A SERVICE ORIENTED SLA MANAGEMENT FRAMEWORK FOR GRID ENVIRONMENTS

C. Marinos, V. Pouli, M. Grammatikou, S. Papavassiliou, V. Maglaris
Network Management & Optimal Design Laboratory (NETMODE)
School of Electrical & Computer Engineering
National Technical University of Athens (NTUA)
Tel: +30-210.772.1451, Fax: +30-210.772.1452
{cmarinos, vpouli, mary, papavassiliou, maglaris}@netmode.ntua.gr

Abstract

Traditionally, network Service Providers specify Service Level Agreements (SLAs) in order to guarantee service availability and performance to their customers. However, these SLAs are rather static and span a single provider domain. Thus they are not applicable to a multi-domain environment and do not enable users with the flexibility to set-up and monitor e2e services. In this paper we present a framework for automatic creation and management of SLAs in a multi-domain environment. The framework is based on Service Oriented Computing (SOC) and contains a collection of web service calls and modules that allow for the automatic creation, configuration and delivery of an end-to-end SLA, created from the merging of the per-domain SLAs. We also present a monitoring procedure to monitor the QoS guarantees stipulated in the SLA. The SLA establishment and monitoring procedures are tested through a Grid application scenario targeted to perform remote control and monitoring of instrument elements distributed across the Grid.

Keywords: Service Level Agreement, QoS, SLA framework, multi-domain, monitoring
1. INTRODUCTION

A Service Level Agreement (SLA) is a contract between a network provider and a customer defining all aspects of the offered services. It may specify the levels of availability, serviceability, performance and operation conditions. It may also define the procedures and the reports that must be provided to track and ensure compliance with the SLA or describe other attributes of the service, such as billing and penalties in case of SLA violations. Service Level Agreements with the network providers enable users to use services for their applications in a more effective and reliable way. The benefit of using an SLA for an application is that a contract can be provided to the application users for the level of Quality of Service (QoS) that they have requested and the procedures that will be followed in order to enforce this level of QoS. An SLA can also provide a means of describing the network domains’ level of QoS in a unified way, irrespective of the type of technologies that are used to provide QoS across the network.

In Grid environments [1], [17] where applications or users require resources from heterogeneous domains offering various performance guarantees, the issue of acquiring a specified level of QoS is of great importance. The lack of control in traditional IP networks created the need for efficient QoS provisioning mechanisms. Among the dominant ones is the DiffServ framework [2], where marking and policing of flows is implemented at the edge routers according to per-domain SLAs [19]. This way a variety of advanced network end–to–end services is offered across multiple, separately administrated domains [12]. Based on the DiffServ architecture [2], the Premium IP (PIP) service was implemented in European Research & Education Networks (NRENs). Using DiffServ priorities, traffic can be classified as high priority PIP, Best Effort (BE) for standard IP applications and Less than Best Effort (LBE) for non–time critical bulk data transfer. The Premium IP service provides a virtual leased line service as PIP data packets are not expected to experience congestion in the network regardless of the load of the other traffic classes. PIP is suitable for real–time applications, such as Voice over IP (VoIP), IPTV, video conferencing as well as time critical applications such as remote control of instruments and devices [18].

In this paper we present a framework based on SOC [3] that supports the provisioning of advanced end-to-end (e2e) IP network services via the dynamic management of SLAs, in a real multi-domain environment that crosses the NRENs. The framework uses web service (WS) [5] calls to allow for the automatic creation, configuration and delivery of an e2e SLA created by merging the SLAs provided by each domain. We also present a monitoring procedure to monitor the QoS guarantees stipulated in the SLA. Our framework has been deployed in the NREN/GÉANT Advanced Multi-domain Provisioning System - AMPS [9] to offer automatic e2e SLAs for network path reservations.

The rest of the paper is organized as follows. Section 2 presents some related work, while section 3 describes the SLA Framework along with its core components and its architecture for SLA establishment and monitoring procedures. In section 4 the evaluation and testing of the SLA framework is performed through a Grid application scenario targeted to perform remote control and monitoring of instrument elements distributed across the Grid. Finally in section 5 we provide some concluding remarks.

