Available online at www.sciencedirect.com
SCIENCE DIRECT
Computer Communications 28 (2005) 2020–2033
computer communications
www.elsevier.com/locate/comcom

A measurement-based approach for dynamic QoS adaptation in DiffServ networks

Toufik Ahmed[a,b,*], Raouf Boutaba[a], Ahmed Mehaoua[b]

[a]School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ont., Canada N2L 3G1
[b]CNRS-PRiSM Lab, University of Versailles, 45 av. des Etats-Unis, 78000 Versailles, France

Received 29 July 2004; accepted 29 July 2004
Available online 30 September 2004

Abstract

This paper presents a practical approach to managing multimedia traffic in DiffServ network, using network monitoring feedback and control. We exploit the flexibility of multimedia traffic and process network level parameters to adapt the traffic according to the current state of the network. The latter is determined based on reports sent by bandwidth monitors installed on each node of a DiffServ Domain. The bandwidth monitors interact with a policy server which, depending on the network state, decides the policy(ies) that should be enforced by the DiffServ network. The implementation of the selected policies typically leads to accepting, remarking, or dropping the multimedia traffic entering the network. Multimedia streams may be assigned different levels of QoS, as interpreted by the marker at the DiffServ edge router and marked according to network state. To achieve such dynamic QoS adaptation for multimedia applications, we have implemented and evaluated a policy-based management system. Performance evaluation shows that multimedia applications adapt better to network conditions using our approach.

Keywords: DiffServ; Policy; QoS adaptation; Monitoring

1. Introduction

Recent work on Quality of Service (QoS) Management led to the development and standardization of enhanced protocols and services. The Internet Engineering Task Force (IETF) defined the Policy-based Networking (PBN) framework to enable dynamic network configuration and service provisioning. PBN relies on the use of high-level abstract policies which specify how to dynamically configure a network node in a vendor-independent and interoperable manner. Most of the efforts in this area have focused on the Internet Differentiated Services (DiffServ) architecture.

The DiffServ architecture defines four types of data-path elements: traffic classifiers, actions elements, meters and queuing elements [1]. Combining these elements into higher-level blocks creates a Traffic Condition Block (TCB), which can be configured through PBN. This involves the use of administratively prescribed policies that specify what actions should be executed in response to predefined events. Some of the configuration data required by this process do not change frequently and is hence stored in a repository at the level of the policy server. Examples include user information and infrastructure data such as network addresses and name server information. Some other application or traffic specific configuration data changes frequently and is hence more difficult to maintain. This is the case for the data required to mark the traffic entering the network (audio, video and data traffic) appropriately. Marking the traffic entering a DiffServ domain commonly consists to set the DiffServ Code Point (DSCP) field in the header of these traffic packets appropriately. The IP address, the port number or a combination of these is generally used to identify the traffic entering the network. The network administrator needs only to specify the traffic management

* Corresponding author. Address: CNRS-PRiSM Lab, University of Versailles, 45 av. des Etats-Unis, 78000 Versailles, France. Tel.: +33 1 39 25 40 76; fax: +33 1 39 25 40 57.
E-mail addresses: tad@bbcr.uwaterloo.ca (T. Ahmed), tad@prism.uvsq.fr (T. Ahmed), rboutaba@bbcr.uwaterloo.ca (R. Boutaba), mea@prism.uvsq.fr (A. Mehaoua).

0140-3664/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2004.07.043

policies referring to the traffic in question, e.g. using address information, and specifying how this traffic will be marked when entering the DiffServ domain. The user in-profile traffic is marked and treated in the network according to the user profile negotiated when subscribing to the service. Out of profile traffic (traffic in excess of agreed upon user profile) in turn is dropped or marked as best effort traffic. This approach is static and therefore does not address the application requirements as a function of the changing network state. Also, the static nature of this approach may lead to inefficient utilization of the network resources.

In this paper, we define a policy based framework for dynamic bandwidth allocation based on network state and application QoS requirements. The implemented system allows to dynamically configuring Diffserv routers with adequate marking and traffic shaping policies to accommodate multimedia traffic with varying QoS requirements. The traffic management policies are dynamically decided and enforced in the DiffServ network thanks to a PBN infrastructure and to network feedback provided by bandwidth monitors installed in the network. This proposed dynamic bandwidth management approach improves system responsiveness to network events (e.g. congestions) and allows for better QoS adaptation depending on multimedia applications characteristics.

This paper presents our approach, its implementation and performance evaluation. It is organized as follows. Section 2 presents the background and related works. Section 3 presents our framework for dynamic QoS policy decisions. The system implementation and experiments are described in Sections 4 and 5 respectively. Section 6 presents the conducted performance evaluation. Finally, Section 7 concludes this paper.

2. Background and related works

In this section, we describe some background related to DiffServ architecture and policy-based network management.

2.1. Differentiated services

Quality of service provision within the Internet has been the subject of significant research and deployment efforts recently. IETF has concentrated its effort on two approaches: IntServ and DiffServ.

The Integrated Service (IntServ) model is motivated by the desire for applications to choose among multiple, controlled levels of delivery service for their data packets [2]. The integrated service framework defines two classes of service, the Controlled-Load [3] and Guaranteed [4] and relies on a resource reservation protocol such as RSVP [5].

The Differentiated Services (DiffServ) model [1] uses a small, but well-defined set of building blocks from which a variety of aggregate router behaviors may be designed to provide quality of service [6]. IP packets are tagged with distinct labels before entering an IP DiffServ domain and will receive particular forwarding treatment at each network node along the path. This set of routing functions is called Per-Hop Behavior (PHB). The PHB is characterized and established according to the Differentiated Service Code Point (DSCP) value located in the packet's header. Currently, a small number of PHBs has been standardized by the IETF DiffServ working group. The most well known are Expedited Forwarding (EF) [7] and Assured Forwarding (AF) [8].

The key difference between Intserv and DiffServ is that Intserv provides end-to-end QoS service on a per-flow basis while DiffServ offers better scalability through flow aggregation and class differentiation over large timescales.

2.2. Policy-based network management (PBNM)

PBNM is a software tool used for managing network resources to provide QoS in IP networks. In the Differentiated Services framework, it is used for configuring DiffServ routers in an administrative domain. The tool provides the means to allocate resources to a particular user as specified in the Service Level Agreement (SLA) with this user. It consists of the following elements (Fig. 1):

- Policy editing console: allows the network administrator to define and to edit the policies applicable in her/his administrating domain. A Web-based console is commonly used here.
- Policy decision point (PDP): a policy server that retrieves policies from a repository and makes decisions on behalf of Policy Enforcement Points (PEPs). Policy decisions enable service differentiation, setting of QoS configuration, QoS provisioning, and efficient use of bandwidth.
- Policy enforcement point (PEP): a network device, such as a router, a switch or a firewalls that enforces policy decisions using access control lists, queue management algorithms and other means. It receives configuration policies from the PDP using the COPS protocol [9] (COPS-RSVP [10] or COPS-PR [11]).
- Policy repository: a Lightweight Directory Access Protocol (LDAP)–compliant directory server where the policies are stored.

