Network Monitoring Session Description

Augusto Ciuffoletti
augusto@di.unipi.it
INFN-CNAF
Via B. Pichat - Bologna - ITALY

Antonis Papadogiannakis, Michalis Polychronakis,
{papadog,mikepo}@forth.ics.gr
FORTH
Heraklion - Crete - Greece

May 4, 2007

Abstract

Network Monitoring is a complex distributed activity: we distinguish agents that issue requests and use of the results, other that operate the monitoring activity and produce observations, glued together by other agents that are in charge of routing requests and results.

We illustrate a comprehensive view of a such architecture, taking into account scalability and security requirements, concentrating on the definition of the information exchanged between such agents.

We address scalability by introducing monitoring sessions activated on demand, with a declared preference for passive monitoring tools, and security by enforcing authenticated communications at every step. A scalable protocol for public key diffusion is introduced in a companion paper.

1 Introduction

When we consider the information exchange related to Network Monitoring, we see that the main actors involved are the producers of monitoring data, and the consumers. We refine such view by considering consumers as parts of a complex activity that manages the tasks submitted by users: we call such distributed activity *Workflow Management* (here including also the monitoring activity successive to task allocation) and Workflow Management Agent (also WMA) the local agents that cooperate in its implementation.

While *allocating the resources* for user tasks, the interest of such agents is for snapshots of recent performances as well as for static capabilities of resources; if a reservation oriented approach is used, resource allocation is carried out by scheduling resource capabilities, without any need of a monitoring activity. In contrast, while *running a user task*, the behavior of the resource must be permanently monitored, in order to guarantee an appropriate quality of service and for accounting purposes.

Such considerations narrow our interest to a subset of what is often considered as Network Monitoring: we exclude the maintenance of pointwise historical traces, needed to respond to unanticipated requests, and instead we consider monitoring activity to be dynamically configured according with WMA requests. As a consequence we do not consider the design of a *repository* for network observations, while we are only marginally interested to the availability of generic aggregated statistics of dynamic behaviors and of static properties of network elements. Instead, we concentrate on the dynamic configuration of the monitoring activity, and to the transfer of streams of observations from producers to WMA.

On the side of the distributed functionality in charge of managing the production of Network Monitoring data, we introduce specialized agents (the Network Monitoring Agents, NMA) in charge of controlling local

capabilities. Such agents are located according with a partitioning of the whole Grid: each partition, a domain in our terminology, is a set of Grid components characterized by a uniform connectivity with the rest of the system. Such abstraction is often used in the Internet architecture, so we have opted for an overloaded term to indicate it. However, it is worth stating that a Network Monitoring domain does not necessarily correspond to a DNS domain, or to a routing AS or area. Equivalence with such existing entities can be stipulated whenever non contradicting the principle of uniform connectivity.

The principle of uniform connectivity is used to justify the collection of aggregate statistics and of static capabilities for network elements between domains, thus limiting monitoring activity. As anticipated, such information is mainly directed to task allocation, which should be preferably addressed using anticipate reservation. In such case, the uniform connectivity requirement may become less stringent.

The rationale behind the introduction of NMAs is the localization of the capabilities and of the workload related to network monitoring. NMAs act as proxies for addressing monitoring requests, and manage the streaming of monitoring data for the whole domain.

Each domain may contain one or more NMAs, which may be responsible for the observation of distinct Network Elements, or related to distinct administrations living within the same domain. They are responsible of controlling Network Monitoring Elements located inside the domain. Network Monitoring Elements (NME) represent resources provided for monitoring the network using appropriate tools.

Figure 1 summarizes the above architecture in a simple system consisting of three domains (large ovals labelled with the domain ID), each with a NMA (a small circle on the border of each oval). Two NMEs are included in domains "FORTH" and "INFN-CNAF", while the other domain "INFN-NA" contains a WMA.

In the design of a NME we remark a relevant distinction between passive and active techniques, that impacts the scalability of the whole architecture. Since passive techniques are notably less intrusive than active ones, we prefer the former, although the latter should be provided as a fallback solution. For instance, in case of a simple request of connectivity monitoring between two sites, the option of a slow ping should be provided in case passive monitoring is not available. Other scenarios should address passive techniques.

