Authorization of a QoS Path based on Generic AAA

Leon Gommans, Cees de Laat, Bas van Oudenaarde, Arie Taal

Advanced Internet Research Group, Department of Computer Science, University of Amsterdam.

Abstract

For data intensive Grid applications, such as shown at iGrid2002, users may require short-lived guaranteed high bandwidth connections. These types of connections, providing a certain Quality of Service (QoS) will need to be authorized and provisioned, often through multiple administrative domains. We present a case study of a Bandwidth on Demand service that provides a QoS path based on Generic Authorization, Authentication, Accounting, that represents a first step forward towards a multi domain solution.

Keywords: QoS, Bandwidth on Demand; multi domain; Authentication Authorization Accounting

1. Introduction

In this paper we discuss a middleware solution for the authorization of a Quality of Service (QoS) path with a focus on high bandwidth. QoS represents value and, when used in an on demand fashion, needs policy control to allow cost effective usage. QoS path creation was first introduced with the Resource ReSerVation Protocol (RSVP) [1]. During the QoS path creation routers exchange RSVP requests and have the possibility to pull a policy decision from a policy server using Common Open Policy Service (COPS) [2]. In our approach a network of AAA servers communicate Bandwidth on Demand (BoD) requests and use the agent sequence as defined in the Generic AAA framework [3].
An advantage of the agent sequence is that the underlying equipment does not have to carry as much intelligence as equipment using the sequence. Network equipment such as layer 3 routers, layer 2 switches or layer 1 cross-connects are capable of providing a QoS path with different levels of granularity. Because of the agent sequence our focus is on a switched-like architecture as the more appropriate solution to authorize a QoS path.

2. Generic AAA architecture

An AAA server may be involved in handling one or more of three basic functions. The first function is Authorization of resource usage. Secondly, it verifies identities (Authentication). Finally, the AAA server may log key attributes of its functions so they can later be used for Accounting purposes.
The architecture of a Generic AAA server is explained in [4].
We consider the authorization of a QoS path through multiple administrative domains. As each administrative domain implements the authorization of its resources to an AAA server,
more than one AAA server needs to communicate by means of AAA requests in order to authorize a QoS path.
A number of definitions exists for terms with respect to policies, see for example [5] and [6]. In this paper we will introduce the term Driving Policy. In our applied model, an AAA server fetches a Driving Policy when it receives an AAA request. For each type of AAA request there exists a corresponding Driving Policy that instructs the AAA server how to deal with the request, e.g. a Driving Policy has to check the pre-conditions of the actions to be performed and how to deal with the post-conditions of these actions. We distinguish between specific actions and generic actions. Specific actions are performed by a so-called Application Specific Module (ASM), whereas generic actions are delegated to the generic part of an AAA server. The provisioning of a path within a single domain is typically performed by an ASM. Providing the date is an example of a generic action. The module that is responsible to execute a Driving Policy is the Rule Based Engine (RBE) and is contained in the Generic part of an AAA server.
Fig. 1 shows the different components of an AAA server. The behavior of the generic part of an AAA server is determined by the combination of Driving Policies, ASM's and AAA requests. This implies that behavior can be easily adapted.

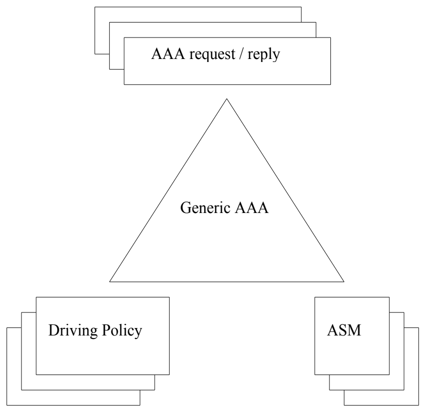


***Fig. 1** The different components of an AAA server.*

3. Authorization / Control models

In the context of Generic AAA we consider a QoS path to be build out of two elementary QoS elements: network nodes (vertices) and network links (edges), where the relevant parameters of a network node or network link are under the control of an AAA server. This means that these parameters are governed by a set of policies residing in the Policy Repositories of the AAA servers in control.
In general a QoS path may span multiple administrative domains. The following terminology is introduced to facilitate the model discussion. We introduce the term QoS segment as the part of a QoS path that is under control of a single administrative domain. Each QoS element is authorized by an AAA server. Those elements of a QoS path belonging to a single AAA server are called a QoS component.
The AAA server controlling QoS elements should also maintain the state of the assigned part of the QoS path condition.
In this way, the whole QoS path can be seen as a guaranteed end-to-end connection.

