Investigating Bandwidth Broker's inter-domain operation for dynamic and automatic end to end provisioning

Christos Bouras and Dimitris Primpas

Research Academic Computer Technology Institute, N.Kazantzaki Str. & Department of Computer Engineering and Informatics, University of Patras, 26500 Rion, Patras, Greece
E-MAIL: bouras@cti.gr, primpas@cti.gr

Abstract

This paper deals with a distributed bandwidth broker that we try to extend in order to perform inter-domain operation. The basic issues for inter-domain operation are discussed and we try to approach the most demanding issues as the selection of the best inter domain routing path. Generally, we discuss three models for inter domain routing through bandwidth broker, analyzing their advantages. Also, a very important point that affects the inter-domain operation is the SLAs between adjacent domains and the capability that a bandwidth broker should have to ask and perform dynamic negotiation. Finally, we analyze the best model and present how it should be incorporated in the existing distributed implementation.

Keywords: bandwidth broker, SLA, end to end provisioning

1. Introduction

The last years many critical applications have been appeared widely, demanding specific network characteristics in order to operate effectively. For this purpose, many service providers and researchers started studying Quality of Service issues. Currently, 2 basic architectures have been proposed by IETF, the Integrated Services (IntServ) and the Differentiated Services (DiffServ). The DiffServ (Black et al. 1998) has been widely adopted and many mechanisms and QoS services have been implemented. The most critical issue in the deployment of a QoS service is the ability to provision the network and provide the service in end to end basis. Otherwise, the deployment is quite complicated, as many parties should work and also there are many issues that can finally degrade the overall performance that the applications experiences. In this direction, the automatic provisioning and management of a network and ideally all the connected networks through pre-agreed interfaces is the next big challenge. This goal can be achieved through the bandwidth brokers. The last years, several researchers work on the issue of bandwidth brokers, dynamic QoS provisioning and inter domain operation. Many research papers have been presented, focusing on various models and architectures (Braun and Khalil 2001), (Hwang and Revuru 2003), (Zhang et al. 1999).

This paper presents a summary on the latest architectures on this issue, an overview of our implementation in NS-2 (using a distributed model) and finally the module for the inter–domain operation. The rest of the paper is organized as follows. Section 2 describes the bandwidth brokers and their usage, while section 3 presents the architecture of an existing bandwidth broker that we have implemented in NS-2 and operates in single domains. Section 4 presents the topic of the inter-domain operation, focusing on all the related issues regarding the successful end to end provisioning and efficient utilization of the resources. Next, section 5 describes the design of the adaptation of the existing bandwidth broker in order to operate in

a multi domain mode, using SLAs and keeping its operation secure. Finally, section 6 describes the conclusions and the future work that we intend to do in this area.

2. Bandwidth broker's architecture

A bandwidth broker is a "service" that provisions a backbone network and manages the supported QoS services. Actually, a bandwidth broker has many interfaces as it receives demands for QoS services that it manages; it processes those demands and decides if it can satisfy them. In case that the answer is positive, the bandwidth broker configures the network devices (routers, switches etc) to provide the bandwidth guarantees. The process of each demand needs information from the network configuration - condition as well as from the network's policy and management as the existing SLAs etc. Service Level Agreements (SLAs) are generally contracts between network service providers (called interdomain or inter-ISP SLA). The interconnecting networks negotiate SLA with respect to the services to be provided at the network boundary. The subset of SLA that provides the technical specifications such as QoS, aggregate traffic profile, and PHB resource allocation is referred to as the Service Level Specification (SLS). An SLA and SLS can be static or dynamic. Static SLSs are negotiated on a long-term regular basis (monthly-yearly). Dynamic SLS implementation requires a network aware middleware agent, and capable of signaling to update the interconnection SLS dynamically (by minute or hour). In modern networks, such tools are developed, allowing end to end provisioning and dynamic SLA negotiation.

