Performance Evaluation of the Managed Bandwidth Service with QoS Guarantees

Christos Bouras[1,2] Vaggelis Kapoulas[1,2] Dimitris Primpas[1,2]
[1]*Computer Technology Institute, 61 Riga Feraiou Str., 26221 Patras, Greece*
[2]*Department of Computer Engineering and Informatics, University of Patras, 26500 Rion, Patras, Greece*
TEL: +30-(2)610-{960375, 960355, 996954}
FAX: +30-(2)610-960358
e-mail: bouras@cti.gr, kapoulas@cti.gr, primpas@cti.gr

Abstract

This paper describes a solution to extend the Managed Bandwidth Service (MBS), which is provided on a backbone network using MPLS VPNs, to non-MPLS domains and also evaluates it. It is based on the Class Based Weighted Fair Queueing mechanism (CBWFQ) [4] and describes the way it should be implemented to provide guaranteed bandwidth connections. In particular, CBWFQ can provide assured bandwidth connections when simultaneously leads to efficient bandwidth utilization. In addition, the proposed solution is being tested on simulation environment (using the Network Simulator) in order to evaluate its performance characteristics. On the simulation tests had been reserved specific amount of bandwidth for specific flows and their throughput was measured in order to understand if the proposed solution works well. Finally, the results are very well and prove that this method can guarantee bandwidth for every flow in all network conditions (congested or un-congested). At last, the paper presents a technical example of router configuration for implementing the MBS service according to the basic proposed solution.

Keywords: Managed Bandwidth Service, Quality of Service, Simulation, Performance Evaluation

1. Introduction

The idea of the Managed Bandwidth Service is to provide connections between two specific points with guaranteed bandwidth. Usually, the MBS connections are established between routers on the network and aren't extended to the end users. This happens mostly because the methods that can extend the connections to the end users are inefficient and add considerable overhead to the local networks. The most recent years, many providers and technology institutes try to implement Managed Bandwidth Service as their "clients" have show a large interest about this service. In addition many national organizations, National Research Educational Networks (NRENs) and Geant [14] made the decision to provide MBS service. The available solutions in order to implement this service are few and depend on the network technology that every network provider uses. The most common and practical solution is the use of ATM permanent virtual circuits (PVCs), which can guarantee specific bandwidth to a 2- point connection, on ATM networks. This solution was adopted by Geant, on TEN-155 and by many national networks as Greek research network (GRNET) [15] that used ATM infrastructure. On next generation networks, where is expected the use of MultiProtocol Label Switching (MPLS) [19] technology, the most common solution will be the implementation of virtual private networks (VPNs)[6]. MPLS is a new efficient technology that provides traffic-engineering, virtual private networks and constraint based routing that can lead to load balancing of a network. The main idea of virtual private networks is to provide connections between specific points on a public network using the public infrastructure. The traffic on the VPN is totally separated from the whole network traffic virtually and the other users don't understand the existence of VPNs. The MPLS VPN technology provides many abilities such as the implementation of VPNs [6] on layer 2 of ISO/OSI model and on layer 3 too. GEANT is planning to use layer 2 MPLS VPNs to provide MBS connections on its backbone, using Juniper infrastructure and especially the cross circuit connection (CCC) method [8] [13]. NRENs are expected to follow a similar schema, which will allow them to interconnect their service with Geant's. Greek Research Network has decided to use layer 3 MPLS VPNs and interconnect the MBS connections with Geant at the Geant's PoP [1][15][14]. According to this approach, the MBS service will use the OSPF routing protocol [20] and RSVP-TE protocol [21] in order to

implement the MBS connections. This paper presents a solution (method) to extend the MBS service, which is implemented with that way, on non–MPLS domains and try to approach closer the end users. This method is based on the DiffServ mechanism Class based Weighted fair Queuing (CBWFQ), that can provided minimum guaranteed bandwidth to specific traffic flows. The method is described analytically and also its performance is evaluated. This evaluation has been done with the use of the network simulator (NS-2), where we have performed various simulation tests. In every test we had configured this method to reserve bandwidth for specific flows and we also added background traffic in the network is order to bring it in congestion. Finally, we measured the throughput of every flow, which presents us the bandwidth that it used. So, in all the tests, this method worked very well and provided guaranteed bandwidth connections.

