ON THE INTEGRATION OF PASSIVE AND ACTIVE NETWORK MONITORING IN GRID SYSTEMS

Sergio Andreozzi, Augusto Ciuffoletti, Antonia Ghiselli
INFN-CNAF
Viale Berti Pichat 6/2, 40126, Bologna, Italy
sergio.andreozzi@cnaf.infn.it
augusto@di.unipi.it
antonia.ghiselli@cnaf.infn.it

Demetres Antoniades, Evangelos P. Markatos, Michalis Polychronakis
Institute of Computer Science, Foundation for Research & Technology – Hellas
P.O. Box 1385, 71110, Heraklio, Greece
danton@ics.forth.gr
markatos@ics.forth.gr
mikepo@ics.forth.gr

Panos Trimintzios
European Network and Information Security Agency
P.O. Box 1309, 71001, Heraklio, Greece
panagiotis.trimintzios@enisa.eu.int

Abstract This paper focuses on the integration of passive and active network monitoring techniques in Grid systems. We propose a number of performance metrics for assessing the quality of the connectivity, and describe the required measurement methods for obtaining these metrics. Furthermore, the issue of efficiently representing and publishing the measured values is considered. We show that it is important to have both active and passive monitoring strategies applied to Grid systems; and when we do have both strategies it is necessary to have an a priory hybrid design. Finally we depict the tradeoffs introduced by this approach and the description of the components for a domain oriented monitoring infrastructure that supports both passive and active monitoring tools in Grid systems.

Keywords: Grid connectivity, network monitoring, active monitoring, passive monitoring, connectivity performance metrics, distributed database.

1. Introduction

The Grid computation system paradigm extends the traditional distributed computing approach towards the coordination and sharing of computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations. In order to setup an optimal execution environment for a Grid application, knowledge about the status, characteristics and composition of the various resources is required. In current systems, monitoring and understanding of characteristics, status and availability of computing and storage resources has been extensively explored (e.g., see [1]) and working solutions on large-scale systems exist (e.g., see [11]). In contrast, monitoring of communication resources is at an early stage, mainly due to the complexity of the infrastructure to monitor and of the monitoring activity.

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 and 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 Grid Monitoring Architecture (GMA) [3], defined in the context of the Global Grid Forum (GGF) [8], the overall network infrastructure monitoring can be divided into three distinct phases: the *production* of observations, their *publication*, and their *utilization*. The three activities tightly interoperate based on carefully designed interfaces among them, although each of them uses different tools. Network monitoring tools are used for the *production*, powerful databases and publication services following different delivery and data models are used for the *publication*, and various other techniques, such as administration and workflow analysis visualization tools, are used for the *utilization*.

In this paper, we focus on network monitoring from the Grid viewpoint, and we concentrate on tools related to the *production* and *publication* activities of observations. For the *production* activity, we propose a number of metrics related to the quality of the Grid connectivity. We also describe the monitoring techniques that are required for obtaining these metrics. We qualitatively discuss both the accuracy with which we can derive each metric, as well as the complexity and overhead induced by the measurement process. For the *publication* activity, we are mainly interested in the efficient representation of both active and passive monitoring metrics. Our primary concern is the scalability when producers are increasing in number and monitoring data output. In order to limit the quantity of observations that need to be published, we also propose a *domain-oriented* overlay network.

The rest of this paper is organized as follows. In Section 2, we classify existing network monitoring tools and techniques. Section 3 describes the proposed network monitoring architecture, comprising passive sensors distributed at ingress and egress points of Grid resources, and presents performance metrics that can be derived using single or pairs of passive monitoring sensors. Section 4 presents the current Grid connectivity monitoring architecture based on active network monitoring. In Section 5 we describe the issues and potential approaches for the integration of passive network monitoring into the *publication* infrastructure, which currently supports *only* metrics derived using active monitoring, such as the Round Trip Time (RTT). Section 6 addresses security and privacy concerns related to our integrated monitoring architecture. Finally, Section 7 concludes the paper.

2. Classification of Network Monitoring Techniques

In this section, we classify network monitoring approaches based on two different criteria. We first look into the distinction between *path*- and *link*-oriented monitoring. Then, we classify network monitoring approaches based on whether they use *active* monitoring or *passive* monitoring strategies.

