Issues about the Integration of Passive and Active Monitoring for Grid Networks

S. Andreozzi[2], D. Antoniades[1], A. Ciuffoletti[2], A. Ghiselli[2], E.P. Markatos[1],
M. Polychronakis[1], P. Trimintzios[1]

[1] FORTH-ICS
P.O. Box 1385
71110, Heraklion, GREECE,
{ptrim,mikepo,danton,markatos}@ics.forth.gr
[2] CNAF-INFN
Via Berti Pichat 6/2
40126 - Bologna, ITALY
augusto@di.unipi.it,{sergio.andreozzi,antonia.ghiselli}@cnaf.infn.it

Abstract. In this paper we discuss the issues arising with the integration of passive and active monitoring techniques for Grid network infrastructure monitoring. Our proposal is related to the monitoring information *production* and *publication*. Initially, we present the context of our work within the Grid world. We enlist the range of different techniques to perform measurements and obtain monitoring data. We propose a number of interesting performance metrics of the quality of the Grid infrastructure connectivity, and the related passive and active monitoring techniques that are required in order to obtain these metrics. We qualitatively discuss both the accuracy with which we can measure each metric, as well as the complexity and overhead induced by the monitoring activity. We also look at the impact of the information that various measurement metrics may have on other modules of other actors in a Grid monitoring infrastructure. We also discuss into the issues behind the efficient representation and publication of monitoring from both passive and active techniques. We show that, when these techniques are applied to Grids, they should be merged with a hybrid design strategy. Finally, we discuss the tradeoffs introduced by such approach, and describe the components that support a domain oriented Grid monitoring infrastructure that supports both passive and active monitoring tools.

1 Introduction

Grid computations require three kinds of resources: Storage, Computing, and Communication resources. Grid aware applications need to know the characteristics of such resources in order to setup their execution environment. With current tools, monitoring and understanding the availability and status of Storage and Computing resources is sufficiently precise, can be easily translated into database schemas, and could be used for early experiments in system resources optimization. Monitoring of Communication resources, on the other hand, is much more complicated since we have the complexity of the interconnection infrastructure, which includes monitoring large-scale services, while we also want to manage and control the overall measurement infrastructure.

Monitoring the network infrastructure of a Grid has a vital role in the management and the utilization of the Grid itself. While it gives to maintenance activities the basic information for identifying network problems and diagnosing the cause, thus contributing to Grid fault tolerance, it also provides to Grid-aware applications the ability to undertake actions in order to improve performance, as well as resource utilization. In the latter category, we also include accounting activities that are important when Grid resources are shared by different administrative authorities.

According to the Global Grid Forum (GGF) schema [3], the overall network infrastructure monitoring activities can be divided into three distinct activities, according to the treatment of what we call the *observations*: their *production*, their *publication*, and their *utilization*. Although the three activities tightly interoperate, based on carefully designed interfaces between them, each of them uses different tools that are appropriate for the specific activity. Network monitoring tools are used for the *production* activity, powerful databases for the *publication*, and various other techniques, such as visualization tools for administration and workflow analysis for Grid aware applications, for the *utilization*.

In this paper we are focusing on Grid network infrastructure monitoring tools related to the monitoring *production* and *publication* activity of the GGF. For the *production* we are proposing a number of interesting metrics of the quality of the Grid infrastructure connectivity, and the monitoring techniques that are required in order to obtain these metrics. At this stage of our work, we qualitatively discuss both the accuracy with which we can measure each metric, as well as the complexity and overhead induced by the monitoring activity. We are also looking at the impact of the information that various measurement metrics may have on other modules of other actors in a Grid monitoring infrastructure.

For the *publication* activity, which is done via powerful databases, we are mainly interested in the efficient representation of both active and passive monitoring metrics. The issue of interest here is the induced complexity when the various monitoring *producers* are increasing in size and monitoring data output. Scalability is one of our main concerns. Being able to extend the monitoring coverage of the the Grid to hundreds of nodes requires the careful design of a distributed hierarchical *publication* database architecture. In this work we have as a starting point a *per-domain* architecture and will try to look into making the database hierarchical.

