End-to-End Provision of Policy Information for Network QoS

Volker Sander
Central Institute for Applied Mathematics
Forschungszentrum Jülich GmbH
52425 Jülich, Germany
v.sander@fz-juelich.de

William A. Adamson
Center for Information Technology Integration
The University of Michigan, Ann Arbor, MI 48109, U.S.A
andros@umich.edu

Ian Foster
Mathematics and Computer Science Division
Argonne National Laboratory, Argonne, IL 60439, U.S.A.
Department of Computer Science
The University of Chicago, Chicago, IL 60637, U.S.A.
foster@mcs.anl.gov

Alain Roy
Department of Computer Science
The University of Chicago, Chicago, IL 60637, U.S.A.
alain@cs.uchicago.edu

Abstract

High-end networked applications such as distance visualization, distributed data analysis, and advanced collaborative environments have demanding quality of service (QoS) requirements. This paper focuses on making policy decisions when users attempt to make reservations for network bandwidth across several administrative network domains that are controlled by a bandwidth broker. We present a signalling protocol that facilitates the establishment of a distributed policy decision point as well as the establishment of a direct signalling channel between the source and end domains.

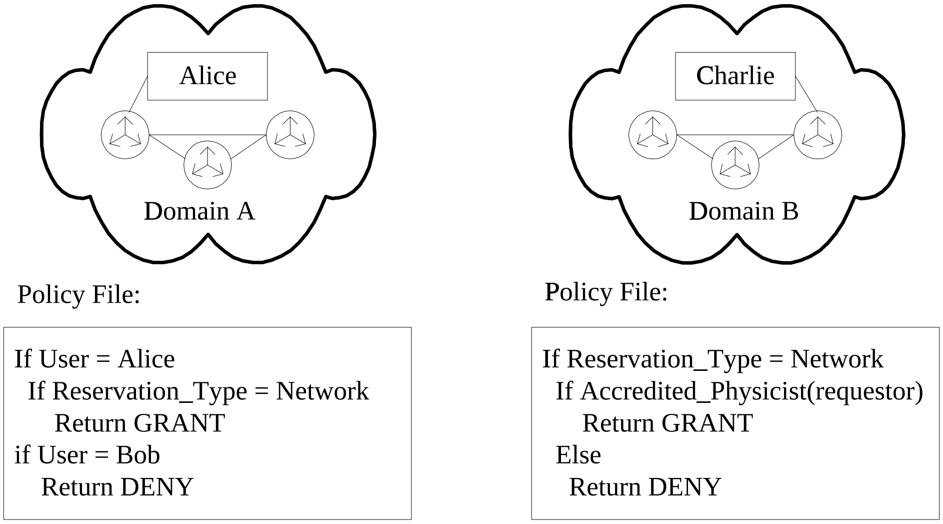
1 Introduction

Emerging computational Grid environments promise new capabilities for problem solving and effective distance collaboration [8]. However, applications that use Grid technologies often place substantial demands on scarce—and typically shared—resources such as networks, storage systems, and computers. In order for these applications to work to the satisfaction of their users they need performance guarantees for the resources they use. Reservation capabilities are an important part of any technical solution to this problem [9, 20, 19].

Because these resources are both scarce and shared, a system of rules for resource use, or *policy*, is often associated with a resource to regulate its use [24]. *End-to-end* performance guarantees typically require the *co-reservation* of several distinct resources. A number of technical issues complicate the co-reservation process:

- *Policy heterogeneity.* Co-reservation can require negotiation with resource owners in each of several distinct administrative domains. Each domain may have different policies governing who can use its resources and for what purposes, and different trust relationships with individual users. For example, in Figure 1, domain A's policy might state that "Alice can use the network, Bob cannot," while domain B's policy is that "only accredited physicists can use the network."

Published in *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC), August 2001*



Policy File:

```
If User = Alice
  If Reservation_Type = Network
      Return GRANT
if User = Bob
    Return DENY
```

Policy File:

```
If Reservation_Type = Network
  If Accredited_Physicist(requestor)
      Return GRANT
  Else
    Return DENY
```

Figure 1. Different domains may have different reservation policies.

- *Trust heterogeneity.* Scalability demands that every resource should not have a direct trust relationship with every user. While some domains know about individuals (e.g., domain A), others must be able to delegate responsibility for personal trust relationships to third parties. (For example, domain B agrees to provide resources to anyone whom a third party accredits as a "physicist.")

- *Interdomain policy dependencies.* A policy expressed in one domain can be dependent on policy decisions expressed in other domains. For example, for reasons for presented below, domain A may wish to enforce the policy "I will only authorize a reservation if reservations have also been approved for all other resources in the end-to-end path." Or, domain B might only authorize bandwidth greater than 10 Mb/s if domain A has committed to shaping the traffic in a certain way.

- *Scalability.* If a set of applications creates many parallel flows between the same two end-domains, it is infeasible to negotiate an end-to-end reservation for each one.

In this paper, we present an innovative co-reservation architecture that addresses these issues. This architecture incorporates two principal elements:

- An *interdomain signalling protocol* supports the communication of reservation requests and associated authentication and authorization information between resource managers in different domains.

- Support for *tunnels* [4] allows an entity to request an aggregate end-to-end reservation. Users authorized to use this tunnel can then request portions of this aggregate bandwidth by contacting just the two end domains—the intermediate domains do not need to be contacted as long the total bandwidth remains less than the size of the tunnel.

In the rest of this paper we address these issues in the context of network reservations. First, we review some background material, then present our architecture, and finally present a protocol that facilitates the secure propagation of the information relevant to authorization.

2 Background: Differentiated Services and Bandwidth Brokers

Making network reservations is a complicated process. Several methodologies have been proposed and two basic approaches have been developed within the Internet Engineering Task Force (IETF). The first approach, as exemplified by the RSVP protocol [3] and Integrated Services [2] model, requires that a reservation request be propagated through each router that will handle the traffic for a reservation. There are some scaling problems with this approach, including the fact that each router normally has to recognize each packet belonging to a reserved flow and treat it specially [15].