2.1 Link versus Path Monitoring

An important issue that emerges when considering network monitoring is related to the monitoring granularity. We consider two main alternatives: (1) *Single link* is appropriate for maintainers that require a fine-grained view of the network in order to localize problems; nevertheless, it is not suitable for most of the Grid-aware applications, since they require end-to-end observations and typically cannot derive the necessary information from the correlation of measurements regarding multiple single links; (2) *End-to-end path* gives a view of the system that is filtered through routing; this may be sometimes confusing for maintainers, but is appropriate for Grid-aware applications.

The scalability of the two approaches is dramatically different. Let N be the number of resources in the system. A link oriented monitoring system grows with $O(N)$, since a Grid can be assimilated to a bounded degree graph. On the other side, an end-to-end (or path-oriented) approach, grows with $O(N^2)$, since, as a general rule, each resource has a distinct path to any other resource. This consideration would exclude the adoption of an end-to-end path approach, but there are issues to be considered with the single-link approach. First, the edges of each link are often black boxes containing proprietary software; there may be no way to add sensors for monitoring purposes, or even to simply access the stored data. Second, 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 considerations given above, it is obvious that no single approach is the most appropriate for all monitoring purposes. We propose to complement the two strategies in order to limit their drawbacks. Our strategy is to introduce an overlay network that clusters networked services into *domains*, and restricts monitoring to inter-domain paths. This approach, which resembles the inter/intra domain routing dichotomy in the Internet, strikes a balance between the two extreme design strategies outlined below:

- An *end-to-end path strategy* offers to Grid oriented applications a valuable insight of the path connecting two resources. However, this insight does not include the performance of the local network, which usually outperforms inter-domain paths, and the address space is still $O(N^2)$. Nevertheless, it must be considered that N now stands for the number of domains, which should be significantly smaller than the number of resources.

- A *single link strategy* provides maintainers with a reasonable localization of a problem. Regarding accounting, as long as domains are mapped to administrative entities, it gives sufficient information to account resource utilization.

In essence, a *domain-oriented* 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 implied overlay view cannot be derived from a pre-existent structure. For instance, the Domain Name System (DNS) is not adequate to map monitoring domains, since the same DNS subnetwork may in principle contain several monitoring domains, and a domain may overlap with several DNS subnetworks. Thus, the overlay network, or *domain partition*, must be separately designed, maintained, and made available to users, as explained in Section 5.

2.2 Passive versus Active Monitoring

Another classification scheme that is often used when dealing with network monitoring distinguishes between active and passive monitoring techniques. The definition itself is rather slippery, and often a matter of discussion. For this work, we adopt the following classification criterion: a monitoring tool is classified as *active* if it induces traffic into the network, otherwise it is classified as *passive*.

Passive monitoring is more appropriate for monitoring gross connectivity metrics like link throughput; it is also needed for accounting purposes. 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 monitoring approaches, compared to active monitoring, 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

Active monitoring is more effective for observing the network sanity and is suitable for application oriented observations, such as jitter, when related to multimedia applications. On the other side, this approach implies an unavoidable network overhead due to the injected probe packets which compete with user traffic.

Passive monitoring tools can give an extremely detailed view of the network's performance, while active tools return a response that combines several performance figures. As a general rule, effective network monitoring should exploit both techniques. In the following two sections we discuss both passive and active monitoring in the context of the data *production* for Grid infrastructures.

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. Besides monitoring a single link, emerging applications can benefit from monitoring data gathered at multiple observation points across a network. Such a distributed monitoring infrastructure [15]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 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 domains, with each domain operating one or more monitoring sensors. Each sensor may monitor the link between the domain and the Internet (as in domain 1 and 3), or an internal link of a local sub-network (as in domain 2). An authorized user, who may not be located in any of the participating domains, can run monitoring applications that require the involvement of an arbitrary number of the available monitoring sensors.