The rest of this paper is organized as follows. In Section 2 we look into the various alternatives for classifying monitoring tools and techniques, namely the passive versus active monitoring techniques, and the end-to-end versus the single-link approaches. In Section 3 we describe the potential architecture of using a number of passive monitors distributed at ingress and egress points of each Grid cluster. Also in this section we identify new performance metrics that can be derived with the use of a single or pairs of passive monitoring sensors.

In Section 4 we look at the current status of the Grid connectivity active monitoring architecture, which we will integrate the passive architecture presented in Section 3. In Section 5 we describe the issues and the potential approaches for integration of the metrics induced by passive monitoring into the *publication* infrastructure which currently supports *only* metrics derived from active monitoring, such as the round trip time (RTT). Section 6 addresses a number of security and privacy concerns related to our integrated monitoring architecture. Finally, Section 7 summarizes the main goals and issues addressed in this paper.

2 Classification of Monitoring Approaches and Techniques

In this section we will look into the different existing monitoring approaches. First we will look into the distinction between end-to-end and single link monitoring. As a second classification attribute we will use the intrusiveness of the monitoring technique, i.e., if it induces additional network traffic, where we will describe the active and passive approaches to deriving monitoring metrics.

2.1 Finding a Compromise Between Link and Path Monitoring

One issue that emerges when considering network monitoring is related to the scope of monitoring. We envision two main alternatives:

single link - it gives the view from a single observation point. It is good for maintainers, that need a fine grain view of the network in order to localize a problem, but inappropriate for Grid-aware applications, that may need end-to-end observations; Note that correlation of the information from multiple single links may provide monitoring metrics appropriate for some Grid applications, as we show in this paper.

end-to-end path - it gives a view of the system that is filtered through routing: this may be sometimes inappropriate for maintainers, but is essential for Grid aware applications.

However, one key feature that differentiates the two approaches is the cardinality of complexity: let N be the number of resources in the system, in the case of a link oriented approach this is $O(N)$, since the Grid can be assimilated to a bounded degree graph, while in the case of a path-oriented approach the address space is $O(N^2)$, since, as a general rule, each resource has a distinct path to any other resource.

This complexity consequence seems to exclude the adoption of a end-to-end path approach, but there are other problems with the single-link approach:

- edges are often black boxes that contain proprietary software: there may be no way to modify or add code for monitoring purposes, while there are many problems even to simply access the stored data;

- deriving an end-to-end path performance metric from single-link observations requires two critical steps: to reconstruct the link sequence, and, even more problematic, to obtain time correlated path performance compositions from single-link observations;

From the above it is obvious that no single approach is the most appropriate for all monitoring data *production* purposes. It seems therefore appropriate to complement the two strategies, in order to limit their drawbacks.

One strategy is to introduce an overlay network that groups network clusters into a single *domain*, and restricts monitoring to inter-domain paths. Such strategy, which resembles intra-/inter-domain routing separation of the Internet, finds a compromise between the two extreme design strategies outlined above:

- like an *end-to-end path strategy*, it offers Grid oriented application a valuable insight of the path connecting two resources. However, such insight does not include the performance of the local network (which, by the way, usually outperforms inter-domain paths), and the address space is still $O(N^2)$, although now N stands for the number of domains, which should be significantly smaller than the number of resources;
- like a *single link strategy*, it provides the maintainers with a reasonable localization of a problem. As for accounting, as long as domains are mapped to administrative entities, it gives sufficient information to account resource utilization.

In essence, a *per-domain approach* limits the complexity of the address space into a range that is already managed by routing algorithms, avoids path reconstruction and has a granularity that is compatible with relevant tasks. The basic idea behind this approach is to group nodes into *domains* and use both the *domain-to-domain path* and the *single domain* strategies.

However, this *per-domain approach* it introduces an overlay view that cannot be derived by a pre-existent structure: the Domain Name System (DNS) structure is not adequate to represent the whole monitored domain, since the same DNS subnetwork may in principle contain several monitoring domains, and a domain may also overlap several DNS subnetworks. The overlay network (or *domain partition*) must be separately designed, maintained, and made available to users. We will consider issues related to this component in section 5.