The alternative Differentiated Services [1] approach was developed in reaction to the perceived scaling difficulties with RSVP. Instead of having each router recognize each reservation, only the first router recognizes packets on a per flow base, and then marks the packet as belonging to a *traffic aggregate*. Each subsequent router then recognizes the traffic aggregates and treats them in some pre-defined way. By carefully limiting the traffic admitted to the traffic aggregate, QoS guarantees for bandwidth can be provided [18].

Approaches for limiting the traffic in order to provide guarantees are still being developed, but an emerging mechanism is the *bandwidth broker* (BB) [4]. A BB provides admission control and configures the edge routers of a single administrative network domain.

Whenever the network reservation end-points are in different domains, a specific contract between peered domains comes into place, used by BBs as input for their admission control procedures. A service level agreement (SLA) regulates the acceptance and the constraints of a given traffic profile. Service Level Specifications (SLS) are used to describe the appropriate QoS parameters [22] that an SLA demands. End-to-end guarantees can then be built by a chain of SLSs.

3 Co-Reservations and Inter-BB Signalling

It is unlikely that a single bandwidth broker will control more than one domain, because each administrative domain wishes to have control over the resources it owns. A network reservation for traffic traversing multiple domains must therefore obtain multiple network reservations, as shown in Figure 2. Here, Alice wants to make a network reservation from her computer in *source domain* A to Charlie's computer in *destination domain* C. Somehow she needs to contact and negotiate a reservation with BB_A and BB_C as well as the *intermediate (or ISP) domain*, BB_B. We describe two approaches to making this happen.

Approach 1: Source-Domain-Based Signalling Alice, or an agent working on her behalf, can contact each BB individually (Figure 3). A positive response from every BB indicates that Alice has an end-to-end reservation. However, there are two serious flaws with this methodology. First, it is difficult to scale since each BB must know about (and be able to authenticate) Alice in order to perform authorization. Furthermore, if another user, Bob, makes an incomplete reservation, either maliciously or accidentally, he can interfere with Alice's reservation. Such a misreservation is illustrated in Figure 4.

We know of two multi-domain reservation systems that adopt this approach. We ourselves have developed what we call an *end-to-end reservation API* within our General-purpose Architecture for Reservation and Allocation (GARA) system. GARA provides advance reservations and end-to-end management for quality of service on different types of resources, including networks, CPUs, and disks [10, 11]. It defines APIs that allows users and applications to manipulate reservations of different resources in uniform ways. For networking resources, GARA implements a bandwidth broker as described above. A library provided by GARA implements an end-to-end network API that facilitates end-to-end reservation for its users. Receiving source and end-point of a network reservation, the library determines the relevant BBs and propagates the request to each them either sequentially, or if optimized, concurrently. Our implementation of this API guarantees that all necessary domains are contacted, but of course there is nothing to stop a malicious user from modifying our implementation to skip a domain. Furthermore, Alice still has to be known by all related BBs.

The STARS system [13] adopts a variant of this approach, in which a separate source domain entity—the reservation coordinator (RC)—performs the end-to-end reservation. This strategy alleviates the problems noted above, in two respects: first, in many situations it may be feasible for the RC to be "trusted" to make all necessary reservations; second, all bandwidth-brokers need not be aware of all end-users. However, we still require a direct trust relationship between all intermediate and possible end-domains.

Approach 2: Hop-by-Hop-Based Signalling. The problems just noted motivate us to define an alternative approach, in which reservation requests are propagated between BBs rather than all originating at the end domain. As shown in Figure 5, this means that Alice only contacts BB_A, which then propagates the reservation request to BB_B *only if* the reservation was accepted by BB_A. Similarly, BB_B contacts BB_C. With this solution, each BB only needs to know about its neighboring BBs, and all BBs are always contacted. In addition to the hop-by-hop based signalling approach, Figure 5 also demonstrates the use of the GARA API to couple a multi-domain network reservation with a CPU reservation in domain C.

Note that source-domain-based signalling may be faster than hop-by-hop based signalling, because the reservations for each domain can be made in parallel

4 Policy Information

A complicating factor in a multi-domain environment is that different domains may wish to enforce different policies concerning who can use their resources. For example, in Figure 6, the three BBs specify three different policies:

- The source domain BB, BB$_A$, specifies that Alice is allowed to use as much bandwidth as she wants, up to the maximum available, except during business hours when she is restricted to 10 Mb/s.
- The intermediate domain BB, BB$_B$, specifies that up to 10 Mb/s can be allocated to anyone who is a member

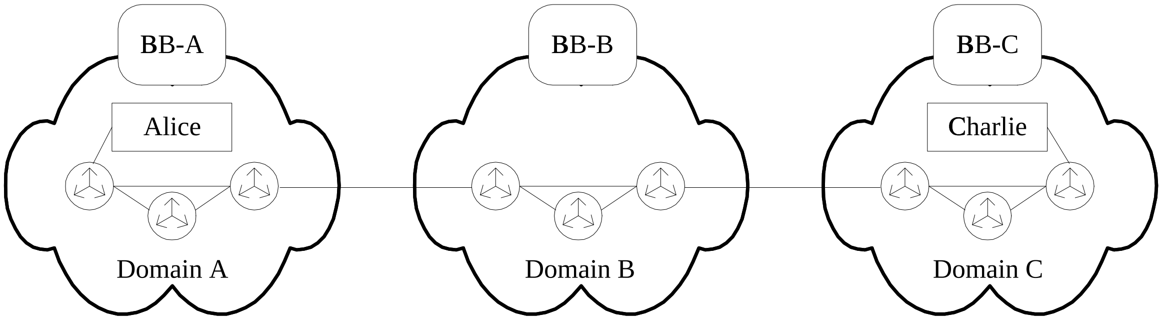


Figure 2. The multi-domain reservation problem. Alice needs to contact three BBs to make a network reservation from her computer in domain A to Charlie's computer in domain C.