To enforce security, NMEs should accept controls only from local NMAs. In their turn, NMAs should accept requests only from peer NMAs, as well as from local WMAs. In that sense domain partitioning improves the flexibility and expandibility of our network monitoring architecture.

The next section analyzes the activity of the NMA, and describes step by step the life-cycle of a monitoring request.

2 The operation inside the Network Monitoring Agent

The purpose of a NMA is to coordinate the monitoring of the networking resources used by the computation coordinated by the WMA. More precisely, we distinguish four distinct activities:

- to accept (*proxying*) network monitoring requests coming from WMAs providing the description of the monitoring activity. Such requests may come either from a WMA inside the same domain, or from another NMA. In either case the request must be authenticated.
- to route the request to another NMA which is able to control an appropriate NME;
- to coordinate the monitoring activity carried out by NMEs;
- to support the streaming of Network Monitoring data to the requesting WMA, possibly through other NMAs.

In the case of *proxying*, a WMA that coordinates a given computational activity will produce a number of *Network Monitoring Session Descriptions*. Such data item is exhaustively described in next section.

Concerning the *request routing* activity, the WMA will forward session descriptions to the local NMA, which will authenticate the request, and forward it to the appropriate NMA. We do not detail how such request is routed, but consider that this operation is based on the accessibility of a database containing

Network Monitoring Agents Descriptions. Such data items map NMAs to domains, define their monitoring capabilities, as well as their connectivity with other NMAs.

The *control of Network Elements* requires knowledge of Network Monitoring capabilities available on NMEs within the local domain.

In order to support the *streaming* of Network Monitoring results, a data channel is built between the NMA in charge of coordinating the monitoring session and the NMA proxying the WMA. In principle such path may traverse several NMAs, and should consider the possibility of optimizing the path in case the same information is requested by many different WMA tasks.

In conclusion, we have identified 3 data structures supporting our Network Monitoring architecture:

- a *local directory* that supports authentication of requests from WMAs in the local domain, as well as the description of local NMEs;
- a *global directory* that supports mutual authentication of NMAs;
- a *network monitoring session description* which contains the description of a single session.

While the design of the *local directory* does not address any challenging aspect, the other two have distinct reasons of interest from a research point of view. The implementation of a *global directory* implies the solution of a number of problems concerning distributed processing, while the *description of a monitoring session* should flexibly cope with the diversity of network monitoring requests.

Here we focus on the latter problem, addressing the reader interested in the former to a specific article [2]: in the next section we introduce the data structure describing a monitoring session as an instance of an *XML Schema Description* document.

3 The XML schema of a Network Monitoring Session

The complex type `NetworkMonitoringSessionType` (its XSD is in the appendix) is the frame for a monitoring request, whose attributes are a sort of header for the Session Description:

SessionId It is a way to identify and refer to a session. Its syntax can be constrained into a URI-like form using an appropriate pattern, which is not considered here;

StartAt It is the requested time when to start the monitoring activity;

Duration It is a timeout, in case the Session is not explicitly closed by the requesting WMA;

BandwidthLimit It is used for negotiation of the multicast facilities, and corresponds to an upper bound of the traffic generated by the monitoring activity, in bytes/second;

Priority Its usage is similar to the above.

Elements are a more composite description of the monitoring activity, which consists of a sequence of elements with complex types:

RequestFrom The agents (possibly more than one) that request the activity. This information is used to generate or extend the multicast tree, as well as to check privileges;

Route The indication of the route the stream is going to follow, represented as a tree of NMAs. The case study in section 3.1 exemplifies its management;

NetworkElement A session monitors a single domain-to-domain path (this is the meaning associated to a Network Element, more restrictive than in RFC2216);

MeasurementStream The description of the low level network monitoring activity. Such data should be passed to the back-end supported tool, which results in the production of a stream of data of known content and syntax.