The rest of the paper is organized as follows: section 2 describes the problem and the available technical solutions while section 3 presents the proposed solution with all the technical details. Section 4 presents a small description for Network Simulator (NS-2) and the simulation experiments and results that have been done in order to evaluate its performance. Finally, section 6 describes our future work and section 7 presents our conclusions. In appendix, there is a simple implementation on a real network with CISCO [12] routers.

2. Managed Bandwidth Service: The available technical solutions

Usually, the backbone networks are connected with other smaller networks (sub-networks), which belong to organizations or universities etc and the end users are connected on these, or through LANs to these, as can also be shown on Figure 1. So, an issue that must be investigated is the ability to extend the MBS connections from backbone networks, where in our case it is an MPLS domain, to sub-networks (non-MPLS domains) and reach closer the edge routers, where the end users are connected. This attempt has a big difficulty, because all networks that are connected on a backbone use different infrastructure, and technologies. So, the proposed solution must be quite general to cover all the cases. Besides, the solution is necessary to simplify many issues such as the way that the whole service will be administered and the way exactly will be interconnected with the MBS connections on the backbone.

The MBS service on the backbone is implemented using MPLS VPNs layer 3 and the appropriate signalling protocols. In particular, the OSPF will calculate the routing path and the RSVP-Traffic Engineering will make the signalling and bandwidth reservation. Our proposed solution tries to extend that MBS service to the non – MPLS sub-networks. The available solutions that satisfy all the above cases and can be used are based on QoS mechanisms and particularly using IntServ and DiffServ architecture.

The IntServ architecture uses the RSVP protocol and reserve resources across a specific path for a flow or aggregate of flows. The RSVP protocol has the advantage that it can provide strict QoS guarantees but in other side the bandwidth management and utilization it succeeds is not optimal. Besides, its function increases the network load because it requires the exchange of many packets to establish connections and update every time its status on every router.

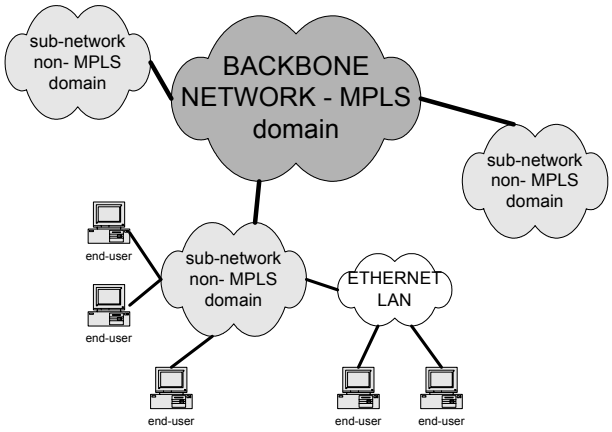

Figure 1 The problem representation

Instead, the DiffServ architecture doesn't make any strict reservation but simply classify the packets on class of services and try to treat each traffic class differently to provide finally QoS guarantees (delay, jitter, packet loss, and bandwidth). The mechanisms that it uses in order to provide those guarantees are classification, marking, metering, policing and queue management. Especially, the Expedited forwarding per hop behavior [2] can provide bandwidth guarantees and the general idea is to simulate the leased lines. A service that belongs to EF-based DiffServ architecture is IP- Premium that can be used to provide MBS connections, but it has a drawback that it is not general to suite on all network cases. Instead, it requires specific network technology, infrastructure (router series) and operating systems (IOS). Generally, both these architectures (IntServ and DiffServ) can be used in the attempt to provide management bandwidth service on a sub- network. The main solution that we propose is based on the DiffServ architecture and in particular uses the class based weighted fair queueing mechanism (CBWFQ) [4], a software-based mechanism that is independent of network infrastructure and it is supported by many router models (series) and IOSs.