2.3. Measurements

Measurement is a crucial function for Internet traffic engineering and network management. A framework for traffic engineering in IP-based networks is presented in [12]. Different types of measurements have been identified and are either passive or active. Passive measurement gathers the statistics of the network from Management Information Bases (MIBs), whereas active measurement injects test packets into the network (e.g. ping packets). Information obtained from these packets are taken as representative of

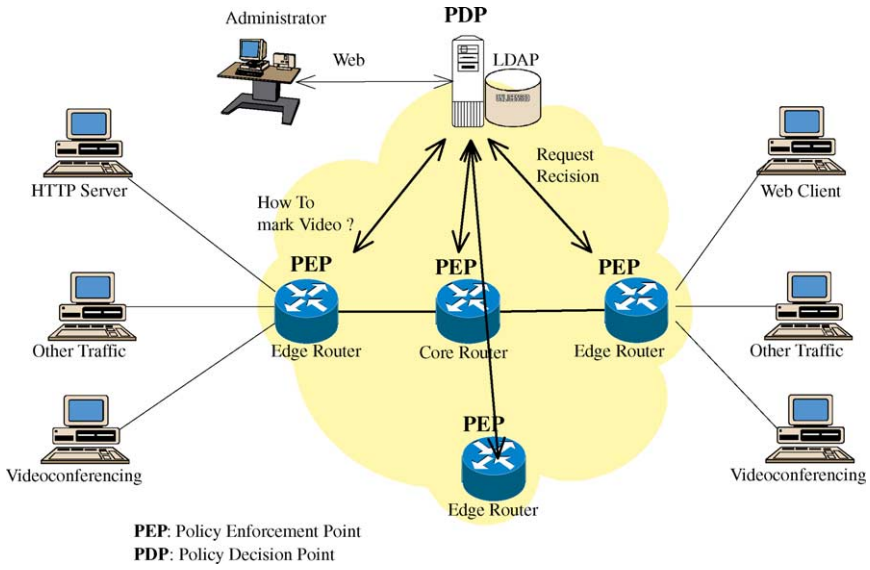

Fig. 1. Policy-based network management.

the network behavior. Several metrics are identified and described in [13] and include:

- Flow-based: gathers information about each flow in the network. This technique is not scalable.
- Interface-based, link-based and node-based: collects information on each interface in the network element.
- Node-pair-based: calculates the performance of the network edge-to-edge.
- Path-based: it is similar to pair-based measurement but operates on a particular path. It is used generally for admission control.

Our emphasis lies on QoS measurements, particularly on the bandwidth utilization metric. The later is chosen for its protocol and media independence.

Scalability is the foremost issue in monitoring bandwidth usage on a particular path. Considering the large size of the Internet, the complexity of path computation and the required amount of information exchange and maintenance are unmanageable. The scalability issue is addressed here through aggregation and hierarchical measurements. Fig. 2 illustrates how hierarchical measurement estimates the traffic matrix for a large ISP. Each router performs passive monitoring of incoming traffic (i.e. BW_{ij}: bandwidth usage

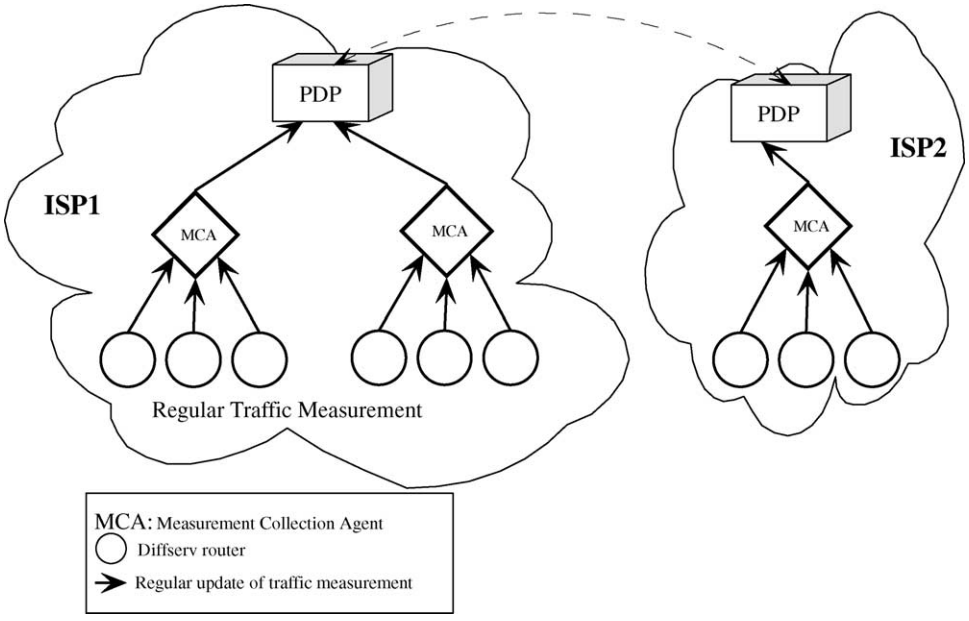

Fig. 2. Hierarchical aggregation of measurement.

for router *i* on its interface *j*). Through regular update, each router provides a partial view of the network. MCA (Measurement Collection Agent) aggregates this partial measurement and forwards the result to the PDP. The PDP combines all of the measurement information into a matrix which provides a global view of the network and from which meaningful feedback information can be generated.

The feedback information is used for dynamic bandwidth allocation to traffic streams. This is achieved by deciding and enforcing the proper Diffserv marking policies for these streams.

3. Dynamic QoS adaptation

Our approach for configuring a DiffServ network consists in dynamically adjusting internal router configuration (filter, scheduler, meter) to deliver QoS to multimedia streams. This is achieved by the monitoring and reporting information sent by monitoring agents installed on each network element. This section first discusses static and dynamic policy decisions. Afterwards, we present an example of our proposed configuration. Finally, we present our QoS management algorithm.

3.1. Static policy decision

A traffic stream receives a predefined treatment based on predefined policies. This treatment is interpreted in a Diffserv network as a PHB [7–8]. This task is done by the TC (Traffic Control) function, which assigns the correct DSCP to the client's traffic according to it SLA [6]. Recall that each client defines it requirements and these are translated into a SLAs. The allocation of resources (QoS) remains static and can lead to bandwidth wastage.

Algorithms such as *Time Sliding Window Three Colour Marker* (TSWTCM) [14] and a *Two Rate Three Color Marker* (TRTCM) [15] can be used to mark IP packets processed by the edge router to receive a particular PHB. Such algorithms meter the traffic stream and mark packets based on the measured throughput rather than the characteristics of the traffic (delay, loss, jitter, etc.). Instead, in our architecture resources are allocated to traffic streams based on the nature of the information being transmitted.

To receive a particular treatment, the user specifies her/his profile **TSpec** (Traffic Specification). TSpec specifies the temporal properties of a traffic stream selected by a classifier. It provides rules for determining whether a particular packet is in profile or out of profile. The Meter uses a Token Bucket to control user traffic. The following is a non-exhaustive list of profile parameters:

1. *Token bucket rate r* (bps): the rate at which the tokens are accumulated in the bucket.
2. *Bucket depth b* (bytes): the bucket size.
3. *Peak rate p* in bits per sec (bit/s): defines the maximum rate at which packets can be sent in short time intervals.

An *Excess Treatment* parameter describes how the service provider will process excess traffic (i.e. out of profile traffic). The process takes place after Traffic Conformance Testing. Excess traffic may be dropped, shaped and/or remarked. Depending on the particular treatment, more parameters may be required (e.g. the DSCP value is needed for re-marking and the shapers buffer size is needed for shaping). All of these actions are predetermined once the network element is configured, and these actions do not change over time. Fig. 3 gives an example of how user traffic is treated using static policy configuration. In this example, the user sends traffic which does not conform to her/his Traffic Specification. Edge router controls this traffic using a token bucket. Non-conforming traffic will be dropped always. For the conforming traffic, the appropriate marking is done by the edge router.