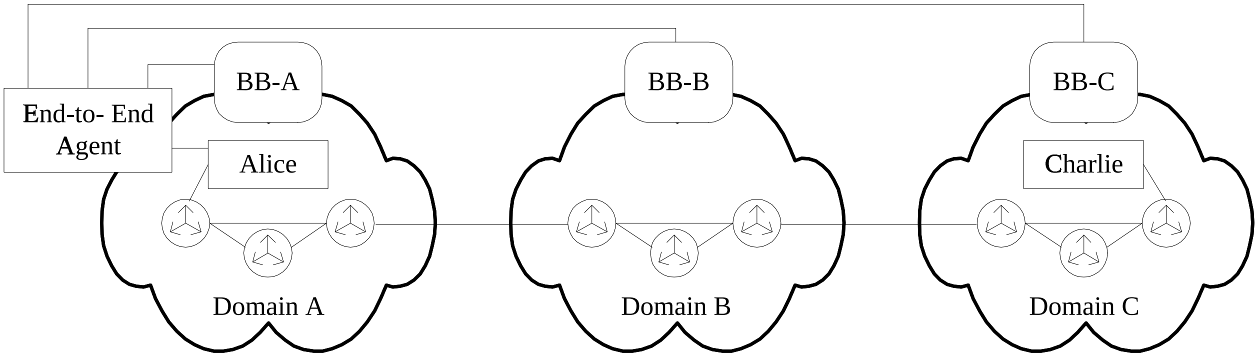


Figure 3. Source-domain-based signalling is controlled by a source domain entity that contacts all related BBs directly.

of group "ATLAS experiment" or who can provide a capability provided by community "ESnet."

- The destination domain BB, BB_C, specifies that it will only accept reservations above 5 Mb/s, if the requestor can provide a capability provided by community "ESnet" and if she can present a valid reservation for a computing resource in domain C.

In general, we see that a BB making a decision must be able to consider:

- request parameters, e.g., the destination domain and the amount of bandwidth required,
- authentication information, e.g., a public domain credential for the originating user,
- authorization information such as assertions regarding group memberships (perhaps originating from the user or the source domain) and cryptographically signed capabilities issued by various authorities,
- SLA information such as traffic engineering parameters of up stream BB's.

We assume that either the end user or the bandwidth broker of the source domain acting on behalf of the end user contacts a policy server such as an Akenti [21] server. This policy server provides the related policy information based on the request, its use conditions, and the identity of the requestor. This policy information is propagated along with the user's request. However, the propagation protocol should not make strong assumptions on the actual syntax of

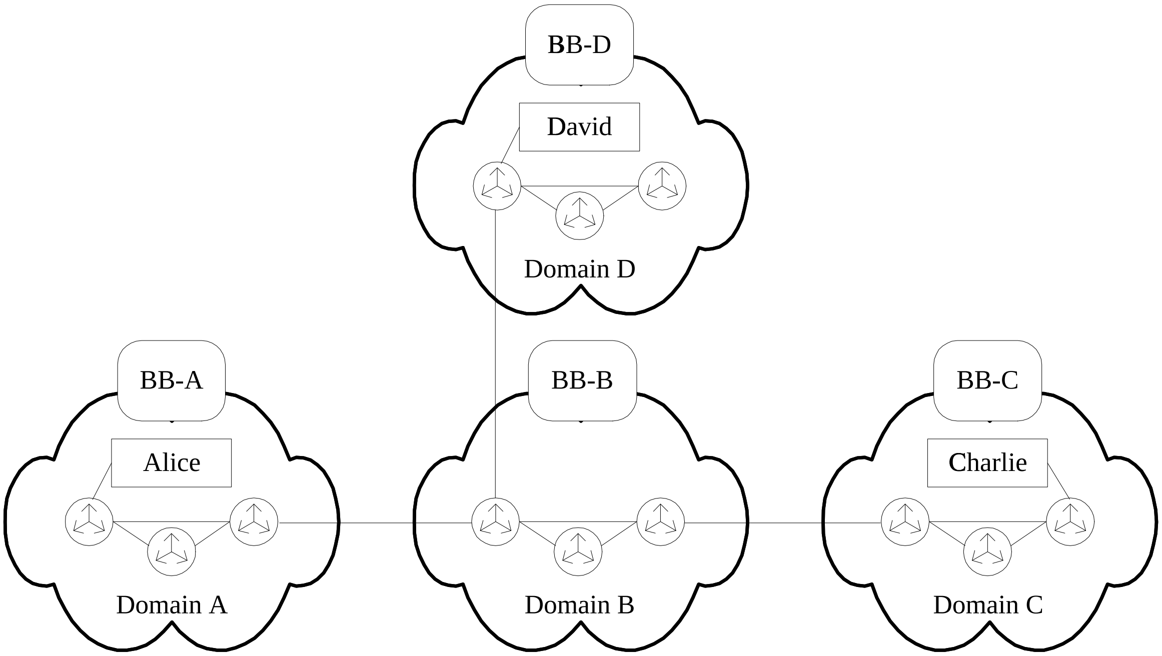


Figure 4. David, a malicious user in domain D, makes a reservation in domains D and B, but fails to make a reservation in domain C, even though he will be sending his reserved traffic to Charlie in domain C. Domain C polices traffic based on traffic aggregates, not on individual users, so it cannot tell the difference between David's reserved traffic and Alice's reserved traffic. Therefore, there will be more reserved traffic entering domain C than domain C expects, causing it to discard or downgrade the extra traffic, thereby affecting Alice's reservation.

this policy information. It should handle simple attribute-value pairs which might be signed by the assigning entity as well as capability certificates.

By separating authentication and authorization issues one can facilitate the flexible propagation of different policy related information. As long as the protocol ensures that the end-entity can approve the integrity and the authenticity of the received information, authorization decisions can be made without depending on specific features of the language expressing the policy attributes. Therefore, the same propagation protocol can be used for different policy representations. The important aspect here is

- the protocol is independent of policy syntax, and
- the different domains need to agree on the syntax, and
- the syntax in the figures is therefore just an example

¿From that perspective, the actual syntax of the use conditions and capabilities [21] described as policy file in Figure 6 and the illustrated format of the policy information represents one example scenario of the propagation protocol.

Notice that an implication of this discussion is that our architecture must provide mechanisms for communicating information securely between BBs. We discuss this mechanism in Section 6.