This mechanism extends the weighted fair queueing mechanism (WFQ) [10] and allows the classification of packets on specific classes. Its operation is simple and will be described briefly. First of all, CBWFQ mechanism creates a policy map and in each policy map defines classes. Each class contains packets that are associated in a class based on several criteria. The classification is implemented with the definition of class maps, where are configured the parameters for each matching criterion. According to these class maps, the supported matching criteria are access control lists, the used protocols and the input interfaces. Next, for each class that is defined in a policy map there are some characteristics, which should be assigned. In particular, the most important is the assigned bandwidth that is the minimum guaranteed bandwidth for the traffic class on congestion. This characteristic leads to that the class based weighted fair queueing mechanism is appropriate to support MBS connections, as it provides minimum guaranteed bandwidth. In addition, for each traffic class is defined the maximum queue limit if the queue is tail drop or is configured the weighted random early detection mechanism (WRED) instead. The CBWFQ mechanism also creates a default class to manage packets that do not belong to any other classes and reserve for them the remaining bandwidth on the link. By default, the remaining bandwidth should be at least the 25% of the initial bandwidth of the link but there is the ability to be changed with the use of max-reserved-bandwidth command. In particular, this command specifies the maximum allowable reserved bandwidth by the classes and the remaining is assigned to the default class.

3. The implementation of MBS service for GRNET's sub-networks

In our attempt to extend Managed Bandwidth Service on networks that are interconnected with an MPLS backbone domain (GRNET domain), the CBWFQ mechanism is the most capable to use. Our proposed implementation is the creation on every non- MPLS networks a policy map and a new policy class per each requested MBS connection. On each policy class the assigned bandwidth parameter will be equal to the requested bandwidth from the MBS connection exactly. In addition, on each class should be used tail drop queue mechanism and the queue limit should be defined as the maximum admissible. In other case, there would be used the WRED mechanism which drop packets randomly to avoid congestion. Another very important point is the definition of the matching criteria for the packet classification. The available solutions are many, the used protocol, the MPLS labels or the definition of access lists. Our proposal is the use of access control list and the definition of access groups where the sender's and receiver's IP addresses must belong.

Except the policy classes that implement the MBS connections, it is necessary to take care about the other traffic on the network. The CBWFQ mechanism requires the configuration of a default class that manages all that traffic. The main goal of default class is that does not administrate them at all, but it only forwards them using the remaining bandwidth from the other classes. Besides, the characteristics of that class can be configured, as the other classes. In particular, the default class can use tail drop queues or WRED mechanism and the queue limit is a user- defined variable. In addition, default class has the ability to use multiple dynamic queues by the flow based weighted fair queueing mechanism that can be run there. Finally, according to our proposal the network provider must decide about the total amount of bandwidth that allows being reserved on every link by the MBS service. Next, every router must be configured to support that decision and the administrator must also be informed to schedule properly the procedure of accepting new MBS connections.

In addition, a very important point for the implementation of MBS service is the selection of the routing path for the MBS connections. This selection must be done from a routing protocol and in our case the best solution is the OSPF. OSPF always returns the best path between 2 nodes, using as criterion the number of hops. In some cases it is not very flexible and can overload some links because the only criterion is the number of hops. A very effective alternative solution is the enrichment of the criterion adding a cost to each link. This feature is supported by all the routers and IOSs and can lead to network load balancing. The default cost that this feature inserts is associated with the total bandwidth of the link (10^8/bandwidth), but the proposed solution is to be assigned explicitly a cost. In particular, we propose this cost to be equal with the total utilized bandwidth at the moment of path's selection. This approach is quite easy to be implemented as the OSPF database keeps an entry (txload) that represents the utilized bandwidth every moment.

Finally, the operation of CBWFQ mechanism is clear and is appropriate according to the above design for the implementation of MBS connections. The only restriction that the CBWFQ mechanism inserts is the number of classes that it allows to be implemented on a policy map. In particular, the allowable classes are up to 64, which is a restriction because it means that only 64 MBS connections can be implemented.

