An Admission Control and Deployment Optimization Algorithm for an Implemented Distributed Bandwidth Broker in a Simulation Environment

Christos Bouras and Dimitris Primpas

Research Academic Computer Technology Institute,
61 Riga Feraiou Str., 26221 Patras, Greece
Department of Computer Engineering and Informatics,
University of Patras, 26500 Rion, Patras, Greece
Tel: +30-2610-{960375, 960316} Fax: +30-2610-960358
{bouras, primpas}@cti.gr

Abstract. This paper describes and tests a distributed bandwidth broker that has been implemented in NS simulator. It focuses on the admission control algorithm, its advantages and drawbacks. Also, the bandwidth broker is tested, managing the IP Premium service and we compare 2 different implementations of the service. Finally it approaches the problem of the optimal location of a bandwidth broker in a backbone network. For this purpose, a new model is proposed that evaluates each node and finally selects the most capable node where the base bandwidth broker should be located.

1 Introduction

A bandwidth broker [1][2] is an entity that operates in a backbone network and is responsible to manage QoS service. Actually, it receives demands for bandwidth allocation; it processes them 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. This area is a widely open research issue, where several research team works on. There are many scientific papers on this area, where several architectures and algorithms have been presented [4][5][6][7].

We have implemented such a bandwidth broker in a simulation environment. The bandwidth broker as it has been implemented follows a generic architecture and is consisted of various modules. Those modules are: an admission control module that also contains a decision module, which describes the algorithm that runs in order to check each request. Additionally, the admission control module has a second module that stores all the necessary information for bandwidth broker's operation and also updates them whenever it is necessary. Besides, there is a module that is available to end users to make their requests. Finally, the implemented bandwidth broker has a module that describes the QoS service that it supports (classification, queue and scheduling algorithms etc) and it configures the network devices accordingly in each accepted demand. The bandwidth broker has been implemented using an independent implementation of each module and now it will be tested in order to evaluate its performance.

P. Lorenz and P. Dini (Eds.): ICN 2005, LNCS 3420, pp. 766–773, 2005.

The rest of the paper is organized as follows: Section 2 has a description of the implemented bandwidth broker focusing on the admission control algorithm. Section 3 presents the QoS service that the bandwidth broker manages and describes the simulation tests that we performed, comparing 2 alternative implementations of the same IP Premium QoS service (using different queue management mechanisms). Next, section 4 approaches the problem of the selection of the optimal node to host the bandwidth broker where we propose a new model that selects the best node using various criteria. Finally, section 5 describes our conclusions as well as the future work that we intend to do on this area.

2 Bandwidth Broker Implementation in NS Simulator

Simulation has always been a valuable tool for experimentation and validation of models, architectures and mechanisms in the field of networking. It provides an easy way to test various solutions in order to evaluate their performance without needing a real network dedicated for experiments. In our case, a bandwidth broker has been implemented and tested on simulation environment (NS-2 [12]) in order to evaluate its performance characteristics and its used mechanisms.

The implemented bandwidth broker [11] on NS-2 followed the classic architecture of a bandwidth broker. The implementation required several changes and additions in the NS structure and source code. In particular, the bandwidth broker that was implemented is based on two new agents, the Edge Bandwidth Broker (BBedgeAgent) and the Base Bandwidth Broker (BBbaseAgent). BBbaseAgent creates BBB packets and consumes BBE packets created by the BBedgeAgent. BBedgeAgent creates BBE packets and consumes BBB packets created by the BBbaseAgent. A BBedgeAgent, which represents a client (user) can send a RAR requesting guaranteed bandwidth between the node it is running and another node The BBedgeAgent that exists on every node simulates a situation where a BB client is connected to a router on a real network. This agent operates as client that makes the communication with the base BB and updates its local router with the configuration modifications according to new admissions. In our case, this agent also stores data regarding the adjacent nodes of the node and communicates with the base BB every time the base BB needs this information. So, the architecture is somewhat distributed as some information is stored locally on every "client" and not centrally on the base BB.

