DARED: a Double-Feedback AQM Technique for Routers Supporting Real-Time Multimedia Traffic in a Best-Effort Scenario

Mario Barbera, Francesco Licandro, Alfio Lombardo, Giovanni Schembra, Giovanni Tusa
DIIT - University of Catania - V.le A. Doria, 6 - 95125 Catania - ITALY
Telephone: +39 095 7382375
Fax: +39 095 338280
Email: (mbarbera, flicandro, lombardo, schembra, gtusa)@diit.unict.it

***Abstract*— The transmission of real-time multimedia traffic on the Internet is becoming a challenging problem for current research in telecommunications networks. In fact, given the variability of the available network bandwidth, it is necessary to have a mechanism to discover it runtime, and to tune the output flow rate of multimedia sources according to the network bandwidth currently available. An additional problem is that current Internet devices are unaware of application-level quality of service specifications. The idea at the basis of this paper consists of using an *Active Queue Management* (AQM) technique in the router buffers and a rate-control algorithm at the source site which collaborate with each other, in order to meet source QoS requirements, depending on the available resources, simultaneously achieving pro-active control of network congestion. The proposed AQM technique, called *Dynamically Adaptive Random Early Detection* (DARED), works on a network architecture characterized by a double feedback: from network to sources, and from sources to network. The paper shows via simulation, how the architecture simultaneously meets different target QoS levels, specified by sources belonging to different classes.**

I. Introduction

In the last few decades an increasing amount of attention has been paid to technologies for the transmission of real-time multimedia traffic on the Internet, since it will represent most of the traffic in the immediate future, due to the high bit rates. The problem is that real-time multimedia applications are usually based on the UDP transport protocol and are not rate-controlled: this can lead to network congestion collapse, with a large number of packet losses. So the challenge for the Internet in the immediate future is to define a sort of flow control for UDP sources, similar to that of TCP sources, in order to avoid, or at least reduce, network congestion, and to meet the quality of service (QoS) requirements of this kind of application. As for TCP, this control should be based on joint cooperation between sources and network devices.

For TCP traffic the solution was to adopt window-based Additive-Increasing Multiplicative-Decreasing (AIMD) rate control at the source, and Active Queue Management (AQM) techniques in network routers to provide timely feedback, in order to prevent congestion.

Unfortunately, the transmission of real-time multimedia traffic on the Internet causes the following two problems:

- losses introduced by droptail routers due to buffer overflow, or further losses introduced by AQM routers to prevent congestion, are not acceptable for UDP-based real-time sources because they can neither implement a loss recovery mechanism by retransmission, nor use losses as feedback information to control the output rate;
- different multimedia applications present different quality of service (QoS) requirements; for example, there are applications which are more sensitive to end-to-end delay, while others require more bandwidth to achieve a better encoding quality; the current best-effort scenario is not able to differentiate between the quality of service required by traffic flows.

The current challenge for the evolution of the Internet in the immediate future is to define new mechanisms at both the source and the network routers.

At the source site, it is necessary to have a mechanism to discover the available bandwidth runtime, and to adapt the output flow rate according to the network bandwidth currently available. As an example, a possible solution already proposed in the literature by the authors [10], [11] is to use a TCP-friendly protocol to discover the bandwidth, and a rate control algorithm to control the source output. More specifically, the authors introduced a Rate/Quality-Adaptive MPEG video source, using TM-5 [12] to control the MPEG source output bit rate. In this way the source has the ability of this algorithm to follow real-time variations in the network bandwidth.

As far as network routers are concerned, the problem is that current Internet routers treat flows generated by sources with different QoS requirements in the same way, and are unaware of application-level QoS specifications. Therefore, another important challenge is to realize network routers which are able to adapt their behavior, in order for the QoS the sources are provided with to be as close as possible to their requirement. This has to be achieved taking into account that the scenario being considered is a best-effort one, and not the DiffServ scenario foreseen for the next generation Internet, where call admission control and policing will be carried out at the network edge to limit network input traffic.

The idea at the basis of this paper consists of using an AQM technique in the router buffers and a rate-control algorithm at the source site which collaborate with each other in order to achieve pro-active control of network congestion, trying simultaneously to meet source QoS requirements depending on the available resources. The result is an enhanced best-effort scenario where QoS is not guaranteed, but in the event of network congestion, routers co-operate with sources to provide them with a QoS as close to their requirements as possible.