4. Performance evaluation

4.1. The Network Simulator NS-2

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, the proposed solution for a bandwidth management service has been tested on simulation environment in order to evaluate its performance characteristics. The simulator that has been used is the Network Simulator NS-2 [16][18] and the expected result is the performance characteristics of the above solution. In particular, the metrics are the throughput of the foreground and background traffic, as they inform us for the utilized bandwidth from each traffic category.

The Network Simulator NS-2 is a free open source simulator that was created at its first version on Information Science Institute. Next, many people used it and tried to develop many new features on it, as we did it [17] [11], trying to extend its DiffServ functionality. Finally, it became one of the most powerful network simulators and now NS-2's current release is the ns-2.26, which is quite stable and supports many new features. NS-2 provides many advantages to its users, such as an easy environment to understand its function, implement new scenarios and test them. In addition, it is a simulator that it is updated very often and also there are many people that work with it and always support its operation by fixing bugs or adding new code.

4.2. The Experimental Procedure

The whole procedure contains various simulation experiments on NS-2, each time following different scenario in order to simulate various network conditions. In particular, the network had background traffic that covered a large fraction of the link bandwidth (almost 80%) and we inserted in the network an aggregate of flows that required specific bandwidth guarantees. That aggregate was treated according to the proposed solution with the use of the already implemented CBQ mechanism in NS-2 and all the traffic on the network was inserted with the "cross traffic" model. The CBQ mechanism, is implemented in NS-2 and its work is to simulate the Class Based Weighted Fair Queuing mechanism that many manufacturers provide to their software, as CISCO does The CBQ mechanism should be configured on every router independently, something that provides a big advantage, as it is not necessary to use complicated topology in order to simulate a real network's condition. So, we simulated various scenarios and finally measured its throughput in order to evaluate its performance. The throughput of every flow represents the bandwidth that uses every time, so it is the best metric for the evaluation of the solution's performance.

4.2.1. Experiment 1

The scenario's of the first experiment is quite simple: we inserted traffic with average rate over 3500Kbps and reserved 2000Kbps (at minimum) with the CBQ mechanism. In addition, there was background traffic with average rate almost 8000Kbps and the link was 10Mbps. So, according to the router configuration, the remaining bandwidth of the background traffic was at most 8000Kbps. If the CBWFQ mechanism is not used, then the throughput of the foreground traffic should not be stable, but with variations because of the network's congestion and the packet drops. Otherwise, if the CBWFQ mechanism is used, then the throughput of the foreground traffic should be stable, and of course more than the rate of 2000Kbps (the minimum guaranteed), and the variation should be presented on background traffic's throughput.

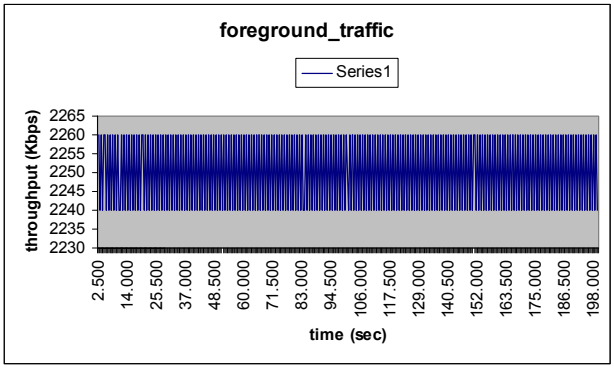

Figure 2 Foreground traffic's throughput at experiment 1

Figure 2 and Figure 3 presents the experimental results with the throughput of the foreground and the background traffic's respectively. According to those, foreground traffic's throughput is always more than 2200Kbps when we had configured the CBQ mechanism to guarantee at least 2000Kbps and the network is congested. So, the mechanism reserved 2000Kbps for the foreground traffic and additionally used free amount of bandwidth in the link, when there was.

On the other hand, the background traffic used the free bandwidth and adjusted its rate according to that. Consequently, the proposed solution seems to work well on this scenario and can guarantee minimum bandwidth for an aggregate of flows.