2. RELATED WORK

To date several frameworks have been proposed to support the creation and management of services based on SLAs, however these frameworks do not apply in multi-domain environments. For instance, in [6] a Contract Framework is presented and in [7] a generic framework for negotiating SLAs is also introduced. Specific components automate large parts of the negotiation process with the ability for the user to maintain control. In a related work, Garg [8] presented an SLA-based framework for QoS provisioning and dynamic capacity allocation. The proposed SLA framework allows users to buy a long term capacity at a pre-specified price with a three tier pricing model with penalties. On the vendors’ side, DiffServ [2] is primarily indicated as the preferred technology to achieve SLA management and delay, jitter, packet–loss, throughput and availability are considered as the most vital SLA technical parameters for an IP service.
With respect to the above proposals, our framework differentiates from the others in the sense that it uses service oriented techniques to provide an automated mechanism for the generation, configuration and delivery of an e2e SLA applied in a multi-domain environment. This e2e SLA is created by merging the per-domain SLAs according to specific merging rules, described in the next section. Our SLA framework offers QoS in the various applications that require resources spanning multiple domains. Along with the SLA establishment procedure we provide a monitoring procedure to monitor the QoS guarantees stipulated in the SLA. The SLA framework has already been deployed on top of the AMPS, a multi-domain reservation tool and tested in real QoS–enabled networks, such as the NRENs and GÉANT to offer e2e SLAs for the network path reservations.

3. ARCHITECTURE

In this section we provide some requirements and the Service Level Specification (SLS) in 3.1; in 3.2 we present the framework, while in sections 3.3 and 3.4 the SLA establishment and monitoring procedures are introduced respectively.

3.1. REQUIREMENTS AND SLS

In principle, the SLA framework provides an end-to-end (e2e) SLA that has to contain all necessary parameters to provide end-to-end control for a service instance specification between two endpoints. The e2e-SLA establishment should be transparent to the end-users, regardless of the underlying network technologies and intermediate domains.

For the generation of an e2e SLA by the SLA framework several issues have to be examined:

- Different ISPs must agree to cooperate with each other, authorize and configure the framework’s communication between other cooperating ISPs and clearly specify what actions are taken and by whom.

- Depending on the service request type (uni-directional, bi-directional), a uni-directional or bi-directional end-to-end SLA should be delivered by the SLA Framework. This distinction is necessary since the downstream (closer to the destination) domain may not wish to participate in the e2e marking and policing setup requested by the upstream (closer to the source) domain.

- In order to honor the guarantees stipulated in the e2e-SLA, continuous monitoring, for the whole service duration, of all the SLA parameters in the contract between the user and the service providers is needed. The e2e monitoring SLA will be based on the investigation and monitoring of all the per-domain SLAs along the path and must not have negative impact on the network performance.

- Due to the fact that we are dealing with a multi-domain environment where QoS provisioning is each domain’s responsibility, the e2e SLA will be derived by the merging of each domain’s SLA, as shown in Figure 1.
An SLA is a set of technical and non-technical parameters agreed between the consumer and the provider. An SLA contract can be formed by two different parts: (i) an administrative and legal part that contains information that does not depend on the particular service (e.g. user authentication, availability information, privacy and security aspects), and (ii) the Service Level Specification (SLS) part, which is a set of parameters and their corresponding values that define the technical parameters of the service provisioning for a specific flow instance. While SLAs represent a description of a required service, SLSs are more formal documents containing a list of technical parameters. An SLS can contain information as shown in Table 1.

### Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Instance Scope</td>
<td>This field should contain the technical information of the ingress interface and the egress interface of each domain involved in the path.</td>
</tr>
<tr>
<td>Flow Description</td>
<td>This field should contain the DSCP value, the IP source-destination address pair along with the protocol and the ports that the application is targeted to.</td>
</tr>
<tr>
<td>End-To-End Performance Guarantees</td>
<td>The SLA performance guarantees field of the end-to-end SLA is proposed to adopt the guarantees derived from the individual SLAs involved in the service provision across the path. More specifically:</td>
</tr>
<tr>
<td></td>
<td>- Capacity will be less or equal to the minimum of the capacities in each SLA instance: $C_{e2e} \leq \min{C_i}$</td>
</tr>
<tr>
<td></td>
<td>- MTU value supported by the end-to-end SLA will be less or equal to the minimum MTU value of the chain of individual SLAs: $MTU_{e2e} \leq \min{MTU_i}$</td>
</tr>
<tr>
<td></td>
<td>- One-Way-Delay is an additive metric and therefore the corresponding field in the $e2e$ SLA will be produced as follows: $OWD_{e2e} \geq \sum{OWD_i}$</td>
</tr>
<tr>
<td></td>
<td>- IPDV$_{e2e}$: we treat IPDV as an RMS value and use its square as an additive parameter.</td>
</tr>
<tr>
<td></td>
<td>- Packet loss: will be the sum of all the packet losses.</td>
</tr>
</tbody>
</table>