Starting from the numerous works already present in the literature regarding AQM techniques, a novel AQM technique, called *Dynamically Adaptive Random Early Detection* (DARED), is proposed to be implemented in network routers with the following targets:

- to adapt its behavior to incoming requests from sources belonging to different QoS classes, in order to satisfy, as much as possible, their different QoS requirements;
- to adapt the working point to the time-varying traffic conditions of the network;
- to work in combination with adaptive real-time multimedia applications like, for example, the Rate/Quality-Adaptive MPEG video sources proposed in [10], [11];
- to avoid losses, which cannot be recovered by re-transmission, as is done, for example, by TCP sources;
- to be compatible with other routers, network systems and sources not implementing the same technique.

This technique works on a network architecture characterized by a double feedback: the first feedback is generated by routers which, by marking packets on incipient congestion, signal the incoming situation to the traffic sources, and require them to decrease their rate; the second feedback is generated by the traffic sources which communicate their degree of satisfaction concerning the received QoS to the routers, which can dynamically modify their parameters in order to satisfy them. The paper proposes this architecture and, via simulation, analyzes its performance in a wide range of network and traffic conditions.

The paper is organized as follows. Section II provides a brief background to Active Queue Management. Section III describes the transmission system proposed. Section IV analyzes performance. Finally, Section V concludes the paper.

II. Active Queue Management Algorithms

Some years ago, in [2] it was demonstrated that the performance of IP networks loaded by TCP traffic improves when queues are actively managed through timely defined policies. Many AQM policies have been proposed in the literature on the basis of this observation. The main idea behind the Random Early Detection (RED) algorithm [2] is to notify responsive sources about incipient congestion, so that their output rates can be reduced before the buffer overflows, leading to the loss of a great number of packets. The Internet Engineering Task Force (IETF) has produced an Internet Draft [1], with recommendations for the deployment of RED-like Active Queue Management, as a pro-active approach to efficiently notify responsive flows of incoming congestion.

The RED AQM technique randomly discards incoming packets according to a linear drop function, $p_{AQM}(\tilde{m})$. This is a function of the estimated average queue length, $\tilde{m}$, obtained with an Exponential Weighted Moving Average (EWMA) filter. Let us indicate the queue length immediately after the arrival of the $(n-1)$-th packet as q_{n-1}, and the estimated average queue length at the time of the $(n-1)$-th arrival as $\tilde{m}_{n-1}$. The average queue length at the time of the n-th arrival, $\tilde{m}_n$, is estimated in two different ways according to the value of the instantaneous queue at the arrival of the n-th packet, q_n: if its value is non null, the average queue length is estimated as follows:

$$\tilde{m}_n = (1 - w_q) \cdot \tilde{m}_{n-1} + w_q \cdot q_{n-1} \tag{1}$$

where w_q is the EWMA filter parameter; otherwise, if the queue is empty (i.e. $q_n = 0$), the average queue length is estimated as follows:

$$\tilde{m}_n = (1 - w_q)^h \cdot \tilde{m}_{n-1} + w_q \cdot q_{n-1} \tag{2}$$

where h represents the number of packets that might have been transmitted during the time when the queue was empty, and is estimated as:

