CMST 11(2) 91-100 (2005)

DOI:10.12921/cmst.2005.11.02.91-100

COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY **11(2)**, 91-100 (2005)

Towards multi-domain monitoring for the European research networks

Jeff W. Boote[1], Eric L. Boyd[1], Jerome Durand[2], Andreas Hanemann[3], LoukikKudarimoti[4], Roman Łapacz[5], Nicolas Simar[4], Szymon Trocha[5]

[1] *Internet2, 1000 Oakbrook Drive, Suite 300, Ann Arbor, MI 48104, USA*
e-mail: {boote|eboyd}@internet2.edu

[2] *GIP Renater, 151 boulevard de l'Hôpital, 75013 Paris, France*
e-mail: Jerome.durand@renater.fr

[3] *German Research Network, c/o Leibniz Supercomputing Center, Barer Str. 21, D-80333 Munich, Germany*
e-mail: hanemann@lrz.de

[4] *DANTE, 126-130 Hills Road, Cambridge CB2 1PG, United Kingdom*
e-mail: {loukik.kudarimoti|nicolas.simar}@dante.org.uk

[5] *Poznań Supercomputing and Networking Center, Noskowskiego 12/14, 61-704 Poznań, Poland*
e-mail: {romradz|szymon.trocha}@man.poznan.pl

Abstract: We propose the creation of a multi-domain measurement framework with dynamic characteristics identical to that of the network as a whole. Our approach recognises and facilitates the ability of independent network entities to set policies and limits on the use of measurement resources locally while encouraging and facilitating the use of such resources by users interested in network paths that traverse remote administrative domains.

Key words: networks, end-to-end performance, multi-domain monitoring, monitoring framework, active monitoring, passive monitoring

1. INTRODUCTION

In recent years, research and education networks dramatically increased the core bandwidth and introduced new services which have been demanded by users to allow for the deployment of Grid services and innovative applications. The trouble is that users located in different networks who encounter an end-to-end performance problem do not have standard access to network performance characteristics. These characteristics are needed in order to help the users understand and solve their issues. Available bandwidth, transmission delay, jitter and packet loss are examples of this. Today, networks make use of numerous tools for monitoring a variety of these characteristics. Each network domain along the path between the two users has its own set of performance data and its own policies to access this data which are often restricted to itself or its users.

In practice, a problem may arise while using videoconferences to share lectures between different universities. In case of quality degradation in the network connection to a remote lecture, it has to be determined which administrative domain is responsible. Often, it may be quite easy to verify whether the problem is located inside the local area network. But the situation may be much more complex if the problem is located in any of the other domains involved. In this case, information has to be requested from the staff of the intermediate backbone networks, which may be quite time consuming. This slow problem-discovery sometimes makes it impossible to provide important parts of a lecture to the participating students.

An extensive user survey for a group of European National Research and Education Networks (NRENs) has shown the lack of a unified performance measurement system. Such a system should enable the user to check the network performance by having access to the provider's measurement data concerning connection performance characteristics. In addition, it should be useful to locate and resolve performance problems. In the survey, users have declared their willingness to make the necessary performance information of their domains available for end-to-end monitoring.

The approach presented here is the first result of the collaboration between the GN2 Joint Research Activity 1 (JRA1) [1] and Internet2 [2] Performance Architecture and Technology team (PAT). Our goal is to provide a framework for monitoring network connections. The infrastructure design developed and the implementation will follow a consistent approach that respects the multi-domain organisation of the networking environment and identified user requirements.

The rest of the paper is organised as follows. In Section 2, a brief overview of the monitoring framework addressed in the project is given. Details about the middle

layer of this framework where a service-oriented architecture involving different kinds of services will be applied are given in Section 3. A use-case for the collaboration of services in a single domain environment is presented in Section 4. Its extension with respect to multi-domain environments is explained is Section 5. The implementation of a prototype for the system, which is called PerfSONAR (Perfmance focused Service Oriented Network monitoring ARchitecture), and the further project schedule are the subjects of Section 6. Section 7 outlines related work in the context of multi-domain network monitoring, while the last section concludes the paper.

2. MONITORING FRAMEWORK

The general monitoring framework, which is refined during JRA1 is depicted in Fig. 1. The main requirements for the framework design, which have been identified in the requirement phase of the project, are flexibility, security, scalability, and fault tolerance.

The Measurement Points are the lowest layer in the system and are responsible for measuring and storing network characteristics as well as for providing basic network information. The measurements can be carried out by active or passive monitoring techniques. The Measurement Point Layer of a domain consists of different monitoring components or agents deployed within the domain. A monitoring agent provides information on a specific metric (*e.g.*, one-way delay, jitter, loss, available bandwidth) by accessing the corresponding Measurement Points. Each network domain can, in principle, deploy Measurement Points of its choice.