Taking the simplest unidirectional QoS path between two nodes, three different controls / authorization models can be distinguished: Individual Control, Partial control, and Full control.

3.1 Individual Control model
Each element, network node or network link, is managed individually by an AAA server. The AAA servers execute their own and independent set of policies. This model offers "individual control" of the simplest element. According to the used terminology, each QoS element is a QoS component.

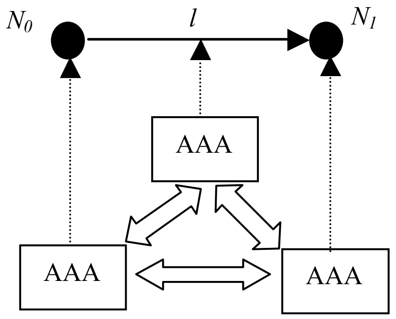


***Fig. 2** Individual Control model*

3.2 *Partial Control model.*
A single set of policies controls the usage of both a network node and its outgoing network link (fig. 3) or the set of policies controls a network node and its incoming network link (fig. 4). This requires specific policies to describe the negotiations with the neighboring element.
Both figures exist of two components.

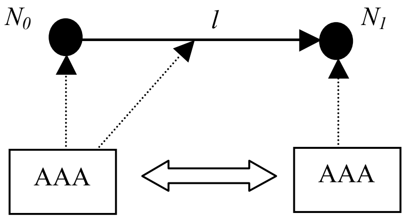


***Fig. 3** Partial Control model with an outbound connection.*

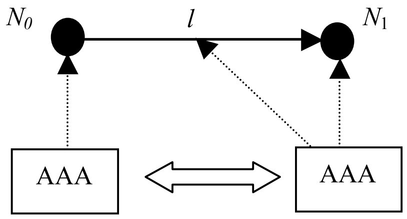


***Fig. 4** Partial Control model with an inbound connection.*

3.3 Full Control model.
A single AAA server controls all three network elements of the simple QoS path, i.e. the QoS component is a segment.

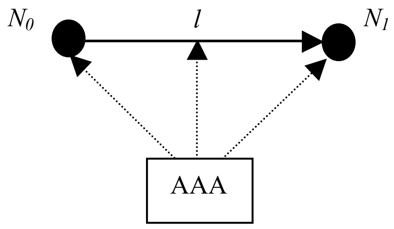
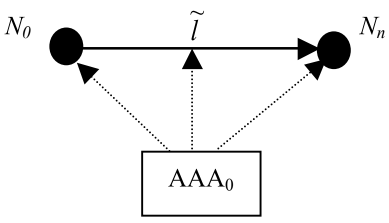


Fig. 5 *The Full Control model.*

4. Authorized path discovery.

To show how an authorized QoS path through multiple domains can be established, we make the following assumption. AAA servers, especially proxy AAA servers have an underlying mechanism that advertises the connections they can establish. Although not fully researched yet by our group, we expect to be able to re-use mechanisms such as BGP [7] (or proposed extensions to this protocol that is referred to as Optical BGP) to advertise reachability of networks within administrative domains. A purposely build *BGP* ASM will enable an AAA server to obtain a view on topology and discover which AAA servers should be contacted along the QoS path.
Each AAA server will act as an agent or broker for its sub-domains.
As the start for the setup of an authorized QoS path between two nodes, N_0 □ D_0 and N_n □ D_n , one of the control models shown in fig. 2-5 can be selected. If D_0 and D_n are different administrative domains, the start situation will be the simplest QoS path according to the Individual Control model or according to the Partial Control model. If D_0 and D_n turn out to be the same administrative domain, a simplest QoS path according to the Full Control model may also be an appropriate choice. In general multiple domains are in play, i.e. the network link of the start situation will be a logical network link instead of a physical network link. Authorization of a QoS path between N_0 □ D_0 and N_n □ D_n is established through the following number of steps. Without loss of generality we take the Full Control model as the starting situation, with $D_0 = D_n$ and logical network link $\tilde{l}$, see figure 6. N_0 *and* N_n are authorized by AAA_0, but for the logical network link $\tilde{l}$ a physical solution must be found.