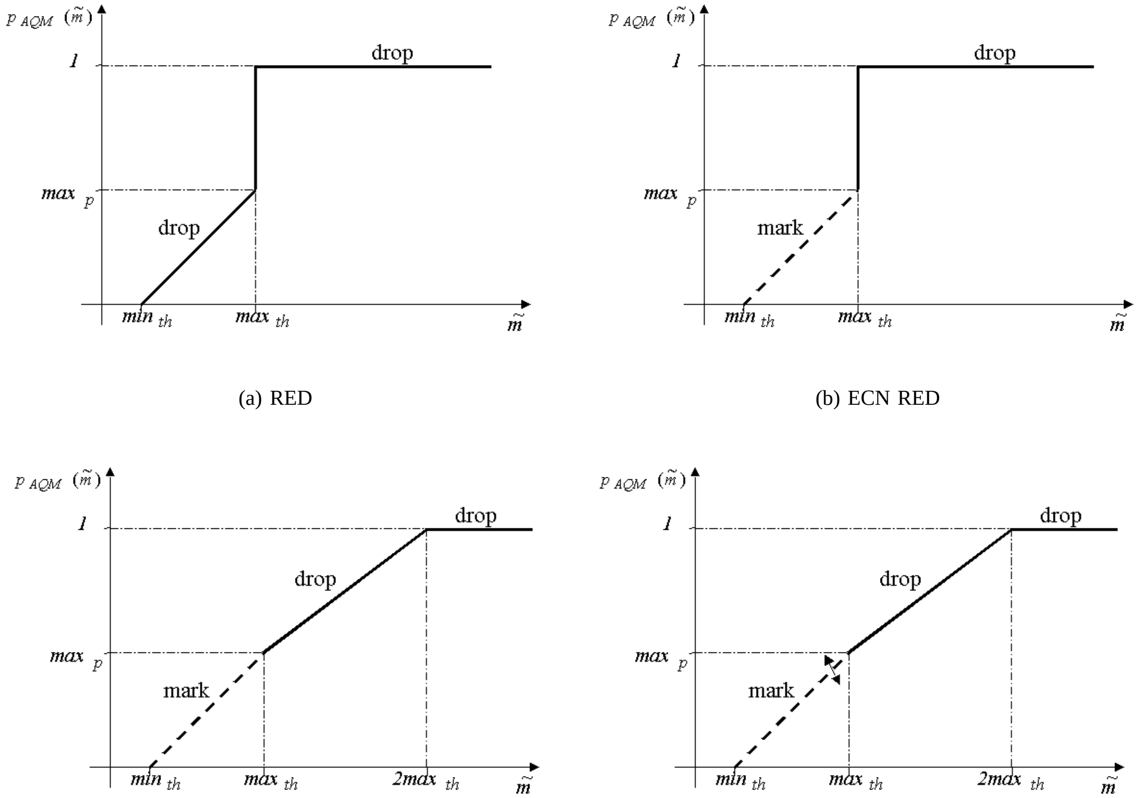
$$h = C_p \cdot (t_n - t_0) \tag{3}$$

C_p being the underlying link bandwidth in packets per second, t_n the time of the (n)-th arrival and t_0 the start of the queue empty time.
The drop function used by RED routers, shown in Fig. 1(a), is the following:

$$p_{AQM}(\tilde{m}) = \begin{cases} 0 & \text{if } \tilde{m} < min_{th} \\ \frac{(\tilde{m} - min_{th}) \cdot max_p}{max_{th} - min_{th}} & \text{if } min_{th} \le \tilde{m} \le max_{th} \\ 1 & \text{if } \tilde{m} > max_{th} \end{cases} \tag{4}$$

When the estimated average queue length is less than a given threshold, min_{th}, all packets are accepted in the queue. The random packet dropping probability linearly increases from 0 to max_p as the estimated average queue length grows from min_{th} to max_{th}; packets are discarded with a probability of one when either the estimated average queue length exceeds max_{th}, or the instantaneous queue length has reached its maximum value. Of course, if RED is too severe, a large number of losses are introduced, and therefore link utilization is penalized.

In addition, early discarding may introduce a huge number of losses in the attempt to prevent congestion, which may be deleterious for both TCP-based data traffic and UDP-based multimedia traffic. In fact, in the first case sources are able to recover losses, but this may cause unacceptable delays due to re-transmissions. In the presence of UDP applications which usually transmit information units (e.g. video frames) segmented into a certain number of packets, on the other hand, the huge number of packet losses introduced to prevent congestion may cause many information units to be discarded



(a) RED

(b) ECN RED

(c) Gentle ECN RED

(d) Gentle ECN Adaptive RED

Fig. 1. Behavior functions of the RED-based AQM techniques.

at their destination; this problem becomes very important for video sources, which use inter-frame correlations to achieve high video compression. To solve this problem, simultaneously providing early congestion notification without unnecessary packet losses, Explicit Congestion Notification (ECN) was proposed [3]. As shown in Fig. 1(b), instead of discarding packets as in the original version, RED routers implementing this facility mark them with the aim of alerting sources to incoming congestion when the estimated average queue length is between min_{th} and max_{th}, while they apply the original RED in other cases. Of course, this mechanism assumes that all sources are able to change their output rate according to these notifications. This is the case, for example, of TCP sources, or sources using rate-control algorithms like the Rate Adaptation Protocol (RAP) [9] or TCP-Friendly Rate Control (TFRC) [8]. If some traffic sources are not able to modify their output rate according to the notifications received, the ECN mechanism is not applied to their flows, so that the RED router discards packets instead of marking them.

Another important observation made during the evolution of the RED-based AQM techniques regards the sensitivity of performance to parameter settings and traffic load variations; if, in fact, the traffic load grows when the value chosen for max_p is too low, the working point of the average queue length can go up to max_{th}, causing a sharp increase in the dropping rate and a decrease in throughput until the average queue value goes down, below max_{th}, again. To avoid these oscillations around max_{th}, another option was proposed for RED: the *gentle* RED [4], [5]. This technique is based on a variation of the drop function in the range between max_{th} and twice max_{th}, linearly increasing from max_p to 1. The behavior function of the RED with the ECN and gentle options, is shown in Fig. 1(c). Furthermore, in order to overcome the difficulties in RED parameter configuration, an adaptive version of RED was proposed in [6], [7]. As depicted in Fig. 1(d), through a slow adaptation of max_p based on the value of the estimated average queue length, Adaptive RED (ARED) [7] ensures that RED is well configured and its performance more stable under a wider range of traffic loads. The parameter max_p is changed using an AIMD policy to maintain the average queue length around a target value halfway between min_{th} and max_{th}. In this way ARED makes the average delay stable and predictable in advance. The configuration of the ARED thresholds min_{th} and max_{th} and of the EWMA filter parameter w_q is based on knowledge of the link capacity.