We opt to indicate one single network element in accordance to the fact that a given session is implemented by a single Network Monitoring Agent. It is impossible to guarantee such fact if several Network Elements are monitored within the same session.

Advanced passive network monitoring tools that are able to observe distinct characteristics of traffic flowing between given endpoints may incorporate such data into a single stream.

The flexibility of the scheme is based on the definition of the type used to describe the `MeasurementStream`, which is where the monitoring tools are indicated and configured. As a general rule, a single frame in the stream will contain several numerical values produced (quasi) synchronously by the same tool activation.

A `MeasurementStream` element contains one or more `CharacteristicStream` elements, each containing the description of a tool activity. Such elements are passed untouched to the NME, each of them corresponding to a frame series in the stream.

Each `CharacteristicStream` element includes a choice of elements containing the controls specific for a given network monitoring tool. Note that we do not consider abstract "characteristics", for instance *roundtrip time*, but make explicit reference to the operational description of their computation. In other words, a ping is a ping, and not a roundtrip time. The WMA is free to use it as a roundtrip time, but it cannot confuse it with a roundtrip time measured during a TCP connect (which is not simply a *protocol* difference). The use of a trade mark (e.g. linux-ping) is OK, but in many cases a more abstract reference to the methodology used to measure it (e.g., ICMP ping) is preferable. The tool wrapper may accept both a tool specific name or a methodology to indicate the same operation. The WMA may indicate either a methodology or a tool specific name, and the NMA should not interfere with such indication. Descriptive statistics (historical average, stddev etc.) are indicated as tool dependent options.

Generic elements are the following:

SamplePeriod The granularity of the time axis, in seconds;

SourceIP A specific monitored IP: this details the monitoring below the *network element* level. Several *SourceIP*'s may be indicated, if the tool supports this, but all should be included in the same source domain: it is the responsibility of the WMA to ensure compliance. The role of the source in the measurement depends on the specific tool (see *SourceDomain*).

DestinationIP same as above.

Concerning the tool specific element, we outline the example of two external XSD documents describing a trivial ping, and a passive monitoring session.

The trivial ping (see the XSD in table 1) is characterized by the endpoints and by a ping frequency, already indicated in the `CharacteristicStream`. Such data is complemented with the length of the packet. Two distinct characteristics can be requested: the roundtrip time, and the packet loss rate.

A sophisticated passive network monitoring tool (we envision a prototype based on the MAPI monitoring library [12]) is shown in table 2. Based on the source and destination addresses, and optionally on the protocol name and the type of a specific application, we can filter and monitor the traffic we are interested in. The `ProtocolName` element can be any network protocol at the transport layer (such as TCP and UDP) while `ApplicationName` may correspond to any Grid-related application (such as HTTP, GridFTP, and Globus). The identification of a specific application in the Grid network traffic can be as simple as looking for a static port number, or more complex based on deep packet inspection, application-level protocol decoding, or other heuristics. The measurement frequency is defined using the `SamplePeriod` element, that is part of the `CharacteristicStreamType`.

Other options for the passive monitoring tools include requests for anonymization of sensitive fields in the results (e.g., IP addresses) and use of a third host, whenever needed, for gathering and correlating the results.

```
<schema
 xmlns="http://www.w3.org/2001/XMLSchema"
 xmlns:pt="http://www.di.unipi.it/~augusto/schema/PingTool.xsd"
 targetNamespace="http://www.di.unipi.it/~augusto/schema/PingTool.xsd">


 <annotation>
  <documentation xml:lang="en">
   Network Monitoring Tool Ping.
   Copyright CoreGRID. All rights reserved.
   Version 0.0
  </documentation>
 </annotation>

<complexType name="PingOptionsType">
 <sequence>
  <element name="PacketSize"
                type="integer"
                minOccurs="0"/>
 </sequence>
 <attribute name="CharacteristicId"
                type="pt:PingCharacteristicIdType"
                use="required"/>
</complexType>

<simpleType name="PingCharacteristicIdType">
 <restriction base="string">
   <enumeration value="RoundTrip"/>
   <enumeration value="PacketLoss"/>
 </restriction>
</simpleType>

</schema>
```

Table 1: Trivial Ping Options

The interested reader finds in companion papers the description of the techniques used to measure round-trip time [8], packet loss rate [9], available bandwidth, and per-application bandwidth usage [1].

