapper, is a static mapping. It performs the allocation decision on static thresholds. For example, if a delay constrains is expressed and EF bandwidth is still available and sufficient, an EF bandwidth corresponding to the required ones will be allocated. All packets belonging to this flows will be marked EF. Adaptive algorithms attempt to fulfil the requirements of the flows while optimising the classes utilization. The classes are ordered by level of performance they offer. Adaptive mappers are thus responsible for monitoring the performances of the flows and dynamically allocating class resources. An example of such an algorithm is described in the next section.

1) *Ack-based earliest deadline first TCP stream ordering*: The Ack-based mapping algorithm aims at ensuring that TCP data transfers are completed within their specified schedule. It tracks the achieved performance of each TCP stream, and tries to compensate if it gets lower than the requested performance. The streams are ordered according to their deadlines: the stream with the earliest deadline is taken care first. This algorithm assumes that the application is able to specify the schedule it plans to follow for it's data transfer. The schedule is specified in terms of:

- start date: specifying when the emission is going to start. It's expressed in seconds, relative to the reception of the QoS request.
- stop date: specifying when the emission is going to finish. It's expressed in seconds, relative to the reception of the QoS request.
- rate: specifying the average rate that the application expects between the start and stop dates. It's a number of megabits per second.

The following actions are taken by the programmable QoS services, that correspond to the performance measurement, adaptive control and enforcement components mentioned in the previous section.

a) *Ack-based transmission tracking*: The system evaluates the amount of data transmitted at a given moment in time using the TCP acknowledgment messages flowing from the flow's receiver back to the sender.

The acknowledgment number included in the TCP ACK packets is not exactly equal to the actual amount of data successfully received. But it is a good estimate of the amount of transmitted data.

b) *Class mapping*: If, for example, two DiffServ classes BE, EF are available, the principles of the DiffServ allocation algorithm is as follows:

Given Cl the DiffServ class of the current packet flow, Q_{trans} the amount of transmitted data and Q_{req} the requested amount of data to transmit at time t , P the control time period,

```
Cl = BE
for each received ack
    estimate Qtrans

for each P
    if Qtrans < Qreq
        Cl = EF class
    else
        Cl = BE class
```

To avoid oscillations and as accelerated TCP packets lead to packet reordering, the control time period is set to several seconds, corresponding to more than 10 RTT. The ack-based algorithm proceed a coarse grain throughput adjustment useful for long lived sessions.

c) *Flow scheduling*: As different TCP flows competes for the same QoS resource at the edge of the WAN, QoSINUS has to manage and schedule them in order to satisfy both individual and global constrains. We propose an earliest deadline first ordering scheme, to order the flows. Comparing the initial requested schedule with the actual amount of data transmitted for each stream, the system is able to determine which streams are behind schedule and to classify them in terms of urgency. The stream behind schedule with the earliest deadline is given the maximum priority, i.e. is marked in the best performing DiffServ class.

The intuition here is that a stream with a later deadline will have the opportunity to catch up later on. This allows for example, to lower a massive bulk transfer throughput a little bit, when smaller urgent communications have to take place. This approach should thus lead to a more efficient averaged use of the network. But QoSINUS never gives any *strict* guarantees to flows. For example, it may happen that timing request of flows are too strong or the demand is too heavy, making the QoS objectives impossible to reach. In this sense the proposed service can be view as a Best Effort QoS service. However, the active approach, permits to modify easily the algorithms. Several parameters can be dynamically tuned to optimise the service if required. We are studying this aspect.

III. EXPERIMENTS AND RESULTS

To validate the QoSINUS approach, we have develop a prototype in the context of the e-Toile project.

A. The e-Toile VTHD testbed

The e-Toile project [9] is an experimental wide area grid testbed ¹.

The dwdm VTHD (vraiment très haut débit) network infrastructure interconnects the e-Toile grid nodes with access link of 1 to 2 Gbit/s (see Figure 4). The e-Toile middleware relies on existing middleware (Globus [10], Grid Engine...) and integrates new building blocks. The e-Toile middleware aims at fully exploiting the power of the advanced networking infrastructure. One of the original aim of e-Toile is to focus on the High Performance Grid Networking aspects and to evaluate the benefit that grid middleware and applications can get from enhanced networking technologies.