5 Architecture Overview

We can now proceed to define our architecture. We assume a set of BBs that communicate via an inter-BB signalling protocol. A source BB accepts incoming requests that contain the information listed in the preceding section, such as request parameters, authentication information, and authorization information (assertions and/or capabilities).

We introduce an entity called a policy server that encapsulates a BB's admission control procedures. When a request comes in, it is forwarded to the policy server which executes local policy and passes back a result ("yes" or "no") and a modified request. The implementation of this

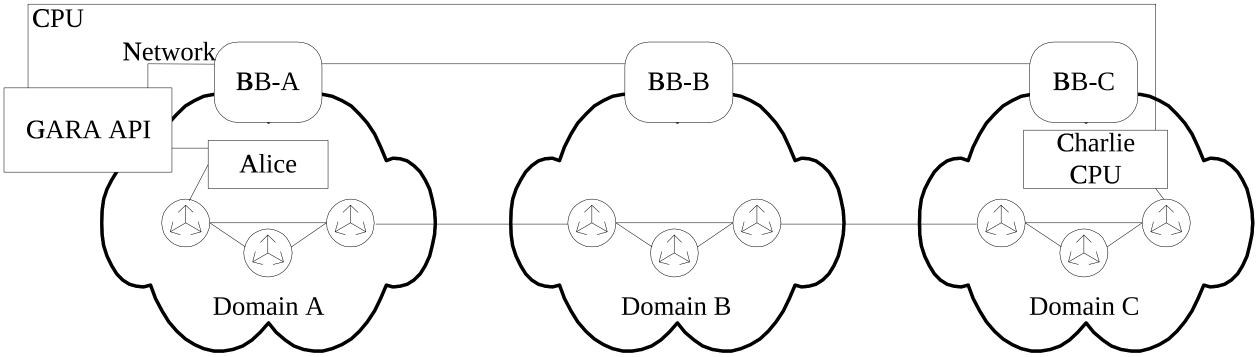


Figure 5. Hop-by-hop-based signalling of QoS demands is done using an authenticated channel between peered BBs among the downstream path to the destination.

policy server is not the focus of this paper, but we note that we would like it to be able to express diverse authorization policies including:

- Authority based on validated *assertions* concerning group membership. In this case, the policy might say "approved if group server P validates the user as a physicist"; if the user's request includes the assertion "I am a physicist", then the policy server verifies that assertion by contacting that group server, passing the user's supplied identity certificate. The group server then verifies whether the user is a member of the group and responds appropriately.
- Authority based on cryptographically signed *capabilities* issued by various authorities [17]. In this case, the policy might say "approved if the user supplies a capability of type C issued by authority A;" if the capability C is supplied, then the policy server verifies its validity and responds appropriately. One representation of capabilities is to encode the capability attributes in the extension field of an ITU X.509v3 certificate [14], issued by a specific community authorization server (CAS) being developed within the Globus project.
- Traditional *access control lists* may also be of interest, expressed in terms of the identities of individuals who are allowed to use resources.

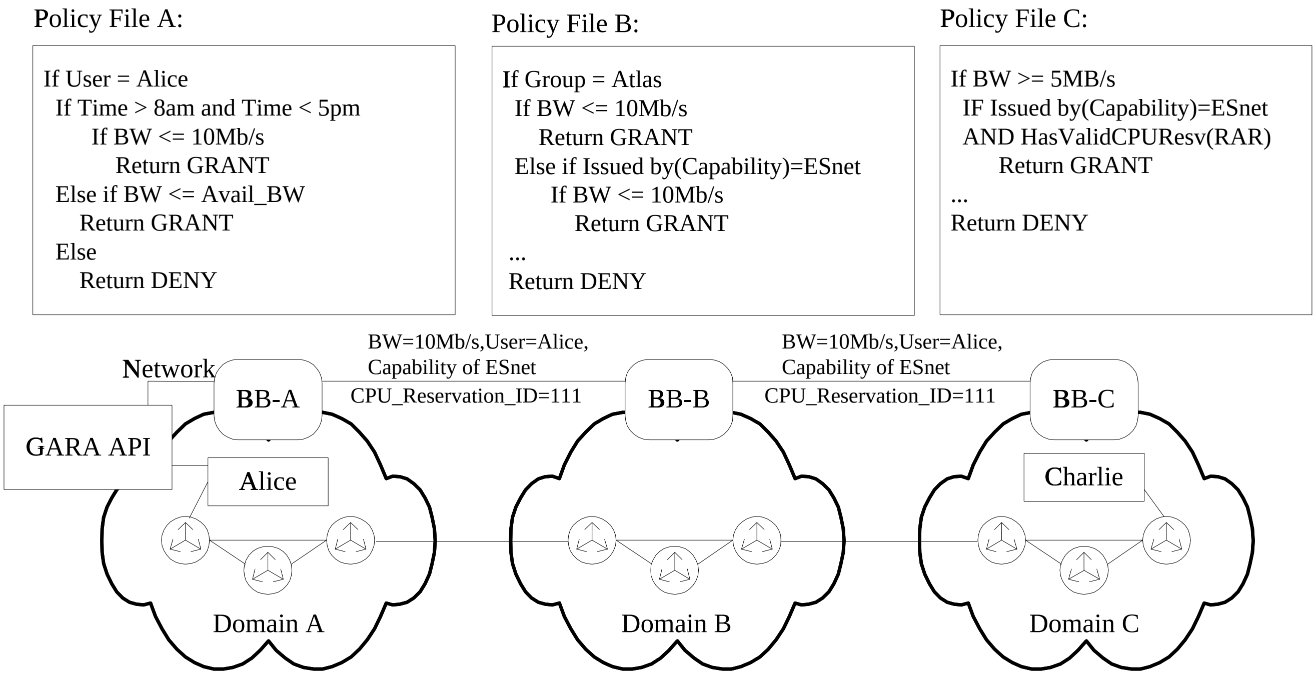
6 A Transitive Trust Model for Signalling Policy Information

Having defined the principal elements of our architecture, we now describe our inter-BB signalling protocol in detail and explain how (a) it ensures secure transmission of information between BBs and (b) establishes direct trust relationships between end domains, as required for the establishment of tunnels. We describe the actions that are performed at the source domain, in intermediate domains, and at the destination domain.

