Active Resource Management For The Differentiated Services Environment*

Ananthanarayanan Ramanathan and Manish Parashar
Department of Electrical And Computer Engineering,
Rutgers University, Piscataway, NJ – USA.

Abstract: *This paper presents a mechanism for active resource management (ARM) in a differentiated services environment. While the differentiated services architecture and the bandwidth broker agent provide a mechanism for QoS management through resource reservation, this mechanism is based on a static provisioning of resources. As bandwidth requirement are typically dynamic, such a static reservation approach can either lead to wasted bandwidth or leave applications, resource starved. The active resource management approach presented in this paper addresses this problem by using active networking techniques to dynamically reallocate resources based on the current state and requirements of the network. An implementation and evaluation of ARM using the NS-2 simulation toolkit is presented.*

Keywords: *Active Resource Management, Quality of Service, Bandwidth Brokers, Differentiated Services.*

1. Introduction

A large percentage of the traffic on the Internet today is either multimedia related or is some form of real time data that is critical to an application. Such time-critical data requires some level of service guarantees. The Internet Protocol (IP), however, is based on best effort and lacks the capability to provide any Quality of Service (QoS). Various solutions have been proposed to address this problem by guaranteeing required resources. These include RSVP, Differentiated Services and MPLS [1].

The Differentiated Services (DS) network architecture provides these QoS guarantees in the most scalable and least complex manner. In a DS domain, there exist two levels of service provisioning, the standard best effort service and the premium service where the clients' requests for guarantees are met. While bandwidth is reserved for each client at a price, these reservations are made without any understanding of the nature of the information. The bandwidth broker tries to understand the parameters involved using service policies made in accordance with the client's requirements. Though this provides a better sense of resource allocation, the result is still static provisioning of resources, and leads to a wastage of bandwidth.

This paper presents the design of the active resource management (ARM) approach. ARM uses active networking techniques at the bandwidth broker to reallocate client's bandwidth on the fly. Active resource management is achieved by keeping a track of the resource usage of each client. A client's traffic rate rarely touches the peak transfer rate, i.e. the rate of transfer for which the bandwidth was reserved. So anytime the users' traffic rate drops below the agreed rate of transfer, a portion of the unused bandwidth can be returned to the pool of bandwidth that is reserved for each type of service. Thus, through over provisioning for the same amount of resources and dynamic reallocation, an increased number of clients can be maintained resulting in a better bandwidth usage. The ARM concept is evaluated using the NS-2 simulator toolkit.

The rest of this paper is organized as follows. Section 2 presents background material. Section 3 describes the ARM approach and architecture. Section 4 outlines an evaluation using the network simulator (NS-2) and some experimental results. Section 5 presents some conclusion.

*The research presented in this paper is based upon work supported by the National Science Foundation under Grant Numbers IIS 9872995 (KDI) and ACI 9984357 (CAREERS) awarded to Manish Parashar.

2. Background And Related Work

Differentiated Services

Differentiated Services Architecture [2] is based on a simple model where the traffic entering a network is classified and conditioned at the boundaries of the network, and then assigned to different behavior aggregates. The individual micro flows are classified at the edge routers, into one of the many classes. A per-class service is then applied at the core of the network. The classification is done at the ingress router, and the packet is marked, using codepoints, as belonging to one of the many classes and injected into the network. The core routers that forward the packet examine this marking and treat the packets accordingly. A traffic meter is used to ensure that the packet conforms to the traffic profile agreed upon by the network provider and the customer.

All packets with the same codepoint are grouped together and are known as a behavior aggregate and are handled the same way. The codepoints (DSCP) are known as per hop behavior (PHB) and are of two types: expedited forwarding (EF), and assured forwarding (AF) [6]. EF PHB was defined to support low loss, low delay, and low jitter. While the AF PHB defines four relative classes of service with each service supporting three levels of drop precedence.

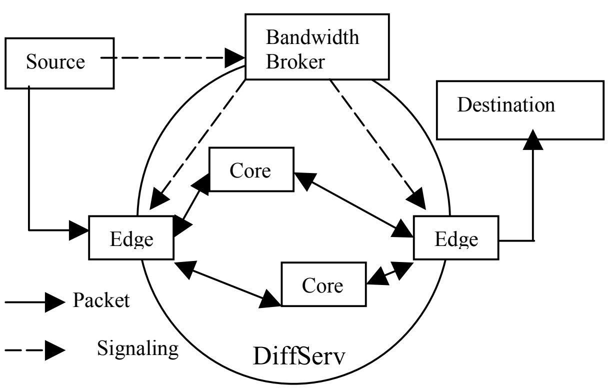

Fig1 A generic DS domain with a Bandwidth Broker

DiffServ is being regarded as a reasonable solution to provide Quality of Service on the Internet. Research is being carried on in various universities, and IETF has a DS working group. Companies like Cisco and IBM provide DS functionality in their routers, and Nortel Networks has evaluated DS using NS-2 toolkit.