The performance problem of the grid network are studied in e-Toile from the different but complementary points of views: high performance file transfers, active network technology, grid scale MPI... A Grid based on a very high speed controlled network such as e-Toile permits to study the limits of the existing communication services and protocols and to validate more efficient approaches that offer the gigabit performance to the grid user level and take into consideration the specific needs of heterogeneous grid flows.

B. VTHD DiffServ implementation

The Gigabit VTHD [11] Backbone provides four DiffServ classes (EF, TCP AF, UDP AF and BE). The principle of the DiffServ configuration in VTHD is that a bandwidth share is statically provisioned in the edge routers and allocated to the four DS classes. Each access point has to control and to shape the traffic injected in each class. Table I gives the absolute rate allocated to each DiffServ class.

DS Class	Share
Expedited Forwarding	10%
Assured Forwarding for TCP	30%
Assured Forwarding for UDP	30%
Best Effort	30%

TABLE I
THE STATICAL BANDWIDTH SHARE OF VTHD DIFFSERV CLASSES.

The next section exposes experiments we realized to validate QoSINUS control and dynamic allocation of the DiffServ classes algorithms. The goal is to show how QoSINUS makes an efficient use of the DiffServ class resources to keep the network transmission time as close as a given T as possible.

¹e-toile is a RNTL project (réseau national de recherche en logiciel) funded by French Ministry of Research

C. Ack-React experimentation

This experiment demonstrates the use of the TCP Ack packets when evaluating the throughput of a transmission and choosing the DiffServ class. It illustrates how the service can keep the network transmission time as stable as possible.

Three sites are involved, in Lyon (I), Paris (II) and Grenoble (III). Those sites are linked by the VTHD network, as shown in figure 4

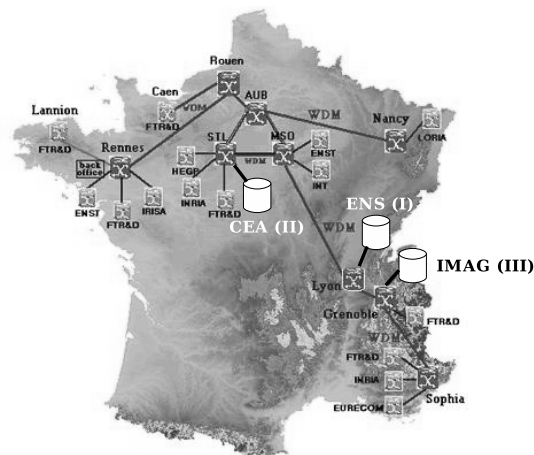


Fig. 4. Network topology

A TCP stream is flowing from Lyon to Grenoble. Background traffic is generated from Lyon to Paris, that produces a congestion on Lyon's router, and affects the TCP stream's performance. The QoSINUS programmable service is deployed at the VTHD border in Lyon, and controls the TCP streams emitted from Lyon to the network.

This test's procedure is as follows. The application, that needs to transmit 486 MBytes of data, sends its QoS request. The stream is scheduled between $t + 10$ seconds and $t + 70$ seconds. The requested rate is 64 Mbit/s. About 11 seconds later, the TCP stream is started. At $t + 20$ seconds, the background traffic is triggered, at 1 Gbit/s. At $t + 57$ seconds, the background traffic is stopped. The application stops sending data whenever its done with the transmission of its 486 MBytes of data.

1) *Impact of the background traffic:* The first step consists in observing the impact the background traffic has on the TCP stream, if nothing is done to protect it. Figure 5 (left) shows how the TCP stream behaves in this case. When no concurrent traffic is emitted, the TCP stream achieves 120 Mbps, and gets ahead of schedule. But it gets much slower (15 Mbps), once the concurrent traffic is started. The stream completes the transmission behind schedule.

2) *Reacting to the Ack packets:* The same procedure is repeated with the DiffServ class mapping algorithm is turned on. Figure 5 shows what happens in this case.