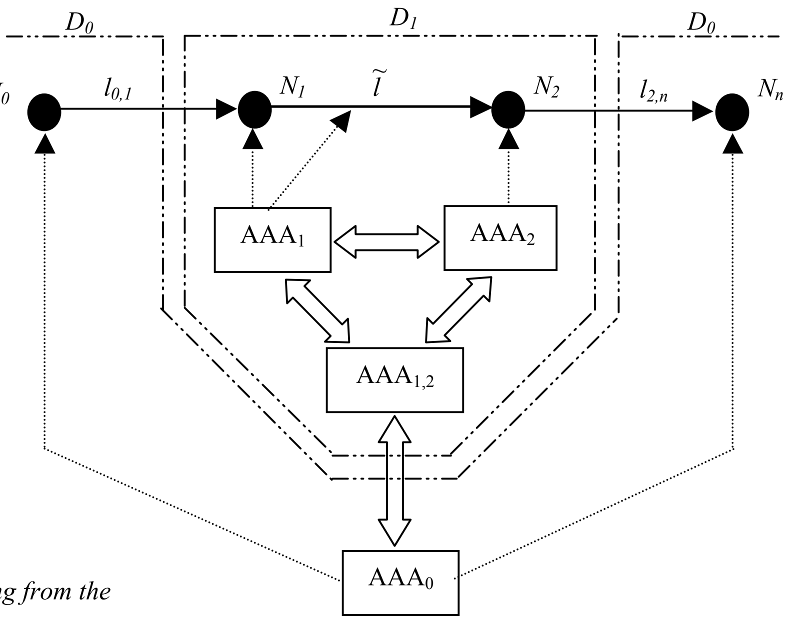


Fig. 6 *The setup of a QoS path starting from the Full Control model.*

Therefore, AAA_0 will contact the AAA server that advertises a connection between N_0, N_n.
In fig. 6 it is $AAA_{1,2}$ that advertised the required connection. Server $AAA_{1,2}$ acts as a proxy server for two other AAA servers: AAA_1 and AAA_2. AAA_0 and $AAA_{1,2}$ exchange requests to authorize the network links $l_{0,1}$ and $l_{2,n}$. Fig. 6 shows that for the authorization of $l_{0,1}$ $AAA_{1,2}$ resorts to AAA_1 and for the authorization of $l_{2,n}$ $AAA_{1,2}$ resorts to AAA_2. This whole process is started by a Driving Policy residing at AAA_0.

According to the situation depicted in fig. 6 this process of authorization will continue via AAA_1, as this AAA server still controls a logical network link.
As soon as all logical network links of the QoS path are authorized, this process of authorization has come to an end.

The provisioning of the complete QoS path might be established by different approaches. One approach is to wait for provisioning until the whole QoS path is authorized. It is also possible to choose for an approach of successively provisioning. More research is necessary to describe the role of policies in both of these approaches.

5. AAA server authorization interactions

For any QoS path a particular AAA server will be the root of a decision network (tree) needed to authorize usage of the QoS elements involved. All AAA server interactions are policy driven. Policies may ask for a set of credentials or tokens to be present in the request that represents somebody else's authorization or delegation of authority. For example these credentials or tokens may have been issued by an advance reservation system as a proof of a reservation. Policies may involve parties in the decision that may or may not be hidden from the requesting party. For instance a budget- or payment authorization may be requested. Such checks may additionally involve parties unknown to the requestor.

Next to the content of an AAA request, that may need to carry a token or credential that represents an authorization, also the integrity, peer authenticity and sometimes confidentiality of a request must be ensured. Here we rely on existing security mechanisms such as provided thru the GSS-API [RFC 2743] that offers support for these aspects. More specifically, toolkits such as Globus 's Grid Security Infrastructure does also rely on the GSS-API. Some level of interoperability is offered when the GSS-API is deployed with Public Key certificates issued by a GRID Certification Authority. The VOMS project [8], within the EU DataGRID and DataTAG project researches the definition of roles within Grid environments. Thru collaboration with the VOMS project we plan to integrate the concept of roles. A user role may define that a particular user is authorized to allocate bandwidth.