Bandwidth Broker:

All agreements between the customer and the service provider (SLA) are used to define the policies that map to different PHBs'. A service provisioning policy indicates how traffic conditioners are configured at the edge routers and how the traffic streams are mapped to the DS behavior aggregates. To add some intelligence to the provisioning mechanism, a central agent is defined which keeps track of the resources in the DS domain and accordingly define the SLA's.

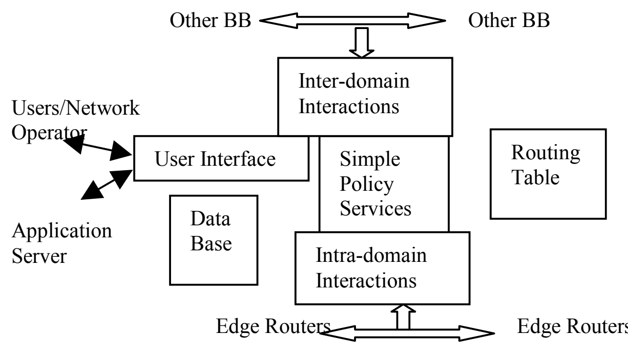

Fig2 Functional decomposition of bandwidth broker

The Functional decomposition of a bandwidth broker is shown in Fig2 [5]. It has a user interface that provides a mechanism by which the user can request resources. The BB uses Intra-domain Interactions to configure the edge routers, while the peering BBs' have Inter-domain Interactions to request resources and make admission control decisions to enable end-to-end flow of traffic. A routing table is maintained to access inter-domain routing information before allocating the resources. A database is used to store information about all the BB's parameters such as SLA's, current reservations, configuration of routers, DSCP mapping and policy information.

The Bandwidth Broker has been designed to add intelligence to the DiffServ, to help optimize the existing resources. An advisory committee has been initiated to define the protocols implemented by the broker [3].

3. Active Resource Management

In the DS model, policies are predefined, and accordingly resources are allotted to the particular client. The clients' peak traffic rate, time for which the service is required, delay and jitter are some of the parameters for defining a policy. For certain applications where the data is in the form of streaming media, the traffic rate is bursty and rarely at peak transfer rate. In such situations, a portion of the allocated bandwidth remains unused, and no other client can use it. In order to allocate these resources intelligently, a broker agent is used. The client defines his requirements using SLA's. The agent maintains a database of parameters, in accordance with which reservations are made and the DSCP for those services are assigned. The end result is still static reservation, where bandwidth once allotted to a client is used solely according to that client's traffic flow.

Thus arises a need to reuse the bandwidth wasted on each reservation that is made and if possible re-allot it to another client. The basic concept is that by effectively knowing when a client is sending packets and how much of this allotted bandwidth is being used at any given time, the excess bandwidth can be reallocated without loss of promised service. In order to measure the traffic rate of every client, the bandwidth broker agent can use the meter that is attached to each flow. For example, the TSW Tagger [8] is a meter that measures the average traffic rate, using a specified window size for the TSW2CM and TSW3CM [9] policers. With the knowledge of incoming traffic, different DSCP's are defined for various traffic rates. So when the broker agent notices a traffic rate that is less than the rate agreed upon, it steps down to a lower DSCP that suits the current rate. The remaining unused bandwidth is restored to the pool of available bandwidth. Thus allowing more reservations. For the worst case scenario where all the clients send in traffic at their peak rate, the additional bandwidth is provided by dipping into the pool of bandwidth belonging to the best effort services. Conditions for allocation are: limit the number of reservations allowed per class, and fix the amount of bandwidth that must be reserved for best effort services. By doing so, we limit the number of premium service reservations to reduce the amount of unused bandwidth and provide a threshold of tolerance within which we can add or remove bandwidth as needed.

An Illustrative example

The test network includes two DS domains, and a client belonging to a heterogeneous network architecture.

