# Management of Inter-domain Quality of Service Using DiffServ Model in Intra-domain

Sara Bakkali, Hafssa Benaboud, and Mouad Ben Mamoun

LRI, Faculty of Sciences at Rabat, Mohammed V-Agdal University,
Rabat, Morocco
bakkalisara@gmail.com,
{benaboud,ben_mamoun}@fsr.ac.ma

**Abstract.** During the last decade, Internet has experienced enormous evolution. This evolution concerns the huge quantity of traffic circulating over Internet and also the important diversity of these traffics types. Each type of traffic requires a specific QoS parameters. This point may represent a serious concern mainly due to the difficulty in ensuring QoS for traffics that cross multiple domains or Autonomous Systems (ASs). To solve this problem several researchs and studies has been proposed. In this paper, we describe a new mechanism that we proposed in a previous work to solve this problem. Our mechanism ensures the end to end QoS requirements over multiple ASs. This method keeps the same values of QoS parameters required by the traffic, even during its passage across several ASs. This paper explains the problem of inter-domain QoS, describes our new approach, we give then a case study of our solution using DiffServ model in intra-Domain. Simulation Results show the improvement of network performance when using the new mechansim and when comparing them to those of the standard case.

**Keywords:** Inter-domain routing, QoS, DiffServ.

## 1 Introduction

Today, traffics circulating on networks are very diversified and require a specific parameters in terms of bandwidth, delay and other necessary parameters. View the limited network resources, it was necessary to find a mechanism for a QoS management within a network. To solve this problem a various models have been implemented to ensure QoS whithin the same network or in intra-domain case. However, in Internet which is an inter-domain network the problem is not resolved yet. In this context we present this paper which describes in details the new method that we prorposed in [1]. This method ensures the end-to-end QoS constraints for services across multiple domains. Services involved in our approach include real time services such as voice and video telephony and conference, as well as services those requiring high capacity interconnections like links between scientific sites or cloud services, which are provided by different domains or AS's.

J. Świątek et al. (eds.), *Advances in Systems Science*,
Advances in Intelligent Systems and Computing 240,
DOI: 10.1007/978-3-319-01857-7_69, 

The remainder of this paper is organized as follows. In section 2 we present related works and define the inter-domain QoS problem. Next in section 3, we describe our approach that ensures end-to-end QoS over multiple domains. Then in section 4, we present a simulation of our approach using DiffServ model. And finally, in section 5, we conclude this paper and give future works.

## 2 Related Works and Inter-domain QoS Problem

### 2.1 Inter-domain Problem

Several solutions and technologies have been proposed and implemented to provide QoS within the same domain (AS), such as IntServ (Integrated Services) [2] model, DiffServ (Differentiated Services) [3] model or even MPLS [4]. However, a serious problem is posed when the traffic crosses another domain (AS). This problem is mainly due to the fact that QoS constraints, required by the client and which the operator undertakes to provide (usually specified in the Service Level Agreement, SLA), are defined in the classes of service. While the definition of the classes of service is assured by the domain administrator, they are consequently specific to each domain, and are valid only within this domain. In this case, in the transition to another domain the QoS constraints offered to the traffic will not be the same as in the source domain, therefore the QoS required by the client at the beginning will not be provided from the end-to-end until its destination.

### 2.2 Related Works

A various studies and several solutions have been proposed to ensure QoS in inter-domain; each solution suggests a specific approach to treat the problem. Among these solutions, is an extension of traffic engineering in MPLS (Multi-Protocol Label Switching) architecture for inter-domain's use called inter-domain MPLS Traffic Engineering [5]. This solution is mainly based on the label's exchange between edge routers, and bandwidth reservations using enhanced version of Resource Reservation Protocol (RSVP). Also, MESCAL project (Management of End-to-end QoS across the Internet At Large) [6] introduces a new architecture for inter-domain QoS management. However, it focuses only on financial management between customers and operators and between operators. Likewise, a complete model has been proposed in [7] to provide management functionality for End-to-End QoS by combining a routing procedure, a common set of QoS operations and an information model. However, it's specific to dedicated point-to-point connections. Authors of [8] and [9]treated the path computation aspect of the inter-domain QoS routing by providing a new algorithm named HID-MCP (Hybrid Inter-Domain Multi-Constraint Path for inter-domain multi-constraint QoS paths computation. Nevertheless, this solution concerned only paths pre-computation or computation, and didn't offer a complete approach to ensure end-to-end inter-domain QoS.