2.1 The Admission Control Algorithm

A very important module in a bandwidth broker is the admission control. There are several algorithms that has been proposed for efficient admission of requests [8][9][10]. But in our case, where the operation is distributed, we designed and implemented a simple distributed admission control algorithm, where the base bandwidth broker agent runs only the main part of the algorithm, in order to ensure the coordination and the proper whole operation.

The system's operation begins when an Edge Bandwidth Broker makes a request asking guaranteed bandwidth of x bps from the node it is running to some other network node. Then, the Base Bandwidth Broker begins to serve the request by running the admission control. It searches the routing tables to find the next hop from the node n0 that made the request to the other end-node nk. Then, the Base Bandwidth Broker sends a query to the Edge Bandwidth Broker that runs on node n0 asking if there is available bandwidth between the nodes n0 and n1. If the answer is positive, the Base Bandwidth Broker finds the next hop n2 from node n1 to node nk and sends a query to node n1 asking if there is available bandwidth between the nodes n1 and n2. If all the answers are positive, this procedure continues until node nk is reached. This means that there exists available bandwidth from node n0 to node nk and the Base Bandwidth Broker will send a positive answer to the Edge Bandwidth Broker that made the request so that node n0 is notified that it is allowed to begin sending data. The procedure will be completed after the Base Bandwidth Broker sends to all the Edge Bandwidth Brokers that lay on the path n0,n1,…,nk, messages informing them to reduce by x bps the available bandwidth to the links that lay on the path. In case one of the Edge Bandwidth Brokers sends a negative answer, because there is not sufficient available bandwidth on a link, the Base Bandwidth Broker sends a negative answer to the node that made the initial request and the procedure ends there.

Sequentially, after the successful admission of a new request, the bandwidth broker should run the resource allocation module that configures properly the backbone routers across the path to provide the admitted guarantees.

2.2 Advantages and Disadvantages of Admission Control Algorithm

The implemented admission control algorithm has many advantages and some drawbacks. In particular, this module (admission control) is operated distributed, as parts of this algorithm run in the clients and a part and the basic synchronization in the base agent. Also, this admission control algorithm only needs simple data structures in the base and edge bandwidth broker agents. Each edge bandwidth broker must store information only for its links that manages. Initially, this information is the maximum bandwidth of the link that is available for the QoS service and the reserved bandwidth. The maximum available bandwidth for reservation on the link is determined by the network dimensioning. On the other hand, the base bandwidth broker agent needs to store more information as the nodes that are managed by the bandwidth broker, the links that each node manages and some data structures that should be used during the processing of every request. The nodes that participate in the bandwidth broker operation can be stored using only an array that should be updated each time a node introduce itself in the bandwidth broker operation or delete itself from the bandwidth broker. Also, this makes the algorithm highly extensible due to the fact that the necessary information for a new node and link is stored locally (in the client agent) and therefore, the bandwidth broker operation can cover new nodes simply when the new node (client agent) introduce itself by an appropriate message. Finally, during the process of every request, the base bandwidth broker has access to network modules, as the routing tables (routing information) and uses temporarily (for the process of each specific request) some information from there.

The drawback of this algorithm is that is works based on the current routing schema and does not provides any kind of load balancing that might be necessary when it operates in a large backbone network. In particular, the base bandwidth broker uses the classic OSPF routing protocol that is configured normally (uses the classic Dijstra algorithm that calculates the minimum path without using costs for the edges). Therefore, this module might lead to rejection of requests in case that the basic minimum path is full and alternative paths are not taken into account. This problem can be solved by running an optimization algorithm when the network approaches such situations. This optimization algorithm can run additionally in bandwidth broker's operation, reconfigure periodically the admitted requests and examine again the rejected requests searching for alternative paths. Such an optimization algorithm is in our future plans to implement. The basic idea of the algorithm is to reroute some of the admitted requests from alternative routing paths, when of course the guaranteed bandwidth and delay characteristics are satisfied.