Figure 1: End-to-end SLA merging from per-domain SLAs
losses across the individual domains: $PL_{e2e} \geq \sum_{i} (PL_i)$

<table>
<thead>
<tr>
<th>Monitoring Infrastructure</th>
<th>This field should contain information on the monitoring capabilities of each domain in terms of which parameters are monitored, the points where measurements are possible, the availability of measurements.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliability</td>
<td>Reliability should define allowed mean downtime per unit of time for the service provision and maximum allowed Time-To-Repair (TTR) in case of breakdown for the provision of the e2e service described by the e2e SLA, as the worst figures among the individual SLAs along the end-to-end path</td>
</tr>
</tbody>
</table>

Table 1: Service Level Specification (SLS)

3.2. SLA FRAMEWORK DESCRIPTION

The SLA framework can be deployed in network reservation systems in order to provide end-to-end SLAs for network reservations. For the establishment and monitoring procedures to be feasible, the SLA framework should be installed in each domain across the end-to-end path in order to acquire from each domain its respective SLA and provide finally to the end-user the e2e SLA created by the merging of the per-domain SLAs, according to the merging rules defined in Table 1 above. Well defined APIs will carry out the WS communication between the SLA Frameworks of the involved domains, including all required intra-domain entities and the administrator who will be responsible for the SLA management within each domain. The components for the inter-domain communication can be categorized into the following levels as shown in figure 2:

- **Web Interface**: through this web interface the domain administrator can manage and configure the SLA Framework.
- **SLA Communicator**: this component is responsible for the intra-communication between the different entities of each framework. By calling the appropriate web services, each SLA communicator of the SLA framework can collect required data and interchange messages.
- **SLA Manager**: through this entity, the SLA Frameworks in the different domains inter-communicate with each other and transfer the appropriate data via Web Services.
- **SLA Core Engine**: this component collects the information from the Web Interface and the SLA Manager and generates either the per-domain SLAs or the e2e SLA if it is the first framework in the path.
- **SLA Persistence Layer**: this layer is responsible for the communication of the core components with the SLA database.
- **SLA DB**: each SLA Framework maintains its own database where local SLAs of the domain, internal data of the framework and end-to-end SLAs are stored. The data can be viewed and edited via the administrative functions of the Web Interface.

Figure 2 below shows the SLA Framework Architecture and its components.
For enabling the EGEE user to have access to the SLA framework and request an SLA, a User Interface (UI), as shown in figure 3, is provided. Through this interface the EGEE user, after having been authenticated, can fill in his request with the desired level of QoS. Afterwards the UI will communicate via Web Services with the SLA Communicator (figure 2) to forward the user request to the main components of the SLA framework in order to proceed with the generation of the corresponding SLA.

**Figure 2: SLA Framework Architecture**

**Figure 3: User Web Interface**

3.3. **ESTABLISHMENT**

In the following section we will describe the SLA generation life-cycle. It involves three phases: Creation, Publication and Delivery.
- **Creation Phase:** The end user triggers the end-to-end SLA provisioning, when he requests a service. More specifically, the user requests, through the UI, an e2e SLA for a network reservation spanning multiple domains. The service requests are received through the SLA Communicator, which forwards the request to the SLA Manager to communicate it: a) to the next domains and also b) to the local domain’s SLA Core Engine component in order to provide the per-domain SLAs. Each domain evaluates the network reservation request, produces the local SLA by filling in the corresponding Administrative Part and Service Level Specification, and finally, stores the respective SLA to its local database (SLA db in Figure.2).

- **Publication Phase:** After the creation phase, the newly defined per-domain SLAs need to be published in order to be available for the merging of the e2e SLA. Publication consists of the transfer of the per-domain SLAs to the first SLA Framework in the reservation path, which is made through the SLA Manager component.