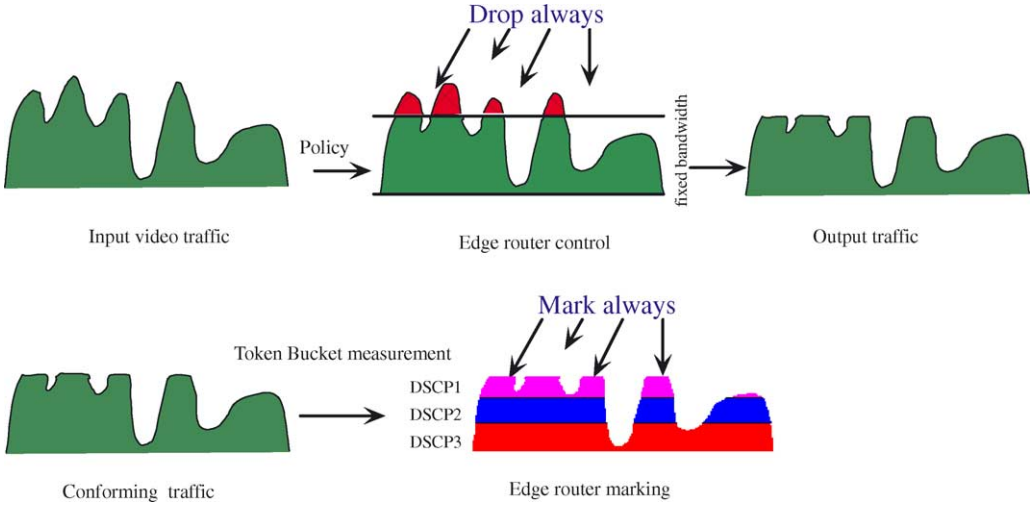

Fig. 3. Static policy decision.

More flexible resource allocation can be achieved by dynamically determining the control action to be performed by the network element. This way different traffic conditioning actions may be performed on the in profile traffic and out of profile packets. For example, control actions may be decided dynamically according to current network status. This approach promotes network adaptation to application characteristics in addition to adapting applications to network conditions as proposed in works [16–18].

3.2. Dynamic policy decision

In the static approach, out of profile traffic is simply dropped, remarked or assigned a new profile. These actions are decided when determined when the network element is configured. For example the *Policing Rule = drop out of profile* packets can be applied to all the packets which are out of profile regardless of whether the network is capable or not of transmitting these packets. Conforming traffic is always marked with the same way because of the token bucket-based marking use at the edge routers.

Fig. 4 shows example actions decided for application on out of profile and in profile packets. These actions depend on the network state (e.g. link utilization).

3.3. Self-configuration of DiffServ domain

When a network element is started, its local PEP requests from the PDP the policies concerning DiffServ traffic marking. The PDP may also proactively provision the PEP, in reaction to external events such as those generated by the bandwidth monitor.

Fig. 5 shows the steps involved in the configuration of a DiffServ domain. These steps are as follows:

Step 1: On edge router initialization, the local PEP requests from the PDP all policy decisions concerning DiffServ QoS Management (filtering, classes, queuing discipline, and actions for out of profile traffic). All incoming packets are processed according to these pre-installed policy decisions.

Step 2: When the bandwidth monitor located on the core router detects a significant change in the available bandwidth, it informs the PDP of the current bandwidth availability.

Step 3: The PDP makes new QoS management decisions and transmits the corresponding policies to the edge router PEP.

These steps allow appropriate configuration of different policies for the same traffic.

We introduce the following policy rule: On event: If <profile> then <action>.

- A profile is used to determine when a policy rule applies to a particular traffic.
- An action is performed by the PEP to any traffic with a given profile. Examples of actions are marking, accepting and rejecting traffic.

Example of policy rule:

- Rule 1: Mark DSCP value EF on all packets with source addresses from 193.51.25.1 to 193.51.25.255 with priority 0
- Rule 2: Mark DSCP value AF11 on all packets with destination address 200.200.200.100 with priority 1

3.4. Example of application

Assume that an audio application has subscribed to a given DiffServ class (an Expedited Forwarding Class). The audio traffic is defined by a particular profile. In this example, DiffServ class simply means that the audio stream will be marked with the appropriate DSCP (EF PHB).

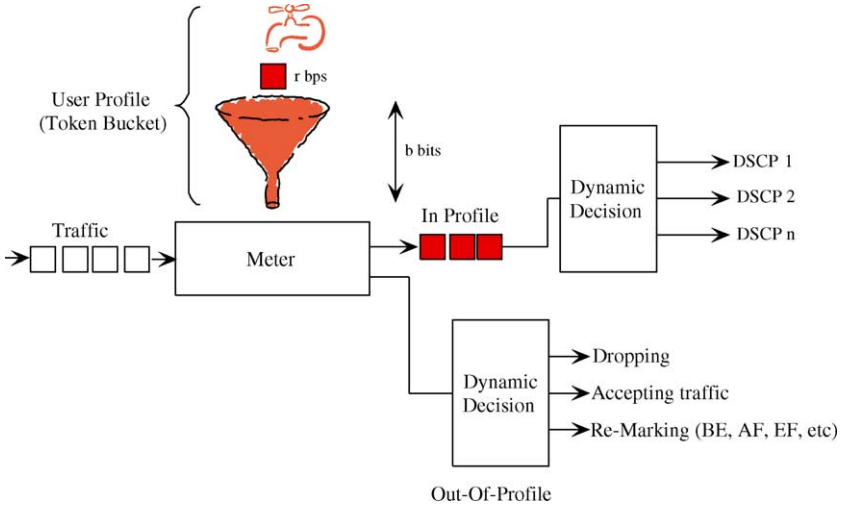

Fig. 4. Dynamic policy decision.

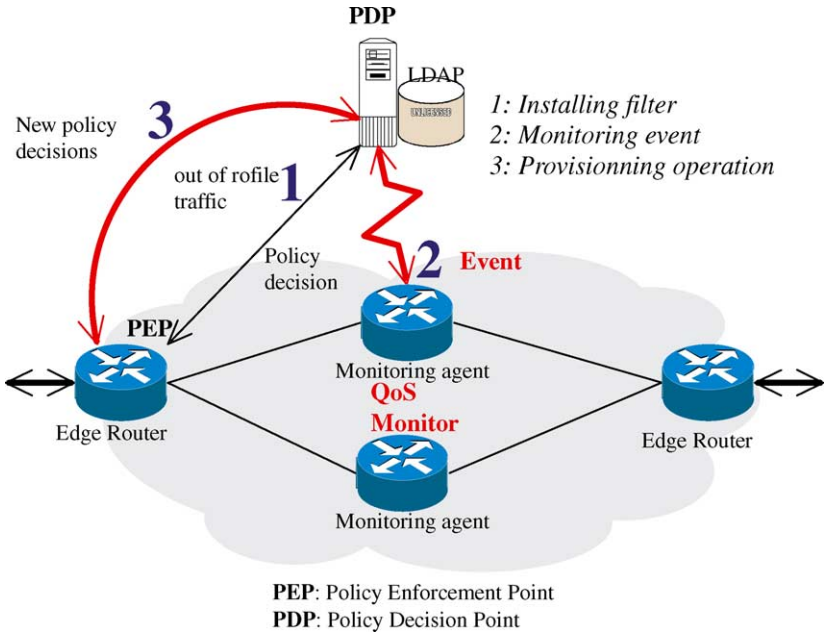

Fig. 5. COPS-PR with monitoring event.

The Administrator of the DiffServ domain configures the environment to support the Gold, Silver, Bronze and other services. Such configuration can be done through a Bandwidth Broker.

Supporting different classes of service in the core network requires putting in place classifiers, which cause the devices to examine the DiffServ mark on the packet and then treat the packet accordingly.

These configurations do not change frequently because they are not associated with specific application or traffic but with the network management.