All these inter-domain solutions do not provide to client's traffic the same QoS required as in its source domain. In this context, we introduce this paper which presents a solution that offer to client's traffic the same QoS constraints even in passing to another domain.

## 3 Proposed Solution Description

### 3.1 Approach Definition

To solve the problem mentioned above, and to ensure continuity of QoS constraints offered to the client even after the transition to other domains, we introduce a new method that provides a new mechanism for inter-domain traffic treatment. The basic idea in our approach is to designate in each domain a server responsible for the management of the different classes of service, named the **Class Manager** (CM). On this server we define a table, named **Class Table** (CT) that contains all information concerning the different classes defined in this domain (such as bandwidth, loss rate, delay, etc.). Once the CM of each domain filled its CT, it sends it to the neighbouring domain. In this way, each CM has all the information about its neighbours classes of services, and then, upon receiving a packet from the neighbouring domain, the router in the current domain can classify it in a class that has the same characteristics as the source class. In this manner, the client flow retains the same QoS constraints throughout its path to the destination, and receives the same treatment from end-to-end.

### 3.2 CT Table Structure

The class table is structured according the following fields:

1. AS number: to identify domain associated with the class.
2. Class number: to identify the class of service.
3. Bandwidth: to indicate the percentage of bandwidth allocated to the class.
4. Priority: to specify the priority level of the class.
5. Queue-limit: to specify the maximum number of packets that the queue can hold for this class.

We note that, to ensure a certain correspondence between the CT tables of the different domains, we define in the CT table only class parameters common between various router's constructors, which are basic parameters used by the different constructors to characterize a class of service, other parameters more specific and appropriate for each router's constructor are not considered in the CT table. The parameters used in the CT table must be specified in the agreement established between the domains as we will explain later in this paper.

### 3.3 Sending Information from Routers to CM Sever

As we have already mentioned, routers classify the customer traffic by applying mechanisms of the adopted QoS intra-domain model. Informations concerning parameters relatives to every class defined on a router are in the router configuration file. We propose that the routers execute a PerlScript to retrieve information concerning classes of service from the configuration file, and to place them in a new file. This file will be sent to the CM server. Once the CM server receives all routers files, it regroups them in a file named CT, that represent the class table in which are stored informations concerning all classes of service defined in the domain.

### 3.4 Exchanging Tables between CM Servers

The communication between all domains CM servers uses the TCP protocol. So, in order that a CM sever cen send its CT table to the neighbouring domain CM server and receive its CT, they establish at first a TCP session. Once the session TCP is established, the first message exchanged between both CM servers is the identification message, which allows each CM server to become identified by its neighbour, by sending its IP address and AS number. The identification message format is presented in the following figure:

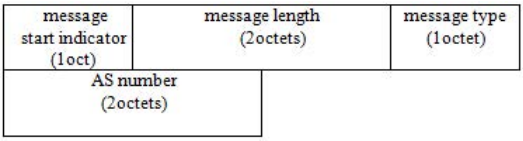


**Fig. 1.** Identification Message Format

After the identification, CM servers exchange their CT tables by sending a set of messages to announce their classes of services, called announcement messages. Every message contains various parameters values relatives to every class defined in the domain.

The format of every message is as follows:

Information contained in every message as soon as it's received by the CM server it's registered in its CT table. This way when the CM server receives the totality of messages, it will have all information concerning all classes defined in the neighbouring domain.

The last type of message is the update message, which is sent by a CM server when there is an addition or modification of a class of service defined in its domain. The update message has the same structure as the announcement message.

Once a CM server receives its neighbour CT table, it diffuses it to the routers of its domain. Hence, all domain routers will possess all information about classes

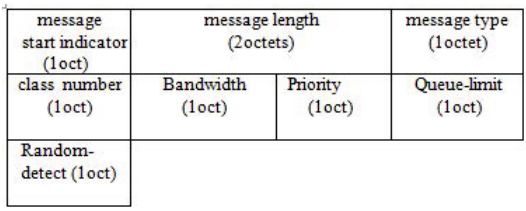


**Fig. 2.** Announcement Message Format

of service defined in the neighbouring domain, and can use this information to create and configure classes of service which will have same values of QoS parameters. According to these classes of service packets coming from the neighboring domain will be classified with the same QoS constraints and will be forwarded in the current domain.

### 3.5 Agreements between Domains

The proposed solution is mainly based on agreements established between domains. Indeed, the information exchanged between domains in CT tables is very important and very sensitive information and the domain administrators have to negotiate and establish an agreement that will manage relations between domains so that the exchange of CT tables takes place with no problem. The agreement also defines how the table's exchange will be charged.