2.2 Passive and Active Monitoring Techniques

There is another classification that often emerges when dealing with network monitoring tools. This classification scheme distinguishes between active and passive monitoring. The definition itself is slippery, and often a matter of discussion. For our purpose, we will adopt the following classification criterion:

> *a monitoring tool is classified as* active *when it induces traffic into the network, otherwise it is* passive.

One feature shared by passive monitoring tools, which we prefer to call *traffic observers*, is that they give an extremely detailed view of the performance of the network, while active tools, which here we call *network benchmarks*, return a response that combines several performance figures (for instance consider using the usual ping, which includes delay, loss, etc.).

As a general rule, effective network monitoring should exploit both kinds of tools:

- a network benchmark is more effective to monitor network sanity;
- network benchmarks are also suitable for application oriented observations (like jitter, when related to multimedia applications);
- traffic observers are more appropriate to monitor gross connectivity metrics, like link available bandwidth;
- traffic observers are needed for accounting purposes

Simplifying, while a network benchmark is required to discover an unused network resource, a traffic analyzer is needed to detect that another is nearly congested.

In the following, we will discuss both passive and active monitoring in the context of monitoring data *production* for Grid infrastructures. Then we will also look into the *publication* of monitoring information and the issues behind using *domain-wide* measurement databases. Finally, we will discuss some unwanted but unavoidable security and privacy issues that arise from the Grid network infrastructure monitoring activities.

3 Passive Network Monitoring for Grid Infrastructures

Passive traffic monitoring has become increasingly vital for network management, as well as for supporting a growing number of automated control mechanisms needed to make IP-based networks more robust, efficient, and secure. Passive network monitoring techniques analyze network traffic by capturing and examining individual packets passing through the monitored link, allowing for fine-grained operations, such as deep packet inspection.

The main benefit of passive network monitoring, compared to active monitoring techniques, is its *non-intrusive* nature. Active network monitoring techniques incur an unavoidable network overhead due to the injected probe packets, which compete with user traffic. In contrast, passive network monitoring techniques passively observe the current traffic of the monitored link, without introducing any network overhead. At the same time, passive monitoring provides accurate measurements of fine-grained traffic characteristics.

Besides passive network monitoring applications based on data gathered at a single observation point, emerging applications can benefit from monitoring data gathered at multiple observation points across a network. Such a distributed monitoring infrastructure [1] can be extended outside the border of a single organization and span multiple administrative domains across the Internet. In such an environment, the processing and correlation of the data gathered at

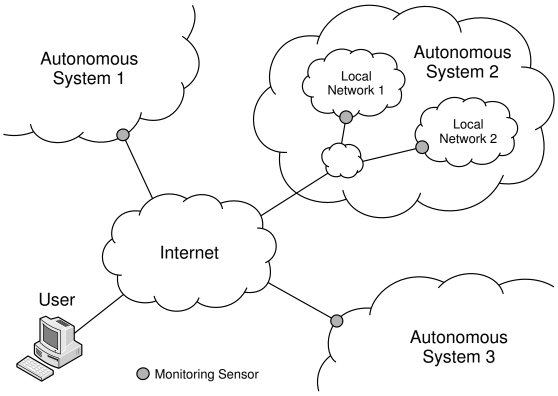

Fig. 1. A high-level view of a distributed passive network monitoring infrastructure.

each sensor gives a broader perspective of the state of the monitored network, in which related events become easier to identify.

Figure 1 illustrates a high-level view of such a distributed passive network monitoring infrastructure. Monitoring sensors are distributed across several autonomous systems (AS), with each AS operating one or more monitoring sensors. Each sensor may monitor the link between the AS and the Internet (as in AS 1 and 3), or an internal link of a local sub-network (as in AS 2). An authorized user, who may not be located in any of the participating Autonomous Systems (ASes), can run monitoring applications that require the involvement of an arbitrary number of the available monitoring sensors.

A passive monitoring infrastructure, either local or distributed, can be used to derive several end-to-end connectivity performance metrics, which are useful to the Grid applications for assessing the status of the Grid infrastructure connectivity and take effective balancing decisions. Some of these metrics could be measured using intrusive active monitoring techniques, but passive techniques have the benefit of not injecting any additional traffic into the network. However, there are also several metrics measurable by passive monitoring techniques, that cannot be measured using active monitoring. In the following sections we enlist several of these metrics, classifying them based on the number of passive monitoring observation points required to derive them.