When the audio application signs up for the service, the edge router is configured to mark the application's traffic with the appropriate PHB. Based on the IP address and/or the port number, the administrator sets a policy that marks traffic coming from that address with EF PHB.

In order for incoming customized traffic (to the audio application) to receive DiffServ treatment (e.g. feedback traffic, RTCP, client commands) a policy must be deployed at the opposite edge of the DiffServ domain.

When the audio application starts sending the data, the edge router has to ensure that: (1) the data sent by the audio server does not exceed what the application has subscribed to (SLA) and (2) marking conforming traffic (in profile traffic) with the appropriate PHB (EF PHB in this example). In case out of profile traffic is received, the edge router requests a decision from the PDP. The PDP knows the current network state because it receives monitoring information from the monitors installed in the network. It decides a new policy rule (e.g. dropping, marking or accepting out of profile traffic) based On the current network state.

A similar scenario can be envisaged for compressed video traffic. Assume that a video server sends a unicast multi-layer video stream to a video client. The server sends one base layer and one or several enhancement layers. The enhancement layers improve the video quality but cannot be decoded without the base layer. Using our QoS adaptation, the administrator can configure the edge router to add/drop the enhancement layers depending on the resource availability in the network. The administrator specifies an edge router policy to admit the video base layer before accepting other enhancement layers. Similar mechanisms that adapt video rate to network condition exist such as those described in [19–21]. However, these mechanisms are implemented in the video server, not at the edge router.

3.5. Traffic management algorithm

We define three policy rules: Rule1, Rule2 and Rule3 for handling multimedia traffic (see Fig. 6). The PDP selects one of these rules depending on information periodically received from the monitors. The measurements computed by the laters concern bandwidth usage link utilization in the network. The level of traffic shaping is determined using an *Exponentially Weighted Moving Average* (EWMA) approach, presented in Section 3.6. Our algorithm described below uses the predefined policy rules to make a decision depending on bandwidth availability in the network. This algorithm can be easily adapted for the audio and video applications previously described.

3.6. Computing bandwidth consumption

The EWMA *(Exponentially Weighted Moving Average)* technique can be used to detect uncontrolled behaviors. EWMA is an Exponential Smoothing technique that

```
Initialization:
Start Bandwidth Monitor Mi for each Router i to calculate the available
bandwidth BW_i
Lambda ←— 0.2      // Fixed value for historical data
X ←— 50% link capacity     // Initial value of EWMA
Loop:
      BW ←— max(BW_1, BW_2, …, BW_i)
  //EWMA available bandwidth X
  X ←— (1-lambda) * X + lambda * BW
  if  X < Min_th then
       Rule1: Accept out-of-profile traffic
       Mark   traffic with DSCP1
  else if Min_th<=X<Max_th then
       Rule2: Remark out-of-profile traffic with DSCP2
        Mark in profile traffic with DSCP1
       // Example EF will be remarked to AF11 and so on.
  else
       Rule3: Drop out-of-profile Traffic
        Mark in profile traffic with DSCP2
  End.
End loop
```

Fig. 6. A simple algorithm using dynamic policy decisions.

employs an exponential smoothing parameter to place emphasis on recent observations or past history. Our algorithm uses a EWMA filter to compute bandwidth consumption. Bursty traffic can cause a transient congestion, but the bandwidth consumption is not affected by this since its value is smoothed. EWMA statistics are used to respond dynamically to the changing value of the time series, which is, in our case, the bandwidth consumption measured periodically in bottleneck links.

The collected statistics have the following form: $X_t = \lambda \mathrm{Bw} + (1-\lambda)X_{t-1}$

where:

- X is the mean of historical data (target result)
- Bw is the observation at time t (current bandwidth consumption)
- $0 \leq \lambda \leq 1$ is a constant that determines the importance of historical data in the EWMA.

Small values of λ (e.g. 0.2) allow to detect small shifts in bandwidth consumption and larger values (between 0.2 and 0.4) for larger shifts [22].

Policy decisions depend on the EWMA statistics computed by each network monitor and sent to the PDP to be aggregated for the whole DiffServ domain.

4. Implementation

Our prototype consists of three modules that perform Dynamic QoS adaptation in a Diffserv domain. These are: a Web-based policy enabled bandwidth broker, a network monitoring system, and a PBN system (i.e. a Policy Decision Point and a collection of Policy Enforcement Points). Fig. 7 shows the main components implemented.

4.1. Web-based policy-enabled bandwidth broker

The administrator uses the web interface to configure the DiffServ domain and to enter a new bandwidth management policy or to edit an existing one. A Java Servlet engine is used to store all the information in a repository. We use an OpenLDAP [23] server running on Linux to handle dynamic web-based configuration. Other optional features, such as validation, verification, and conflict detection are not yet implemented in the current prototype.

Fig. 8 shows a simple web-based interface of the bandwidth broker. It illustrates the edge router configuration, specifically the filter configuration and PHB setting for the traffic entering the network.

4.2. Network monitoring system

Network monitoring provides network status in terms of resource availability. The network monitoring system collects and maintains up-to-date information about network resource consumption/availability.

The system implementation consists of monitoring agents, written in Java, each of which collects information on the interfaces of the associated router. The collected information consists of real-time traffic flow measurements at the input and the output of each interface. This way, the agent augments the functionality of PEP by reporting monitoring information to the PDP in the form of COPS Report State Messages. Based on agents feedback, the PDP delivers to the PEP a set of new policy decisions. Policy decisions are made thanks to the algorithm described in Fig. 6.

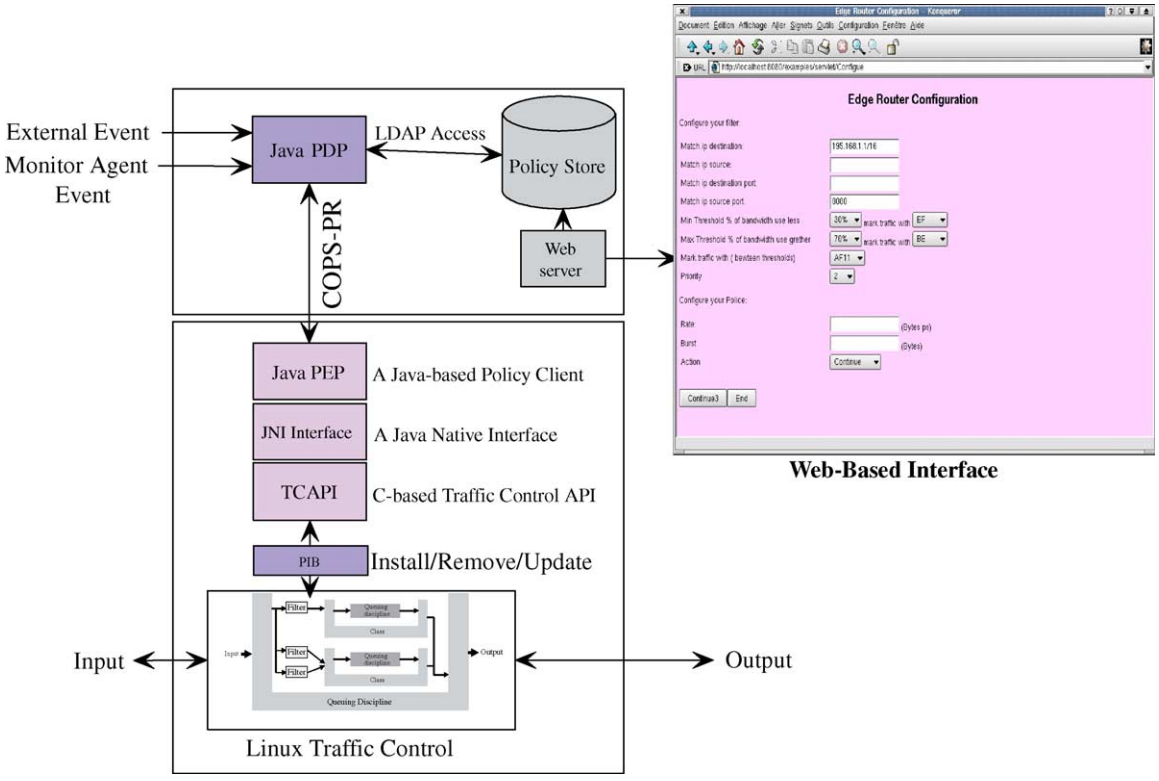

Fig. 7. Implementation architecture.