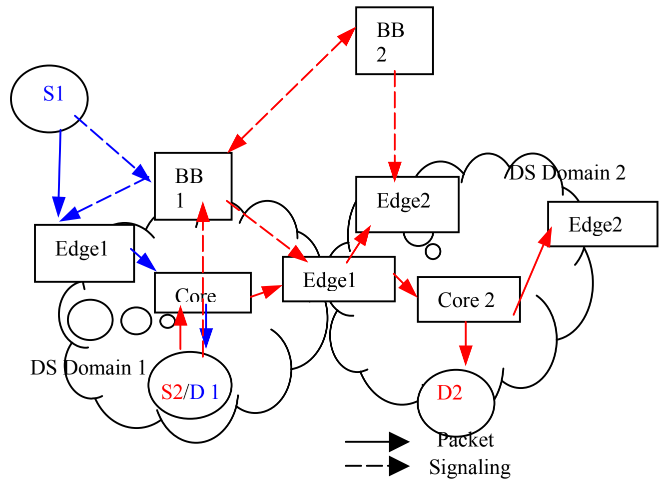

Fig3 Test network showing different scenarios

When source1 requests service, it contacts the BB1 on DS1 enroute to the destination1. If the destination1 is within this domain, as shown in Fig3, the BB1 looks into its database, and decides upon the best available bandwidth and other parameters, defines the SLA, and assigns a DSCP for the traffic flow. The Edge routers are configured to mark the packets from this client with the correct codepoint, while the core routers just forward the packets. The BB1 also assigns a set of lower DSCPs, which define slightly lesser bandwidth requirements. If the available bandwidth goes down below a level, the BB uses the meter to check the traffic rate from the source, and if the rate is any lesser than the bandwidth allotted then it steps down the service to a lower DSCP which provides only the required amount of bandwidth, and the remaining unused bandwidth is returned to a pool of available bandwidth. Source S1 is in DS1 domain and the destination D2 is DS2 domain; the source contacts the BB1 of its domain. The BB1 then looks at the database and the routing table to figure out the downstream edge router and the peering BB2, and sends a Resource Allocation Request (RAR) based on the SLA it has with the source. On confirmation of the reservation the above process is followed.

4. Implementation and Evaluation of the ARM Algorithm using NS-2

We have implemented the ARM algorithm on the NS-2 toolkit. Using the NS-2 toolkit with the DiffServ patch provided by Nortel Networks, we generated DS domains and created suitable test networks [4].

The DiffServ implementation has three modules, the edge and core router modules, and the policy module. The policy class handles the creation, manipulation and enforcement of edge router policies. A policy defines the treatment the packets will receive at an edge router. Policies are set using Tcl [7] scripts. The policy class uses a policy table to store the parameter values. The table has various fields such as SLA, current reservation, router configuration, policies, and DSCP mappings.

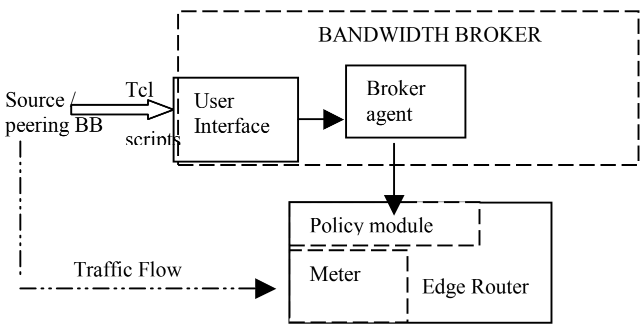

Fig:4 Modular breakup of BB and its interaction in traffic flow

The bandwidth broker is used to configure the policy module of the DiffServ. We define the main broker agent and a user interface. The broker agent provisions according to the SLA's using Tcl scripts and it checks the type of the incoming flow (AF/EF) and associated bandwidth reserved for such flows and then makes its allocations. If allocation fails, it looks for an alternate SLA. This achieves static provisioning. The Active Resource Management (ARM) algorithm keeps a track of each client's average traffic rate using a meter that is associated with each flow and is defined in the policy table. For every DSCP a set of alternate DSCP's are also defined. Each DSCP corresponds to a different traffic rate and the algorithm switches the current DSCP marking of the packet flow according to the traffic rate indicated by the meter.

The algorithm is currently being tested using the same set of experiments, conducted on DS domains without the broker, with the broker and with a broker implementing the ARM algorithm.

5. Conclusion

There is a need of guaranteed services for real time media and mission critical traffic that cannot be provided by standard IP methods. The Differentiated Services framework provides a suitable, scalable and less complex means for providing these guarantees and with the help of the bandwidth broker agent, a level of intelligent resource provisioning is achieved. But to reach a level of optimization of these resources provided by any DiffServ domain, this paper presents ARM. An Active Resource Management algorithm that reallocates the bandwidth reserved for specific clients when not used by them, to other clients, but returning the bandwidth when needed, and in certain cases providing more bandwidth that agreed upon, so as to maintain the flow. Thus, it provides optimum usage of the limited bandwidth that is available. We have implemented ARM using NS-2 and are currently evaluating it.

References

[1] Chris Metz. *"IP QoS: traveling in the first class on the Internet."* IEEE Internet Computing, Vol. 3, No. 2, March/April 1999

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

[3] http://qbone.internet2.edu/bb/

[4] P. Pieda, J. Ethridge, M. Baines and F. Shallwani. *"A Network Simulator Differentiated Services Implementation"* Open IP, Nortel Networks.

[5] K. Nichols, V. Jacobson and L. Zhang. "A Two Bit Differentiated Services Architecture For The Internet" RFC 2638, JULY 1999

[6] Heinanen, J., F. Baker, W. Weiss, and J. Wroclawski. "Assured Forwarding PHB Group." RFC 2597. June 1999.

[7] Brent B. Welsh, "Practical programming in Tcl and Tk" Third Edition.

[8] Clark, D. and W. Fang. "Explicit Allocation of Best Effort Packet Delivery Service." MIT Lab for Computer Science

[9] Fang, W., N. Seddigh, and B. Nandy. *"A Time Sliding Window Three Color Marker"* (Internet draft). March 2000.