## 4 Simulation Using DiffServ Model in Intra Domain

After the description of our approach in the previous sections, we present a simulation of a sample application to better understand the approach and its operation, and also to prove the efficiency of its principle. In this example we treat the case where the network uses DiffServ model to provide QoS in intra-domain and we consider a client with sensitive, important and expensive application that needs to use the resources in the neighbouring domain. Firstly, we will briefly present DiffServ model, to describe then the architecture and the scenario of the simulation.

### 4.1 DiffServ Model

DiffServ is a service model that ensures the QoS requirements in a network. The client's flows in a network are treated by creating differentiated service classes with different priorities[10]. The main advantage of DiffServ over other models, is its simplicity and robustness especially in large-scale implementations. This robustness is due to the fact that in DiffServ are two types of routers: routers in the network core (core router) and the edge routers, only the edge

routers that handle complex treatment of the flows that require resources and consume bandwidth and time. The edge routers perform classification, control and marking operations, and calculate an 8-bit DSCP (DiffServ Code Point) label that indicates the packet's class of service[11].

As mentioned above, we consider a client with a sensitive, important and expensive application that needs to use the resources in the neighboring domain. The use of DiffServ model allows classifying the client traffic in a class of service that responds to all the required QoS constraints, but only within its AS source. However, in some cases, this sensitive traffic must pass to the neighbouring domain and then, it loses the QoS values assigned to it in its own domain as agreed. The Requiered QoS is not provided from end-to-end. Obviously, in DiffServ model, the definition of classes of service is assured by the domain administrator, so, they are specific to each domain, and are valid only within this domain.

### 4.2 Simulation Topology Description

To solve the problem mentioned above we use our approach proposed in Section 3. Then, this simulation objective is to show performances of this approach, which consists in the fact that the user traffic is classified in classes of service with the same parameters even if they are located in two different domains. For that, we use the network simulator ns2 to compare two simulation scenarios, in both cases we consider two networks that use the diffserv model for QoS management in intra-domain, in the first scenario the two networks use classes of service with different parameters (it is the case in conventional networks), and in the second case both networks use same parameters for their classes of service (which is the principle of our new method). The topology we simulate is presented in figure 3.

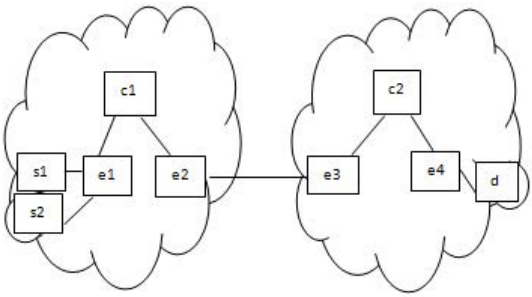


**Fig. 3.** Simulation Topology

Simulation parameters in the first case are the following: On the node s1 tcp agent is configured to emit ftp traffic, and on the node s2 an udp agent is configured to send cbr traffic. In the first network we define two classes of service,

the first one with the DSCP code 10, in which we classify the tcp traffic and the second with the code DSCP 11 in which we classify the udp traffic. The queue size of the two classes is 50 packets, they have two levels of priority (virtual queue), and a token bucket policer with CIR=100 kbps(Committed Information Rate) and CBS=10bytes (Committed Burst Size) for the first class, and CIR=300 kbps and CBS=40 Kbytes for the second class. In the second network we also define two classes of service but with different parameters, the first with DSCP 10 in which we classify the tcp traffic and the second with DSCP 11 in which we classify the udp traffic. The queue size of both classes is 20 packets, they have a two levels of priority (virtual queue), and a token bucket policer with CIR=1 Mbps (Committed Information Rate) and CBS=3 Kbytes (Committed Burst Size) for the first class, and a CIR=3 Mbps and CBS=10 Kbytes for the second class.

Simulation parameters in the second case are the following: On the node s1 tcp agent is configured to emit ftp traffic, and on the node s2 an udp agent is configured to send cbr traffic. In each network we define two classes of service; the two classes defined in the first network have the same parameters as those defined in the second one (to respect the principle of our method). The first class is defined with DSCP code 10 in which the tcp traffic is classified and the second class is defined with DSCP code 11 in which the udp traffic is classified. The queue size of both classes is 50, they both have two levels of priority (virtual queue), and a token bucket policer with CIR=100 kbps and CBS=10 bytes for the first class, and CIR=300 kbps and CBS=40 Kbytes for the second class.