4.3. The PBN system

This system is composed of a PDP and a set of PEPs communicating using the COPS-PR protocol. All system components are implemented in Java. A PEP is associated with the edge router interfaces where the marking process is performed. The PEP is notified of policy changes via COPS provisioning operation. The received policy information is transformed into a form suitable for the device (e.g. using a Linux DiffServ Traffic Control API). After this, all incoming packets to this device will be marked according to the new policy.

The PDP is responsible for decision making. It uses network feedback to make the appropriate decision. Our implementation is limited to one Diffserv domain.

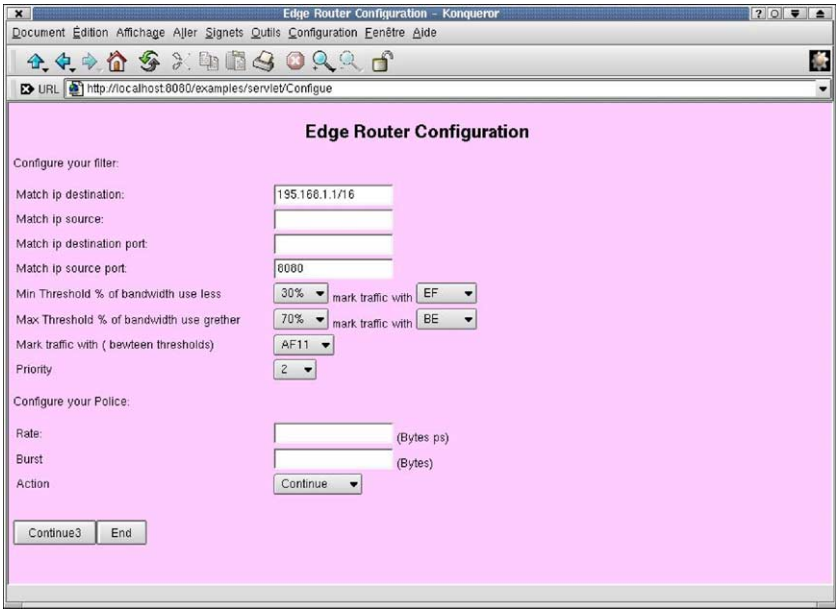

Fig. 8. Snapshot of the bandwidth broker Web interface.

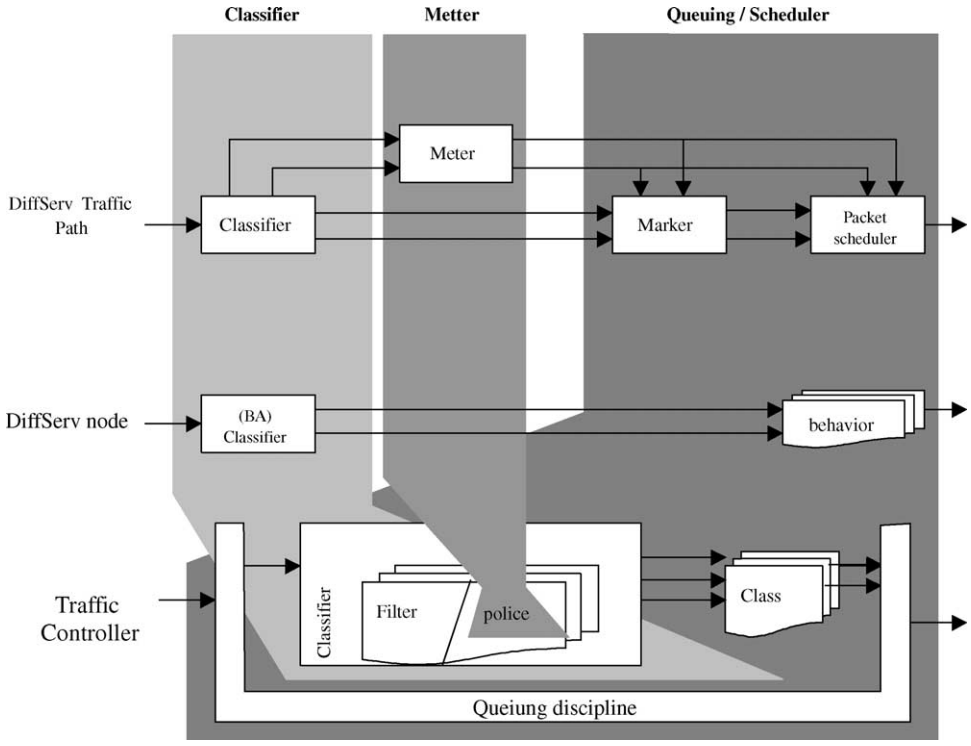

Fig. 9. DiffServ router implementation.

4.4. DiffServ router implementation

The DiffServ router implementation is a based on the Linux traffic control implementation described in [24,25]. Each element of the Diffserv router is viewed as a set of components that can be manage via the interfaces specified in [26–28]. The following elements are included in our router implementation (see Fig. 9):

- Queuing disciplines: the queue determines the order in which data is transmitted. Each network interface has a queuing discipline associated with it, which controls how packets are treated.
- Classes: traffic can be divided into classes according to certain rules. Each class maintains a queuing discipline to serve it packets.
- Filters: to put packets into classes we use filters. Filters are used to distinguish among different classes of packets and process each class in a specific way.
- Policing: used to control the amount of traffic from a given class.

5. Performance evaluation

This section describes our testbed and performance evaluation. The network administrator uses the Web interface of the policy enabled bandwidth broker to configure the edge and core routers according to a predefined set of policies. Suppose that the administrator's domain can handle EF, AF11 and BE class only. The administrator configures the filters accordingly. The task of the filter is to mark the traffic entering the network with the appropriate PHB according to user profile. The administrator also chooses how to handle out of profile traffic by tuning two control thresholds (Min_th and Max_th).

5.1. Experimental testbed

Fig. 10 shows our testbed. The user transmits a customized traffic (multimedia traffic) across a Differentiated Services network. The network is composed of DiffServ capable routers. We use Linux-based IP routers with DiffServ implementation [24–25]. The testbed consists of two edge routers connected through 10 Mbps Ethernet links.

Linux Traffic Control supports several control actions that can be performed on user traffic. These actions are summarized below:

1. *Continue*: can be used to 'Remark' the traffic to another class of service.
2. *Drop*: This is a very aggressive option that simply discards a particular traffic.
3. *Pass/OK*: Pass on traffic. These options may be used to disable a complicated filter while leaving it in place.
4. *Reclassify*: Most often comes down to reclassification to Best Effort. This is the default action.