The TCP stream is affected the same way as before by the background traffic. But the system detects that the achieved transmission performance gets lower than the requested one (at $t + 28$ seconds). The system reacts accordingly, and

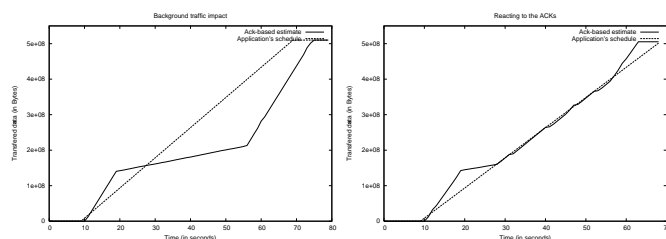


Fig. 5. ACK based adaptation turned off (left) and on (right)

alternatively marks the stream's packets in EF, to boost the transmission rate, and back to Best Effort, when it catches up. In this case, the achieved performance curve sticks to the requested performance curve and the transmission is able to end within schedule.

IV. RELATED WORKS

This work is related to several similar efforts. Within the GGF community, different groups focus on standardization of interfaces to reserve or specify expectation from the network. The GRAAP is proposing an interface where the concept of SLA is expressed. The QoSINUS API is similar, but does not aim at establishing a firm contract between the application and the network resource. Our API serves as a way to express needs and goals, but no SLA result. In QoSINUS, only best effort QoS is aimed. Such goal allows more flexible and extensible QoS solutions.

A work for defining the concept of Grid Network Service is also in progress within the GHPN group of GGF [12]. QoSINUS is proposed as an instance of such distributed and independent Grid netservice. Any Grid application can involve it to improve its performance. The Grid network service GARA [13] shares with QoSINUS the same goals in terms of end to end QoS provisioning. However, existing DiffServ implementation of GARA [14] does not rely on an adaptive packet marking approach. GARA reserves and guarantees resource allocation with classical DiffServ semantic. With its *Best Effort QoS* approach, QoSINUS may be more scalable as it does not suppose a multi domain deployment of DiffServ services. In many cases, Grid application may accept performance decrease at the cost of time completion extension, leaving flows that are more time sensitive using the network resource. In QoSINUS, the network resource is considered as a grid resource that can be scheduled but also *pre-empted*.

Within the Internet community, a large amount of effort for adaptive packet marking [15] has been done in the past. These works mainly focused on enhancing the Assured Forwarding service that is not able to offer expected and negotiated guarantees in a pure DiffServ approach. Adaptive algorithms executed in proxies located in network have been proposed [8], [16]. These proxies, opposed to our QoSINUS agents do no serve the heterogeneous needs of grid flows in a *global grid resource management* view, but aim at guaranteeing rate obtained by TCP flows marked in AF. For example, the Expedited Forwarding PHB is not used as in QoSINUS

where the packet acceleration capacity is exploited. QoSINUS approach can be also compared with [17]–[19], but is more dedicated to grid applications.

V. CONCLUSION AND PERSPECTIVE

This paper examined the problem of flexible and scalable mapping of grid flows QoS requirements to the available IP QoS network services. It presented a flexible service named QoSINUS, designed for dynamically manage the QoS classes offered at the edge of IP domains to serve the needs of grid applications. This service offers a flexible, transparent and efficient network QoS control to end grid applications. An original adaptive marking algorithm and a scheduling algorithm for the QoS resource have been proposed and are very promising. The results obtained on an experimental grid test bed show the benefit a Grid application can expect from IP QoS services available in current academics network infrastructures. In the future we will compare the performances obtained with the different mapping algorithms under the same utilization and load conditions. An other perspective is to verify the earliest deadline first approach is correct, and that it allows for a more efficient use of the network. Experimentations involving several parallel TCP streams will be conducted in this sense. The programmable network approach and the modular architecture of QoSINUS permit us to easily deploy and improve such algorithms and to apply the principle with other QoS approaches. We plan to explore this service in the context of a multi-domain environment where packet code points may be modified at peering points. We expect that our transparent edge to edge adaptive approach will allow a low cost and transparent end to end performance improvement and control in such complex environments.