A bandwidth broker is also responsible for the inter-domain communication with the bandwidth broker of adjacent domains. This procedure is quite complicated, as it requires direct communication between the 2 adjacent BBs and also a special agreement between the 2 domains that should be taken into consideration. The communication between the bandwidth brokers of 2 domains may include the exchange of signaling, routing or SLA information. The exact information that is exchanged depends on the model and the architecture of the implemented bandwidth broker. In each domain there is a Bandwidth broker entity that manages the domain's devices and services. A bandwidth broker contains several modules that are necessary for its transparent and efficient operation; an inter-domain interface. It is used for communication with adjacent BBs, an intra-domain interface that is used for communication with the service components that are located inside the domain that the BB controls, a routing table interface that is used so that the BB knows the network topology and the routing paths, a user/application interface that allow the user and applications to send requests to the BB, a policy manager interface for the implementation of complex policy management or admission control and a network management interface that is used for coordination of network provisioning and monitoring.

3. Existing architecture of Bandwidth Broker

Such a bandwidth broker that operates in single domain only has been implemented in NS-2 simulator (Zhang et al. 1999), (Sander 2000), following the generic architecture and is consisted of various modules. This bandwidth broker consists of many "devices" in the network that belongs to 2 basic categories: the clients and the basic bandwidth broker server. A client operates on every node of the network and makes the communication with the base server. This client is responsible to manage the local links, provide the interface to the user and applications to make new requests and also update its local routers with the configuration modifications according to new admissions. This client also stores data regarding the adjacent nodes of the node and communicates with the base BB every time the base BB needs this

information. On the other hand, the base bandwidth broker is responsible to synchronize the whole operation and inform accordingly all the clients. It also keeps statistics for the requests and the whole operation, like the active nodes etc. So, the architecture is somewhat distributed as every "client" executes some threads locally and stores some information there, but always synchronized by the base bandwidth broker server.

The system's operation begins when an Edge Bandwidth Broker makes a request asking prioritization for given traffic class with rate of x bps from the node the client is running to some other network node. Then, the Base Bandwidth Broker searches the routing tables to find the route from node n0 that made the request to the other end-node nk. Next, the Base Bandwidth Broker sends a query to the Edge Bandwidth Broker clients that run on nodes who manage the links across the routing path, asking if there is available bandwidth. If all the answers are positive, means that there is available bandwidth from node n0 to node nk, the Base Bandwidth Broker will send a positive answer to the source. The procedure will be completed after the Base Bandwidth Broker sends to all the Edge Bandwidth Broker clients that lay on the path n0,n1,…,nk, messages informing them to make the bandwidth reservation on the network devices.

3.1.Provisioning model

The implemented bandwidth broker manages a DiffServ based QoS service (Black et al. 1998) that tries to provide bandwidth guarantees as well as minimum delay and jitter. This service is well known as IP Premium and can be supported in IP or either MPLS domains. The requests that the bandwidth broker can handle, should have declared the 2 end points (source and destination aware). The implementation of the service is based on a strict filtering and policing of the traffic in incoming interface and then proper marking. The packet marking is done in DSCP field of IP packet header, but the service can be easily implemented on MPLS domains too, as the MPLS marking is proportional to DSCP marking. In particular, every time an IP packet inserts the MPLS domain, then during the assignment of the MPLS label, the value of the IP precedence (3 most significant bits of DSCP) are copied on MPLS Experimental field. Next, all the authorised traffic is forwarded on all routers through high priority queues that are activated by default. Finally, the traffic is served to the destination experiencing the minimum delay and jitter that the DiffServ network can achieve.

This bandwidth broker, supporting the IP Premium service, has been implemented in ns-2.26, where several enhancements have been done in order to support the necessary mechanisms (Ceid DiffServ NS web site, 2006). The operation of the bandwidth broker has been tested through a number of simulation tests and the overall performance is very positive. Also, an evaluation of the impact of distributed nature of the bandwidth broker has been done and the overall conclusion was that it works fast but its speed can be accelerated if the distribution of the clients and the base bandwidth broker are optimal (Bouras and Primpas, 2005).