3.1 A case study: monitoring Processor to Storage connectivity

A simple example illustrates the request of an active monitoring session between a Storage and a Computing Element to monitor their connectivity through an ICMP ping (see table 3).

The origin of the Network Monitoring Session descriptor is the WMA represented as a green circle inside the INFN-NA domain (see figure 1). The WMA has no hints about the Network Monitoring Architecture, so it delivers a bare MeasurementStream instance to the local NMA.

At this point the Measurement Stream is encapsulated into a Network Monitoring Session description, and routes the request to the known NMA at one end of the Network Element. The identifier of the forwarding NMA is placed in the route stack.

The NMA in the INFN-CNAF domain discovers that it cannot handle the request: there is no ping wrapper on the Computing element, and therefore the monitoring activity cannot be carried out. It forwards the NetworkMonitoringSession instance to the known NMA on the other Network Element endpoint, FORTH, pushing its own address on the stack.

The next NMA discovers that the storage element is equipped with a ping wrapper: therefore it extracts the MeasurementStream description from the Session description, and delivers it to the NME co-located with the Storage Element. It also discovers that it is adjacent to the NMA in the INFN-NA domain, and eliminates the intermediate INFN-CNAF agent from the Route stack.

The NME activates a ping process, formatting the data coming from such process according to its specifications, and forwarding successive frames to the local NMA, which in its turn encapsulates the frames by indicating the Session they belong to and passing them to the next NMA in the stack.

In our case this is the NMA located at INFN-NA, which decapsulates the data and passes it to the WMA, which is able to unmarshall the data contained in the datagram according with tool specifications, and process the data.

The WMA finally interrupts the monitoring session notifying the local NMA, which propagates the request according to the route stack known to it. When the request reaches FORTH NMA, it stops the monitoring activity on the computing element. Alternatively, FORTH NMA will perform the same activity when the "Duration" timeout expires. Intermediate NMA's will suspend and remove the registration of the session from their soft state.

4 Related works

The coordination of a network monitoring infrastructure is a matter of active research. The first effort in this sense is probably the Network Weather Service [13], which still offers relevant suggestions. However, such prototype indicates but solves only partially the real challenges of a coordinated network monitoring architecture: scalability and security.

Successive studies mainly focussed towards the publication of network monitoring results in view of retrospective analysis: this option limits the application of such infrastructures to those scenarios where monitoring requests are *planned* and concentrate on a restricted subset of routes. Without such limit any solution is deemed to unscalability, since the number of routes grows with the square of the number of resource elements in the network.

Such scenario is nonetheless of great practical relevance: administrative monitoring, as well as accounting or diagnosis fall into the category of a monitoring task that concentrates on few routes, known a priori. To cite some of the works on this trail, we cite the Globus MDS [11], and EGEE network performance monitoring architecture [5].

In this paper we explore another facet of the problem, which is relevant to cope with *unplanned* monitoring requests. The interest for such aspect of network monitoring is that monitoring requests from the agents responsible for the coordination of Grid jobs cannot be anticipated, they extend to a limited lifetime, they

```
<schema
 xmlns="http://www.w3.org/2001/XMLSchema"
 xmlns:am="http://www.di.unipi.it/~augusto/schema/MAPIMonitoringTools-0.1.xsd"
 targetNamespace="http://www.di.unipi.it/~augusto/schema/MAPIMonitoringTools-0.1.xsd">


 <annotation>
  <documentation xml:lang="en">
   Passive Network Monitoring Tools (FORTH).
   Copyright CoreGRID. All rights reserved.
   Version 0.0
  </documentation>
 </annotation>

 <complexType name="MAPIMonitoringToolsOptionsType">
  <sequence>
   <element name="ProtocolName"
                type="string"
                minOccurs="0"/>
   <element name="ApplicationName"
                type="string"
                minOccurs="0"/>
   <element name="Anonymize"
                type="string"
                minOccurs="0"/>
   <element name="ThirdParty"
                type="string"
                minOccurs="0"/>
  </sequence>
  <attribute name="CharacteristicId"
                type="am:MAPIMonitoringToolsCharacteristicIdType"
                use="required"/>
 </complexType>

 <simpleType name="MAPIMonitoringToolsCharacteristicIdType">
  <restriction base="string">
    <enumeration value="RoundTripTime"/>
    <enumeration value="PacketLossRate"/>
    <enumeration value="AvailableBandwidth"/>
    <enumeration value="UsedBandwidth"/>
  </restriction>
</simpleType>

</schema>
```