6.1 Source Domain

Recall that in our model a user (or agent acting on their behalf) signals a reservation request to the BB in the user's administrative network domain. In addition to the basic bandwidth request, such as 10 Mb/s of guaranteed bandwidth, this request may include additional information such as a cost that the user is willing to accept and assertions and capabilities as described above. When such a user request arrives, the source domain BB performs four steps:

- The BB contacts the policy server to verify that the user-provided information is correct, and that the user is authorized to make the request in the local domain.
- The BB receives additional domain-wide information from the policy server. This information is used to identify additional constraints that might have to be added to the reservation request. This may include groups in which the end-domain resource requires membership, additional cost offers for the particular request, any information relevant for traffic engineering purposes for downstream domains, or specific requirements derived from the contract with the peered domain, such as parameters for treatment of excess traffic or reliability parameters expected for this service [22].



Policy File A:

```
If User = Alice
  If Time > 8am and Time < 5pm
        If BW <= 10Mb/s
            Return GRANT
  Else if BW <= Avail_BW
      Return GRANT
  Else
      Return DENY
```

Policy File B:

```
If Group = Atlas
  If BW <= 10Mb/s
      Return GRANT
  Else if Issued by(Capability)=ESnet
        If BW <= 10Mb/s
            Return GRANT
 ...
 Return DENY
```

Policy File C:

```
If BW >= 5MB/s
  IF Issued by(Capability)=ESnet
  AND HasValidCPUResv(RAR)
        Return GRANT
...
Return DENY
```

Figure 6. A multi-domain environment where each bandwidth broker enforces a specific list of reservation policies. The user Alice is making a network reservation request, referring to an existing CPU-reservation in domain C. Each BB will evaluate each request with respect to its local policy file.

- The BB decides whether or not the request can be satisfied within the local domain, based both on the traffic profile and the policy constraints.

- If the reservation request can be granted locally, the BB forwards the request to the next BB in the network path, along with any additional information that was added. This additional information facilitates a signalling path tracing as well as the propagation of identity information. This allows the establishment of a direct mutually authenticated channel between source- and end-domain when the tunnel is actually used.

6.2 Intermediate Domain

Whenever an intermediate BB receives a message from the upstream BB, it checks whether the requested traffic profile conforms to the related SLA, and, if this is the case, it may add additional information such as capabilities and policies, and will forward the request downstream. It may use SLS-related information added by any upstream domain, if they exist and are relevant for this decision.

Whenever a request is denied by one domain, the event is propagated upstream to inform the user of the reason for the denial.

6.3 Destination Domain

The BB of the ultimate end domain makes the final authorization decision based on its local policies, using (as in the case of the intermediate domains) any or all relevant information supplied in the request, whether request parameters, identify certificate, assertions, or capabilities.

6.4 The Signalling Protocol

We now look more closely at the security issues associated with our inter-BB signalling protocol. There are two issues:

- Messages between BBs should be mutually authenticated

- Because trust is not transitive in general, the protocol must accomplish a trustworthy model for transporting the policy and additional information end-to-end.

The direct signalling between peer BBs used in the above description can easily be secured using SSLv3/TLS [5]. While SLAs are used to regulate the services between two domains, we extend this agreement by adding information to facilitate the trust relationship between two peered BBs. This information includes the certificates of the peered BBs as well as the certificate of the issuing certificate authority, all used during the SSL handshake.

A common way to ensure the integrity and authenticity of messages is to use digital signatures. In our case, this works as follows. A user requesting a service augments the request with any relevant additional information, such as a supplied reservation handle, and signs the resulting augmented request with her private key before it gets propagated. The source domain's BB might further augment the request—such as information received from a policy server—and sign the resulting larger request with its own private key. A complete request therefore is comprised of a collection of information, each signed by the entity that added it. The signatures both assert the authenticity of the information and allows for the tracking the path taken by a request as it moves from BB to BB.

When a request is received at the destination domain, the BB checks local policy and resource availability. If these checks succeed, then the BB adds its own signed policy information and propagates the modified request to the previous intermediate domain BB, again using SSL/TLS. The approval therefore propagates back to the source domain, with each intermediate domain referring to local SLA and SLS information as it verifies that it can approve the request.

Establishment of tunnels is supported by a resource allocation request (RAR), which is the dynamic establishment of a direct signalling channel between source- and end-domains. Because of this direct connection, it must be possible for the end-domain to derive the identity of the source domain's BB.

One technical problem raised by this approach is access to public keys. The approval of a digital signature requires access to the public key of the signing subject. This access can be accomplished by one of the following techniques:

- Distribute all relevant certificates within all requests. Supposing that the issuing authority is known and trusted, one can check the authenticity of the signature. However, there is a question of whether there is a way to facilitate the approval of a signature of entities without a direct trust relationship and in the absence of cross-signed CAs. We address this problem by having each domain add the certificate of the upstream domain—known because of the SSL handshake—and sign it. This web of trust allows each domain to access a list of key introducers [12] when deciding whether to accept the public key stored in the certificate.

- Maintain a certificate repository accessible through secure LDAP. Upon receipt of the reservation specification, C would extract the distinguished name (DN) of A from it, and would search in the certificate repository for the related public key. It is important to note that there has to be a strong trust relationship with the repository.

- Completely decouple the distribution of policy information from BB-to-BB communication, i.e., transport it out of band.

- (Restricted) delegation mechanisms could be used to propagate authorization attributes, by having each BB impersonate the caller's identity.

While each of these solutions has interesting characteristics, we believe that the first solution is to be preferred because it offers a flexible framework for trust decisions supporting different security levels.

To describe the proposed mechanism and its advantages, we introduce the following notation:

- *res_spec* reservation specification of the user

- *Capability_Cert* denotes authorization information in any valid representation. The information is typically signed by an issuer, i.e. a policy or an authorization server. Examples are Attribute Certificates [6], capabilities [17], or Impersonation Certificates [23] containing authorization attributes in its extensions. We will use *Capability_Cert*$'_A$ to indicate that entity A has issued a capability. A detailed description of this procedure can be found at the end of this section. Note that the delegation is only performed when capabilities are transported. For other representations this field might be empty.