- **Delivery Phase:** Once the per-domain SLAs have been defined and published, the first SLA Framework in the reservation path merges all the per-domain SLAs, according to the merging rules defined in Table 1, in order to establish the end-to-end SLA. Afterwards, it stores the e2e SLA to the local SLA database and communicates via WS to inform all other SLA Frameworks in the chain through the SLA Manager. If a domain in the path does not deliver its respective SLA, the first SLA Framework informs the client that an SLA cannot be delivered for the requested IP service, so that the client can make a new request for a different path or with different level of QoS.

### 3.4. Monitoring Procedure

In an inter-domain environment, mechanisms for monitoring the performance parameters for a specific service instance must be available. Such information is critical for both the end user and the service provider, since the former is the entity that requests the service and the later is the entity that delivers the service. Evaluating the monitoring results, a user can ask for SLA re-negotiation if the negotiated service level is not the appropriate one.

For the SLA monitoring, several monitoring tools will be used. The entity that will be responsible for the SLA monitoring, apart from the end user, should be the one that provides the SLA framework administrator.

The SLA monitoring and troubleshooting procedures that are followed are described below:

- In order for the administrator to monitor an SLA it should access the monitoring framework and provide all the required parameters (e.g. time period over which the network monitoring for the SLA applies, source and destination IP addresses, metrics that need to be measured, such as OWD, achievable bandwidth etc). The measured data can be viewed in various ways (data table, matrix, time plot, histogram) so the administrator can choose the way in which the resultant data is displayed. It should be mentioned that the domains which need to be measured, must be informed in advance about what kind of measurements will be taken, in order for the available configurations and deployments of the monitoring tools to take place.

- In order for the SLA guarantees to be honoured, some alarm triggering mechanisms were developed that provide the ability to set thresholds on the measured metrics and accordingly generate alarms in case of threshold violations.

- In case an SLA metric is violated due to a problem (e.g. bandwidth falls below the specified value in the SLA template) the monitoring framework tries to identify in which domain the problem occurred by taking intermediate measurements if needed and accordingly notify the Technical Contact(s) (provided in the Administrative Part of the end-to-end SLA) of one or more of the domains along the path in order to solve the problem. Those involved in a reservation, such as users/VOs/RCs, can install and/or run their own application oriented tests in order to verify the performance of their service and
if it is degraded. In case of faults, perceived by them, the problem must then be reported and a trouble ticket should be opened. It should be mentioned that the administrator must refer to the specific SLA instance of the reservation path for which the problem exists when reporting a problem to a domain, so that the reservation details, as registered in the corresponding SLA instance, are taken into consideration for troubleshooting.

- In each domain, the fault-handling and trouble ticket procedures specified in the corresponding SLA must be followed. Normally, the domain(s) in which the problem occurs will be able to troubleshoot it.

- After troubleshooting they should report back to the administrator what were the causes of the problem and if normal operation has been restored.

- Finally, actual failure indication response times (as declared by a domain in the corresponding SLA) must be crosschecked by the administrator against the promised values.

The flowchart for the above procedure is shown in the figure below:
In order for the monitoring procedure to be feasible, some monitoring requirements must be available:

- Network performance data should be available for every SLA that corresponds to an end–to–end reservation.
- The SLA should be monitored on an e2e basis, with the point of measurements as close to the end point as possible.
- For every SLA monitored, the following metrics should be collected: OWD (One Way Delay), RTT (Round Trip Time), IPDV (IP Delay Variation), Packet Loss, Capacity.

There should also be possible to make on demand measurements frequently in order to provide an up–to–date view of the network performance but in any case they should not greatly impact the network performance.
4. GRID EXPERIMENT

In this section we describe the Grid experiment that was performed in order to test and evaluate the SLA establishment and monitoring procedures that were described in the previous sections. The application scenario was taken from the GRIDCC [4] project and considers a remote control application to monitor the instruments distributed across the Grid.

The GRIDCC project produced middleware is based on a web services architecture and not only extends the middleware being developed by other Grid projects but also develops completely new middleware where needed. An innovative idea of the GRIDCC project is the IE (Instrument Element), which provides an abstraction of the instruments in order to access and control them via Web Services in a unified and standardized way from the VCR (Virtual Control Room). The VCR is the User Interface (UI) that allows users to build complex workflows and submit them to the execution services or even connect to the resources directly in order to control the instruments in real time [16].