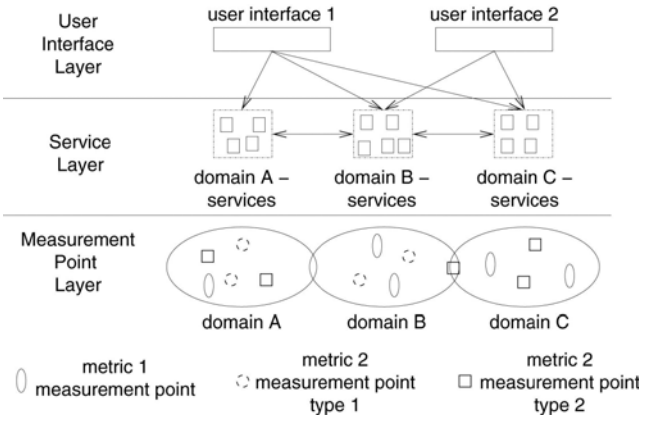

Fig. 1. JRA1 architecture proposal

The Service Layer is the middle layer of the system and consists of administrative domains. It allows for the exchange of measurement data and management information between domains. In each domain, a set of entities (services) is responsible for the domain control. Each of them is in charge of a specific functionality, like authentication and authorisation, discovery of the other entities providing specific functionalities, resource management or measurement of network traffic parameters. The interaction of the entities inside a domain as well as the access to the Measurement Point Layer or other domains may not be visible to the end user. Some of the entities contain an interface which can be accessed by the User Interface Layer.

The User Interface Layer consists of visualisation tools (user interfaces) which adapt the presentation of performance data to be appropriate for the needs of specific user groups. In addition, they may allow users to perform tests using the lower layers of the system. From the user interface perspective, the Service Layer provides an additional level of abstraction to hide the differences between Measurement Points deployed in the different domains.

The aim of the design is to provide the main functionalities in the Service Layer as independent entities to allow for increased flexibility for the system: existing elements may be replaced easily or new ones inserted. Even if the number of entities is large, each one can be identified and invoked using discovery functionalities.

3. SERVICE LAYER AND MEASUREMENT POINT LAYER SERVICES

There are three general categories of performance measurement data, *i.e.*, active and passive measurement results as well as network state variables that can be thought of as data producers. From the user or network administrator point of view, analysis tools, threshold alarms, and visualisation graphs can be thought of as data consumers. Between data producers and data consumers is a pipeline of aggregators, correlators, filters, and buffers, which can be thought of as data transformers and data archives. Data producers, consumers, transformers and archives are all resources that need to be discovered and (possibly) protected from over-consumption using authentication and authorisation. A services-based measurement framework implements each of these roles as an independent service: Lookup (LS), Authentication (AS), Measurement Archive (MA), Transformation (TS) and Resource Protector (RP). These services form the Service Layer. Measurement Point (MP) services and also form Measurement Point Layer of monitoring components. Users of any service, whether they are end user applications or other services, are classified as clients. Providers of any service are classified as servers. Therefore, many services can be both client and server, depending upon the context. To achieve this, all data providers implement a publisher interface and all data consumers implement a subscriber interface. When a data flow is requested, the consumer provides a handle to a subscription interface if it wants a push interaction. If it does not provide a subscription

handle, the data producer creates a publisher interface that the consumer can poll.

3.1. Measurement Point service

The Measurement Point service (MP) creates and publishes measurement data by initiating active measurement tests, querying passive measurement devices or capturing packets [3]. A common interface to these capabilities is required for ease of integration into the monitoring system as a whole. Measurement data of interest includes at a minimum: active delay, loss, jitter, processing information retrieved from network equipment, flow-based measurements, active stress-type achievable bandwidth measurements, active probe-type available bandwidth measurements and passive packet capturing (*e.g.*, NetFlow). MPs use measurement setup protocol to allow the user to request measurements to be made for a specified set of parameters and then publish the results of these measurements to one or more subscriber interfaces. Legacy capabilities (*e.g.*, existing active measurement tools, Netflow and SNMP) can be "wrapped" within an MP.

In a server interactions scenario, the MP accepts measurement requests and uses a push method of data publishing. In such an approach, the client has to provide, in advance, one or more subscriber handles to send the results directly to it. It is also possible to send data indirectly *via* a Transformation service. In a client interactions scenario, the MP registers its own presence with an LS and publishes measurement data to subscribers. The MP may send resource availability and authorisation requests to the RP.

3.2. Lookup service

Services register their existence and capabilities, subject to locally-determined policies and limits, with a Lookup service (LS). Services register using the join protocol. They may register for a limited period of time or leave without disrupting the interaction of other services. Clients discover needed services by querying an LS, using the lookup protocol. The first LS is found by one of several approaches, including multicast, well-known servers or internal configuration. Once an LS is found, additional LSs are identified by querying the first one. LSs register themselves with other LSs and are organised using peer-to-peer distribution techniques.