In the following subsections, we present the configuration and the parameters of edge and core routers in

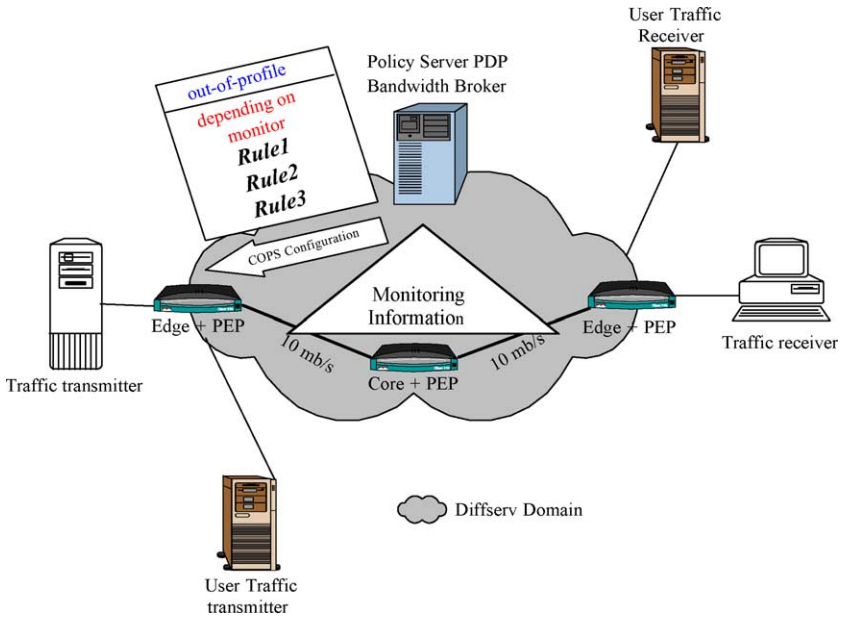

Fig. 10. Experimental testbed.

the testbed. The entire configuration is done using our policy configuration tool.

5.2. Edge router configuration

Edge routers accept traffic into the network. They characterize, police, and/or mark user traffic. Their major task in our experiment is to perform policing of traffic according to the agreed upon SLA.

In our experimental testbed, edge router configuration is simple. Basically, the edge router limits the amount of EF traffic to 15% of the bandwidth capacity rate, i.e. 1.5 Mbit. This parameter can be modified using the Web-based bandwidth management interface. EF is more demanding in terms of latency and packet loss and has been therefore selected for policing. The router must make sure that the departure rate configured for EF is greater than the arrival rate and that the queuing delay is minimized. The EF flow is bounded and isolated in our configuration. For the Edge Router we used a simple CBQ (Class-Based Queuing) [29] discipline to classify the incoming traffic.

5.3. Core router configuration

Core routers are configured to perform packet classification based on DSCP, packet scheduling, and queue management, policing and packet dropping.

We used CBQ as our packet scheduler as proposed in [29]. For CBQ, a single set of mechanisms is proposed to implement link sharing and real-time services. In our implementation, CBQ is used to classify EF, AF, and BE traffic so that each user can get appropriate resources based on packet marking.

Our CBQ mechanisms include:

- A classifier to classify arriving packets. The classification is based on DSCP field in the IP header,
- A scheduler to determine the order in which packets from the various classes will be sent. The Linux Kernel implements several queuing disciplines (e.g. RED 'Random Early Detection' or GRED 'generalized RED'). The GRED queuing discipline is used to support multiple drop priorities as required for the AF PHB group. One physical GRED queue is composed of multiple VQs (Virtual Queues). GRED can operate in *RIO (*RED with In/Out bit) *mode*[30], with coupled average queue estimates from the virtual queues, or in *standard mode* where each virtual queue has its own independent average queue estimate as required for RED [31]. In our testbed, we used GRED as the queuing discipline for AF classes, since our marking algorithm takes into account these properties to give different level of QoS for multimedia traffic.

Using our policy-based bandwidth management system, we allocated 1.5 Mbit/s for each AF class (i.e. AF1, 2, 3 and 4), all of which are bounded. We allocated 3.5 Mbit for the best effort traffic which dis also allowed to borrow any available bandwidth.

6. Performance analysis

In our experiments, a customized traffic (video traffic) transmitted from a video server to video client through a Diffserv network. This traffic crosses a DiffServ network. We load the network using *n* IP traffic generator each one composed of a traffic transmitter and a traffic receiver.

The traffic transmitter generates a UDP packet of 1024 bytes with IP and UDP headers according to a Poisson distribution with parameter $\lambda = 128$ packet/s which gives 1 Mbit/s per traffic generator. In our experiment, and since our Ethernet links are 10 Mbit/s, we have chosen $n = 5$, $n = 7$ and $n = 10$ in order to load the network differently each time. Each source can be either on or off during an exponentially distribution on/off period with an average of $\lambda_{on} = \lambda_{off} = 1$ s.

We compare the above scenario when using our algorithm and when not using our algorithm.

Policing is performed at the edge of the network for video traffic, based on the video server <IP_adr, Port_-number> information. The applicable policy is determined according to the traffic profile Tspec (traffic profile). Tspec takes the form of a token bucket (r,b) and the following optional parameters: a peak rate (p), a minimum policed unit (m), and a maximum datagram size (M).

The token bucket and peak rate parameters require that the traffic obeys the rule that over all time periods, the amount of data sent cannot exceed $M + \min[pT, rT + b - M]$ [32]. M is the maximum datagram size, and T is the length of time period. Datagrams which arrive at an element and cause a violation of the $M + \min[pT, rT + b - M]$ bound are considered out of profile (non-conformant) and require a dynamic decision from the PDP.

6.1. Out-of-profile experiment

In this experiment, we set the parameters for the token bucket to be $r = 1$ Mbit/s and $b = 2$ K, for user traffic. This means that user traffic must not exceed 1 Mbit/s, otherwise considered out of profile.

For testing purposes, we transmit an out of profile traffic (not conform to TSpec). This traffic is at a constant bit rate of 1.5 Mbit/s. The token buckets accept only 1 Mbit/s, therefore, the 0.5 Mbit/s are considered out of profile. The in profile traffic will be marked with EF PHB whereas the out of profile traffic will be marked either by EF or AF11 or dropped (according to the network status).

Fig. 11 shows the network load during the period of the experiment (180 s). This load represents the traffic sent from the n traffic generators to the receivers. This measure has been taken from the ingress interface of the core router, which corresponds to the bottleneck link. During to first 60 s there are only $n = 5$ traffic generators that can be either on or off. From period 60 to 120 s there are $n = 7$ traffic generators. In the last 60 s (from 120 to 180 s) the number of the traffic generators is $n = 10$.

The PDP makes the decision according to the smoothing value of the bandwidth usage (i.e. EWMA). This decision is a policy rule sent directly to the edge router of the DiffServ network.

In our experiments, we set the value of Min_th = 4 Mbit and the value of Max_th = 7 Mbit. These two values help us determining network congestion level to generate the feedback (please refer to the algorithm described in Fig. 6). The read time of the bandwidth consumption performed by the bandwidth agent is set to 1 s.

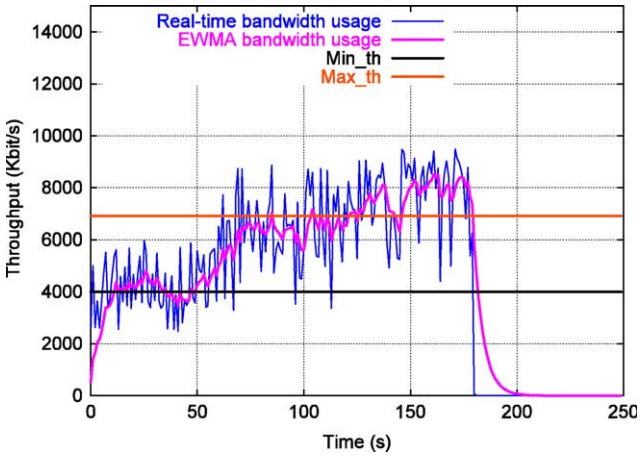

Fig. 11. Bandwidth usage in Core Router.