### 4.3 Simulation Results

By simulating the architecture already presented with the parameters that we have detailed above, we obtain the results listed in figure 4. These results concern the end-to-end calculation of three main parameters to estimate the network performances; the throughput, the delay and the loss rate. By analyzing the results presented in the figure 5, which represent the average values of the different calculated parameters, we note that the use of the new method principle (the second simulation case) decreased significantly the end to end throughput, which means a decrease of the end to end link utilization rate for both types of traffic (tcp and udp) that allows a better optimization of network resources while improving the conditions for routing traffic since delay and loss rate also decreased. We also plot the instantaneous variation of the previous parameters, to compare the two scenarios.

The figures 5 and 6 represent the throughput variation in function of time. During all the duration of simulation (6 seconds), we note that the throughput values in the second case of simulation (which represents the new method) are lower than those of the first case (which represents an ordinary network). According to the figures 7 and 8 we also note a significant decrease in the instantaneous loss rates values for both types of traffic (tcp and udp) comparing the second case simulation with the first case.

| | Throughput | | Delay | | Loss Ratio | |
|---|---|---|---|---|---|---|
| | TCP (FTP) | UDP (CBR) | TCP (FTP) | UDP (CBR) | TCP (FTP) | UDP (CBR) |
| Usual network | 1287,79 | 6260,14 | 0.0238 | 0.0313 | 3.26087 | 39.8078 |
| With new method | 815,851 | 2996,05 | 0.0061 | 0.0129 | 1.39 | 18.05 |