- *pkey*$_A$ private key of entity A

- *cert*$_A$ X509 certificate of entity A

- *sign*$_{pkey_A}$*(attributes)* adds a signature to the given attribute list using the private key of entity A

- *DN*$_A$ distinguished name of entity A

We assume that the BB in domain A receives the following information from User U:

$$\begin{aligned} RAR_U = & sign_{pkey_U}(\{res\_spec, DN_{BB_A}, \\ & Capability\_Cert'_{CAS}, Capability\_Cert'_U\}) \end{aligned}$$

Because RAR_U was received through a mutually authenticated channel, we assume that the BB in domain A has access to the user's certificate. This information facilitates the approval of the received capability certificates, which were issued by some authorization servers, because the granted capabilities were passed to BB_A using the user's private/public key pair as proxy key. Once the request was approved, it is extended with the user's certificate and the DN of the downstream BB, as well as with additional policy information, if necessary, and signs the new message using its private key:

$$RAR_A = sign_{pkey_{BB_A}}(\{RAR_U, cert_U, DN_{BB_B}, \\ Capability\_Cert'_A\})$$

When BB_B receives this, it adds BB_A's certificate and the distinguished name of the downstream's BB to RAR_A. If necessary, it will add additional policies and capabilities, signs the whole message, and transmits it to C:

$$RAR_B = sign_{pkey_{BB_B}}(\{RAR_A, cert_A, DN_{BB_C}, \\ Capability\_Cert'_B\})$$

Note that BB_C is able to check the signature of RAR_B because it does have access to the certificate of BB_B exchanged during the SSL handshake. Additionally, BB_B introduces the public key of BB_A by transmitting its certificate. BB_A however, as source of the request, did approve the SLA with domain B by listing the DN of BB_B in its request. BB_C can now decide whether it trusts BB_B's introduction.

Now let us assume that RAR$_N$ specifies the message submitted by the n-th bandwidth broker of the path between source and end-domain. Furthermore, let us assume that the n+1-th bandwidth broker is not the one of the end-domain. Then we can describe the message created by the n+1-th bandwidth broker as:

$$RAR_{N+1} = sign_{pkey_{BB_{N+1}}}(\{RAR_N, cert_N, DN_{BB_{N+2}}, \\ Capability\_Cert'_{N+1}\})$$

While the proposed protocol permits direct access to the transported information whenever appropriate, it also makes it possible to check signatures without a direct trust relationship. For example, in the case above, let us actually resolve what BB_C would receive:

$$sign_{pkey_{BB_B}}(\{sign_{pkey_{BB_A}}(\\ \{sign_{pkey_U}(\{res\_spec, DN_{BB_A}, \\ Capability\_Cert'_{CAS}, Capability\_Cert'_U\}), \\ cert_U, DN_{BB_B}, Capability\_Cert'_A\}), \\ cert_A, DN_{BB_C}, Capability\_Cert'_B\})$$

BB_B's certificate is approved by the SLA and the SSL handshake. BB_C can therefore be sure that the bandwidth broker of domain B has approved the receipt of the message

$$sign_{pkey_{BB_A}}(\{sign_{pkey_U}(\{res\_spec, DN_{BB_A}, \\ Capability\_Cert'_{CAS}, Capability\_Cert'_U\}), \\ cert_U, DN_{BB_B}, Capability\_Cert'_A\})$$

from a trusted entity presenting certificate cert$_A$. Note that C also knows that B has established this trust relationship based on a contract, i.e., B has signed a contract that enforces it to trust the subject capable of using cert$_A$. Depending on the level of trust C is actually requiring, this permits a trust relationship to A, as B was

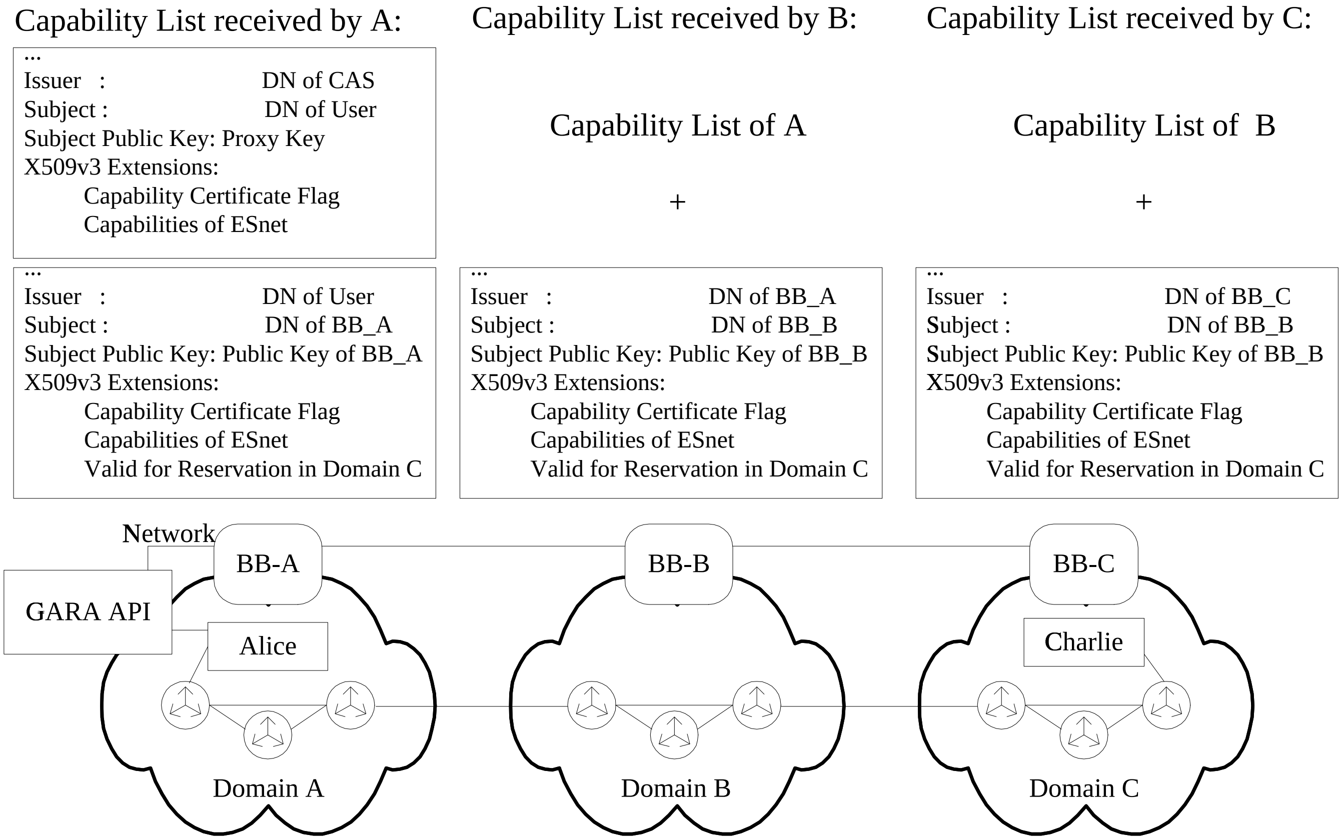
- approving that BB_A was able to use the private key corresponding to cert$_A$
- has a trust relationship to A based on a contract
- has a known trust relationship to C based on a contract