3.1 Metrics Using a Single Observation Point

In this section we discuss elementary, i.e., not summarized, measurement metrics that can be derived with the help of a single passive monitoring observation point, located usually at the link that connects the domain this the rest of the Grid infrastructure.

Network-level Round-Trip Time The network Round-Trip Time (RTT) is the time taken for a packet to traverse the network from the source to the destination and back. RTT is one of the simplest network connectivity metrics, and can be easily measured using active monitoring tools like for example `ping`. However, it is also possible to measure RTT using solely passive monitoring techniques. One such technique is based on monitoring the TCP connections that goes through a link [8]. RTT can be estimated more accurately based on the time difference between the `SYN` and `ACK` packets exchanged during the three-way handshake of a TCP connection.

Application-level Round-Trip Time Besides the network RTT time, passive monitoring allows for measuring the RTT time at the service level, i.e., the time that a client has to wait in order to receive a response from a remote service for a particular request. For example, web server response time, as perceived by the end user, can be measured by monitoring the traffic between the user and the web server. By inspecting the contents of the packets, one can distinguish a request for a particular page and the relevant reply, and then compute the service response time based on their time difference. Similar techniques are used in EtE [7], which measures service performance characteristics using passive monitoring.

Note that the application-level RTT is composed of the network-level RTT plus the delay in the server. Both these metrics could be measured, the first one by `ping`s or our suggested technique in section 3.1, and the latter with host-based resource availability tools, but the composed metric will not be as accurate as our direct approach since the time of the other the composed metric does not have the time correlation aspects.

Throughput Being able to observe the actual data transferred between end points, passive monitoring can provide traffic throughput metrics at varying levels of granularity. The aggregate throughput provides an indication for the current utilization of the monitored link. Based on the current conditions, i.e., the throughput seen by the active connections, this metric may provide the means to estimate the future aggregate throughput. Consequently, as a proportion of the total link capacity, it provides an estimate for the available bandwidth of the link.

Besides aggregate throughput, fine-grained per-flow measurements can be used to observe the throughput achieved by specific applications. This metric can be measured using the appropriate filters based on known ports, or specified IP addresses, or both. Even for applications that do not use predefined ports, protocol-inspection techniques can be used to identify the traffic they produce, and quantify it [10].

Retransmitted Packets In case that packet loss cannot be measured (e.g., because only one observation point is available—see Section 3.2), the amount of retransmitted packets provides a good indication of the quality of the route towards their destination.

Packet loss ratio can be measured using a single monitor by tracking the packets that are sent multiple times during a given time window. However, storing all the outgoing packets that passed through the link during the time window is a highly resource-consuming task, especially for high speed links.

Furthermore, comparing each new packet to the already captured packets for finding duplicates is a very computationally-intensive task. Techniques similar to those used in trajectory sampling [6] can be used in order to keep only digests of the packets, in order to reduce the space requirements, and search them more efficiently.

Packet Reordering Packet reordering, as reported in [9], can play a significant role in degrading application throughput, even and in small occurrence. In order to measure the percentage of reordered packets, a single passive monitor can observe the sequence field of incoming TCP packets. Since this kind of monitoring uses only header-level information would be computationally inexpensive, but also could help to avoid highly reordering links in order to achieve maximum application throughput.

3.2 Metrics Using Multiple Observation Points

In this section we discuss measurement metrics that can be derived with the help of a a pair of passive monitoring observation points, each located at the link that connects the domain to the rest of the Grid infrastructure, or more monitoring points across several domains.

One-Way Delay and Jitter The one-way delay is the time taken for a packet to traverse the path form the source to the destination. The asymmetric routing that commonly occurs within the Internet makes this metric important for some applications. The one-way delay can be measured using two passive monitors located at the source and destination network domains. When the same packet passes through both monitors, the one-way delay can be measured from the difference in the time each monitor observed the packet. For such measurements, the clocks of the monitors have to be synchronized, e.g., using the Network Time Protocol (NTP) or synchronizing with the Global Positioning System (GPS), depending on the required accuracy.

A closely related metric is the *variation* in the one-way delay of successive packets, commonly referred to as jitter. Jitter is particularly important for real-time applications, since it predetermines the sizes of the relevant stream buffers.