Table 2: MAPI options

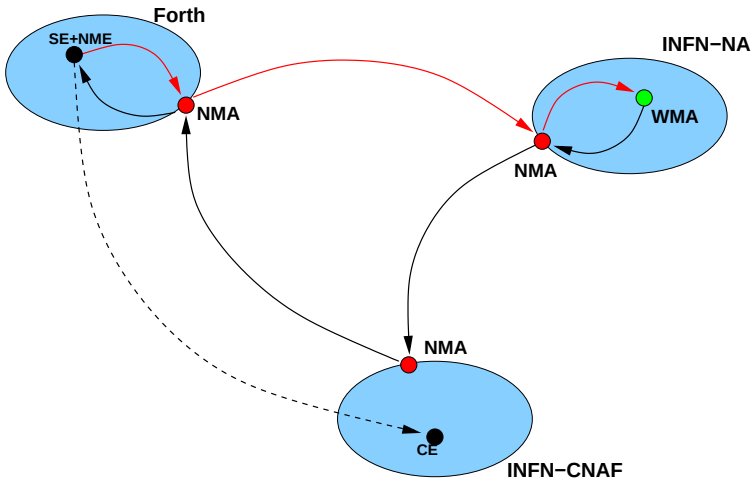

Figure 1: Information flow related to a ping session: the green circle indicates a WMA, black arrows indicate the flow of a Network Monitoring Session description representing a request, red circles represent NMAs, black circles represent monitored sites, and red arrows represent the data stream from the NME to the WMA.

```
<?xml version="1.0"?>

  <nmsd:NetworkMonitoringSession
    xmlns:nmsd="http://www.di.unipi.it/~augusto/schema/
                NetworkMonitoringSessionDescription-0.4.xsd"
    SessionId="456@this.NMagent.ip"
    StartAt="2007-09-17T12:00:00.000-01:00"
    Duration="2H">
    <RequestFrom TaskId="WF245" WorkflowMonitoringAgentId="OurBroker@FORTH"/>
    <Route>
      <NextAgent Agent="NMAgent@FORTH" Index="1"/>
      <NextAgent Agent="Theodolite@CNAF" Index="2"/>
    </Route>
    <NetworkElement SourceDomain="FORTH" DestinationDomain="CNAF"/>
    <MeasurementStream>
      <CharacteristicStream CharacteristicStreamId="1">
        <SamplePeriod>5</SamplePeriod>
        <SourceIP>processor_1.ics.forth.gr</SourceIP>
        <DestinationIP>ftp.cnaf.infn.it</DestinationIP>
        <PingOptions CharacteristicId="RoundTrip">
          <PacketSize>2048</PacketSize>
        </PingOptions>
      </CharacteristicStream>
    </MeasurementStream>
  </nmsd:NetworkMonitoringSession>

```

Table 3: XML instance for the example in figure 1

have a moderate (if any) need of historical data, mainly to improve measurement robustness. Such aspect of network monitoring is far less studied, but exhibits a number of challenges: flexibility, since new requests must be activated dynamically for scalability reasons, and security, since network monitoring is an expensive activity, and requests must be authenticated.

Our approach to this aspect of network monitoring is marginally related to the past experience with *planned* network monitoring. The problems raising in the two cases are too different to justify a common solution: one for all, *unplanned* network monitoring in principle does not need a measurements database, while *planned* network monitoring relies on the availability of a powerful repository for measurements (think for instance to the R-GMA [4] architecture). Therefore we aimed at a different approach.