A passive network monitoring infrastructure, either local or distributed, can be used to derive several performance metrics useful to Grid applications for

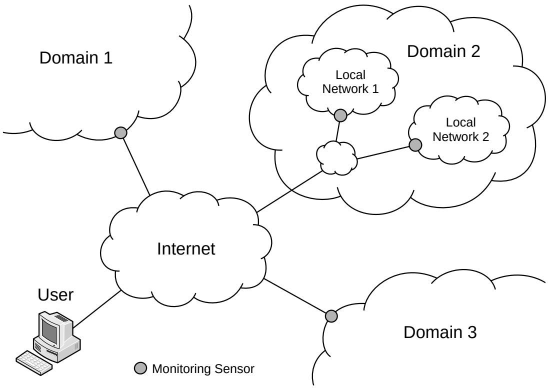

Figure 1. A high-level view of a distributed passive network monitoring infrastructure.

assessing the status of the Grid infrastructure connectivity and taking effective balancing decisions. Although some of these metrics could be measured using active monitoring techniques, passive techniques have the benefit of not injecting any additional traffic into the network. Furthermore, 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, classified based on the number of passive monitoring observation points required to derive them.

3.1 Metrics based on a Single Observation Point

In this section, we present basic metrics that can be measured using passive monitoring from single observation point. This observation point can be located usually at the link that connects the domain with the rest of the Grid infrastructure.

3.1.1 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 pass through a link [10]. 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.

3.1.2 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 by the network-level RTT plus the delay in the server. Both these metrics could be measured: the first by `ping`s or using the technique in Section 3.1.1; the second by means of host-based resource availability tools. Nevertheless, the composed metric will not be as accurate as the direct approach since the latter does not have to deal with time correlation aspects.

3.1.3 Throughput.

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, 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 [13].

3.1.4 Retransmitted Packets.

In case that packet loss cannot be measured (e.g., because only one observation point is available, see Section 3.2.2), 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, reduce the space requirements, and search them more efficiently.

3.1.5 Packet Reordering. Packet reordering, as reported in [12], can play a significant role in degrading application throughput, even 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, it would be computationally inexpensive, and also could help to avoid highly reordering links in order to achieve maximum application throughput.

3.2 Metrics based on Multiple Observation Points

In this section, we discuss metrics that can be derived using either 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 distributed across several domains.

3.2.1 One-Way Delay and Jitter. The one-way delay is the time taken for a packet to traverse the path from 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.

3.2.2 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; they can also be lost due to routing system problems, or due to poor network conditions that may result to damages in the datagram. The packet loss ratio is a very important metric, since it affects data throughput performance and overall end-to-end quality.

In passive monitoring observation points, packet loss can be measured using two cooperating monitors at the source and destination network domains. The two sensors will 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 must 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.

3.2.3 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, meaning that the domain is not available. By passively counting the unestablished connections, both in network and application level, can give us an indication of the availability of a particular domain or service. Correlating the results from several monitoring points can be a good measurement of the availability.

4. Active Network Monitoring for Grid Infrastructures

Active tools induce test traffic 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 end (the ‘target’) cooperates by returning some kind of feedback. The `ping` tool is a 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 applications like multimedia streaming. A `ping` that uses `UDP` packets or a short `ftp` session may be used to gather the necessary information for optimal file transfers. Since active tools report the same network performance that the application would 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 a relevant property for a dynamic entity, such 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, which 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 active monitoring techniques can be divided into two categories: (1) ‘packet oriented’, related to the behavior induced by single packet transmissions between the measurement points; (2) ‘Stream oriented’, related to the behavior induced by a sequence of packets with given characteristics such as the timing and the length of the packet stream or the content of individual packets.

In the first category, we find RTT, `TCP` connection setup characteristics and one-way figures of packet delay and packet delay variation. In the second category, we find `ftp` transfer of a randomly generated file of given length, or a back-to-back sequence of `UDP` packets.

A relevant feature shared by active monitoring tools is the ability to detect the presence of a resource, disregarding if it is used or not, since they require 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

The domain overlay database is a cornerstone of a domain-based architecture. The structure of this architecture reflects a view of a Grid focusing on 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 National Institute for Nuclear Physics [9]. GlueDomains follows a domain-oriented approach, as defined in Section 2.1. The measured values are published using the Globus Monitoring and Discovery Service (MDS) [14]. MDS is the information services component of the Globus Toolkit that provides information about the available resources on a Grid and their status. This service is the official information service of a large-scale Grid such as the LHC Computing Grid [11]. The published information is rendered through GridICE [2], a Grid monitoring tool.