Note, that both these metrics can be measured with active monitoring techniques, which suffer from the trade-off between accuracy and amount of additional test traffic injected into the network. The passive monitoring approach discussed here does not add any additional traffic, while it is as accurate as the synchronized clocks in the monitoring observation points.

Packet Loss Ratio Packet loss occurs when correctly transmitted packets from a source never arrive at the intended destination. Packets are usually lost due to congestion, e.g., at the queue of some router, routing system problems, but also poor network conditions may result to datagram damage. The packet loss ratio is a very important metric, since it affects data throughput performance and overall end-to-end quality.

Facilitating passive monitoring observation points, packet loss can be measured using two cooperating monitors at the source and destination network domains. These monitoring will keep track the packets that have been sent from the source network but have not arrived to the destination after a timeout period. The timeout period has to be greater than the one-way delay between the domains, though to be on the safe side for extreme delays, values greater than RTT should be used.

Service Availability The domain and service availability metric is a major concern for Grid users. For example, in the case where a `SYN` packet does not have a `SYN-ACK` response, that means that the domain is not available. So passively counting the unestablished connections, both in network and application level, can give us an indication of the availability of a particular domain/service. Correlating the results from several monitoring points can be a good measurement of the availability.

4 Active Monitoring for Grid Infrastructures

We will now focus our discussion on the use of active monitoring tools and techniques for Grid connectivity and network infrastructure monitoring data *production*. Basically, such tools induce a test traffic benchmarks into the Grid connectivity infrastructure, and observe the behavior of the network. As a general rule, one end (the *probe*) generates a specific traffic pattern, while the other (the *target*) cooperates by returning some sort of feedback. For example, the `ping` tool is the most well known representative of this category.

Disregarding the characteristics of the benchmark, an active monitoring tool reports a view of the network that is near to the needs of the application: for instance, a `ping` message that uses the Internet Control Message Protocol (ICMP) gives an indication of raw transmission times, useful, for instance, for multimedia streaming, while a `ping` that uses TCP packets or a short `ftp` transfer may be used to gather information that needed in order to optimize file transfers. Since active tools report the same network performance that the application will observe, their results are readily usable by Grid-aware applications that want to optimize their performance.

The coordination activity associated to active monitoring is minimal: this is relevant for a dynamic entity, as a Grid, where join and leave events are frequent. A new resource that joins the Grid enters the monitoring activity simply by starting its *probe* and *target* related activities. However, join and leave activities introduce security problems, that are further addressed in Section 6.

Most of the statistics collected by active tools have a local relevance, and need not be transmitted elsewhere: as a general rule, they are used by applications that run in the domain where the probe resides. A distributed *publication* engine may take advantage of that, exporting to the global view only those observations that are requested by remote *consumers*.

Network performance statistics that can be observed using an active tool can be divided into two categories:

packet oriented: which observe the behavior induced by single packet transmissions between the measurement points. Besides packet round trip delay, using appropriate probes one may also observe TCP connection setup characteristics and one-way figures of packet delay and packet delay variation;

stream oriented: which observe the behavior induced by a sequence of packets with given characteristics. Such characteristics may bind timing, length, content with variable levels of tolerance. For instance, an `ftp` transfer of a randomly generated file of given length, or a back-to-back sequence of `UDP` packets of given length.

A relevant feature shared by active monitoring tools is the ability to detect the absence of a resource, disregarding that it is used or not, since it requires an active participation of all actors (probe, target and network). This not only helps fault tolerance, but may also simplify the maintenance of the Grid layout, which is needed by Grid-aware applications.

Since active monitoring consumes some resources, security rules should limit the impact of malicious uses of such tools: this issue is also covered in Section 6.

5 The Domain Overlay Database

As discussed in the introduction the *publication* of monitoring data is another of the three important Grid infrastructure monitoring activities. The main tools to realize the *publication* activities is by using powerful databases.

The domain overlay database is a cornerstone of a domain-based architecture. The structure of such architecture reflects a view of the Grid oriented to network performance, and its implementation addresses performance and scalability.