6. Case study

For the case study presented we will focus on scientific Grid users, who have a need for an on-demand high bandwidth QoS path. These users fall into a specific category of users that sets them far apart from regular Internet users. Classifying users for our purpose is a 3 dimensional problem: The number of users can both be considered against the amount of bandwidth needed and the amount of destinations in need to be reached. In our case we are targeting a possible solution for users that

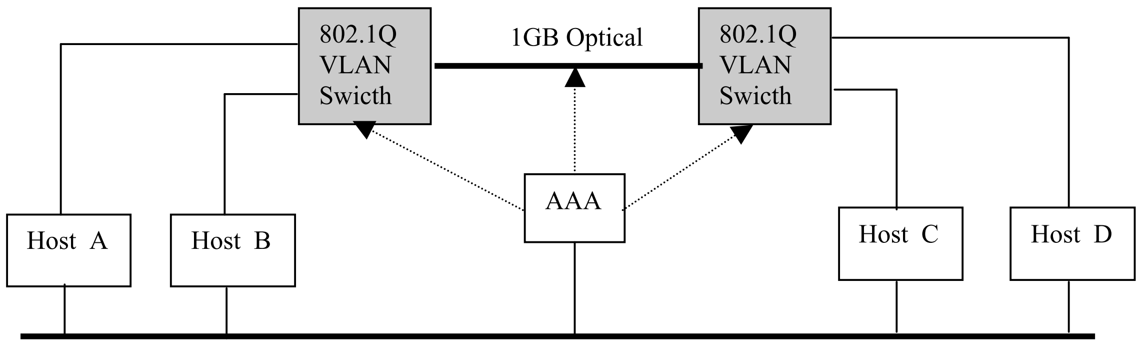


Fig. *7 iGrid 2002 demo setup*

are both in need of very high amounts of bandwidth and are very limited towards the need to reach destinations. When considering the bandwidth usage only, these users may fall into the category C as mentioned in the article "The rationale of Optical Networking" [9]. Also considering the fact that applications in this space may use application specific (adapted) protocols to reach optimal performance, this demonstration shows the authorization of a QoS path using layer-2 switches. With data transfer programs such as GridFTP in mind, the shown configuration allows the creation of a by-pass channel that can be used as the data-channel in parallel with the regular Internet connection which in this case can be used as the control channel.

The actual technology choice for this model was motivated by availability of layer-2 switch technology donated by Enterasys Networks. The concept used however may be applied to any layer 1, 2 or 3 technology that is capable of creating QoS path.abstractions.

The usage of the agent model effectively separates the control plane and data forwarding plane. As most of the intelligence exists in the control plane, deployement of less intelligent layer 1 or 2 equipment in the data forwarding plane seems a more suitable solution.

In our test bed we have two 802.1Q VLAN switches interconnected by an optical fiber. Each switch connects two hosts, see fig. 7. The network link is implemented as a Gigabit Ethernet connection. One of the hosts will send an AAA request for a BoD service via the regular Internet to the AAA server. The AAA server will fetch the BoD Driving Policy that describes the plan of actions in order to achieve the authorization and provisioning of the connection. After success a QoS path is provisioned as a private network between the requesting hosts and the targeted host.

More implementation details are discussed the following section.

6.1. AAA messages

Both the top level data objects to be carried in an AAA protocol message and the various types of AAA protocol messages have not formally been defined yet. Data objects and message types must be defined in an abstract way without regard to encoding. The message types defined could be thought of as the Use Cases (in the Unified Modeling Language sense) for a generic AAA server. The data objects specify what kinds of information can be routed among the AAA servers in the Generic AAA infrastructure.

We have chosen to express the different types of AAA messages in XML. The AAA protocol consists of request-reply pairs. XML schemas define these request-reply pairs. An XML schema provides a means for defining the structure, content and the semantics of an XML document. In order to keep the discussion simple an AAA service request for bandwidth might look like:

```
<Request type="BoD" >
  <AuthorizationData>
    <Credential>
      <Type>simple</Type>
      <ID>JanJansen</ID>
      <Secret>#f034d</Secret>
    </Credential>
  </AuthorizationData>

  <BodData>
   <Source>100.10.20.30</Source>
   <Destination>110.1.2.3</Destination>
   <Bandwidth>1000</Bandwidth>
   <StartTime>now</StartTime>
   <Duration>20</Duration>
  </BodData>
</Request>
```

The AAA server returns a simple answer to the invoker whether the authorization has succeeded or not. This answer is added as a text node to the XML reply shown below:

```
<Reply type="BoD" >
  <Answer>
    <Message></Message>
  </Answer>
</Reply>
```

6.2 Driving policies

In our policy language a Driving Policy is of the form 'IF(Condition) THEN (ActionList) ELSE (ActionList)'. The Driving Policies can be expressed as nested if-then-else structures, i.e. an if-then-else structure might be part of a Condition as well as part of an Action List.