**Fig. 4.** End to End Average QoS Values

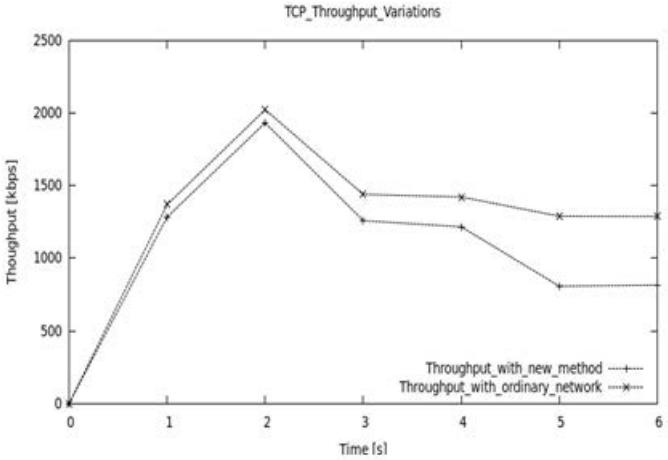


**Fig. 5.** TCP Instantaneous Throughput Variations

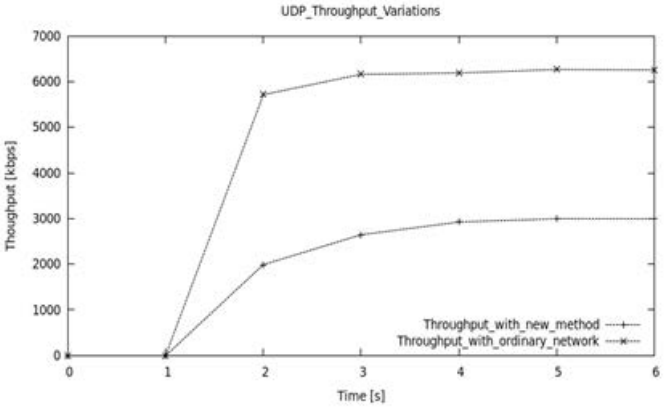


**Fig. 6.** UDP Instantaneous Throughput Variations

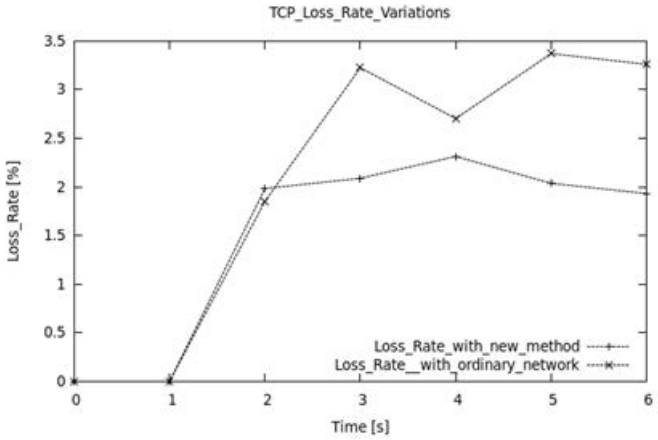


**Fig. 7.** TCP Instantaneous Loss Ratio Variations

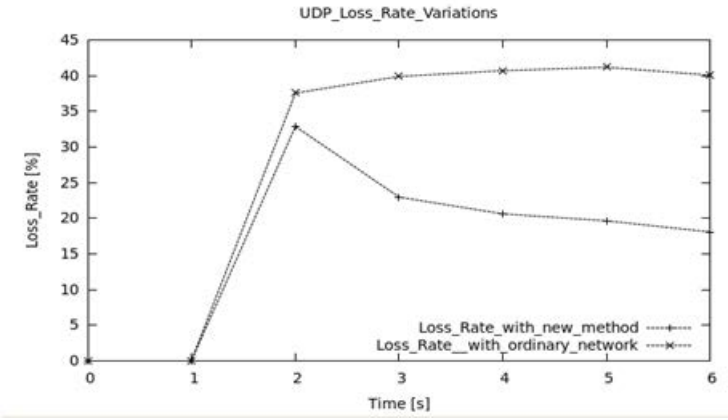


**Fig. 8.** UDP Instantaneous Loss Ratio Variations

We note that the simulation results are illustrative, and allow us to deduce that the use of our new method principle; which consists on keeping the same QoS parameters even in another domain; has improved network performance by reducing the delay and loss rate and also has ensure a better optimization of network resources by reducing the utilization rate of the end to end link.

## 5 Conclusion and Future Work

In this paper, we proposed a new mechanism which ensures end-to-end QoS over multiple AS. We described it and we gave details of its operation and its components. We gived then a simulation of this approach using the DiffServ model for

providing QoS in intra-domain network. Simulation Results show the improvement of network performance when using our mechanism and of course, traffic keeps the same QoS provided by its own domain when it is destined to another AS. The next step of our research will focus on evaluating performance of the proposed approach in other environments by taking into account various models proposed in intra domain, and also on studying and proposing a mechanism for securing this approach.

## References

1. Bakkali, S., Benaboud, H., Ben Mamoun, M.: On Ensuring End-to-End Quality of Service in Inter-Domain Environment. In: Gramoli, V. (ed.) NETYS 2013. LNCS, vol. 7853, pp. 326–330. Springer, Heidelberg (2013)
2. Shenker, S., Partridge, C., Guerin, R.: Specification of Guaranteed Quality of Service. IETF Informational, RFC 2212 (September 1997)
3. Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: An Architecture for Differentiated Services. IETF Informational, RFC 2475 (1998)
4. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. IETF Informational, RFC 3031 (2001)
5. Farrel, A., Vasseur, J.-P., Ayyangar, A.: A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering. IETF Informational, RFC 4726 (2006)
6. Howartha, P., Boucadairb, M., Flegkasa, P., Wanga, N., Pavloua, G., Morandb, P., Coadicb, T., Griffinc, D., Asgarid, A., Georgatsosen, P.: End-to-end quality of service provisioning through inter-provider traffic engineering. Computer Communications 29, 683–702 (2006)
7. Yampolskiy, M., Hommel, W., Danciu, A., Metzker, G., Hamm, K.: Management-aware Inter-Domain Routing for End-to-End Quality of Service. International Journal on Advances in Internet Technology 4, 60–77 (2011)
8. Frikha, A., Lahoud, S., Cousin, B.: Hybrid Inter-Domain QoS Routing with Crankback Mechanisms. In: Balandin, S., Koucheryavy, Y., Hu, H. (eds.) NEW2AN 2011/ruSMART 2011. LNCS, vol. 6869, pp. 450–462. Springer, Heidelberg (2011)
9. Frikha, A., Lahoud, S.: Hybrid Inter-Domain QoS Routing based on Look-Ahead Information. IRISA's Interne Publications, PI 1946 (2010)
10. Serban, R.: La gestion dynamique de la qualit de service dans linternet, archives inria el-00408686, version 1, Universit de NICE SOPHIA-ANTIPOLIS UFR SCIENCES (2003)
11. Nichols, K., Blake, S., Baker, F., Black, D.: Definition of the Differentiated Services Field (DS Field)in the IPv4 and IPv6 Headers. IETF Standars, RFC 2474 (1998)