Also, the admission control algorithm exchanges many packets of 64 bytes (from the base bandwidth broker agent to the edge bandwidth broker agents and vice versa) that are crucial for the whole operation. These packets use TCP transport protocol and therefore their transmission is as secure as possible. Also the packets have been marked appropriately to use the high priority QoS service in order to achieve minimum delay and jitter and therefore accelerate the whole operation of the bandwidth broker. The general responding time of the admission control module depends on the request parameters (how far in the topology is the 2 edge nodes of the request) and also on the location of the base bandwidth broker as it coordinates the whole operation. Therefore, in cases where the base bandwidth broker is located on a node that is included in the routing path between the 2 nodes, the packets that should be exchanged traverse less links and the processing time is reduced accordingly. This problem, of the most suitable location of the base bandwidth broker (in that distributed operation), is approaches in section 4, where we propose a model that can select the node that should host it.

3 Description and Testing of Bandwidth Broker's QoS Service

The implemented bandwidth broker manages a QoS service (the IP Premium) that tries to provide bandwidth guarantees as well as minimum delay and jitter. The original ns-2.26 [12] functionality supports a limited number of features for packet classification and queue management, therefore, we have already enhanced the simulator with additional functionality [3][13] in order to simulate the IP premium service's operation. In particular, the classification is done using the DSCP field of the IP header and also we implemented the Modified Deficit Round Robin Scheduling Algorithm (MDRR) [3] and changed the whole queue management mechanism to enqueue packets based on DSCP. The QoS service, as it has been implemented, classifies the packets for each class that has been admitted by the bandwidth broker with DSCP value 46. Then, when the packets are inserted in the network, we apply strict token bucket policy in order to be sure that the transmitted rate agrees with the admitted rate. Next, on all the network nodes, the queue management mechanism is properly configured. The used queue management mechanism is a high priority queue on every node that is used for all the admitted traffic classes. Additionally, instead of priority queueing, the MDRR mechanism can be used.

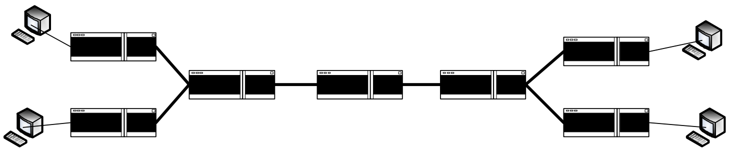


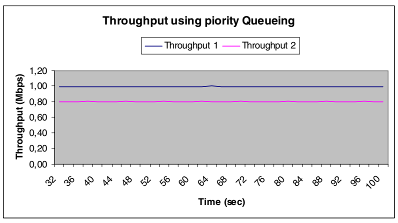
Fig. 1. The network topology

We conducted several tests aiming to evaluate the bandwidth broker's operation when the QoS service (IP Premium) is implemented using the Priority queueing mechanism first and after the MDRR. The topology that was used for those experiments is presented in **Fig. 1**. Each router has an edge client operating locally and also the middle one also contains the base bandwidth broker agent.

3.1 Testing the BB Using the Priority Queueing Algorithm

The bandwidth broker has been configured to manage the IP Premium QoS service, implemented using the priority Queueing as the queue scheduling algorithm. In this case, we performed a set of tests to investigate the operation and finally the guarantees that can provide. For this purpose, the measures that are performed are concentrated on the achieved throughput, delay and jitter. Therefore, we simulated the scenario were the backbone links are all 10Mbps and the bandwidth broker manages 2Mbps on each link for QoS requests. At this point, 2 sources requested 1Mbps and 800Kbps respectively and were successfully admitted by the network as the total bandwidth was available. Finally, the throughput that the 2 flows experienced was exactly the requested and the packet's delay was extremely low.