The architecture we propose is an evolution of [3] and its design has been influenced by Internet streaming protocols: the basic requirements are those announced in [6], but our embrional solution for the request of a Network Monitoring Session is also inspired to the Internet SIP [7] protocol. We also take into account the RTP [10] protocol as for the components of a network monitoring request. In analogy to the *application profiles* introduced in RTP, that characterize the payload in a flexible and expandable way, we opted for a *monitoring tool* oriented description, instead of a *characteristic oriented* approach. Just like in the case of RTP, the *neutrality* of an approach that leaves to monitoring tool designers the freedom to introduce new measurements that do not exactly match existing characteristics, and to workflow managers designers the ability to use them, leaves space to research and new products in the rapidly evolving field of network monitoring tools.

5 Conclusions

We introduce a distinction between planned and unplanned network monitoring activities: we claim that each of them exhibits challenging aspects, and requires distinct solutions, although the latter is receiving less attention than the former from the research community.

The fact that unplanned activities are requested by Workflow Management Agents introduces the need of a scalable and flexible authentication scheme. Once they are activated their output should not be stored for future use, but directly delivered to the requester with a lightweight streaming protocol. The request and reply protocol should be flexible and allow the integration of new monitoring tools.

In this paper we address a fundamental step in the design of a solution for the management of unplanned monitoring activity, which consists in the definition of the information needed to describe a single monitoring session, and the scope of such entity. In order to give an intuitive framework, we outline the architecture of the network monitoring infrastructure, identifying the actors and their inter-play.

A Network Monitoring Session Schema

```
1  <schema
2   xmlns="http://www.w3.org/2001/XMLSchema"
3   xmlns:pt="http://www.di.unipi.it/~augusto/schema/PingTool.xsd"
4   xmlns:am="http://www.di.unipi.it/~augusto/schema/AppmonTool.xsd"
5   xmlns:nmsd="http://www.di.unipi.it/~augusto/schema/
6               NetworkMonitoringSessionDescription-0.4.xsd"
7   targetNamespace="http://www.di.unipi.it/~augusto/schema/
8                    NetworkMonitoringSessionDescription-0.4.xsd"
9   elementFormDefault="unqualified"
10  attributeFormDefault="unqualified">
11
12  <import namespace="http://www.di.unipi.it/~augusto/schema/PingTool.xsd"/>
13  <import namespace="http://www.di.unipi.it/~augusto/schema/AppmonTool.xsd"/>
14
15  <annotation>
```

```
 <documentation xml:lang="en">
  Network Monitoring Session Description.
  Copyright CoreGRID. All rights reserved.
  Version 0.1
 </documentation>
</annotation>

<element name="NetworkMonitoringSession"
               type="nmsd:NetworkMonitoringSessionType"/>

<element name="comment" type="string"/>

<complexType name="NetworkMonitoringSessionType">
 <sequence>
  <element name="RequestFrom"
               type="nmsd:WorkflowMonitoringTaskType"
               maxOccurs="unbounded"/>
  <element name="Route"
               type="nmsd:RouteStackType"
               minOccurs="0"/>
  <element name="NetworkElement"
               type="nmsd:NetworkElementType"/>
  <element name="MeasurementStream"
               type="nmsd:MeasurementStreamType"/>
 </sequence>
 <attribute name="SessionId"
               type="string"
               use="required"/>
 <attribute name="StartAt"
               type="dateTime"
               use="required"/>
 <attribute name="Duration"
               type="duration"
               use="required"/>
 <attribute name="BandwidthLimit"
               type="nonNegativeInteger"
               default="0"/>
 <attribute name="Priority"
               type="nonNegativeInteger"
               default="0"/>
</complexType>

<complexType name="WorkflowMonitoringTaskType">
 <attribute name="TaskId"
               type="string"/>
 <attribute name="WorkflowMonitoringAgentId"
               type="string"/>
</complexType>

<complexType name="RouteStackType">
 <sequence>
 <element name="NextAgent" minOccurs="0" maxOccurs="unbounded">
  <complexType>
   <attribute name="Agent"
               type="string"/>
   <attribute name="Index"
```

```
                type="nonNegativeInteger"/>
   </complexType>
  </element>
  </sequence>
 </complexType>

 <complexType name="NetworkElementType">
  <attribute name="SourceDomain"
                type="string"
                use="required"/>
  <attribute name="DestinationDomain"
                type="string"
                use="required"/>
 </complexType>

 <complexType name="MeasurementStreamType">
  <sequence>
   <element name="CharacteristicStream"
            minOccurs="1" maxOccurs="unbounded">
    <complexType>
     <sequence>
      <element name="SamplePeriod"
                type="float"
                minOccurs="0"/>
      <element name="SourceIP"
                type="string"
                minOccurs="0"
                maxOccurs="unbounded"/>
      <element name="DestinationIP"
                type="string"
                minOccurs="0"
                maxOccurs="unbounded"/>
      <choice>
       <element name="PingOptions"
                        type="pt:PingOptionsType"/>
       <element name="AppmonOptions"
                        type="am:AppmonOptionsType"/>
      </choice>
     </sequence>
     <attribute name="CharacteristicStreamId"
                type="string"/>
    </complexType>
   </element>
  </sequence>
 </complexType>

</schema>
```