The lookup protocol of the service network defines the kinds of queries a client can make when looking for a resource. The LS is not a simple name-based directory service. Queries about the services are based upon attributes such as service type, required authentication attributes and service capabilities, as well as more complex constructs, such as network location or community affiliations.

When the service acts as a server the LS accepts requests for service related information, registration and de-registration requests (including advertisements from other LSs announcing their existence), and keep-alive requests. In a client interactions scenario, the LS registers its own presence to other LSs. The service can also work in peer-to-peer networks where LSs share directory information with other LSs. The still-to-be-defined peer-to-peer distribution algorithm will define which individual peer LS instances need to have cache references.

3.3. Measurement Archive service

The Measurement Archive service (MA) stores measurement data in database(s) optimised for the corresponding data type and publishes measurement data produced by MPs and/or TSs. In addition to providing a historical record for analysis, the MA serves to reduce queries to the MP by effectively offloading the publication to multiple clients. The MA makes use of a set of protocols: storage setup protocol which is used to setup the MA to accept and store measurement data from a publisher (*e.g.*, an MP) and measurement data retrieval protocol to get measurement data from the MA using the client.

In case the MA is perceived as a server, it accepts and stores setup requests as well as publication requests.

The publication request includes a subscription handle and the results are sent directly to the client (or indirectly *via* a TS). As a client, the MA registers its own presence with an LS, subscribes to an MP, other MA, or TS and publishes measurement data to subscribers. The MA may send resource availability and authorisation requests to the RP.

3.4. Authentication service

The Authentication service (AS) provides the authentication functionality for the framework as well as an attribute authority. The AS supports clients with multiple identities, including individual identities that represent different roles at different times. Role-based authentication using attribute assertion-style authorisation protects the privacy of the user [4]. This typically means that a handle is created to provide additional information about the attributes of that user, and that resources can use that handle to make queries about the user subject to privacy policy. Communities of multiple administrative domains that accept each others' authentication can be formed by federating ASs. Federation details are held solely in the AS and hidden from other services within a given administrative domain. In other words, the "trust" relationship within a domain is between the domain's services and the local AS domain, while the "trust" relationship between any two federated domains is managed by the ASs.

In a server interactions scenario, the AS accepts authentication requests and attribute requests via its interfaces. In client interactions scenario it registers its own presence with an LS and may query other ASs for attributes of a federated identity.

3.5. Transformation service

The Transformation service (TS) performs a function (*e.g.*, aggregation, correlation, filtering or translation) upon measurement data. The TS subscribes to one or more servers and publishes to one or more clients, making it a key component of a data pipeline within a measurement framework. For example, a TS might compress datasets from more recent, high-resolution data to less recent, low-resolution data and publish that data to an MA service. A TS also might read from multiple data publishers to create a specific correlation. A very simplistic data analysis example would be a threshold detection operation that then pushed data out for the purposes of triggering a Network Operations Centre (NOC) alarm.

By considering TS as a server, it accepts publication requests. If the request includes a subscription handle, the results are sent directly. If no subscription handle is included, the TS returns a publisher handle to the client, which is then responsible for initiating dataflow. When TS acts as a client, it registers its own presence with an LS, subscribes to one or more MPs, MAs, or TSs and publishes measurement data to subscribers. The TS may send resource availability and authorisation requests to the RP.

3.6. Topology service

The Topology service (ToS) is a specific example of a TS used to make topological information about the network available to the framework. It collects topological information from a variety of sources (*i.e.*, multiple MPs) and uses algorithms to deduce the network topology. The ToS also reflects multiple network layers, from the domain level through wavelengths at the physical level. Understanding the network topology is necessary for the measurement system to optimise its operation. For example, the LS relies on the ToS to determine MPs that are "closest to" interesting network landmarks (*e.g.*, routers). Thus, in the same way that a host may query for an MP instance that has a particular set of properties, a service component can also ask about node proximity. Additionally, the Topology service may be used for overviews/maps that illustrate the network with relevant measurement data.

3.7. Resource Protector service

The Resource Protector service (RP) is used to arbitrate the consumption of limited resources such as network bandwidth. It also has a scheduling component to deal with the consumption of time-dependent resources. When measurement activities are involved, resources may be related to the measurement infrastructure or real network resources. The RP can allocate portions of a resource based upon configuration rules and can schedule the time-dependent resources. Services that consume resources contact the associated RPs to allocate them. Because RPs reduce scheduling flexibility, RPs should only be deployed to protect limited resources. In other words, some MPs do not have to contact an RP at all.

Authenticated requests provide a way of making attribute assertion queries back to the authenticating entity. A handle is included within the Authentication Token that is sent with the request. This makes it possible for the RP to determine what rights a particular resource requestor has to the given resource without fully divulging the identity of the requestor.