[1]



[2]

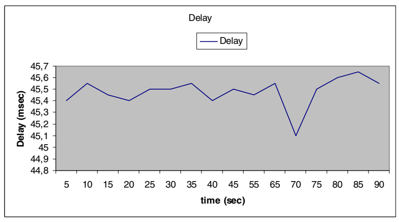


Fig. 2. Throughput and Delay using the IP Premium service with Priority Queuing

3.2 Testing the BB Using the MDRR Algorithm

The MDRR is an alternative queue scheduling mechanism that can provide various operations as it has many characteristics. The bandwidth broker has been tested to evaluate the operation of the IP Premium QoS service using MDRR. The topology that was used is again the topology presented in Fig. 1 and the measurements also focus on the achieved throughput, delay and jitter. The final results are the same as in Priority Queuing for the throughput but the delay was measured a little lower as Fig. 3 shows.

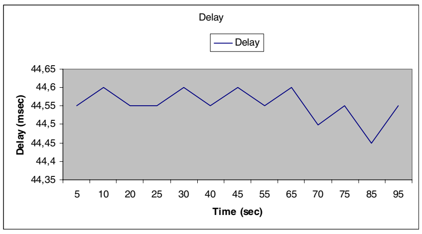
[1] 

Fig. 3. Delay using the IP Premium service with MDRR

Comparing the results from the experiments with the two mechanisms, it is obvious that the bandwidth broker manages very well the IP Premium service that provides the absolute guarantees either with MDRR either with Priority Queueing. The only noticeable difference is that the delay is a little bit smaller when the IP Premium service is provided using the MDRR mechanism. In order to take a decision about the implementation of the IP Premium service and next test it in a backbone network, we should take into account other advantages of the above mechanisms. In this case, the MDRR mechanism seems more powerful than Priority Queueing, due to the fact that except from a high (strict) priority queue, it can support many other queues that can guarantee specific bandwidth (without delay and jitter guarantees).

4 Optimization of Bandwidth Broker's Operation in a Backbone Network

A very important point in the operation of a bandwidth broker is to decide which node should host the base bandwidth broker agent. This decision is more crucial for the efficient operation of the implemented bandwidth broker, when the operation is distributed and the base bandwidth broker agent communicates with all the clients collecting information from the processes that are executed there. In addition, the selection of the location of the base bandwidth broker agent should also take into account the traffic that pass through each node, the importance of each node etc. For this reason, we tried to approach this problem by creating a model that evaluate each node and the adjacent links and according to the weights tries to find the best node to locate the base bandwidth broker. In other words, the problem is to find the root of the graph, where the root is the most important node in the network and most of the packets for the operation of the bandwidth broker will reach it quickly, without passing many links.

This model uses 6 criteria to evaluate the importance of each node in the network operation that are:

- Users. It represents the number of sub-networks and therefore the number of the users that are connected in this node.
- Node equipment. This criterion approaches the capabilities of the specific node. In particular, the grade for this arises from the evaluation of the technology of the routers and the technology and capacity of the backbone links on this router.
- Adjacent nodes. This criterion specifies the importance of the node, taking into account the number of backbone links that are connected on this router.
- Servers. Each node is evaluated by the number of the servers that are connected on it and runs critical and famous services of the network. Except from servers, they can be GRID clusters, VoIP gateways, gatekeepers or any other machine that implies that there is strong possibility for many requests targeted in this node.
- Routing. In this case, the node is evaluated for its importance in the whole routing in the network.
- Interconnection. Finally, the last criterion is used for the condition that this node is an interconnection point with a bigger backbone network and therefore there will be requests from the adjacent bandwidth broker.

Each criterion should be evaluated in the scale from 1 to 10. The evaluation should be done in the same time for all the nodes and the gradation in each one should be analogical. Finally, the weight of each node arises as the sum of all the criteria. In case that there are 2 or more nodes with the same weight, the criteria are taken into account with the following order: Routing, Interconnection, Servers.