As this is not the proper place to fully explain the policy language we will restrict ourselves to the example below in order to convey its expressiveness. A Driving Policy for an AAA server that accepts the above request for bandwidth might look like:

```
IF
(
  ASM::Authorizer.authorize(

  Request::AuthorizationData.Credential.
                                     Type,
  Request::AuthorizationData.Credential.
                                        ID,
  Request::AuthorizationData.Credential.
                                    Secret
                            )
)
THEN
(
  IF
  (
    ASM::RM.CheckConnection(

            Request::BodData.Source,

            Request::BodData.Destination
                            )
  )
  THEN ( ActionList1  )
  ELSE
  (  Reply::Answer.Message =
     "Request failed: Bad source or
      destination"
  )
)
ELSE
(  Reply::Answer.Message =
     "Request failed: Authorization not
      successful."
)
```

This Driving Policy calls for authorization by an ASM called 'Authorizer' that contains a public method 'authorize'. Three arguments have to be passed. The RBE can deduce that all arguments can be retrieved from the incoming request. If this call returns a false value the action in the ELSE-part instructs the RBE to add a fail text node to the XML reply. After success the ActionList of the THEN-part is executed. This ActionList contains a single action, an if-then-else structure. A Resource Manager (RM) ASM is called to check if a route between both end points of the QoS path can be established.
In this fashion a number of pre-conditions are checked before the final call for bandwidth is made:

```
RV =
  ASM::RM.BoD
  (
    Request::SwitchData.Source,
    Request::SwitchData.Destination,
    Request::SwitchData.Bandwidth,
    Request::SwitchData.StartTime,
    Request::SwitchData.Duration
  )
```

By assigning the return value to the variable RV, the return value is available in subsequent actions of the Driving Policy.

6.3 ASM structure

In our Bandwidth on Demand demonstration, we implemented a Resource manager ASM needed to deliver the BoD service. A Resource Manager (RM) ASM is responsible for tracking the state and the discovery of the resources. The RM ASM also guarantees QoS by avoiding oversubscribtion of the GigE network link.
For 802.1Q, the RM ASM administers the used VLAN tags to form the QoS path between the hosts. The RM ASM has also awareness of the destinations based on the primary IP address of the end station and it is able to discover the switch port the secondary interface is connected to.
As such, the AAA server does control a simple QoS path within a single domain.

7. Conclusions and future work

The demonstrated case represents a first step into the usage of Generic AAA for authorization within a single domain. In this example a QoS path is authorized using a set of Driving Policies executed by a RBE involving ASM's.
Generic AAA mechanisms should allow the support of many combinations of different types of applications. Generic AAA ultimately must support for example the combined authorization of a pay per view movie including QoS path based delivery across the network and a pizza to go along with it. If one of these items cannot be delivered, the entire transaction should not be authorized to proceed. Although such vision may not be realized anywhere soon, it does set the direction for future work.
The next steps will be to put research into creating multi-domain scenario's involving the pictured individual and partial control models to form more comprehensive networks. The path discovery functions that indicate which AAA servers should be contacted does play an integral role. Further research will also be put into building complex decision networks where items such as scalability, stability and performance will be investigated.

Acknowledgements
Enterasys Networks donated the equipment for this research. Leon Gommans and Bas van

Oudenaarde are funded by the IST Program of the European Union (grant IST-2001-32459) via the DataTAG project. Additional funding was provided by the Dutch National Research network SURFnet. The authors whish to thank the IRTF AAA ARCHitecture Research Group for the fruitfull discussions, in particular John Vollbrecht and Dave Spence for their neverending collaboration. The authors would like to thank Joe Mambretti and Jeremy Weinberger for their support and contributions during iGrid2002. .

References

[1] RFC2205 Resource ReSerVation Protocol (RSVP) Version 1 Functional Specification, R. Braden et. al.,Sept 1997.

[2] RFC2748 The COPS (Common Open Policy Service) Protocol, D. Durham et. al, Jan 2000.

[3] RFC2904 AAA Authorization framework, J. Vollbrecht et. al. Aug 2000.

[4] RFC2903 Generic AAA Architecture, C. de Laat et. al. Aug 2000.

[5] RFC3060 Policy Core Information Model -- Version 1 Specification, B. Moore et.al. Feb 2001.

[6] A Policy Description Language, J. Lobo et. al. July 1999 in proceedings of the American Association for Artificial Intelligence

[7] RFC 1771 A Border Gateway Protocol 4 (BGP-4), Y. Rekhter et. al. March 1995.

[8] see references at project page at http://grid-auth.infn.it

[9] The rationale of Optical Networking, Cees de Laat, Erik Radius, Steven Wallace, IGRID2002 paper submitted to Future Generations Computer Systems, November 2002.