3.2. Data structure

The bandwidth broker uses 3 separate data structures, where 2 of them are maintained by the base bandwidth broker and the last one on each client. In particular, every client maintains a data structure where keeps information regarding the links (backbone or access) that provisions and manages. This structure keeps a link id, the bandwidth of the link and the maximum allowed reservation. The latter is the result of the dimensioning algorithm that has been used in the network. Additionally, the structure keeps the current reserved bandwidth as

well as an id on each QoS request that has been submitted and is routed from this link. The basic bandwidth broker has 2 other structures too, the first one for the QoS requests and the second for the synchronization of the clients. The first maintains a record per QoS request where it saves the id, the 2 end points, the requested bandwidth, the time period for the request and the result of the admission algorithm (accepted or rejected request). Finally, the structure stores the routing path of each admitted request as well the DSCP marking values. At last, another data structure maintains information for the bandwidth broker's operation. In particular it stores the relationship of each link of the network and the address of the local client bandwidth broker that is responsible for this link. In case a link does not "participate" in the whole operation, which can be caused by various reasons, then the address of the responsible client is null.

4. Extension to multi-domains

When a request is submitted, it should at least contain source and destination addresses and also the requested bandwidth. Next, the basic module of the bandwidth broker should parse the source and destination addresses and try to identify the sub- networks where these addresses belong to. The source naturally resides in the ISP's network, the destination might well be in another ISP's domain. That domain might not be the next connected domain to the ISP and there might be one or more domains in between. If both the source and destination addresses are in the stub networks of the same ISP domain, the Broker that maintains the domain can find the ingress and egress routers by some simple lookup in the related databases (as explained above at the operation of the distributed bandwidth broker). If the destination is in domain other than the source domain, then the Broker must identify the final domain that has the destination IP addresses. In addition and depending on the architecture of the inter-domain protocol, the intermediate domains (if any) that will be traversed by the traffic to the destination.

The most important and challenging issue during the processing of such requests is the investigation of the best path for the end to end communication, taken into account the possible SLAs between domains as well as traffic engineering characteristics. This is a challenging problem in case there is at least 1 intermediate domain and for this purpose there are 3 models that the bandwidth broker can follow. The first one is called centralized and according to this model, the decision about the routing of the request, in particular the domains that the traffic will traverse in order to reach the destination domain is made by the source domain. In order to take this decision, every bandwidth broker should have access to a database that stores the topology and all the peering agreements (SLAs) between ISPs etc. Next, it makes "queries" asking for the requested resources across the paths. It makes this procedure sequentially until it finds all paths that can admit the request. Next it decides regarding the best one according to the criteria that have been specified in the bandwidth broker.

The second model is "peer to peer" and actually the source domain sends the requests that have destination to another domain in every adjacent domain, under the condition that there are available resources from source to the respective egress points. Next, every domain receiving such a message checks if it is the destination domain or it is immediately attached to it. If the answer is positive then the bandwidth broker checks if it can guarantee the requested resources form the ingress point to the destination of the egress point to destination domain. If there are the necessary resources, then the message is forwarded there. In case that the domain is an intermediate and not attached to the destination, then it broadcasts the request to all

adjacent domains, which it has the requested resources to reach them (from its ingress point to the egress point to next domain). From the broadcasting have been isolated the domains from which it has received the message in order to avoid loops. This procedure is repeated and finally all the possible routes between the source and destination are declared to the source domain. Then, the source domain decides the best routing according to specific criteria that have been declared in the routing model. Those criteria can be the minimum path, or the minimum SLAs fulfilment or the path that leads to better load balancing in the network.

Finally, the third model is the simplest one, as it finds the routing path according to normal routing (through a traceroute command). This model has the drawback that does not take into account traffic engineering issues and alternative paths that may have available resources instead the normal path that may be congested.