To allow for flexibility in deployment, MPs may request resources from a list of RPs (potentially different ones, depending upon the particular resources needed to perform the given test). Each RP can be configured with a list of higher-level RPs. This allows for hierarchical organisation of RPs, which can be useful for certain types of resource protection. For example, each MP on a host might be configured with its own RP. Each of these RPs might be configured to contact a single RP that is used to protect the resources of the host itself.

If the RP service acts as a server, it accepts authorisation and resource availability requests. If it acts as a client, the RP registers its presence with an LS. The RP service may request authorisation and resource availability for other resources from other RPs. The RP may request additional attribute information about an authenticated identity from an AS.

4. SINGLE DOMAIN USE-CASE

Because the framework is designed as a set of deployable services, it is possible to envision many different deployment scenarios. Most existing measurement frameworks work only within a single administrative domain or between very well coordinated domains [5]. One of the goals of this framework is to allow for measurements between loosely coordinated domains. Because each administrative domain is represented by a set of independent resources, the concept of a domain is really a function of the deployment of those independent services. The following sub-sections describe how a specific deployment could support measurements within a single domain.

4.1. Service registration

All services register themselves with an LS to participate in the framework. There is a well-defined interface on the LS that accepts registrations or updates from all the other services available in the infrastructure.

If the service does not receive an acknowledgment message from the LS, it assumes the LS is unavailable, in which case the service must register with another available LS (even one from another domain – Fig. 2).

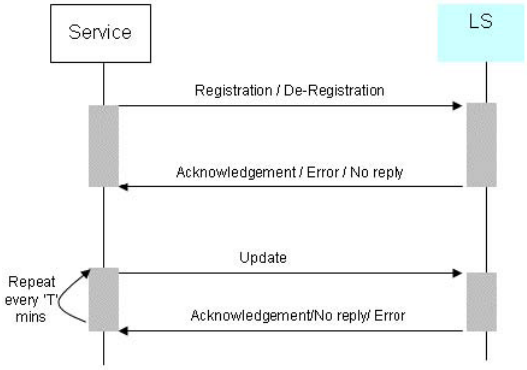

Fig. 2. Service registration message sequence

4.2. Authentication and authorisation

The first thing a client must do to use resources from the framework is to authenticate. Figure 3 shows the control flow for a measurement request where the client has credentials for the same authentication domain as the MP. The example below shows a request of service from an MP,

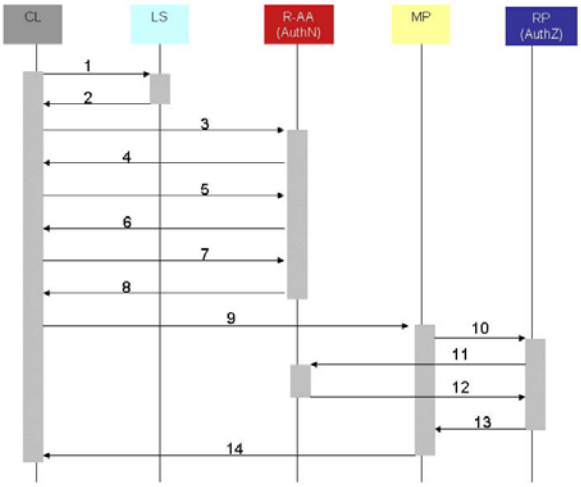

Fig. 3. AA message sequence

but may be applied more generally to any other kind of service that requires authentication and authorisation.

Sequence of actions:

1. Client (CL) queries LS for MPs that match a given criteria.
2. LS returns a list of candidate MPs including an indication of the authentication domains that manage authentication for each one (each MP can be managed by more than one domain). LS also returns the address of an AS that can authenticate for each of the returned authentication domains.
3. CL contacts the Authentication Service that manages authentication for the resource domain (R-AA-Service) and requests an authentication token blessed for use in the resource domain (R-AuthRealm).
4. R-AA-Service returns a list of known (federated) authentication domains and asks CL to choose one for authenticating.
5. CL specifies the domain it has credentials for @R--AuthRealm.
6. R-AA-Service manages identities for R-AuthRealm, so R-AA-Service asks CL for identity credentials.
7. CL presents credentials.
8. If credentials are valid, R-AA-Service creates a handle that can be used to request additional attributes about the identity, subject to attribute release policies in R-AuthRealm (this is done to protect the identity of the requestor). This handle is returned to CL encoded as an AuthToken blessed by R-AuthRealm (R-AuthToken).
9. CL requests a measurement from MP. Request includes the R-AuthToken.
10. MP requests resources from Resource Protector (RP). The R-AuthToken is passed along in the request.
11. RP needs more information about the identity requesting the resources and makes an attribute query to R-AA-Service using the R-AuthToken handle.
12. R-AA-Service releases only as much information about CL identity as is allowed.
13. RP returns resource availability (allowed/disallowed).