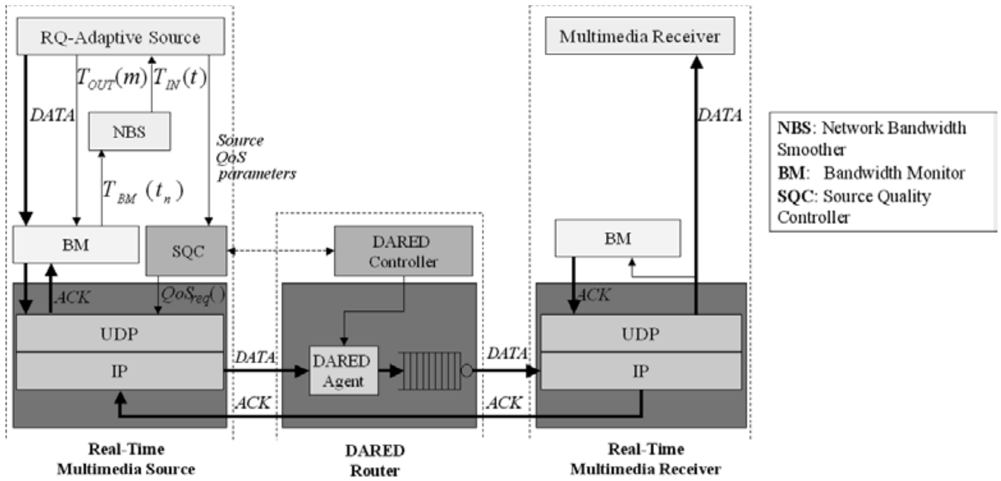


Fig. 2. Rate/Quality controlled multimedia transmission system.

III. Description of the System

In the previous section we discussed the principles behind the RED AQM technique and the subsequent modifications to improve its performance. The main aim of AQM techniques in network routers is to participate in congestion control, for the early detection and prevention of network congestion [1].

Accounting that in the current best-effort Internet no attention is paid to the different QoS requirements of heterogeneous applications, in this paper we propose a novel RED-based AQM technique, called *Dynamically Adaptive Random Early Detection* (DARED), working with joint co-operation between sources and network routers. DARED has the aim of guaranteeing distribution of the available network resources to flows of different classes with different declared QoS requirements, to provide them with a QoS as close to the requirements as possible. This will be done without any explicit resource reservation, according to the principles at the basis of the current best-effort Internet service. DARED is defined in such a way that compatibility with the past is guaranteed; that is, not all network routers need to be DARED routers, but they can be introduced into the network progressively. With this in mind, the system we propose is shown in Fig. 2, where the main blocks are highlighted: the *Real-Time Multimedia Source*, the *DARED Router* and the *Real-Time Multimedia Receiver*. In Section III-A we will describe the functions at the source and receiver sites, while the router architecture will be described in Section III-B.

A. Source/Receiver architecture

In this section we will present the main source and receiver component blocks of the system architecture shown in Fig. 2. At the source site, the main blocks are the *Rate/Quality Adaptive Source* (RQ-Adaptive Source), the *Bandwidth Monitor* (BM), the *Network Bandwidth Smoother* (NBS), and the *Source Quality Controller* (SQC). They are at the top of the UDP/IP protocol stack. The main blocks at the receiver site are the *Multimedia Receiver* and the peer-to-peer entity of the source BM, also located over the UDP/IP protocol stack.

The RQ-Adaptive Source block may be a generic real-time multimedia source, which has the capacity to control its emission rate according to the network bandwidth available.

The BM block continuously estimates the network bandwidth and provides this information to the RQ-Adaptive Source. To this end the two main TCP-friendly algorithms, TFRC [8] and RAP [9], can be used as bandwidth monitors, but with a different aim from their original target, given that our purpose is just to monitor network bandwidth and not to make real-time multimedia output friendly to TCP flows in the network. More specifically, in our system we use RAP with the ECN option, which, as already said in Section II, allows network AQM routers to mark packets when congestion is incoming, instead of discarding them. In addition, we extend the ECN option to the whole range of the estimated average queue length. In this way, all packets arrive at their destination, if not lost due to buffer overflow; however, the rate control algorithm of the RAP protocol considers marked packets to be lost, halving the bandwidth for marked packets notification instead of loss notification.