The events sent by the PDP to the edge router are listed below with the corresponding timestamps (see Table 1).

These events show how traffic is subject to a dynamic behavior in the network. This is an interesting function, since it allows an Internet Service Provider to decide new traffic engineering strategies easily.

Fig. 12 shows the received user traffic with the different PHB colors. In-profile traffic (1 Mbit/s) is always marked as EF whereas the out of profile traffic (0.5 Mbit/s) is dynamically accepted as EF, accepted as AF11 or dropped.

6.2. Marking/remarking strategy for multimedia traffic

In this experiment, we configure the DiffServ edge router to perform marking/remarking for multimedia traffic (MPEG-2 Video). As discussed previously, multimedia

Table 1
List of policies sent by the PDP

Time (s)	Action taken by the edge router (Policy)
0	Rule1: Accept out of profile traffic (EF traffic)
12	Rule2: Remark out of profile traffic with AF11
37	Rule1: Accept out of profile traffic (EF traffic)
38	Rule2: Remark out of profile traffic with AF11
40	Rule1: Accept out of profile traffic (EF traffic)
47	Rule1: Accept out of profile traffic (EF traffic)
103	Rule3: Drop out of profile traffic
105	Rule2: Remark out of profile traffic with AF11
107	Rule3: Drop out of profile traffic
109	Rule2: Remark out of profile traffic with AF11
110	Rule3: Drop out of profile traffic
111	Rule2: Remark out of profile traffic with AF11
112	Rule3: Drop out of profile traffic
116	Rule2: Remark out of profile traffic with AF11
117	Rule3: Drop out of profile traffic
141	Rule2: Remark out of profile traffic with AF11
144	Rule3: Drop out of profile traffic
177	Rule2: Remark out of profile traffic with AF11
179	Rule1: Accept out of profile traffic (EF traffic)

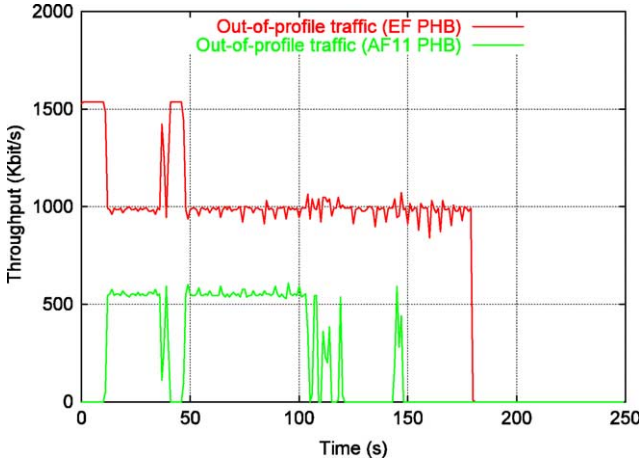

Fig. 12. Received audio traffic with PHB color.

applications generate traffic at varying rates and have a varying ability to tolerate delays and jitter in the network. This flexibility and tolerance with respect to network parameters such as packet loss, delay and jitter are interpreted as a particular PHB invoked by a particular DSCP. Our marking/remarking strategy exploits this flexibility of multimedia flows to adapt the flows based on the state of network and the availability of resources. This action maintains a quantifiable level of QoS even when network conditions are not favorable.

A video profile is declared in the edge router as a token bucket regulated: $r=600$ Kbit/s and $b=2$ K. This means that video traffic must not exceed 600 Kbps otherwise it will be considered as out-of-profile traffic.

For the purpose of this experiment, we transmit a high quality MPEG-2 video stream to see our system reacts. Fig. 13 shows the MPEG-2 video stream sent by the video server. The average rate of this video is 800 Kbps and the peak rate is 1.4 Mbps. Video traffic is not conform to the traffic specification, the excess traffic is considered out-of-profile. The edge router marks the video traffic according to the dynamic policy provisioning. In profile traffic is marked

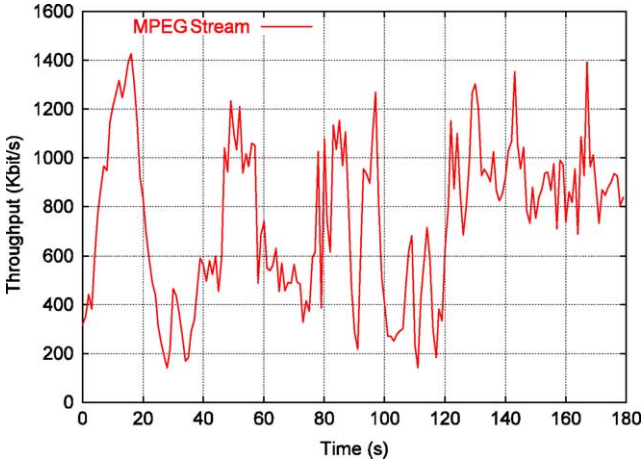

Fig. 13. MPEG-2 video stream sent by the user.

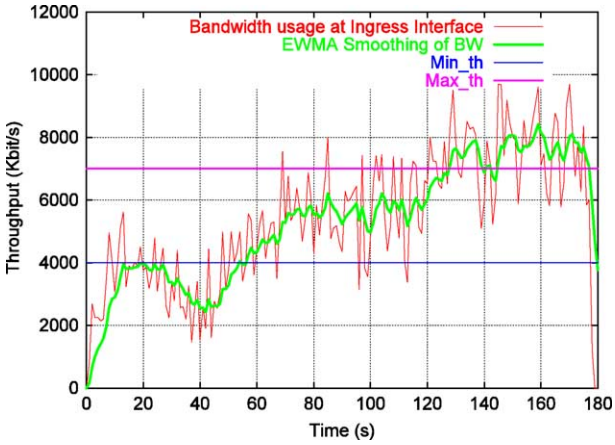

Fig. 14. Bandwidth usage in core router.

with a Gold PHB as long as there is no congestion. When congestion occurs, this traffic is marked with a Bronze PHB. Out-of-profile traffic is marked dynamically either as Gold or Silver or it can be dropped (according to network status).

In this experiment, the network is loaded differently each time. Fig. 14 shows the network load during the period of the experiment (180 s). This load represents the traffic sent from the n traffic generators to the receivers. This measure has been taken from the ingress interface of the core router. During the first 60 s there are only $n=5$ sources on/off. From 60 to 120 s there are $n=8$ sources and in the last 60 s (from 120 to 180 s) the number of the sources are $n=10$. Each source can be either on or off. The same parameters in the first experiment are used for EWMA chart and the congestion threshold.

Our System reacts according to the available bandwidth in the bottleneck link. As shown the video traffic is subject to a dynamic behavior. The edge router admits the video traffic at a reduced QoS level until the required resources become again available. This is very useful for time critical applications. In case of network congestion, the edge router adapts to the network state and reduces QoS rather than completely dropping the traffic.

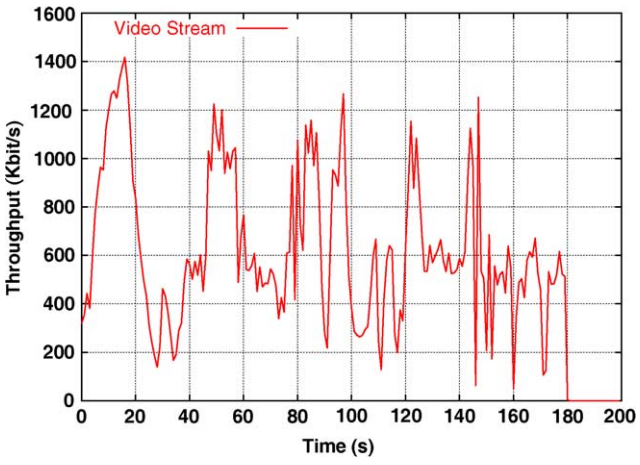

Fig. 15. Received MPEG-2 video Traffic.