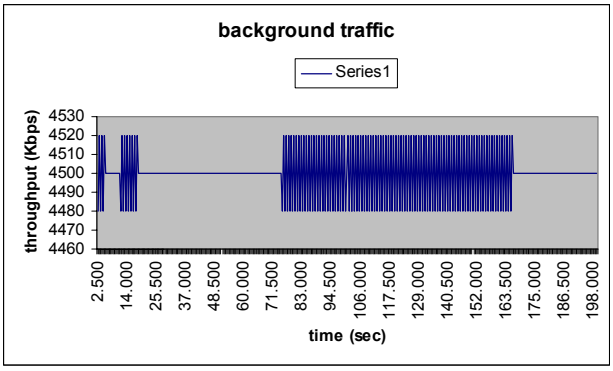

Figure 3 Background traffic's throughput at experiment 1

4.2.2. Experiment 2

In the second scenario, The background traffic was reduced (average rate almost 6000Kbps) and foreground's traffic characteristics remained the same as on scenario 1. In particular, the foreground traffic had average rate over 3000Kbps, but we had only reserved 2000Kbps for it. So, the network was not congested, it was almost fully utilized and the results are presented on the following figures.

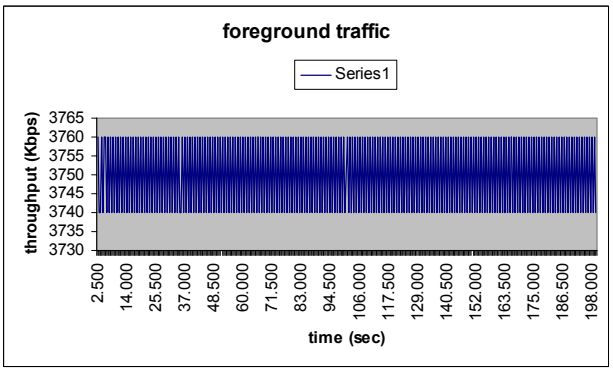

Figure 4 Foreground traffic's throughput at experiment 2

At this experiment, we noticed that foreground traffic's throughput (Figure 4) was not near 2000Kbps, which was arising from the guaranteed bandwidth but was almost 3750Kbps, which is the flow's throughput if all packets are sent. This was caused of the use of CBWFQ mechanism that used the available bandwidth on the link for the foreground traffic. In addition, background traffic (Figure 5) was treated as best effort and its throughput is presented on Figure 5.

As a conclusion, at this experiment, the foreground traffic used its reserved bandwidth and additionally used some of the free bandwidth. In particular, if there is free bandwidth, the CBWFQ mechanism tries to use it fairly for all traffic classes.

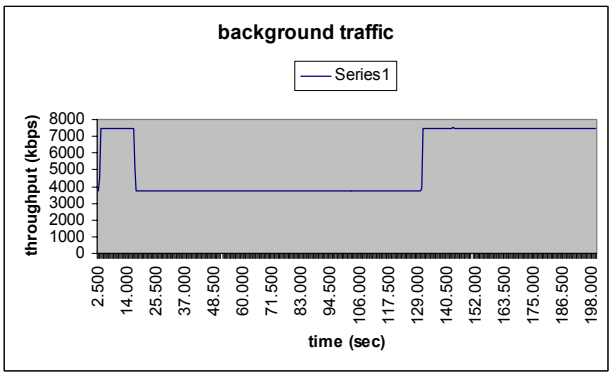

Figure 5 Background traffic's throughput at experiment 2

4.2.3. Experiment 3

Finally, in the third scenario, more flows were added on the experimental network. In particular, there was an aggregate of flows with average rate over 3000Kbps and it had guaranteed bandwidth of 2000Kbps with the use of the proposed solution above. In addition, there was a second aggregate that had average rate over 4000Kbps and had minimum guaranteed bandwidth of 4000Kbps. At last, there was background traffic with average rate almost 5000Kbps and should use the remaining bandwidth that is at most 4000Kbps.