ACKNOWLEDGMENT

This work was funded by the RNTL eToile project of the French ministry of research, by the European Commission program IST-2000-25182 through the EU DataGrid Project, the INRIA project RESO, and the French ministry for research ACI-GRID project. The authors would like to thank all the VTHD and e-Toile project teams, as well as Johan Montagnat for his contribution to this work.

REFERENCES

[1] I. Foster and C. Kesselman, "The grid : Blueprint for a new computing infrastructure," *Morgan Kaufmann Publishers Inc.*, 1998.
 [2] J.-C. Bolot and A. Vega-Garcia, "Control mechanisms for packet audio in the internet," in *INFOCOM (1)*, mar 1996, pp. 232–239.

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *Internet-Draft ietf-avt-rtp-new-01.txt (work in progress)*, 1998. [Online]. Available: rfc1889
 [4] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," Internet Engineering Task Force, Internet Request For Comments RFC 1633, June 1994. [Online]. Available: rfc1633.txt
 [5] S. Zhang, L. Deering and D. Estrin, "RSVP: A new resource ReSerVation protocol," *IEEE network*, vol. 7, no. 5, pp. 8–19, September 1993.
 [6] I. Stoica, S. Shenker, and H. Zhang, "Corestateless fair queuing: Achieving approximately fair bandwidth allocations in high speed networks," in *ACM Computer Communication Review*, vol. 28, no. 4. ACM press, september 1998, pp. 118–130.
 [7] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," Internet Engineering Task Force, Internet Request For Comments RFC 2475, Dec. 1998. [Online]. Available: rfc2475.txt
 [8] N. Seddigh, B. Nandy, and P. Piedad, "Bandwidth assurance issues for tcp flows in a differentiated services network," in *Proceedings of Globecom'99*, Rio de Janeiro, Dec. 1999. [Online]. Available: citeseer.nj.nec.com/seddigh99bandwidth.html
 [9] P. Vicat-Blanc/Primet, G. Romier, and M. Soberman, "Le projet e-toile: dveloppement et mise en oeuvre d'une grille haute performance," in *Proceedings of "Journes Nationales du RNTL 2002"*, 2002.
 [10] I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, no. 2, pp. 115–128, July 1997.
 [11] "Vthd," <http://www.vthd.org/>.
 [12] T. Ferrari and al., "[ghpn-wg] draft-ggf-ghpn-netservices," Global Grid Forum, work in progress, Tech. Rep., 2004.
 [13] V. Sander and A. Roy, "Gara: a uniform quality of service architecture," *GRID Resource Management.*, 2003.
 [14] T. Ferrari and al., "Advance network reservations in grids," *Future generation Computer System: special issue on High performance networks and services for the Grid*, to appear 2004.
 [15] W. Feng, K. Kandlur, D. Saha, and K. Shin, "Adaptative packet marking for providing differentiated services on the internet," in *Proceedings of the International Conference on Network Protocols (ICNP)*, Oct. 1998.
 [16] W. Lin, R. Zheng, and J. C. Hou, "How to make assured service more assured," in *ICNP*, 1999, pp. 182+. [Online]. Available: citeseer.nj.nec.com/lin99how.html
 [17] M. Mellia, I. Stoica, and H. Zhang, "TCP-aware packet marking in networks with diffserv support," *Computer Networks*, vol. 42, no. 1, pp. 81–100, May 2003.
 [18] M. A. El-Gendy and K. G. Shin, "Equation-based packet marking for assured forwarding services," 2002.
 [19] W. Lin, R. Zheng, and J. C. Hou, "How to make assured service more assured," Oct. 1999, p. 182.
 [20] B. Davie, A. Charny, J. C. R. Bennett, K. Benson, J.-Y. Le Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An expedited forwarding phb (per-hop behavior)," Internet Engineering Task Force, Internet Request For Comments RFC 3246, Mar. 2002. [Online]. Available: rfc3246.txt
 [21] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy, "A distributed resource management architecture that supports advance reservations and co-allocation," in *Proceedings of the International Workshop on Quality of Service*, 1999. [Online]. Available: citeseer.nj.nec.com/foster99distributed.html