Checking its own security policy which might limit the depth of an acceptable trust chain, BB_C may accept the public key of cert$_A$, and use this to approve the received information.

We believe that the proposed model offers a flexible and realistic solution for propagating policy information end-to-end. It is flexible because it does not enforce a specific security policy: instead, it offers access to all relevant information. It is realistic because it follows existing trust relations. ¿From an accounting perspective there is already an accepted transitive billing scheme. Whenever a domain actually bills the requesting entity for the use of the network service, SLAs are already used to set up a transitive billing relation in multi-domain networks. When network traffic enters domain C through domain B, it is billed using the agreement between B and C. B as a transient domain, however, would also bill traffic originating from a different domain using the related SLA. Finally, the source domain would bill the traffic against the originator.

6.5 Propagation of Capability Certificates

The model also supports capabilities issued by community authorization services via mechanisms that allow the end-domain to use a granted capability for authorization purposes. Instead of using the private key corresponding to the public key listed in the capability, the BB of the end-domain will use its own private key, together with the full chain of messages. This chaining can be accomplished by



Capability List received by A:

...
Issuer : DN of CAS
Subject : DN of User
Subject Public Key: Proxy Key
X509v3 Extensions:
Capability Certificate Flag
Capabilities of ESnet

...
Issuer : DN of User
Subject : DN of BB_A
Subject Public Key: Public Key of BB_A
X509v3 Extensions:
Capability Certificate Flag
Capabilities of ESnet
Valid for Reservation in Domain C

Capability List received by B:

Capability List of A

+

...
Issuer : DN of BB_A
Subject : DN of BB_B
Subject Public Key: Public Key of BB_B
X509v3 Extensions:
Capability Certificate Flag
Capabilities of ESnet
Valid for Reservation in Domain C

Capability List received by C:

Capability List of B

+

...
Issuer : DN of BB_C
Subject : DN of BB_B
Subject Public Key: Public Key of BB_B
X509v3 Extensions:
Capability Certificate Flag
Capabilities of ESnet
Valid for Reservation in Domain C

Figure 7. Capability certificates received by each bandwidth broker during the proposed end-to-end signalling process.

following the cascaded authorization mechanism proposed by Neuman [17]. In his model each subordinate server signs the received capabilities using the private key of the corresponding public key stored in the capability. Neuman used proxy-key pairs to fulfill this task. In our model, the BB of the source domain uses the public key of the peered downstream domain as public proxy key.

To describe the proposed protocol more precisely, we construct a use scenario (Figure 7) where the user has received a capability certificate by some Community Authorization Server (CAS) during the "grid-login" process. Let us assume that the capability certificate simply contains all capabilities of the ESnet group in the X509v3 extension field. The certificate itself lists a public proxy key, the DN of the user (potentially modified to indicate that this is a capability certificate) and the CAS, as well as the signature of the CAS. In addition to the capability certificate, the user owns the private key corresponding to the public proxy key. Whenever a service is requested, the related server receives the capability certificate and requests prove of the knowledge of the private proxy key. This step, however, can be viewed as authentication. Whenever the authenticity of the capability certificate is approved, a policy engine can directly use its attributes, such as the group membership, to decide whether the request can be granted or not.

In our example the user now requests a network reservation from a host in domain A to a virtual reality device in domain C. To describe the delegation process, we introduced the generic notation Capability_Cert$'$. Here, we clarify the implementation of this notation for capability certificates. To delegate the capability cert to BB_A, the user creates a new capability certificate. The subject of this new certificate is BB_A. Instead of creating a new public key, the SSL handshake of the protocol allows to insert BB_A actual public key to this certificate. Finally, the extensions of the original capability certificate are copied, i.e. the group membership, extended by an additional restriction "valid for RAR". Instead of signing the new certificate with the user's private key, it is signed by using the private proxy key.

BB_A now receives two capability certificates. The original one issued by CAS and the one issued by the user. Note that BB_A can prove that it actually posses the new capability certificate by proving the knowledge of the related private key: $pkey_{BB_A}$. Note that the remaining fields of RAR_U are not needed in this context. Their purpose is to implement the introductory model which facilitates the establishment of a tunnel between source and end-domain.