The GlueDomains [5],[4] prototype serves as a starting point for our study. GlueDomains supports the network monitoring activity of the prototype Grid infrastructure of INFN, the Italian Institute for Nuclear Physics. GlueDomains follows a *domain-oriented* approach, as defined above. Monitoring activity results are published using Globus Monitoring and Discovery System (MDS) [11], which is the information services component of the Globus Toolkit that provides information about the available resources on the Grid and their status, and rendered through the GridICE [2] toolset.

The domain overlay maps Grid resources into domains, and introduces further concepts that are specific to the task of representing the monitoring activity. In order to represent such overlay view, we use the Unified Model Language

(UML) graph outlined in Figure 2. The classes that represent Grid resources are the following:

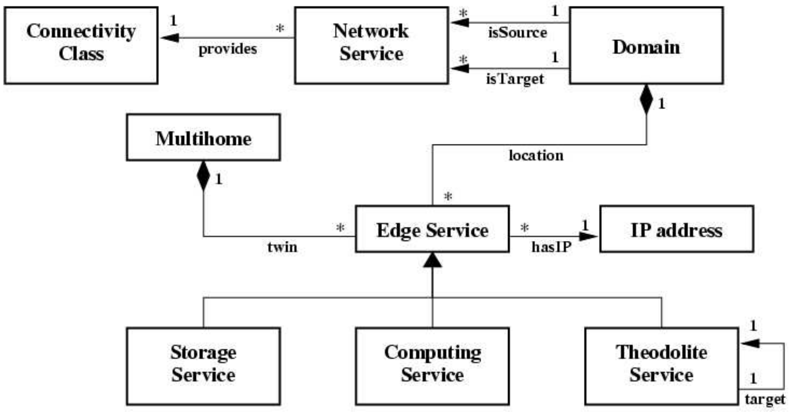

Fig. 2. The UML diagram of the topology database with domain partitioning

Edge Service: it is a superclass that represents a resource that does not consist of connectivity, but is reached through connectivity.

Network Service: represents the interconnection between two Domains. Its attributes include a class, corresponding to the offered service class, and a statement of expected connectivity.

Theodolite Service: a Theodolite Service monitors a number of Network Elements: its attributes include the service class provided by the Network Service it monitors. In GlueDomains Theodolites perform *active* network monitoring.

The following classes represent aggregations of services:

Domain: represents the partitions that compose the Grid. Its attributes include the service class offered by its fabric.

Multihome: represents an aggregation of Edge Services that share the same hardware support, but are accessible through distinct interfaces.

The description of the overlay network is made available through a *topology database*. Such database is accessed rather frequently, since we assume that consumers are interested to statistics between edge services, while the publication engine is addressed by network services entries: a typical query, like the RTT between a storage service and a computing service, entails finding the domains containing the two edge services, and the network service between the two domains, in order to access the publication engine.

Although the dynamic nature of the Grid requires frequent updates of the topology database, it is primarily accessed for reading: this fact suggests the replication of parts of this database inside each domain. GlueDomains, our reference implementation, is based on a centralized topology database, which is adequate to the small scale of the specific Grid. In order to extend the coverage of the Grid to hundreds of domains we need the realize distributed support for the database.

Integration with passive monitoring: this *domain-oriented* database approach within GlueDomains was designed having in mind metrics only *produced* with active monitoring tools. It is clear though that this approach also smoothly fits with the performance metrics structure we described in Sections 3.1-3.2. All measurement data collected by passive monitoring traffic observers can be associated to a specific network service and domain, since basic attributes (like source and destination IP address, service class, etc.) are typically provided by such devices. The knowledge of theodolites as hosts *relevant* from the point of view of network monitoring may indicate the devices performing passive monitoring which packets are more significant, thus opening the way to the cooperation between theodolites and passive traffic observers.

5.1 Description of Monitoring Activities

Also relevant to the management of the monitoring activity is its description. In order to limit human intervention to the design and initial deployment of network monitoring, the description of the monitoring activity should be available to devices that contribute to this task. The next steps are to investigate the possibility of *self-organization* of such activity.

In the case of GlueDomains, theodolite services are the agents that are responsible of network monitoring using active monitoring techniques. The a UML model shown in Figure 3 is centered around such entity, describes the structure of the *monitoring database*.