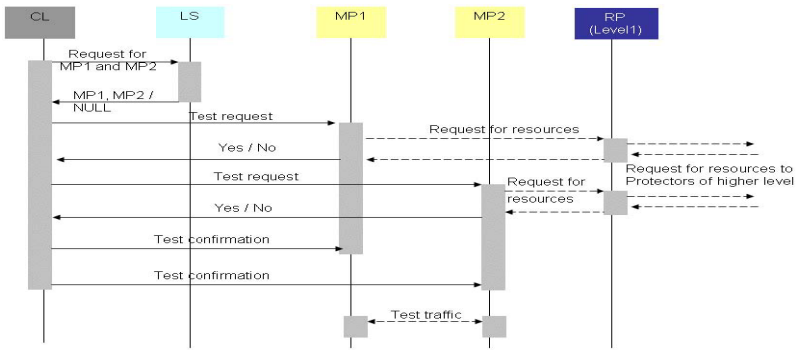

Fig. 4. On-demand test message sequence (test initiation)

This portion includes scheduling of time dependant resources.

14. MP returns response to measurement request.

In the remainder of the single domain examples the authentication details are skipped although the first interaction to be done is to acquire an authentication token.

4.3. On-demand testing

An important goal of the measurement system is the ability to run on-demand tests to divide and conquer an arbitrary network graph. The complexity of supporting on-demand tests requires many interactions between different kinds of services. Figure 4 depicts the on-demand test message sequence.

During a test setup, each MP determines a map of available times for the request and returns that along with a tentative reservation of the first time slot that fits the request. The tentative reservation is held for a time needed to get responses from all MPs that were selected to attend a test.

After the client contacts each MP, it adjusts its request for the next MP to fit within the parameters of the available times it has seen from each MP until it either has a tentative agreement from all MPs for the same time slot, or it determines no common timeslot is available (test cancellation). If the client collects positive responses from all MPs for a defined time period, it sends confirmations for an accepted test.

4.4. Requesting data

Test results are published by data publishers to data subscribers. For archived data, a client asks an LS for the location of measurement data. Then it sends a data query to the appropriate Measurement Archive service which sends data back to the client.

5. MULTIDOMAIN USE-CASE

Most interesting end-to-end performance issues span multiple administrative domains [7]. The framework works equally well for a set of services run by autonomous administrative entities, as it does for services within a single administrative domain. Services in this framework interact directly with the other services they require. Data subscribers and data publishers have no limits placed upon them based upon their relative location to each other. Therefore, the ensuing complications are mainly related to authentication and authorisation, and coordination of distributed measurements.

5.1. Service registration

Multiple administrative domains and multiple coordinated LSs are supported using peer-to-peer techniques [8]. The specific distribution algorithm still needs to be explored. Consideration will be given to algorithms that provide for locality of reference based upon usage patterns.

5.2. Authentication, authorisation and security

Exposing the network measurement infrastructure of a domain to external users opens the domain to potential threats. The installation and configuration of measurement tools is restricted to the domain's administrators or entities closely related to it. Other users wanting to use these tools have to access them through an MP. Authentication and authorisation is necessary to limit the amount of active measurement traffic injected in the network, to prevent such tools from being used for denial-of-service (DoS) attacks, to guarantee a "fair" access to the measurement infrastructure to all authenticated and authorised users and to prioritise the measurements according to locally-defined policies. Authentication and authorisation is also relevant for the other services not directly related to the establishment of a measurement. For example, MAs that provide topological data may only allow local entities to access that data. Likewise, an MP that produces raw Netflow data might only allow subscriptions through an anonymising Transformation service.

5.3. Service requests from a related (federated) domain

Figure 5 shows the control flow for a measurement request where the client has credentials for a different

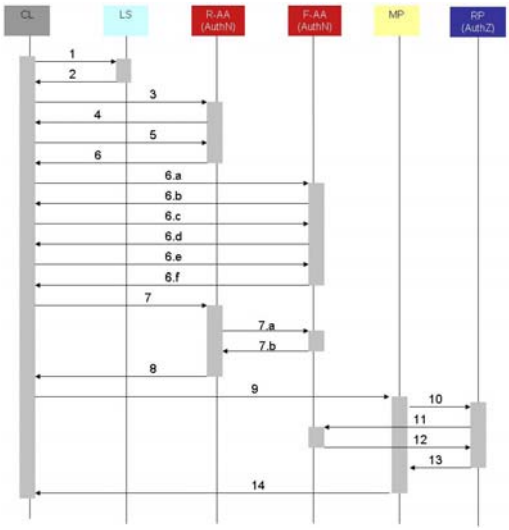

Fig. 5. Measurement from a related domain message sequence

authentication domain from the MP. This request could be to any type of service but is illustrated in this example using an MP. In this case, the two domains (F and R) have created a trust relationship by federating.