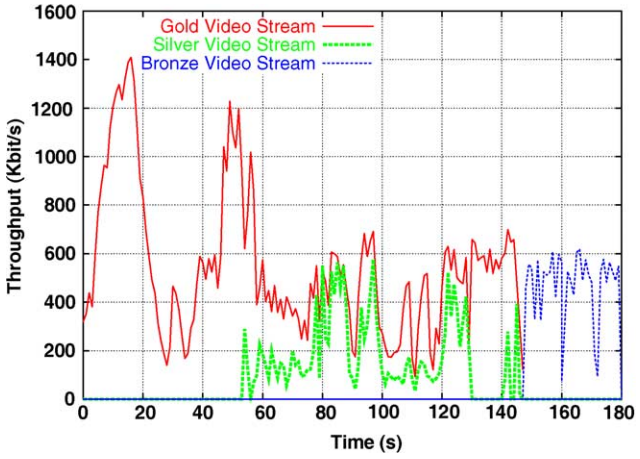

Fig. 16. Different PHB Color.

Fig. 15 shows the video traffic received at the end-user terminal and Fig. 16 shows the different PHB colors assigned to the video traffic when it enters the network.

In this experiment (see Fig. 16), there are three significant time intervals: The first one from 0 s to 56 s when the edge router accepts the video traffic (in profile and out-of-profile). In this case and according to the algorithm, traffic is marked with Gold PHB. The second from 56 to 144 s when the edge router accepts out-of-profile traffic but with a lower QoS level, i.e. marked with Silver and dropped first when network congestion occurs. In-profile traffic is marked with Gold. Note that in this time interval the network is in a congestion state. The last time interval is from 144 to 180 s when the network load increases significantly. In this case, out-of-profile traffic is dropped, and in profile traffic is marked with Silver PHB always.

7. Conclusion

QoS management in the Internet is subject to a large research effort. However, several issues remain to be addressed before a successful deployment of real-time multimedia applications. This paper focused on the issue of dynamic QoS management in a differentiated services network. It presented our approach using network monitoring feedback and control policies. Measurements are processed and used to dynamically adapt QoS parameters for user traffic. A policy-based management approach has been used for the implementation of our system for its ability to handle user traffic dynamically. The sample configuration and policies used in our experiments clearly demonstrate the advantages of our approach. However, several problems inherent to the dynamic nature of our resource management approach require further investigations. One example is the charging scheme for out of profile traffic: Who pays for the service (the sender or the receiver)? An accurate knowledge of the amount of traffic in excess of profile is required in order to establish an adequate payment scheme. Another example is the evaluation of the policy server response time with respect to the traffic management time-scales. These issues are subject to future research.

References

[1] S. Blake, D. Black M. Carlson, E. Davies, Z. Wang, W. Weiss RFC 2475: an architecture for differentiated services, December 1998.
[2] J. Wroclawski, RFC2210—The Use of RSVP with IETF Integrated Services, September 1997.
[3] J. Wroclawski, MIT: RFC2211—Specification of the Controlled-Load Network Element Service, September 1997.
[4] S. Shenker, C. Partridge, R. Guerin, RFC 2212—Specification of Guaranteed Quality of Service, September 1997.
[5] R. Braden, L. Zhang, Berson, S. Herzog, S. Jamin, RFC 2205—Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification, September 1997.
[6] K. Nichols, S. Blake, F. Baker, D. Black, RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, December 1998.
[7] V. Jacobson, K. Nichols, K. Poduri, RFC 2598 An Expedited Forwarding PHB, June 1999.
[8] J. Heinanen, F. Baker, W. Weiss, J. Wroclawsk, RFC 2597: Assured Forwarding PHB Group, June 1999.
[9] D. Durham, Ed, J. Boyle, R. Cohen, S. Herzog, R. Rajan, w, A. Sastry, RFC 2748: The COPS (Common Open Policy Service) Protocol, January 2000.
[10] S. Herzog, Ed., J. Boyle, R. Cohen, D. Durham, R. Rajan, A. Sastry, RFC 2749: COPS usage for RSVP, January 2000.
[11] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith RFC 3084: COPS Usage for Policy Provisioning (COPS-PR), March 2001.
[12] W. Lai, B. Christian, R. Tibbs, S. Berghe, Framework for Internet Traffic Engineering Measurement Internet draft, Work in progress, November 2001.
[13] V. Paxson, G. Almes, J. Mahdavi, M. Mathis, RFC2330: Framework for IP Performance Metrics, May 1998.
[14] W. Fang, Seddigh, B. Nandy, RFC2859—A Time Sliding Window Three Colour Marker (TSWTCM), June 2000.
[15] J. Heinanen, R. Guerin, RFC2698—A Two Rate Three Color Marker (TRTCM), September 1999.
[16] P. Chandra, A.L. Fisher, C. Kosak, P. Steenkiste, Network Support for Application-oriented QoS, Proceedings of Sixth International Workshop on Quality of Service (IWQoS' 98), pp. 187–195, 1998.
[17] J. Bollinger, T. Gross, A Framework-Based Approach to the Development of Network-Aware Applications, IEEE Transactions Software Engineering 24 (5) (1998) 376–390.
[18] F. Chang, V. Karamcheti, Automatic Configuration and Run-time Adaptation of Distributed Applications, Nineth IEEE International Symposium on High Performance Distributed Computing (HPDC) 2000; 11–20.
[19] X. Li, S. Paul, M. Ammar, Layered Video Multicast with Retransmission (LVMR): Hierarchical Rate Control Schemes, Proceedings of Infocom 98, San Francisco, CA, March 1998.
[20] S. McCanne, Scalable compression and transmission over Internet multicast video Ph.D thesis University of California, Berkeley. December 1996.
[21] R. Rejaie, M. Handley, D. Estrin, Layered quality adaptation for Internet video streaming, IEEE Journal of selected areas in communications 18 (12) (2000).
[22] J.S. Hunter, The Exponentially Weighted Moving Average, Journal of Quality Technology 18 (4) (1986) 203–207.

[23] OpenLDAP software available at http://www.openldap.org/
[24] W. Almesberger, Differentiated Services on Linux Home Page http://diffserv.sourceforge.net/
[25] W. Almesberger, J.H. Salim, A. Kuznetsov, Differentiated services on Linux. (work in progress) June 1999.
[26] N. Bjorkman, A. Latour-Henner, A. Doria, The movement from monoliths to component-based network elements, IEEE Communications Magazine 2001; 86–93.
[27] J. Huang, Parlay System and Network Management Working Group Tutorial at www.parlay.org, 2002
[28] J. Biswas, A.A. Lazar, J-F. Huard, K. Lim, S. Mahjoub, L.-F. Pau, M. Suzuki, S. Tortensson, W. Wang, S. Weinstein, The IEEE P1520 Standards Initiative for Programmable Network Interfaces, IEEE Communications Magazine 36 (10) (1998) 64–70.
[29] S. Floyd, et al., Link-sharing and resource management models for packet networks, IEEE/ACM Transactions on Networking 3 (4) (1995) 365–386.
[30] D.D. Clark, W. Fang, Explicit allocation of best effort packet delivery service, ACM Transactions on Networking 1998; 362–373.
[31] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Transactions on Networking 1 (4) (1993) 397–413.
[32] L. Georgiadis, R. Guerin, V. Peris, R. Rajan, Efficient support of delay and rate guarantees in an Internet, ACM SIGCOMM 26 (4) (1996).