GRIDCC provided a good candidate application willing to adopt the SLAs that promised to offer the stringent network QoS guarantees needed for the real-time remote control of instrumentation over the Grid. The testing gave us a real example of a network application and the ability to study the particular requirements of this application and try and identify what the network can offer it.

4.1. DESCRIPTION

The Grid experiment was based on Web Service invocations in order to remotely control, through the IE, the instruments distributed across the Grid environment and then transfer the data produced by these instruments to Grid Resource Centres for further processing.

In order to introduce a real network and preserve the real time behaviour needed for the remote control and data transfer of produced results from the experiment, we need to have a network which can guarantee that its network characteristics (bandwidth, delay etc) remain within a given range when a client located in one site needs to access a server in another site.

Thus, we performed the following test:

- A client in site A logs in to the VCR, described above, and makes Web Service calls (with the same SOAP message payload) to the server which hosts the IE in site B in order to perform remote control of instruments and then initiate probable data transfers of the data produced.
- Both the client and server are NTP time synchronized.
- The test is performed with and without SLA reserved path in order to evaluate the effect of the SLA reservation on the performance.
- Network performance metrics are described in the SLA and are known before the beginning of the test in order to determine an end-to-end time behaviour.
- For the performance testing frequent end-to-end measurements are taken by appropriate monitoring tools in order to evaluate and monitor the level of QoS over the network.
- Results of the network performance metrics are gathered in collective diagrams, in order to have a better control and supervision of the application performance and are also permanently stored to have the possibility to track back the behaviour.
4.2. DEPLOYMENT

In this section, the monitoring tools for running the experiment in the involved domains along with the procedures followed for the deployment of the necessary software are presented.

4.2.1. Monitoring Tools

In order to monitor the network used for the Grid application testing we deployed some monitoring tools to measure specific network performance metrics such as One-Way-Delay (OWD), Internet Packet Delay Variation (IPDV), Packet Loss and Capacity.

- For the monitoring, we installed the PerfSONAR (Performance focused Service Oriented Network monitoring Architecture) monitoring framework [11] which forms an interoperable, distributed performance measurement middleware framework. The release that we installed contains, amongst others, the following tool that was used to monitor the performance metrics needed for the application: Bandwidth Controller (BWCTL) Measurement Point (MP) for measuring the achievable bandwidth.

- One-Way Ping (OWAMP) application for measuring one-way latencies such as OWD, IPDV and packet loss. One way measurements help to better isolate a problem because the direction of the congestion becomes immediately apparent. Since traffic could be asymmetric in the two directions between hosts, one way measurements allow the user to better isolate the effects of congestion for specific parts of the network thus allowing for more informative measurements and in some cases help providers decrease the areas of congestion whenever possible by better allocating resources in their networks.

- RRD (Round Robin Database) Measurement Archive (MA) for collecting the required metrics and providing graphical displays of historical views of the data whenever required.

4.2.2. Deployment Procedure

For the testing of the Grid application scenario, the required software was installed on two nodes (server-client). More specifically, a VCR was installed on the client side and provided a UI for the user to access the instruments and an IE was installed on the server side to provide an interface for the remote control of various instruments. The server node was located in Legnaro, Italy within the INFN [13] resource centre, while the client node used for calling the IE and storing the data produced by the remote control was located in the HELLASGRID [14] cluster in Athens, Greece at IASA [15]. The end-to-end path from IASA to INFIN traversed the Greek NREN GRNET, the Pan-European Interconnection backbone GÉANT and the Italian NREN GARR [10]. As shown in figure 5, we managed to establish three per-domain SLAs across this path, namely from IASA to GRNET, GRNET-GÉANT-GARR, GARR-INFN.
All the monitoring tools described in the previous section were deployed on both nodes in order to run the Grid application between the two nodes and test its performance by measuring specific network metrics. In order to realize the automatic SLA establishment procedure through the SLA framework and evaluate the performance of the reservation mechanism, we conducted two experiments in the same network route; the first one was under Best-Effort (no reservation to the network) while the second one was under a reserved path through the SLA framework which provided guaranteed service, Premium-IP (PIP) enabled. For the two tests we followed the steps described below in each case.