Sequence of actions while requesting measurements from a federated domain:

1. The client (CL) queries LS for MPs that match a given criteria.
2. LS returns a list of candidate MPs including an indication of the authentication domains that manage authentication for each one. LS also returns the address of an AS that can authenticate for each of the returned authentication domains.
3. CL contacts AS that manages authentication for the resource domain (R-AA-Service) and requests an authentication token blessed for use in the resource domain (R-AuthRealm).
4. R-AA-Service returns a list of known (federated) authentication domains and asks CL to choose one for authenticating.
5. CL specifies the domain it has credentials for: @F-AuthRealm.
6. R-AA-Service does not manage F-AuthRealm so it redirects CL to F-AA-System.
   a. CL contacts AS that manages authentication for the client-selected domain (F-AA-Service) and requests an F-AuthToken authentication token for use in R-AuthRealm (as in step 3.).
   b. F-AA-Service returns a list of known (federated) authentication domains and asks CL to choose one for authenticating (as in step 4.).
   c. CL specifies the domain it has credentials for: @F-AuthRealm.
   d. F-AA-Service manages identities for F-AuthRealm, so F-AA-Service asks CL for identity credentials.
   e. CL presents credentials (as in step 7).
   f. If credentials are valid, F-AA-Service creates a handle that can be used to request additional attributes about the identity subject to the attribute release policies in F-AuthRealm. This handle is returned to CL encoded as an AuthToken blessed by F-AuthRealm (F-AuthToken) (as in step 8).
7. CL presents credentials to R-AA-Service. In the federated case, the credentials can be the authentication token from a federated authentication domain, in this case F-AuthToken.
   a. R-AA-Service needs more information about the federated identity requesting access to R-AuthRealm protected resources and makes an attribute query to F-AA-Service using the F-AuthToken handle.
   b. F-AA-Service releases only as much information about CL identity as is allowed.
8. If credentials are valid, R-AA-Service blesses the F-AuthToken for use with R-AuthRealm resources. This effectively creates an R-AuthToken but keeps the handle pointing back to the F-AA-Service for attribute queries.
9. CL requests a measurement from MP. Request includes the R-AuthToken.
10. MP requests resources from the RP. The R-AuthToken is passed along in the request.
11. RP needs more information about the identity requesting the resources and makes an attribute query. The query goes to the F-AA-Service.
12. F-AA-Service releases only as much information about CL identity as is allowed.
13. RP returns resource availability (allowed/disallowed.) This portion includes scheduling.
14. MP returns response to measurement request.

It is important to note that the AuthToken's described above have a limited lifetime, but that they are valid for use with all the services within the given authentication domain. Therefore, the authentication interactions shown in the above diagrams are done infrequently.

Some important aspects about the model shown that are not present in many authentication systems is the assumption that users may have more than one identity. This is needed to support measurements that require more than one MP, where the MPs involved are not within the same authentication domain and they have no common federated relationships. Inter-domain trust relationships are managed by the ASs of the domain. In a framework where there are potentially many deployed services, it is important not to force full distribution of the trust relationships.

5.4. Service requests between unrelated domains

To successfully diagnose real end-to-end network performance problems, it is often necessary to perform measurements across domains that may not have any direct business relationships with each other. Therefore, it is desirable to give users the ability to run tests between MPs on completely unrelated domains. To support this, users need credentials for both authentication domains represented. This does not fundamentally change the authentication and authorisation model represented above. The only requirement is that a client application must be able to manage multiple identities and it may need to authenticate more than once to produce a single measurement result.

6. PROTOTYPICAL IMPLEMENTATION AND PROJECT SCHEDULE

This work is based upon lessons learned from many European and international initiatives and deployed measurement frameworks, including DANTE's perfmonit project [1] and Internet2's piPEs project [9]. The authentication and authorisation piece is influenced by the efforts of GN2 Joint Research Activity 5 -Roaming and Authorization [1] and Internet2's Shibboleth project [10]. The work is also carried out with respect to efforts of the Global Grid Forum (GGF) Network Measurement Working Group (NMWG) [11] to develop schemas for interoperable measurement frameworks.

A prototype is planned for this summer, transforming the generic model into interactive software components

that communicate with each other, having the primary goal to retrieve link utilisation from several networks. We focus our prototype efforts to check that framework design is viable and we will ensure that the communication between selected individual components works. It is also important during this period to investigate and implement technology for exchanging information between a subset of services and the client.

For this, we are building simplified versions of the services to reduce the complexity of the architecture at the first stage. The number of services and their complexity will increase over time while adding additional modules, features and measurement types. The first service is the Lookup service with the functionality necessary to locate other services (*i.e.*, Measurement Archives and Measurement Points). The crucial portion of the prototype system is the MA service, which, in the first place, is a wrapper around Round Robin Databases (RRD) [12] and provides link utilisation statistics through a Web Service [13] interface. The work in progress is to implement the MA to retrieve link utilisation SNMP-based data from existing RRD files upon user request. In the first phase, we implement the script-based approach that will perform the necessary request for data to the Service Layer and will get back data for the user to demonstrate a proof-of-concept. We use a minimum functionality AA, which will always answer positively upon the request to get measurement data. Initially, several MA services will be deployed in several domains, making use of different RRD collections and providing a picture of utilisation of a few research networks, both from Europe (*e.g.*, GÉANT) and from North America (Internet2 and ESnet [14]).