The first two models are actually the most efficient but also the time and process consuming. Both models need updated information for the network's condition and its policy. The first algorithm requires every domain to announce those information centrally (in a database), where all the other domains "query" it to process inter-domain requests. The second algorithm eliminate this situation (announcement of network's condition and operation), by engaging the bandwidth broker server of each domain to answer about the possible paths with available resources. Therefore the operation of each network remains secret and every request that may pass through a domain is processed by domain's bandwidth broker. On the other hand, the second algorithm has a bigger complexity and also needs more time to answer a request as all the possible routes should be checked by asking domains' bandwidth brokers. Therefore, the response time includes the transmission delay of the requests for path finding between the bandwidth brokers of the domains and their execution time locally in each domain. The first algorithm has a big advantage in this issue, as all the information about all domains and their operation is stored locally in a database and therefore all the alternative routing paths can be found by searching the new graph that is produced by adding all domain's topology, the available resources as well as the SLAs between adjacent.

As a conclusion, the first algorithm has better response time but needs to announce internal information periodically to keep the global "network database" updated. The second algorithm keeps that internal information secretly but the process of every request engaged all the bandwidth brokers of the involved domains instead of the bandwidth broker of the first domain that process the request in the first algorithm.

4.1.Structure of exchanged messages for requests

In order to maintain the peering relationship between the bandwidth brokers of adjacent networks and find the best routing paths, special messages are exchanged. Those messages are created and handled only by the bandwidth brokers (using specific format). Message takedown is accomplished via an RAR/RAA pair (He et al. 2004). A RAR message is used to format the request, and is transmitted across the domains. A RAA message is used to acknowledge the request message. Special care is necessary to be taken on security issues, as possible attacks may destroy the existing SLAs, degrade the whole performance of the domains or simply lead to a theft of service. Therefore, inter-domain resource provisioning is achieved through Simple Inter domain Bandwidth Broker Signaling (SIBBS) protocol (He et al. 2004), (Bouras and Stamos, 2006), (Sander, 2000). SIBBS provides a flexible mechanism for aggregating signaling messages among the domains. It was created by Internet2 community and the main goals are to maintain the integrity of the SIBBS messages and also

to provide mutual authentication between the adjacent bandwidth brokers. The SIBBS protocol uses the PKI model and each SIBBS message is signed with the public key of source bandwidth broker. The authentication in the receiver is done using the signature of the source in conjunction with the id of the source bandwidth broker.

5. Adoption of multi domain architecture to existing implementation

According to the distributed nature of the existing implementation and also the freedom that every domain should have on deciding its local policy, the most capable model to implement is the peer to peer model. Therefore, the exchanged information between domains should be adapted to this model. The basic algorithm that should run in this case is the following:

- Each domain has established SLAs with its adjacent domains.
- A domain receives a request and then checks locally for the requested resources from the ingress point (the previous domain) to all the interconnection points that it has. Next, for every local check (to an adjacent domain) that is successful, it sends a request to the bandwidth broker of this domain.
- The next domain makes the same checks. The final domain returns an answer or an intermediate domain sends back a negative answer if it can not provide the requested guarantees.
- Finally, the source waits answers from the adjacent domains that it has forwarded a request.
- Then, it examines all the answers (as each domain can provide several routing paths to the destination) and it selects the one with the laziest SLA fulfilment. The definition of the criteria for the selection of the best path between the alternatives is a quite flexible operation. For our distributed implementation we use the laziest SLAs fulfilment as it leads to load balancing on the interconnection links and the peering information. That load balancing is not extended internally in the domains, but we made that decision as the internal domains usually has alternative paths or more resources that their interconnection links that usually have high cost and their upgrade is more difficult. But in any case, the criteria for the best path can change periodically, depending on the whole network's (including all domains) condition. This can be done through an adaptation model that can recognize such situations and change the criteria for the selection of the routing path.

Generally, this algorithm has a significant large complexity, as it checks every possible routing path and this complexity increases as the interconnection links between domains increase. The problem of inter-domain requests of a bandwidth broker mostly resides to efficient constrained based routing, a problem that has been studied by many researchers (Hwang and Revuru, 2003), (Agarwal et al. 2003). In our case this problem can be modelled as a graph (see Figure 1) where the nodes are the domains of the whole network and the links are the established SLAs. Note that the real interconnection links that do not "follow" an SLA, are not taken into account. Finally, the problem is to find the best path from the source domain to the destination domain without brake any existing SLA.