Next, for each node, we create the “routing” graph for this node to all the others in the network. In particular, we place each node as root and we create all the paths to all the other nodes, using the network’s routing scheme. Therefore, there are N graphs (where N is the number of nodes in the network) that should be examined. Then, we define a new metric for every node, called “special-weight” of node that arises as the weight of this node (that was produced by the above criteria) multiplied with its depth in the graph. In this case, the root of each node has “special-weight” equal to 0. Next, the “special-weight” of the whole graph is the sum of the “special-weight” of all of its nodes. Finally, the problem is to find the graph that has the minimum “special-weight”. We run this model for all the nodes, we create all the N graphs and calculate the “special-weight” for each one. Then, we select the graph that has the minimum calculated “special-weight” and the node that is graph’s root is the node that must host the base bandwidth broker.

5 Conclusions – Future Work

This paper deals with the Bandwidth broker idea and its operation. It focuses on the distributed admission control algorithm that we implemented, mentioning the advantages and drawbacks that we noticed. Also, the paper describes the IP Premium QoS service that the bandwidth broker manages, which we tested with 2 alternative implementations, using the Priority Queuing and the Modified Deficit Round Robin.

The results showed similar behaviour for both mechanisms and also both achieved the requested guarantees. Finally, we tried to approach the operation of a distributed bandwidth broker in a backbone network where there is specific routing schema and also the nodes have different importance (due to the sub networks and the services that they run). There, the most crucial problem that we faced is where the base bandwidth broker should be located, as it affects the efficiency of its operation. Therefore, we propose a model that evaluates the importance of each node, taken into account several parameters and finally select the most suitable node to host the bandwidth broker.

The simulation tests as well as the algorithms that we propose indicated some points for further investigation. Therefore, we have plans for future work that mainly focuses on the simulation and mathematical evaluation of the proposed "host selection" model in order to optimize it. Also, we plan to study and implement an optimization algorithm that will extend the existing admission control algorithm, in order to provide load balancing.

References

1. RFC 2475 "An Architecture for Differentiated Services", S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, December 1998
2. RFC 2905 "AAA Authorization Application Examples", J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence, August 2000
3. C. Bouras, D. Primpas, A. Sevasti, A. Varnavas "Enhancing the DiffServ architecture of a simulation environment", 6th IEEE International Workshop on Distributed Simulation and Real Time Applications, Fort Worth, Texas, USA, October 11 – 13, 2002
4. Manzoor Hashmani and Mikio Yoshida "ENICOM's Bandwidth Broker", Saint 2001 Workshops, pp 213-220, Jan 8-12, 2001, San Diego, USA
5. Bandwidth broker report, University of Kansas
6. QBone Bandwidth Broker Architecture, http://qbone.internet2.edu/bb/bboutline2.html
7. Active Resource Management (ARM) For The Differentiated Services Environment, http://www.caip.rutgers.edu/TASSL/Projects/Adaptive_QoS/ananth/bandwidth.html
8. Przemyslaw Jaskola, Krzysztof Malinowski "Two methods of optimal Bandwidth allocation in TCP/IP networks with QoS differentiation", 2004 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS' 04), San Jose, California, USA, July 25 - 29 2004
9. C. Bouras, K. Stamos, "An Adaptive Admission Control Algorithm for Bandwidth Brokers", 3rd IEEE International Symposium on Network Computing and Applications (NCA04), Cambridge, MA, USA, August 30 - September 1 2004
10. L. Burchard, H. Heiss, "Performance Evaluation of Data Structures for Admission Control in Bandwidth Brokers", April 2002
11. C. Bouras, D. Primpas, K. Stamos, N. Stathis, "Design and implementation of a Bandwidth Broker in a simulation environment", 7th International Symposium on Communications Interworking - INTERWORKING 2004, Ottawa, Canada, November 29 - December 1 2004
12. http://www.isi.edu/nsnam/ns/ns-build.html
13. http://ouranos.ceid.upatras.gr/diffserv/start.htm