Two other phases are targeted in the prototype. The first extension will be to add auto-registration capabilities to the LS, so that any service coming into life could register its capabilities and will automatically be known by the LS, which could provide this information back to all other services when necessary. We also plan to add new measurement capabilities like packet loss and interface errors to the MA. We have considered replacing user scripts with intuitive graphical interface for test setup, data retrieval and presentation.

The work plan calls for the open source development of this architecture, beginning with the first prototype by summer 2005. Development of the entire architecture and early stage deployment is expected to unfold over the next few years as an iterative series of increasingly refined prototypes.

We hope that other national and international networks, such as APAN, ESnet, and CLARA will become increasingly engaged in the effort as it proceeds.

7. RELATED WORK

Many projects and papers address the problem of network monitoring. The IST project INTERMON [15] is focused on inter-domain QoS monitoring as well as on other aspects. They model abstractions based on traffic, topology and QoS parameter patterns and run simulations for planning network configurations. To fulfil these goals, the project has based its entire design on a huge centralised and complex database for topology, flow and test information, collecting all network data in a single location. However, such model is not acceptable in a multi-domain networking environment. It is not conceivable that an entity supersedes the others and has complete control of other networks. Also, while they centralise the collection of predefined measurements, we provide an architecture where any entity could, based an authentication and authorisation rules, schedule new types of measurements and run tests over the multi-domain network. Our project has a more focused goal and has real production constraints from NOCs, requesting data for day-to-day operation of the networks. There are also many constraints for allowing distributed policies among the different networks, for exchange of monitoring data exchange and access to on-demand test tools.

We paid attention to the MonALISA project [16] that has produced a framework for distributed monitoring. It consists of distributed servers which handle the metric monitoring for each configured host at their site and for WAN links to other MonALISA sites. The MonALISA framework provides a distributed monitoring service system using JINI/JAVA and WSDL/SOAP technologies. Our idea shares the method of servers acting as a dynamic service system and providing the functionality to be discovered and used by any other services or clients that require such information. Even though it has similar concepts to our approach, we detailed its application to multi-domain environments with mechanisms for measurements spanning independently managed domains, especially with respect to metrics concatenation and aggregation. The system relies on Java-only JINI [17] and remote method invocation (RMI) technology for the discovery service, a solution that cannot be regarded as open and universal enough. We propose, in our approach, to use Web Services allowing access by components implemented in any technology that is found to be useful.

Besides those, the PlanetLab [18] initiative is also related to our work. It is a huge distributed platform over 568 nodes, located in 271 different sites at the time of this writing. It enables people to be members of the consortium to access the platform, or part of it, to run networking experiments. Most of the projects which run over the PlanetLab infrastructure deal with network monitoring and management in general. Those tests aim at properly designing services at a large scale. The architecture is similar to the one we develop – resources are made available through designed architecture services. PlanetLab proposed a node manager (*e.g.*, access interface for each node) which just allocates local resources based on policies

enforced on the infrastructure service. Although, we identified analogies in the components defined in the two projects, unlike the method described in this paper, the PlanetLab infrastructure service is centralised and relies on a single database and therefore could not be applied as is to the multi-domain environment.

8. CONCLUSIONS

For the efficient use of networks involving different administrative domains, network performance information has to be provided to the end users. We described a measurement framework for characterising the behaviour and usage of the network. Our approach for the implementation of the system is a scalable, distributed service-oriented architecture. While there are many related efforts to the approach proposed in this paper, we believe the way we are incorporating the latest federated authentication and authorisation techniques while retaining the ability to perform measurements between unrelated domains is very powerful and will aid in the deployability of the system.

The design of this framework combines information from different kinds of measurement tools that currently exist and is able to easily accommodate new ones. We hope that, in the long term, the results of our work will encourage cooperation among European, American and other international measurement infrastructures to enable new insights into precise understanding of global Internet behaviour.

Acknowledgement

The perfSONAR work is the outcome of the merging of two efforts from the Internet2 PAT and from the GN2-JRA1. The GN2-JRA1 work inherited from the TF-NGN "perfmonit" activity technical background, which was setup as a support for the TF-NGN Performance and Enhancement Response Team.