The domain overlay maps Grid resources into domains and introduces concepts specific to the task of representing the monitoring activity. We illustrate this overlay view using the Unified Model Language (UML) class diagram presented in Figure 2. The classes that represent Grid resources are the following: ‘Edge Service’, that is a superclass representing a resource that does not consist of connectivity, but is reached through connectivity; ‘Network Service’, representing the interconnection between two Domains; its attributes include a class, corresponding to the offered service class, and a statement of expected connectivity; ‘Theodolite Service’, it monitors a number of Network Elements; in GlueDomains, theodolites perform active network monitoring.

The following classes represent aggregation of services: ‘Domain’, that is a representation of partitions that compose a Grid; its attributes include the service

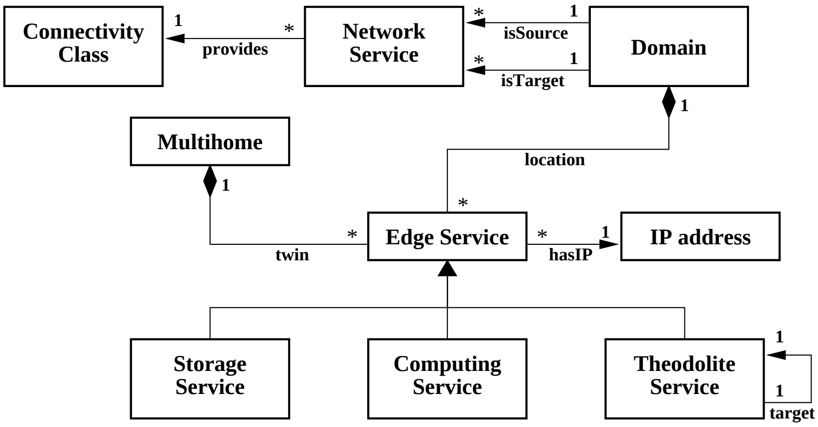

Figure 2. The UML class diagram of the topology database with domain partitioning

class offered by its fabric; ‘Multihome’, that aggregates Edge Services sharing the same hardware support, but being accessible through distinct interfaces.

The description of the overlay network using the above classes is made available through a ‘topology database’ which is used by the ‘publication’ engine in order to associate observations to network services.

Integration with passive monitoring. The *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 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 (e.g., source and destination IP address, service class) are typically provided by such devices. The knowledge of theodolites as hosts relevant from the viewpoint 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 Monitoring Activities Description

The description of the monitoring activity is relevant to its management. In order to limit human intervention in the design and deployment of the network monitoring infrastructure, such a description should be available to devices that contribute to this task, also considering the possibility of self-organization of such an activity.

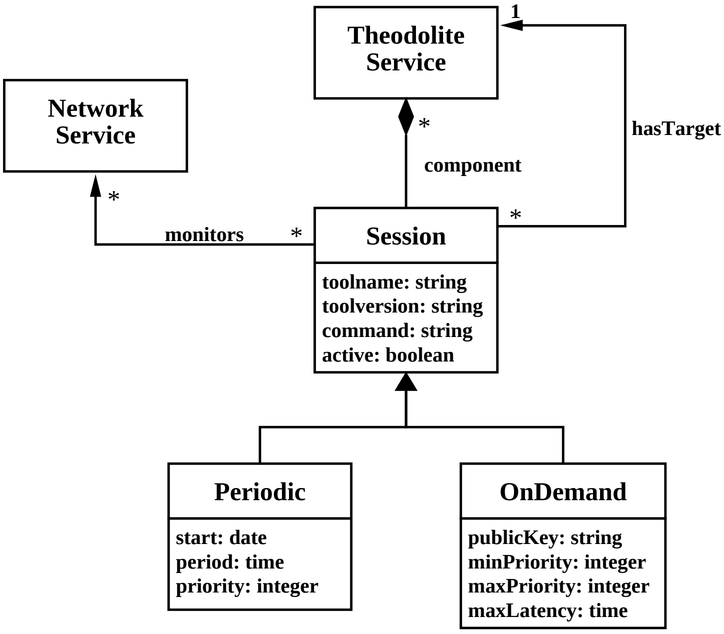

Figure 3. The UML class diagram of the monitoring database

In GlueDomains, theodolite services are agents of monitoring configuration. The UML model shown in Figure 3 is centered around such entity and describes the structure of the monitoring database.

In GlueDomains, active monitoring 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 each session contains details of the monitoring tool and details about the injected traffic.

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

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 them.