### 4.2.2.1. Best Effort service

The first test was realized under a best-effort service in a specific network path. Best effort is the default service level in GÉANT and NRENs. In a best effort network all users obtain best effort service, meaning that there are no specific network performance guarantees provided by the NRENs. Thus the users and applications obtain unspecified bandwidth and delivery time, depending on current traffic load. Thus there was no need to make a service request to reserve a network path neither to offer an SLA because the Best Effort service does not provide any guarantees that data is delivered or that a user is given a guaranteed quality of service level or a certain priority. In this situation we conducted our experiments in the GÉANT/NRENs network without making a path reservation and took the measurements shown in section 4.3.1 (Best Effort case).

In order to monitor the test for the whole SLA duration we installed in both the previous cases the monitoring tools described in Section 4.2.1 and we took measurements of the performance metrics every 10 minutes across the network path as shown in the following section. The experimental results are represented graphically below through the RRD diagrams.

### 4.2.2.2. SLA guaranteed service

The second test was made under a reserved network path with guaranteed QoS characteristics and an e2e SLA was offered to the end user automatically through the SLA framework. More specifically, the SLA request was requested through the site from where the user filled in his request, which was then forwarded to the AMPS in order to make the path reservation. The AMPS is an advanced reservation system that accepts PIP service requests, authenticates them according to the policy module that it has and reserves network resources in each of the domains across the full path. When requesting an SLA the required network metrics, Start and End Times of the reservation, and one IP Source and Destination Address Pair should be specified. In our application scenario we are interested to have guaranteed bandwidth for the transfer of the data produced by the instruments towards the Grid resource for further processing and also the minimum possible delay across the real network is needed to approach the near real time nature of the application. In our case the requested bandwidth was almost 20Mbps, the SLA period was one week and the network path had as source IP: 195.251.54.7 and destination IP: 62.40.126.10. The default MTU in all domains was 1500bytes.

### 4.3. MEASUREMENTS

In the following section we will describe the performance testing procedure followed by the measurements that we took.

For the monitoring procedure we performed frequent (i.e. every 10 minutes) end-to-end measurements with the Bandwidth Controller (BWCTL) and One-Way Ping (OWAMP) applications, measuring the network performance metrics between the two nodes bi-directionally. The monitoring mechanisms were installed in AMPS servers at the edges of the end domains, i.e. IASA-GRNET and INFN-GARR.

More specifically, the OWAMP is used to provide measurements for the OWD, IPDV and packet-loss while the BWCTL is used to measure the achievable bandwidth by sending small packets in order not to affect the network performance of the application. Retrieved data are collected via the RRDtool, stored in an RRD database, and illustrated graphically in diagrams, for the two cases, as shown below.
In the sections below, the first diagrams refer to the Best-Effort experiment and following are the ones concerning the SLA reserved path experiment.

### 4.3.1. Best-Effort diagrams

Below are the daily diagrams for the Best-Effort case. The x-axis declares the days during a week while the y-axis declares the Bandwidth, Packet-loss, OWD and Jitter in each case.

Best-Effort, as mentioned before, is the default service in the GÉANT and NRENs networks meaning that all users obtain Best-Effort service. This service does not provide guarantees for bandwidth and delivery time so the measurements that we will see in the figures below will be measurements showing the current traffic load.

In Figure 6 and Figure 7 we can see the Bandwidth (Mbps) from IASA to INFN and backwards. Its value as we can observe has a lot of variations ranging from 10Mbps to 70Mbps.

![Bandwidth Diagrams](image)

**Figures 6,7:** Bandwidth (Mbps) representation from IASA to INFN and backwards.

In Figure 8 below we see that the Packet Loss during a week is almost 0%, but in many cases presents some peaks in both directions. The difference in these values is due to the asymmetric nature of the two directions between the two nodes in IASA and INFN.

![Packet Loss Diagram](image)

**Figure 7:** Packet loss (%) representation from IASA to INFN and backwards.

- "From packet loss" direction means from IASA to INFN.
- "To packet loss" direction means from INFN to IASA.

In Figure 9 and Figure 10 below, we can see the OWD (ms) from IASA to INFN and backwards. Its average value is about 20ms.
Figures 8, 10: One-Way-Delay (ms) representation from IASA to INFN and backwards.

Finally the jitter (ms) representation is shown in Figure 11. In the IASA- INFN direction we have almost no jitter with some peaks in some cases, while in the opposite direction we have an average jitter of 1ms.

Figure 9: Jitter (ms) representation from IASA to INFN and backwards.