Let us indicate the instant when the n-th rate-change event occurs as t_n, and the new rate calculated at this instant as $T_{BM}(t_n)$. This process constitutes the input to the second main block of the source system, the NBS block. It has the aim of eliminating the high frequencies of this process. This is achieved by filtering the process by means of an EWMA filter with a weighting parameter $\alpha \in [0, 1]$:

$$\hat{T}_S(t_n) = \alpha \cdot T_{BM}(t_{n-1}) + (1-\alpha) \cdot \hat{T}_S(t_{n-1}) \qquad (5)$$

where t_{n-1} and t_n are the instants of the $(n-1)$-th and the n-th rate variations, respectively. Finally, the output of the NBS is obtained as a continuous-time step function derived from $\hat{T}_S(t_n)$ as follows:

$$T_{IN}(t) = \hat{T}_S(t_{n-1}) \qquad \forall t \in [t_{n-1}, t_n[\qquad (6)$$

In this way the process $T_{IN}(t)$ represents the available bandwidth calculated by the NBS block, and its variations can be regulated by the parameter α.

The SQC has the aim of monitoring the main QoS parameters declared by the source, and ensuring that specific QoS signals can be sent by application in the path towards the network routers if their value do not correspond to the range of acceptable values fixed by the source. Based on the chosen metric (e.g. throughput, delay, loss) and level of QoS required by the source, the source is classified by the SQC as belonging to a given QoS class, so that the routers can impact in a different manner on the perceived QoS at the application level. Let ξ_i be the QoS parameter for a source belonging to the i-th QoS class, and $A_i^{(\xi)}$ the corresponding required range for it. For each QoS parameter ξ_i the SQC algorithm works as in Fig. 3:

```
Every Δt seconds:
  monitoring the parameter ξ_i;
  if (ξ_i ∉ A_i^(ξ)) then
    QoS_incr ← true;
  else
    QoS_incr ← false;
  end if
Every packet to be sent:
  c_id ← i;
  if ((now > lastinstant_send + timeout) and (QoS_incr)) then
    send QoS_req(ξ_i, QoS_incr, c_id);
    lastinstant_send ← now;
    timeout ← K · SRTT;
  end if
```

Fig. 3. SQC algorithm.

where:

- Δt is the time granularity chosen to monitor the QoS parameters;
- $QoS_{req}(\xi_i, QoS_{incr}, c_{id})$ is the call from the SQC to the underlying protocols to insert the QoS-increase request in the IP packet header;
- $lastinstant_{send}$ is the instant when the last QoS-increase request was sent;
- $timeout$ is the time granularity chosen to send QoS-increase requests;
- K is a constant multiplicative factor;
- $SRTT$ is the average round-trip-time estimated at the source site.

After many simulations we found that a good value for K is 4, ensuring good responsiveness on the part of the system and avoiding heavy oscillations. In this way, no more than one QoS-increase request can be sent to the network routers in a time shorter than $4 \cdot SRTT$.

So, after each Δt interval, the SQC monitors the ξ_i QoS parameter and, if it does not belong to the range specified by the source for this parameter, it sends a QoS_{req} message, so that the QoS-increase request will be introduced in the IP packet to be sent.[1]

For the case study we consider in this paper, we use the RQ-Adaptive Source block proposed in [10], [11]. It is an MPEG video source whose emission rate is controlled through timely choice of the quantizer scale, in such a way as to respect the network bandwidth calculated by the source BM block. For a detailed description of the Rate/Quality-Adaptive MPEG video source, see [10], [11]. The Rate/Quality-Adaptive MPEG video source uses the Peak Signal-to-Noise Ratio (PSNR) [10], [11], [13], [14], [15] resulting from the quantization process of a given MPEG video source, as the main encoding parameter. This parameter is strictly related to the amount of estimated available network bandwidth. It provides an indication of the distortion introduced by the quantization process and is defined as:

$$PSNR_n = 10 \log_{10} \left[\frac{2^d - 1}{MSE_n} \right] \quad (7)$$

d being the number of bits assigned to a pixel, and MSE_n the mean quantization error for the n-th frame encoding. In order to reduce estimation oscillations, the SQC smoothes the PSNR values measured through an EWMA filter, obtaining the smoothed PSNR, $SPSNR$, as follows:

$$SPSNR_n = (1 - \beta) \cdot SPSNR_{n-1} + \beta \cdot (PSNR_n) \quad (8)$$

where β is the weight of the low-pass filter, $\beta \in [0, 1]$, $PSNR_n$ the instantaneous value of the PSNR process for the n-th frame, and $SPSNR_{n-1}$ is the estimated average value at the end of the previous frame encoding. We chose a value of 0.05 for the parameter β in our simulation. In the case being considered, the Δt interval of the SQC algorithm to monitor the QoS parameters coincides with the frame encoding interval.