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. An 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 should not 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 explore the issues arising from the integration of passive and active monitoring techniques when used for Grid network infrastructure monitoring. Our proposal is related to the monitoring of the *production* and *publication* activities as defined by the GGF.

For the production activity, we propose a number of interesting performance metrics related to the quality of the connectivity of the Grid infrastructure, and the related network monitoring techniques that are required for obtaining 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 induced information that various measurement metrics may have on the modules of other actors in a Grid monitoring infrastructure.

For the publication activity, which is deployed in the form of databases, we are mainly interested in the efficient representation of both the active and passive monitoring metrics. The issues of interest in this case is the induced complexity when the various monitoring producers are increasing in size and the monitoring data output is growing in volume. Scalability is also one of our main concerns. Being able to extend the monitoring coverage of a Grid to hundreds of nodes requires the careful design of a distributed hierarchical publication database architecture. In this work, we propose as a starting point the per-domain architecture, where the Grid infrastructure is divided into domains. 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 first approach towards studying the issues behind the integration of passive and active monitoring. Our target is to reach an integrated system for monitoring the network infrastructure with a Grid-specific point of view. Our second target is to perform a further analysis of the scalability issues of the integrated architecture. In future activities, we aim at making a quantitative scalability assessment and analysis identifying potential bottlenecks. Based on the results of this assessment, we plan to investigate ways to reduce the impact of these bottlenecks. Potential avenues for solving the scalability issues are to use the publish/subscribe model, the threshold crossing/alarming ideas, the ‘divide and conquer’ principle, and techniques from peer-to-peer systems communication.

References

[1] S. Andreozzi, S. Burke, L. Field, S. Fisher, B. Kónya, M. Mambelli, J.M. Schopf, M. Viljoen, and A. Wilson. GLUE Schema Specification - Version 1.2, Dec 2005.

[2] Sergio Andreozzi, Natascia De Bortoli, Sergio Fantinel, Antonia Ghiselli, Gian Luca Rubini, Gennaro Tortone, and Vistoli Cristina. GridICE: a Monitoring Service for Grid Systems. *Future Generation Computer Systems Journal, Elsevier*, 21(4):559–571, 2005.

[3] Ruth Aydt, Dan Gunter, Warren Smith, Martin Swany, Valerie Taylor, Brian Tierney, and Rich Wolski. A Grid Monitoring Architecture. Recommendation GWD-I (Rev. 16, jan. 2002), Global Grid Forum, 2000.

[4] Augusto Ciuffoletti. The Wandering Token: Congestion Avoidance of a Shared Resource. Technical Report TR-05-13, Universitá di Pisa, Largo Pontecorvo - Pisa -ITALY, May 2005.

[5] Augusto Ciuffoletti, Tiziana Ferrari, Antonia Ghiselli, and Cristina Vistoli. Architecture of Monitoring Elements for the Network Element Modeling in a Grid Infrastructure. In *Proceedings of Workskop on Computing in High Energy and Nuclear Physics (CHEP2003)*, La Jolla (California), March 2003.

[6] Nick G. Duffield and Matthias 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] Yun Fu, Ludmila Cherkasova, Wenting Tang, and Amin Vahdat. EtE: Passive End-to-End Internet Service Performance Monitoring. In *Proceedings of the USENIX Annual Technical Conference*, pages 115–130, 2002.

[8] Global Grid Forum, 2006. http://www.ggf.org.

[9] INFN – Istituto Nazionale di Fisica Nucleare, 2006. http://www.infn.it.

[10] Hao Jiang and Constantinos Dovrolis. Passive Estimation of TCP Round-Trip Times. *SIGCOMM Comput. Commun. Rev.*, 32(3):75–88, 2002.

[11] LCH Computing Grid, 2006. http://www.cern.ch/lcg.

[12] Laor Michael and Gendel Lior. The Effect of Packet Reordering in a Backbone Link on Application Throughput. *Network, IEEE*, 16(5):28–36, 2002.

[13] Michalis Polychronakis, Kostas G. Anagnostakis, Evangelos P. Markatos, and Arne 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, April 2004.

[14] The Globus Toolkit 4.0 Documentation. *GT Information Services: Monitoring & Discovery System (MDS)*, 2006. Available at: `http://www.globus.org/toolkit/mds/`.

[15] 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 10th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2006.