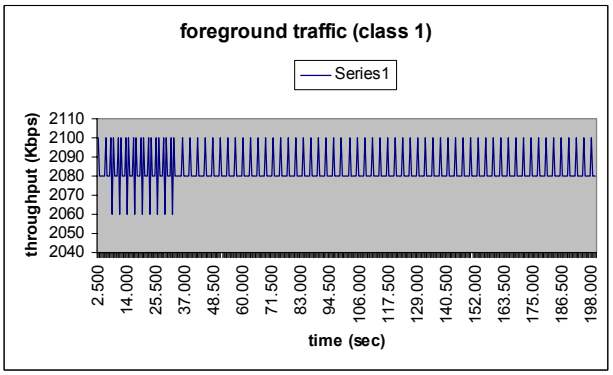

Figure 6 Foreground traffic: first aggregate's throughput at experiment 3

So, at this experiment, the network is congested and also there are many flows that have make bandwidth reservations. The final results (traffic's throughput at Figure 6, Figure 7, Figure 8) are very close to the minimum reserved bandwidth as the figures prove. In particular, the first flow that had made a bandwidth reservation of 2000Kbps has average throughput a little bit more than 2000Kbps (almost 2080Kbps), while the network is congested.

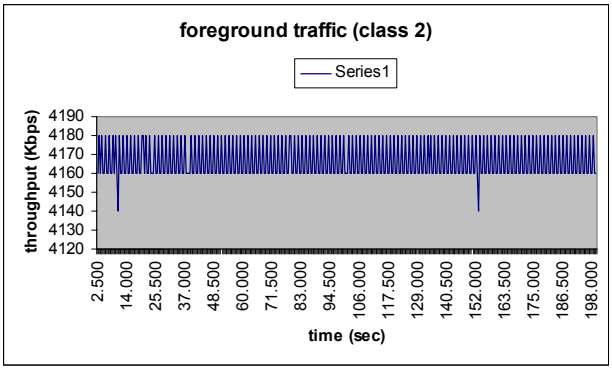

Figure 7 Foreground traffic: second aggregate's throughput at experiment 3

Besides, the second flow that had made bandwidth reservation of 4000Kbps, has similar behaviour, as its average throughput is almost 4160Kbps, a little bit more than the reserved bandwidth.

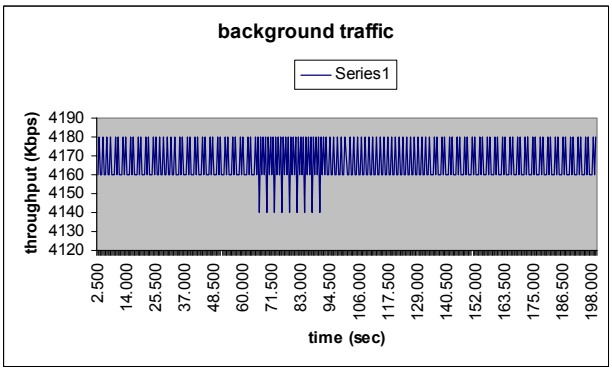

Figure 8 Background traffic's throughput at experiment 3

Finally, the background traffic always used the remaining bandwidth. The CBWFQ mechanism had been configured to treat it as the default class and use the remaining bandwidth but it should not be less than 4000Kbps. So, the figure proves that it also works fine and background traffic has average throughput almost 4160Kbps. It is a little bit more than the reserved bandwidth for the default class but not any more because the network is congested and there is not other free bandwidth.

Consequently, we can make the conclusion that as long as the flows that receive guaranteed bandwidth increase while the network is congested, the proposed solutions seems to lead every flow to use the minimum guaranteed bandwidth. Otherwise, if the network is not congested, the proposed solution is more tolerant and the flows can share the unutilized bandwidth fairly.

All the simulation experiments that were done and are described above, proves that the CBWFQ mechanism, as is planned to be used in order to provide MBS connection, works well. At the simulation environment, the proposed architecture, that provides MBS connections to non-MPLS domains, didn't present any special drawback and we can insist that the only limitation for its use will be the number of the policing classes (up to 64 classes) that can support, according to CISCO [4][12].