References

[1] DANTE homepage including information about GÉANT, perfmonit and GN2 projects. [Online]. Available: http://www.dante.net.
[2] Internet2 homepage. [Online]. Available: http://www.internet2.edu/.
[3] T. Chen and L. Hu, Internet performance monitoring, Proceedings of the IEEE, 90, 1592-1603, Sep. 2002.
[4] Specification of the general architecture, protocols, and message formats of the Shibboleth mechanism. [Online]. Available: http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-arch-protocols-06.pdf
[5] M. Murray and K. Claffy, *Measuring the immeasurable: global Internet measurement infrastructure*, presented at PAM2001.
[6] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, C. Diot, *Packet-level traffic measurements from the Sprint IP backbone*, IEEE Network Magazine, Nov. 2003.
[7] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, *Deriving traffic demands for operational IP networks: methodology and experience*, IEEE/ACM Transactions on Networking, 9, 265-280, Jun 2001.
[8] N. Minar and M. Hedlund, *Peer-to-Peer. Harnessing the power of disruptive technologies*, O'Reilly, Mar 2001, ch. 1.
[9] E2Epi performance evaluation system (piPEs*)* [Online]. Available: http://e2epi.internet2.edu/.
[10] Shibboleth project homepage. [Online]. Available: http://shibboleth.internet2.edu/.
[11] The Network Measurements Working Group (NMWG) of the Global Grid Forum [Online]. Available: http://www-didc.lbl.gov/NMWG/
[12] Round Robin Database. [Online]. Available: http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/.
[13] Web Services activity. [Online]. Available: http://www.w3.org/2002/ws/.
[14] The Energy Sciences Network, Esnet. [Online]. Available: http://www.es.net/
[15] *INTERMON project*. [Online]. Available: http://www.intermon.org
[16] MonALISA project, Caltech. [Online]. Available: http://monalisa.cacr.caltech.edu
[17] JINI network technology. [Online]. Available: http://www.sun.com/software/jini/.
[18] PlanetLab project. [Online]. Available: http://www.planet-lab.org.

JEFF W. BOOTE received his B.Sc. in computer science and engineering from the University of Colorado, Boulder (United States of America) in 1991. From 1991 to 2002 his primary contributions were in the areas of scientific visualisation and graphics, and managing the web engineering group at the National Center for Atmospheric Research (Boulder). Since 2002, he has been a network software engineer for Internet2, with a primary focus on performance measurements. He is the primary architect and developer of the Internet2 piPEs framework and, specifically, the OWAMP and BWCTL measurement tools.

ERIC L. BOYD received his PhD. in computer science and engineering from the University of Michigan, Ann Arbor (United States of America) in 1995. He has worked on performance tool technologies for Hewlett-Packard, Compaq Computer Corporation, and SolidSpeed Networks. He now manages Internet2's Performance Architecture and Technology team, providing leadership for the collaboration between the JRA1 project and Internet2 and evangelising the deployment of Internet2's BWCTL, NDT, OWAMP, and piPEs tools.

JEROME DURAND received a master's degree in networking and telecommunications at INSA Lyon in 2002. He has then joined the IP advanced services team of RENATER. He is one of the initiators and coordinator of the M6Bone project: a worldwide overlay IPv6 multicast network. He is contributing for 6Net IST project, in particular in the multicast and monitoring areas. He is also contributing to the IETF for IPv6 multicast related documents. He is now working on the GN2 project, in the interdomain management and performance measurement research activity.

ANDREAS HANEMANN received a diploma degree (M.Sc.) in computer science from the University of Karlsruhe (TH), Germany. He is involved in the JRA1 performance visualisation starting from the project launch in 2004. Since October 2002 he is responsible for the Customer Network Management project at the Leibniz Supercomputing Center, which provides a network performance visualisation tool for the German Research Network.

LOUKIK KUDARIMOTI received his M.Sc. in Distributed Systems and Networks from the University of Kent (United Kingdom) in 2003. He has been working with DANTE as a network engineer since May 2003 and has been involved with designing and developing distributed monitoring systems for projects such as DANTE's Perfmonit, IST EGEE, and GN2 JRA1.

ROMAN ŁAPACZ obtained his M.Sc. in computer science (Laboratory of Intelligent Decision Support Systems) from Poznan University of Technology (Poland) in 2000. Since 2000, he has been a network software engineer in Networking Department in Poznan Supercomputing and Networking Center. Now he is mainly involved in the work within JRA1 project and cooperation with Internet2 to create general multi-domain measurement system.

NICOLAS SIMAR has been working for Dante since 2000 as a network engineer. He has been working especially on QoS and monitoring. He was involved in the IST SEQUIN project and is currently involved in the IST EGEE project. He now coordinates the GN2 JRA1 activity on multi-domain monitoring.

SZYMON TROCHA received his M.Sc. degree in computer science from Poznan University of Technology (Poland) in 1998. He is a Head of Management Unit in PSNC. He is mainly involved in the network management applications planning and implemention. He is also responsible for traffic analysis and measurement technology research and implementation. Since 2004, he has been leading the JRA1 activity in PSNC.