“From jitter” direction means from IASA to INFN.

“To jitter” direction means from INFN to IASA.

4.3.2. SLA guaranteed diagrams

Following are the weekly diagrams for the SLA reserved path. The x-axis declares the days during the week while the y-axis declares the Bandwidth, Packet-loss, OWD and Jitter in each case.

In Figure 12 and Figure 13 below, we can see the bandwidth measurements from IASA to INFN and backwards respectively. The bandwidth value is almost stable at 16Mbps in each diagram respectively. Compared to the Best-Effort case, we observe a greater stability in the bandwidth measurements reaching the guaranteed bandwidth that we requested.
Figures 10, 13: Bandwidth (Mbps) representation from IASA to INFN and backwards.

In Figure 14 we can observe the packet loss for the two directions (IASA -> INFN, INFN -> IASA). As in the Best-Effort case, packet loss is equal to zero.

Figure 11: Packet loss (%) from IASA to INFN and backwards.

The OWD from IASA to INFN and backwards are shown in Figure 15 and Figure 16 respectively. In the first case (IASA -> INFN) we can see that the average OWD is about 14ms while in the second case (INFN -> IASA) the OWD is about 17ms. Comparing with the Best-Effort case, the OWD in the SLA guaranteed service is a little bit better than the OWD in the Best-Effort. Some picks in the diagrams may be due to false alarms of the monitoring tools.

Figures 12, 16: OWD (ms) from IASA to INFN and backwards.
Finally the jitter (%) representation is shown in Figure 17 below. The jitter is 0%. Again we can see an improvement of the jitter compared to the jitter results shown in the Best-Effort diagrams.

**Figure 13:** Jitter (ms) from IASA to INFN and backwards.

“From jitter” direction means from IASA to INFN.

“To jitter” direction means from INFN to IASA.

### 4.3.3. Comparison between the Best-Effort and SLA reserved path diagrams

From the experimental results we can see that we have an improvement in performance in the reserved path through the SLA framework where the bandwidth and delay variations present smaller variations compared to the best-effort service. That is due to the fact that the service offered through the SLA framework in a PIP reserved path is the best offered service over an IP network that uses the available high priority scheduling techniques to emulate a dedicated circuit. The aim of a guaranteed service is to provide guarantees on bandwidth, one-way-delay (latency), IP Packet Variation (jitter) and negligible percentage of packet loss for IP traffic across one or more interconnected domains. This is of great importance for applications and the primary beneficiaries of this service are the users of applications that demand strict QoS guarantees over the network, e.g. video conferencing and remote control of the Grid instrumentation, due to their real time nature.

### 5. CONCLUSIONS

In this paper we propose a framework for automated Grid SLA management, i.e. the creation, negotiation, publication and monitoring in a multi–domain distributed high performance computing & networking environment. The automated management and monitoring procedures have been presented as they have been designed for a multi-domain system, including Grid resources and high speed interconnecting communication networks. Key to our framework is the provision of per-domain Service Level Specifications (SLSs) in a common template as proposed in our paper.

Our framework provides an e2e automatic SLA service to a Grid end-user and describes SLS specifications (e.g. networking PIP characteristics, monitoring infrastructures and mechanisms, availability guarantees) for each administrative domain involved in the e2e path upon making the reservation. It automates the otherwise manual procedure of finding and contacting each administrative network domain across paths between Grid resources to produce and deliver per-domain SLAs and consequently the e2e SLA.

Real time and interactive Grid applications can benefit from our SLA framework as it offers an automated way to provide and monitor a contract to the end-user specifying that the network performance guarantees and the troubleshooting procedures stipulated in the SLA will be provided and enforced. As an example, a user by signing an SLA can enforce time critical constraints imposed by his experiment.

The proposed framework has been adopted within the EGEE [20] and GÉANT operations and is deployed in the Advanced Multi–domain Provisioning System (AMPS) [9]. The integration of our
e2e SLA Framework in AMPS gave us the opportunity to evaluate and measure its performance in real QoS-enabled networks such as the NRENs and their interconnecting backbone GÉANT.

6. ACKNOWLEDGMENTS
This paper has been partially supported by the 6th Framework Programme for Research & Technological Development of the European Union – EC DG INFSO projects EGEEII, EGEEIII, GN2 (GÉANT) and GRIDCC.

REFERENCES