Now BB_A delegates the received capabilities to BB_B by creating a new certificate. Therefore BB_B receives three capability certificates. One issued by the CAS, one by the user, and one by BB_A. Finally, BB_B delegates this to BB_C which posses four capability certificates. To authorize the request, BB_C can now submit the certificate chain to a policy engine which:

- checks that CAS was issuing a capability certificate for the user,
- checks that the user was able to use the private proxy key during delegation to BB_A,
- checks that BB_A delegated the capability to BB_B, because the new certificate was signed using $pkey_{BB_A}$,
- checks that BB_B delegated the capability to BB_B, because the new certificate was signed using $pkey_{BB_B}$,
- checks that BB_C actually owns the capability certificate by requesting a prove of the knowledge of $pkey_{BB_C}$,
- checks that the validity of all capabilities, i.e. whether some entity did change them inappropriately during delegation,
- uses the ESnet capabilities for authorization purposes.

7 Related Work

The development of a Community Authorization Service is an ongoing effort of the Globus project [7]. The proposed bandwidth broker signalling protocol embeds the delegation of capability certificates issued by an upcoming CAS implementation.

The Akenti [21] project associates lists of Certificate Authorities and administrators with a resource's use policy, expressed in attribute value pairs in a use-condition certificate. The administrators can then create user-attribute certificates each of which associates a user, an attribute and a resource. In order for a user to be granted access to a resource, the Akenti policy engine needs to be presented with multiple user-attribute certificates signed by a CA on the resource CA list, and satisfying all rules in the resource use-condition certificate. While Akenti user-attribute certificates can be encapsulated in the signalling protocol's Capability_Cert, the current Akenti CA trust architecture does not lend itself to utilizing the transitive trust properties of the signalling protocol.

Keynote [16] is designed around authorizing access of local users to local resources, and needs to be extended to operate in the distributed authorization space where users and resources often belong to distinct administrative realms. We are, however, not bound by any previous design, and can take full use of the features of the signalling protocol such as transitive trust and certificate propagation.

The Internet2 project is currently finalizing its design of a bandwidth broker to bandwidth broker protocol [4]. As member of the design team, we are interacting with this specification effort.

8 Conclusions

In multi-domain environments, the establishment of end-to-end network reservations raises challenging technical problems due to the diverse trust relationships and usage policies that can apply. We have described a BB architecture and protocol that addresses these problems. In this architecture, individual BBs communicate via bilaterally authenticated channels between peered domains. Our protocol provides the secure transport of requests from source domain to destination domain, with each bandwidth broker on the path being able to enforce local policies and modify the request with additional constraints. In addition to the description we have provided here, we have separately proposed changes to the SIBBS protocol [4] to incorporate our model of the transport of policy information. The proposal is currently under consideration. In addition, we have recently started an implementation project to extend GARA using this communication model between peer BBs.

References

[1] S. Blake, D. Black, M. Carlson, M. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. *Internet RFC 2475*, 1998.

[2] R. Braden, D. Clark, and S. Shenker. RFC 1633: Integrated services in the internet architecture: an overview. *Internet RFC 1633*, July 1994.

[3] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP)-version 1 functional specification. *Internet RFC 2205*, Sept. 1997.

[4] P. Chimento et al. QBone bandwidth broker architecture. Work in Progress, available from `http://sss.advanced.org/bb/`.

[5] T. Dierks and C. Allen. The TLS Protocol, Version 1.0. *Internet RFC 2246*, Jan. 1999.

[6] S. Farrell and R. Housley. An internet attribute certificate profile for authorization. *Internet Draft draft-ietf-pkix-ac509prof-06.txt*, January 2001.

[7] I. Foster and C. Kesselman. The Globus project: A status report. In *Proceedings of the Heterogeneous Computing Workshop*, pages 4–18. IEEE Computer Society Press, 1998.

[8] I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, 1999.

[9] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy. A Distributed Resource Management Architecture That Supports Advance Reservations and Co-Allocation. In *International Workshop on Quality of Service*, 1999.

[10] I. Foster, A. Roy, and V. Sander. A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation. In *International Workshop on Quality of Service*, 2000.

[11] I. Foster, A. Roy, V. Sander, and L. Winkler. End-to-End Quality of Service for High-End Applications. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, 1999. `http://www.mcs.anl.gov/qos/qos_papers.htm`.

[12] S. Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly and Assiciates, 1994.

[13] G. Hoo, K. Jackson, and W. Johnston. Design of the STARS Network Reservation System. Technical report, LBNL, Orlando Lawrence Berkeley National Laboratory, 2000. `http://www-itg.lbl.gov/QoS/homepage.html`.

[14] I. T. S. S. (ITU-T). The Directory: Authentication Framework. In *Recommendation X.509*, 1997.

[15] A. Mankin et al. Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement: Some Guidelines on Deployment. *Internet RFC 2208*, Sept. 1997.

[16] M.Blaze, J. Feigenbaum, and J. Ioannidis. The KeyNote Trust-Management System Version 2. *Internet RFC 2704*, Sept. 1999.

[17] B. Neuman. Proxy-Based Authorization and Accounting for Distributed Systems. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 283–291, 1993.

[18] V. Sander, I. Foster, A. Roy, and L. Winkler. A Differentiated Services Implementation for High-Performance TCP Flows. *Elsevier Computer Networks*, 34:915–929, 2000.

[19] W. Smith, I. Foster, and V. Taylor. Scheduling with advance reservations. In *Proceedings of the IPDPS Conference*, May 2000.

[20] Q. Snell, M. Clement, D. Jackson, and C. Gregory. The Performance Impact of Advance Reservation Meta-scheduling. In *IPDPS 2000 Workshop, Job Scheduling Strategies for Parallel Processing (JSSPP 2000)*. Springer-Verlag LNCS 1911, 2000.

[21] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari. Certificate-based access control for widely distributed resources. In *Proceedings of the Eighth Usenix Security Symposium*. 1999.

[22] P. Trimintzios et al. An Architectural Framework for Providing QoS in IP Differentiated Services Networks. In *To appear at the 7th IFIP/IEEE International Symposium on Integrated Network Management (IM 2001)*, 2001.

[23] S. Tuecke, D. Engert, and M. Thompson. Internet X.509 Public Key Infrastructure Impersonation Certificate Profile, 2001. `http://www.gridforum.org/security/ggf1_2001-03/drafts/draft-ggf-x509-impersonation-06.txt`.

[24] J. Vollbrecht et al. AAA Authorization Application Examples. *Internet RFC 2905*, Aug. 2000.