5. Future Work

In the future, the MBS service is going to be implemented on the sub-networks that are connected to the Greek Research networks (GRNET) [15] and it will be based on the proposed solution here. GRNET has decided to implement the MBS connections on its backbone network using layer 3 MPLS VPNs and the proposed solution will extend those connections inside the sub-networks (non-MPLS domains). In the future, we plan to perform some more tests that will simulate all GRNET's network (backbone and sub-networks) and we will investigate this method in conjunction with other services. In addition, we will attend all the implementation procedure on GRNET's network. At last, there are also many other issues that can be investigated, like the opportunity to extend more the MBS connections and reach the end users. Besides, a very important point is the automatization of the procedure that implements the connections and the definition of priorities on accepting connections in the case that the service is overloaded.

6. Conclusions

In this paper was presented a solution (design) to extend Managed Bandwidth Service (MBS) on sub-networks (non-MPLS domains) from a backbone network (MPLS domain) and the results from the simulation experiments that have been done in order to evaluate its performance. The solution is based on Class based Weighted Fair Queueing (CBWFQ) and the main idea is to provide each connection using a new policy class and assigned to it the requested bandwidth. The path selection should be done using the OSPF mechanism and as selecting criterion should be used a cost on each link, which corresponds to its usage.

The CBWFQ mechanism comes with a lot of advantages, as it provides classes, which can guarantee specific bandwidth. This bandwidth is the minimum guaranteed on congestion circumstances and during normal conditions every class could use more if it is available. In addition, this mechanism uses a default class for all the other traffic and in our case it will be used for the traffic on the network that does not belong to MBS connections. Finally, another major advantage is that CBWFQ mechanism is supported on many platforms, and especially CISCO maintains that it is supported on many router series and IOSs [4] [12].

In addition, various simulation tests have been done in order to evaluate its performance characteristics. According to them, if the network is congested, the proposed solution based on CBWFQ mechanism seems to lead every flow to use its minimum guaranteed bandwidth. Otherwise, if the network is not congested, the proposed solution is more tolerant and except the guaranteed bandwidth shares the unutilized bandwidth to all the flows fairly.

So, CBWFQ seems to be a powerful mechanism that can provide guaranteed bandwidth MBS connections. Simulation tests also proved its operation and the next step should be the implementation on a real network. Appendix presents a typical implementation, according to the above-presented design, for a real network that uses CISCO infrastructure.

7. References

[1] "Management Bandwidth Service on MPLS domain", C. Bouras, V. kapoulas, D. Primpas, 17th IEEE International Workshop on Communications Quality & Reliability, CQR-2003, Kiawah Island, South Carolina, USA

[2] RFC 2598 "An Expedited Forwarding Per Hop Behavior" V. Jacobson, K. Nichols, K. Poduri, June 1999

[3] RFC 2815 "Integrated Service Mappings on IEEE 802 Networks" M. Seaman, A. Smith, E. Crawley, J. Wroclawski, May 2000

[4] "Class-Based Weighted Fair Queueing", CISCO Documentation

[5] "Subnetwork Bandwidth Management", CISCO Documentation

[6] E. Rosen, Y. Rekhter, "BGP/MPLS VPNs", RFC2547, March 1999

[7] Deliverable D9.5 "Proposal and implementation plan of the migration of current MBS", Geant's Report, Work Package 8

[8] Deliverable D9.12 Service specification for the proposal and implementation plan for the migration of current MBS, Geant's Report, Work Package 8

[9] "IP Quality of Service" Vivek Alwin, CISCO PRESS

[10] C. Dovrolis, D. Stiliadis and P. Ramanathan, "Proportional Differentiated Services: Delay Differentiation and Packet Scheduling", in proceedings of ACM SIGCOMM '99 Conference, Boston, USA, 1999

[11] "Enhancing the DiffServ Architecture of a Simulation Environment" C. Bouras, D. Primpas, A. Sevasti, A. Varnavas, Sixth IEEE International Workshop on Distributed Simulation and Real Time Applications (DS-RT 2002) 11-13 October 2002, pp 108-118.