Video sources are classified as belonging to two different classes based on their degree of sensitivity to $SPSNR$: class-1 sources are more sensitive to $SPSNR$ than class-2 sources. So, in cases of congestion, DARED routers have to treat packets coming from the two classes differently.

Let $A_1^{(SPSNR)}$ and $A_2^{(SPSNR)}$ be the admissible $SPSNR$ ranges for class-1 sources and class-2 sources, respectively. When the SQC estimates an $SPSNR$ value below the lower bound of the acceptable declared range for a class-1 source, a QoS-increase request will be set in the header of the IP packets the source sends. The routers belonging to the path towards the destination receive this information, and modify their behavior in order to be less severe in packet marking towards class-1 packets, thus improving the performance class-1 flows are provided with. If this modification is not sufficient to satisfy the class-1 sources which have issued the request, these sources continue to send further requests until they

[1] To transport the information about the c_{id} (*class identifier*) of every packet, and the possible QoS-increase request, there are different possibilities. Using IP_{v4}, one solution may be to introduce a new IP option in the IP header. Using IP_{v6}, the *flow label* field (20 bits) could be used.

receive the required QoS. Of course, this is to the detriment of class-2 flows. If in the meanwhile some class-2 sources measure an $SPSNR$ value below the lower bound of the acceptable declared range, they start to send class-2 QoS-increase requests, at most one for each $K \cdot SRTT$ seconds as said so far, in order to receive less severe treatment by the network routers. So, as we will formally describe in Section III-B, DARED routers try to find the best working point, defined as the AQM behavior which best meets the requirements of all the sources, that is, the best trade-off between their requirements.

B. DARED router architecture

In this section we will present the DARED router architecture. Two main blocks have to be considered at the router site, with the aim of providing network routers with the ability to adapt their configuration parameters in order to approach the source QoS requirements as close as possible. These blocks, shown in Fig. 2, are the *DARED Controller* and the *DARED Agent*.

The *DARED Controller* identifies the class the incoming packets belong to, and any QoS-increase request from the sources, updating the AQM parameters of the *DARED Agent* if necessary. The *DARED Agent* implements the DARED AQM technique, working on flows passing through the DARED router. It receives packets which have to be forwarded by the router, and marks them according to the DARED algorithm illustrated below.

In the following, without losing in generality, we will describe a DARED router implementing a flow management algorithm when all the sources are RQ-Adaptive MPEG video sources which, as discussed in Section III-A, use SPSNR as their QoS parameter. However, extension to manage other QoS parameters is easy and left for future works. In the considered case, the DARED technique is based on dynamic adaptation of the packet marking probability, which will be different for packets emitted by sources belonging to different QoS classes. The DARED AQM technique is adaptive like Adaptive RED [7] (i.e. it presents the same discarding function and adapts the max_p parameter with the same rules), but presents the two following innovations, to be able to manage multimedia traffic:

1) extension of the ECN option to the whole range of the estimated average queue length, with the aim of reducing or completely avoiding packet losses due to AQM;
2) use of a class-based dynamic marking probability, which is calculated runtime according to the QoS-increase requests received from the RQ-Adaptive Sources in each class.

The class-based dynamic marking probability is calculated as follows. The QoS-increase requests coming from class-1 sources are classified by the *DARED Controller* as positive requests, while the others are classified as negative requests.

At the reception of a new request, the *DARED Controller* updates (i.e. decreases or increases) a global counter, s_g, containing information about the value of the total requests: in this way, s_g is positive when the number of class-1 sources not satisfied in terms of SPSNR is greater than the number of class-2 sources not satisfied for the same reason. When the global counter s_g reaches a given positive threshold, up_{th}, the *DARED Controller* increases the value of a parameter δ with a given increase step, σ, and resets s_g.
Instead, if the global counter reaches a given negative threshold, $down_{th}$, the *DARED Controller* decreases the value of δ with the same step, and resets s_g. The step value which has been demonstrated to be efficient and will be used in the Numerical Results Section is $\sigma = 0.05$. The δ-adaptation algorithm is described in Fig. 4.

Every packet arrival:
  if the packet contains a QoS-increase request **then**
    verifies the class of the incoming packet;
    if $(c_{id} = 1)$ **then**
      $s_g \leftarrow s_g + 1;$
    else
      $s_g \leftarrow s_g - 1;$
    end if
    if $(s_g = up_{th})$ **then**
      $\delta \leftarrow \delta + \sigma;$
      $s_g \leftarrow 0;$
    else if $(s_g = down_{th})$ **then**
      $\delta \leftarrow \delta - \sigma;$
      $s_g \leftarrow 0;$
    end if
  end if

Fig. 4. δ-adaptation algorithm.