The activity, which in GlueDomains is preformed of active monitoring tools, is organized into monitoring *sessions*. Each session is associated to a theodolite, which runs the monitoring tool, and to a monitored network service. The description of the monitoring session contains the description of the monitoring tool, and the details of the benchmark traffic.

The *monitoring database* is accessed infrequently by *producers*, that download the description of their monitoring tasks. This may happen once during a

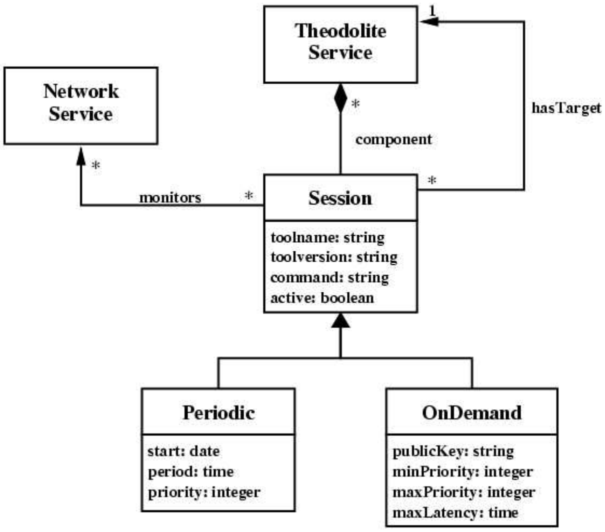

Fig. 3. The UML diagram of the monitoring database

monitoring session, or periodically. Updates are bound to some kind of topology change. Both read and update activities should be restricted to authorized producers, and limited to the records that describe its activity.

Integration with passive monitoring: passive monitoring fits into this diagram, either as a new session class, where the theodolite instructs the remote passive monitoring device about the required activity, or as a new service class, with associated *passive monitoring* sessions. In the former case, an authentication mechanism should be introduced to avoid unauthorized use of passive monitoring devices.

6 Security and Privacy

A large-scale network monitoring infrastructure is exposed to several threats: each component should be able to ensure an appropriate degree of security, depending on the role it plays.

Monitoring sensors may become targets of coordinated Denial of Service (DoS) attacks, aiming to prevent legitimate users from receiving a service with acceptable performance, or sophisticated intrusion attempts, aiming to compromise the monitoring hosts. Being exposed to the public Internet, monitoring sensors should have a rigorous security configuration in order to preserve the confidentiality of the monitored network, and resist to attacks that aim to compromise it.

The security enforcement strategy is slightly different for active and for passive monitoring tools. In the case of passive monitoring tools, the monitoring host should ensure the identity and the capabilities associated with a host submitting a request. Such a request may consist in activating a given packet filter, or in returning the results of the monitoring activity. Each passive sensor should be equipped with a firewall, configured using a conservative policy that selectively allows inbound traffic according with accepted requests, and dropping inbound traffic from any other source. One option is to consider that only theodolite services, whose credentials (for instance their public keys) are recorded in the monitoring database, are able to access passive sensor configuration, and therefore dynamically configure its firewall. Theodolite capabilities may vary according to a specific monitoring strategy.

In the case of active monitoring tools, the target is exposed to DoS attacks, consisting in submitting benchmark traffic from unauthorized, and possibly malicious, sources. One should distinguish between tools that are mainly used for discovery, and those that are used for monitoring purposes. The former should be designed as lightweight as possible, for instance consisting of a predetermined ping pattern: probe's firewall shouldn't mask such packets, unless their source is reliably detected as threatening. The latter might consist in rather resource consuming patterns, and the probe should filter packets according to an IP based strategy: such configuration would be based on the content of the monitoring database.

Both passive and active monitoring tools have in common the need of ensuring an adequate degree of *confidentiality.* In fact, data transfers through TCP are unprotected against eavesdropping from third-parties that have access to the transmitted packets, since they can reconstruct the TCP stream and recover the transferred data. This would allow an adversary to record control messages, forge them, and replay them in order to access a monitoring sensor and impersonate a legitimate user. For protection against such threats, communication between the monitoring applications and a remote sensors is encrypted using the Secure Sockets Layer protocol (SSL). Furthermore, in a distributed monitoring infrastructure that promotes sharing of network packets and statistics between different parties, sensitive data should be *anonymized* before made publicly available, due to security, privacy, and business competition concerns that may arise between the collaborating parties.