[12] http://www.cisco.com

[13] http://www.juniper.net

[14] http://www.geant.net

[15] http://www.grnet.gr

[16] S. McCanne and S. Floyd, "ns Network Simulator", available at: http://www.isi.edu/nsnam/ns/

[17] C. Bouras, D. Primpas, A. Sevasti and A. Varnavas, "DiffServ functionality patches developed for the ns-2 simulator", found at: http://ouranos.ceid.upatras.gr/diffserv/nspatches/description.htm

[18] R. Wielicki, "ns-2 ad-ons page", found at: http://thenut.eti.pg.gda.pl/~rafalw/wfq/

[19] RFC 3031 "MultiProtocol Label Switching Architecture" E. Rosen, A. Viswanathan, R. Callon, January 2001

[20] RFC 2328 "OSPF Version 2" J. Moy, April 1998

[21] RFC 3209 "RSVP-TE: Extensions to RSVP for LSP Tunnels" D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow December 2001

Appendix

Appendix presents a technical example of how the service should be implemented, according to the above design, on a domain that uses CISCO infrastructure with IOS 12.2 including the appropriate router configuration [9]. First of all, when a request for an MBS connection is coming then the system makes the appropriate checks for bandwidth availability. Firstly the system finds the best path that the connection should follow. This operation is done with the use of OSPF mechanism and the additional characteristics of defining a cost to each link. This additional characteristic is defined with the following command **ip ospf cost <cost>**, where CISCO by default uses as cost the cost that arises from the link bandwidth and the relationship 10^8/bandwidth (in bps) Our design says that the cost should arise from the reserved bandwidth on the link and for this case the option <cost> on the command must be related to the txload quantity of OSPF database. This quantity is saved by OSPF in its database and represents the used bandwidth every moment.

After defining the best path that the MBS connection should follow, the network administrator should configure every router across that path. First of all, he creates a policy map (it must be done the first time he tries to implement a class on a router, otherwise he uses the already existing policy map). Next, he defines the name of the class that implements and assigns to this class the requested bandwidth. In addition, administrator specifies for the implemented class the maximum number of packets in queue. In this point, we must notice that according to our proposal, the implemented class (MBS connections) do not use WRED mechanism but drop tail queues. Besides, administrator creates a class map for each class and defines there the matching criterion for packet classification. In this example, we assume that the requested bandwidth is 50Mbps and the requested connection uses the same input interface that can be used as matching criterion (for example, we assume an interface called Ethernet0/1. Instead of that the IP address of sender and receiver can be used, by defining access groups on access lists, as we mention in the design). In addition, example assumes that network uses 100Mbps bandwidth for not MBS connections. The appropriate

configuration for the operations described above is the following:

Router (config)# class-map class-1
Router (config-cmap)# match input-interface Ethernet0/1
Router (config-cmap)# exit

The above configuration creates a class map and defines as matching criterion (for packet classification) the input interface Ethernet0/1.

Router (config)# policy-map policy-map-1
Router (config-pmap)# class class-default default-class-1
Router (config-pmap-c)# bandwidth 100000 (it is the available bandwidth for all the other traffic, except MBS connections)
Router (config-pmap-c)# queue-limit 64 (the defined queue limit must be in number of packets. If queue limit is not defined, router uses the default value of 64 packets)
Router (config-pmap-c)# exit

The above configuration creates a policy map, defines the default class and assigns its allocated bandwidth. In default class can also be used the WRED mechanism instead of drop tail queues. If the bandwidth is not defined, then router applies best effort treatment.

Router (config-pmap)# class class-1
Router (config-pmap-c)# bandwidth 50000 (the assigned bandwidth must be defined in Kbps)
Router (config-pmap –c)# queue-limit 10 (the defined queue limit must be in number of packets. If queue limit is not defined, router assumes the default value of 64 packets)
Router (config-pmap-c)# exit

Finally, the above configuration defines the class, which its matching criterion defined above and allocates its requested bandwidth. All described configuration must be applied to all the routers, across the path, in order to provide the connection between the 2 end points.