After the parameter δ has been calculated, the marking probabilities which will actually be used until the next variation to mark packets coming from class-1 sources, p_1, and from class-2 sources, p_2, are derived from the marking probability, p_r, estimated by the network router with the ARED algorithm. In the computation of p_1 and p_2, we distinguish between the cases when $p_r \in [0, 0.5]$ and $p_r \in]0.5, 1]$. In this way, for a fixed value of p_r and δ, we ensure that a symmetric negative or positive gap is applied to p_r to derive the p_1 and p_2 values, as described in Fig. 5.
For $\delta = 0$ the marking probability is the same for both classes, and coincides with the one estimated by ARED; as δ increases, for a given value of p_r, p_1 decreases towards zero, while p_2 increases towards one. For negative values of δ, on the other hand, class-2 packets are privileged over class-1 packets in the same way: this can be the case, for example, of a strong majority of non-satisfied class-2 sources.

Fig. 6 shows the marking functions applied to class-1 and class-2 packets, and the Adaptive RED behavior function. It is a snapshot obtained when $max_p = 0.5$, and by fixing the adaptation parameter at $\delta = 0.5$. As said so far, in the whole range of the estimated average queue length the congestion control is performed by marking the packets, which are discarded only when the buffer overflows. Moreover, we

Every packet arrival:

p_r is calculated with the ARED algorithm [7];
if ($p_r > 0.5$) **then**
$var_{p_r} \leftarrow \delta \cdot (1 - p_r)$;
else
$var_{p_r} \leftarrow \delta \cdot p_r$;
end if
if (c_{id} = 1) **then**
$p_1 \leftarrow p_r - var_{p_r}$;
else
$p_2 \leftarrow p_r + var_{p_r}$;
end if

Fig. 5. Dynamic marking probability computation.

can observe how the marking probabilities applied to class-1 and class-2 flows, p_1 and p_2, are equidistant with respect to the Adaptive RED curve, which represents the p_r behavior.

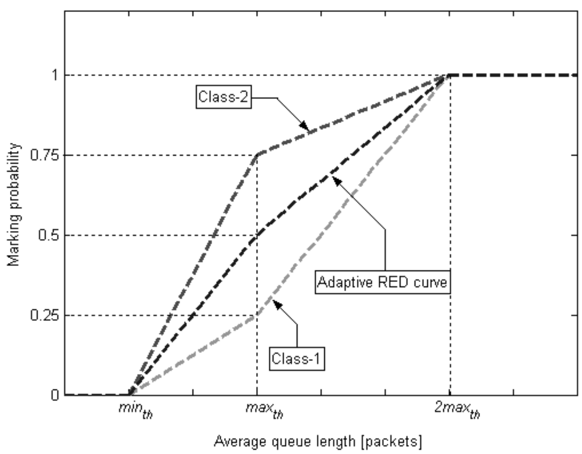


Fig. 6. DARED behavior functions calculated for $max_p = 0.5$ and $\delta = 0.5$.

IV. Numerical results

In this section we will analyze the steady-state and transient behavior of the DARED AQM technique proposed in this paper. The performance analysis was carried out via simulation, using the ns-2 simulator [16]. We used the RAP algorithm [9] as the algorithm implemented in the BM block to estimate network bandwidth. For the video traffic, we considered the adaptive-rate MPEG video source ns-module, generating video traffic characterized by its first- and second-order statistics, available in [17]. In particular we set the parameters of this generator equal to the statistics relating to the first hour of the movie ”Evita”, encoded with the GoP structure IBBPBB, and with a picture format of 384 x 288 pixels. For performance evaluation of the proposed system architecture, we considered the $SPSNR$ as the main QoS parameter, and differentiated video sources into two different QoS classes, as defined in Section III-A. The analysis carried out in this paper is based on a simple topology made up of N video sources attached to a DARED router R1, and N receivers attached to a DARED router R2, as shown in Fig. 7.

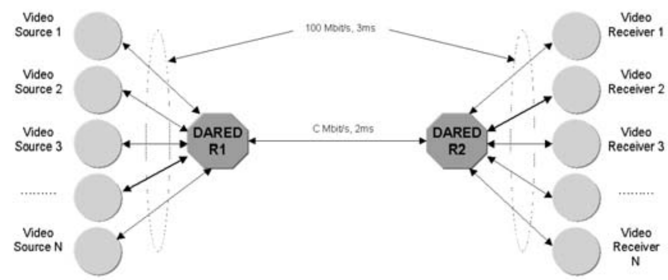


Fig. 7. Simulation Topology.