Also, a very important point is in case that all answers are negative and it is due to SLAs and not to internal domain's resources. Then, the source bandwidth broker identifies (from the answers) those SLAs and asks from the relevant domains if they can make dynamic negotiation in order to fulfil the requests. The selection of the SLA that should be "queried" for dynamic negotiation should be based on the minimum violation. In particular, the bandwidth broker should sort all the paths based on minimum SLA violation (in SLA characteristics) and next it must examine each path, finding the violated SLAs and asking for

dynamic negotiation until it has a successful path or all paths are negative. After the admission of a new request, the inter-domain operation only needs to inform the bandwidth brokers of the involved domains across the routing path. Then those bandwidth brokers inform the clients that are responsible for the relevant network nodes and links that update their status and also perform the configuration of the routers, according to the technology that each domain uses.

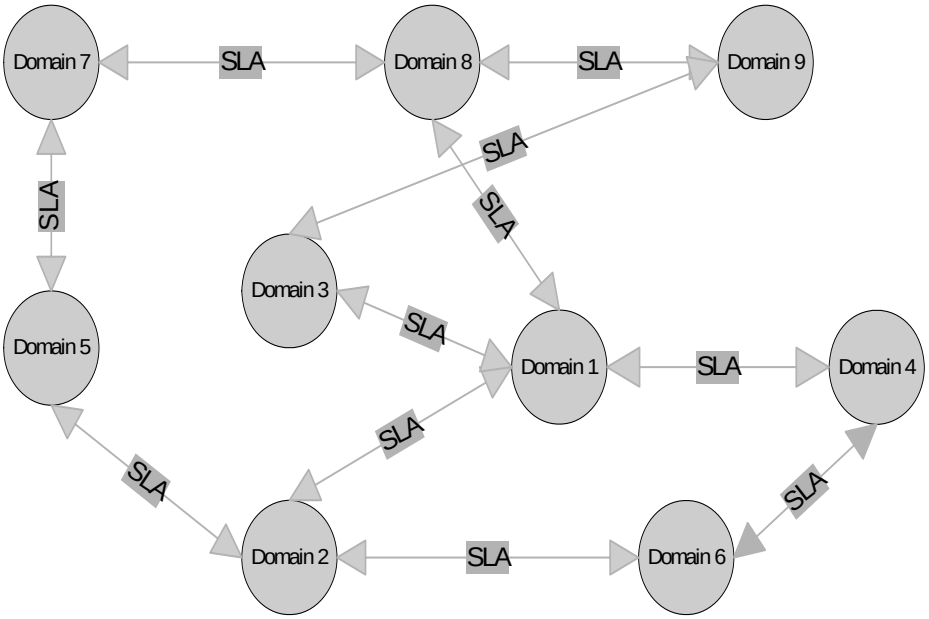

Figure 1: the graph's representation of the problem

5.1.New modules and messages

Our implementation in NS-2 simulator needs some more enhancements in order to support inter-domain operation and the algorithms as presented in the previous section. In particular 3 new modules should be added. The first one should be the main inter domain module that handle all the communication with adjacent domains. This module should be able to create RAR messages targeted to adjacent domains, as well as perform all the operation in the peer to peer model as it was described above. In addition, this module should be able to recognize RAR messages that it receives and process them. Additionally, the dynamic SLA negotiation is an operation that needs an extra module. In particular, the basic class for the inter domain operation should investigate the cases that a request can be admitted if 2 domains make a dynamic negotiation and agree in a higher rate of exchanged traffic. In order to support the dynamic negotiation, the basic module should send a notification to the 2 domain asking for such a negotiation. This will be done by adding in NS-2 a new message type that carries this information. Next, every bandwidth broker will run a new module that will recognize such messages (for dynamic negotiation) and will start this procedure. The procedure will be started by one of the domains in case they have the resources on their interconnection link and also the local policy of the domain is not violated. The information for the local policy is stored in local base bandwidth broker data structure and should be extracted from there. When this negotiation is finished (positive or negative), then the 2 domain will inform the source domain that asked the dynamic negotiation. Finally, a new data structure for keeping track of the inter domain operation should be added on base bandwidth broker class. This data structure should be proportional to the data structure for single domain operation and should enhance it with information about the intermediate domains, the selection of the best path and the possible requests for dynamic SLA negotiations for a request.