From this picture emerges the role of the monitoring database as a kind of certification authority, which is also used as a repository of public keys used by the actors of the monitoring activity: the publication engine, the monitoring tools and the theodolite services. Its distributed implementation is challenging, yet tightly bound to the scalability of the monitoring infrastructure.

7 Summary and Conclusions

In this paper we discussed the issues arising with the integration of passive and active monitoring techniques for Grid network infrastructure monitoring. Our proposal was related to the monitoring *production* and *publication* activities of the GGF. For the *production* we proposed a number of interesting performance metrics of the quality of the Grid infrastructure connectivity, and the related monitoring techniques that are required in order to obtain these metrics. We qualitatively discussed both the accuracy with which we can measure each metric, as well as the complexity and overhead induced by the monitoring activity. We also looked at the impact of the information that various measurement metrics may have on other modules of other actors in a Grid monitoring infrastructure.

For the *publication* activity, which is done via databases, we were mainly interested in the efficient representation of both the active and passive monitoring metrics. The issues of interest here was the induced complexity when the various monitoring *producers* are increasing is size and monitoring data output. Scalability was also one of our main concerns. Being able to to extend the monitoring coverage of the Grid to hundreds of nodes requires the careful design of a distributed hierarchical *publication* database architecture. In this work we proposed as a starting point the *per-domain* architecture and in our future endeavors we will try to look into making the information in database available in distributed fashion among many domains.

This work is a preliminary study of the issues behind the integration of passive and active monitoring. Our target is to reach to an integrated system for Grid network infrastructure monitoring. The second target is look into the scalability issues behind this integrated architecture. Our intent is to make a quantitative scalability assessment and analysis identifying potential scalability bottlenecks. Based on the results of this assessment we will investigate potential ways to reduce the impact of these bottlenecks. Potential avenues for solving the scalability issues will be to use the publish/subscribe model, the threshold crossing/alarming ideas, the "divide and conquer" principle, and ideas from peer-to-peer systems communication.

References

1. *LOBSTER: Large-scale Monitoring of Broadband Internet Infrastructures.* Information available at: http://www.ist-lobster.org.
2. S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. Tortone, and V. Cristina. Gridice: a monitoring service for the grid. In *Third Cracow Grid Workshop*, Cracow, Poland, October 2003.
3. R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, B. Tierney, and R. Wolski. A grid monitoring architecture. Recommendation GWD-I (Rev. 16, jan. 2002), Global Grid Forum, 2000.
4. A. Ciuffoletti. The wandering token: Congestion avoidance of a shared resource. Technical Report TR-05-13, Universit¡E0¿ di Pisa, Largo Pontecorvo - Pisa - ITALY, May 2005.

5. A. Ciuffoletti, T. Ferrari, A. Ghiselli, and C. Vistoli. Architecture of monitoring elements for the network element modeling in a grid infrastructure. In *Proc. of Workskop on Computing in High Energy and Nuclear Physics*, La Jolla (California), March 2003.
6. N. G. Duffield and M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, pages 271–282, 2000.
7. Y. Fu, L. Cherkasova, W. Tang, and A. Vahdat. EtE: Passive end-to-end Internet service performance monitoring. In *Proceedings of the USENIX Annual Technical Conference*, pages 115–130, 2002.
8. H. Jiang and C. Dovrolis. Passive estimation of tcp round-trip times. *SIGCOMM Comput. Commun. Rev.*, 32(3):75–88, 2002.
9. L. Michael and G. Lior. The effect of packet reordering in a backbone link on application throughput. *Network, IEEE*, 16(5):28–36, 2002.
10. M. Polychronakis, K. G. Anagnostakis, E. P. Markatos, and A. Øslebø. Design of an Application Programming Interface for IP Network Monitoring. In *Proceedings of the 9th IFIP/IEEE Network Operations and Management Symposium (NOMS'04)*, pages 483–496, Apr. 2004.
11. The Globus Toolkit 4.0 Documentation. *GT Information Services: Monitoring & Discovery System (MDS)*. Available at: http://www.globus.org/toolkit/mds/.