References

[1] Demetres Antoniades, Michalis Polychronakis, Spiros Antonatos, Evangelos P. Markatos, Sven Ubik, and Arne Øslebø. Appmon: An application for accurate per application network traffic characterization. In *In IST Broadband Europe 2006 Conference*, 2006.

[2] A. Ciuffoletti. The wandering token: Congestion avoidance of a shared resource. In *Austrian-Hungarian Workshop on Distributed and Parallel Systems*, page 10, Innsbruck (Austria), September 2006.

[3] Augusto Ciuffoletti and Michalis Polychronakis. Architecture of a network monitoring element. In *CoreGRID workshop at EURO-Par 2006*, page 10, Dresden (Germany), August 2006.

[4] A. Cooke, A. Gray, L. Ma, W. Nutt, J. Magowan, P. Taylor, R. Byrom, L. Field, S. Hicks, and J. et Al. Leak. R-GMA: An information integration system for grid monitoring. In *Proceedings of the Eleventh International Conference on Cooperative Information Systems*, 2003.

[5] EGEE. *Network Performance Monitoring Architecture*, october 2006.

[6] Sally Floyd, Van Jacobson, Ching-Gung Liu, Steven McCanne, and Lixia Zhang. A reliable multicast framenwork for light-weight sessions and application level framing. *IEEE/ACM Transactions on Networking*, November 1996.

[7] H. Handley, H. Schultzrinne, Schooler E., and J. Rosenberg. SIP: Session initiation protocol. Request for Comment 2543, Network Working Group, March 1999.

[8] Hao Jiang and Constantinos Dovrolis. Passive estimation of TCP round-trip times. *SIGCOMM Comput. Commun. Rev.*, 32(3):75–88, 2002.

[9] Antonis Papadogiannakis, Alexandros Kapravelos, Michalis Polychronakis, Evangelos P. Markatos, and Augusto Ciuffoletti. Passive end-to-end packet loss estimation for grid traffic monitoring. In *Proceedings of the CoreGRID Integration Workshop*, 2006.

[10] H. Shultzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. Request for Comment 1889, Audio-Video Transport Working Group, January 1996.

[11] The Globus Team. Globus Toolkit 2.2 MDS Technology Brief, Jan 2003. Draft.

[12] Panos Trimintzios, Michalis Polychronakis, Antonis Papadogiannakis, Michalis Foukarakis, Evangelos P. Markatos, and Arne Øslebø. DiMAPI: An application programming interface for distributed network monitoring. In *Proceedings of the 10^{th} IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2006.

[13] Rich Wolski. Dinamically forecasting network performance using the Network Weather Service. Technical Report TR-CS96-494, University of California at San Diego, January 1998.