6. Conclusions – Future Work

This paper deals with the inter-domain operation of bandwidth brokers in order to perform end to end provisioning and therefore end to end guarantees through QoS services following DiffServ architecture. Today's networks do not have yet automatic procedures to provide such functionality, but many approaches are under design or testing. In the near future, it is expected that such bandwidth brokers (Qbone web site, 2005), (BMP web site, 2005) will be introduced in network's operation, automating the QoS services and reducing the management cost. The paper presents the relevant aspects for inter-domain operation of a bandwidth broker and describes the 3 models that can be used (the source based, the peer to peer and the normal routing). Also, the paper presents shortly a distributed implementation of a bandwidth broker in NS-2 and finally the enhancements that should be done in order to support inter-domain operation, following the peer to peer model. The most important point in the inter-domain operation is the selection of the optimal path across all domains to serve every request. This problem refers to optimal constrained based routing, but with some particular characteristics as the existence of SLAs between domains that can be dynamically negotiated.

Finally, we already have plans for future work in this area that mainly focuses on the finalization of the implementation of the inter-domain operation and dynamic SLA negotiation. Also, we intend to perform a number of tests in order to investigate the whole operation and performance. Finally, we plan to integrate the bandwidth broker with MPLS traffic engineering characteristics, provided by RSVP-TE protocol.

7. References

Black D., Blake S., Carlson M., Davies E., Wang Z., Weiss W., (1998), "*An Architecture for Differentiated Services*", RFC 2475

Braun T. and Khalil I. (2001) "*Implementation of a Bandwidth Broker for Dynamic End-to-End Capacity Reservation over Multiple Diffserv Domains*", 4th IFIP/IEEEInternational Conference on Management of Multimedia Networks and Services (MMNS)

He Y., Lee B., Lim T.-M., Lim B.-H., Song J., Yeo C.-K., Woo W.-K., (2004), "*Secure Communications between Bandwidth Brokers*", Operating Systems Review 38(1)

Hwang J., Revuru R. (2003) "*Inter-Domain Diffserv Dynamic Provisioning and Interconnection Peering Study Using Bandwidth Management Point - A Simulation Evaluation*", International Conference on Information Systems and Engineering

Zhang L., Ogawa J. O., Terzis A., Wang L., (1999) "*A two-tier Resource Management Model for the Internet*", IEEE Global Internet.

Agarwal S., Chuah C., Katz R., (2003), "*OPCA: Robust Interdomain Policy Routing and Traffic Control*", IEEE Openarch

Sander V., (2000), "*The security Environment of SIBBS*", (http://qbone.internet2.edu/bb/SIBBS-SEC.doc)

Bouras C., Primpas D., (2005), "*An admission control and deployment optimization algorithm for an implemented distributed Bandwidth Broker in a simulation environment*", 4th International Conference on Networking – ICN 2005, pp. 766 - 773

Bouras C., Stamos K., (2006), "*Securing a Bandwidth Broker Architecture*", International Conference on Internet Computing (ICOMP 05)

Ceid DiffServ NS web site (2006), "*DiffServ Quality of Service in NS*", (http://ru6.cti.gr/diffserv-ns/default.htm), (Accessed 24 March 2006)

Qbone web site (2005), "*QBone Bandwidth Broker Architecture*", http://qbone.internet2.edu/bb/bboutline2.html (Accessed 15 July 2005)

BMP web site (2005), "*Bandwidth Management Point (BMP)*", http://imint.syr.edu/bmp/bmp.html, (Accessed 15 July 2005)