Both sources and receivers are connected to the network routers through point-to-point links having a capacity of 100 Mbit/s, and a propagation delay of 3 ms. The routers R1 and R2 are connected to each other through a bottleneck link with a capacity of C Mbit/s, which will be varied in our analysis. The propagation delay of this bottleneck link is 2 ms. Sections IV-A and IV-B describe and discuss the steady-state and transient performance, respectively.

A. Steady-state analysis

The main interest of our study is to demonstrate how the DARED AQM technique is able to assign different amounts of resources to video sources belonging to different classes, in order to satisfy their QoS requirements as well as possible. For the steady-state analysis we assume a total of N=10 video sources: five sources more sensitive to the $SPSNR$ requirement, and five sources less sensitive. The former sources will be classified as belonging to class-1, the latter to class-2. Their declared $SPSNR$ acceptable ranges are assumed to be $A_1^{(SPSNR)} = [47, \infty[$ dB, for every class-1 source, and $A_2^{(SPSNR)} = [44, \infty[$ dB, for every class-2 source.

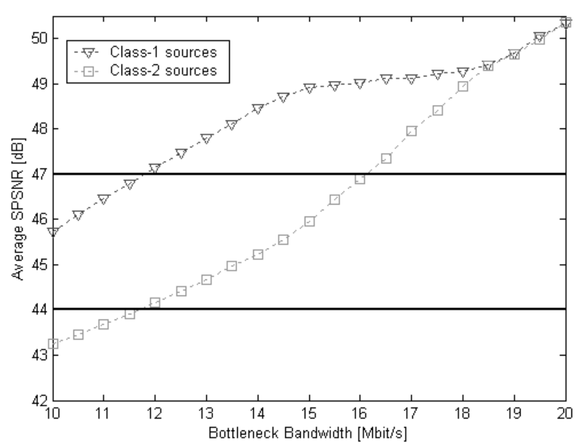


Fig. 8. Average SPSNR vs. the bottleneck capacity.

Fig. 8 shows the average $SPSNR$ obtained by class-1 and class-2 sources with a bottleneck link capacity ranging from $C = 10$ Mbit/s to $C = 20$ Mbit/s. In this figure we can observe that, thanks to the proposed algorithm, even for low capacity values, when $SPSNR$ requirements are not yet satisfied, class-1 sources receive an average $SPSNR$ that is about 3 dB higher than that received by class-2 sources. With a bottleneck link capacity of $C = 12$ Mbit/s, both the class-1 and the class-2 sources achieve the minimum requirements. Therefore any bandwidth increment over $C = 12$ Mbit/s is used by sources from both classes to improve the encoding quality. Moreover, let us note that a gap between the two curves is present for $C > 12$ Mbit/s as well. This is due to the fact that, even if in the long-term period the average $SPSNR$ has acceptable values according to the range specified by the sources, in the short-term period it may happen that the estimated $SPSNR$ results below the lower bound of the admissible range, causing a certain number of requests to be sent by the sources. Of course, this happens more frequently for class-1 sources, and therefore these sources receive more bandwidth, then encode with a higher average $SPSNR$.

Moreover, let us note that, while the link capacity C increases, the slope of the class-1 curve progressively decreases and, on the other hand, the slope of the class-2 curve progressively increases. This is due to a lower probability that the average $SPSNR$ goes down below the admissible ranges, therefore the average $SPSNR$ values achieved by the two classes tend to converge to the same value. For $C > 18$ Mbit/s, all the sources achieve the same average $SPSNR$ value, that is the maximum they can obtain for the considered bottleneck capacity, given that also in a short-term period, the achieved average $SPSNR$ values are to the inside of the intervals $A_1^{(SPSNR)}$ and $A_2^{(SPSNR)}$, and therefore no QoS-increase requests are issued.

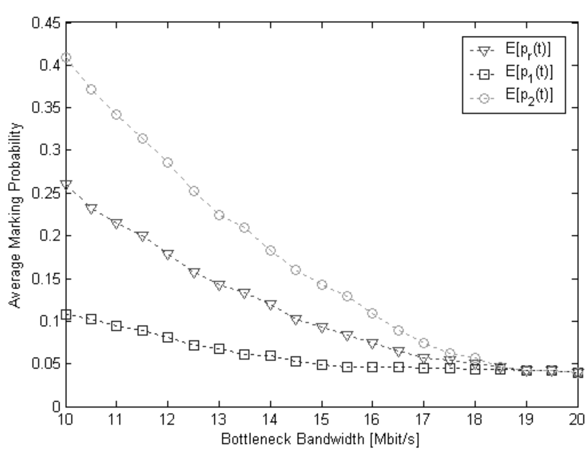


Fig. 9. Average Marking Probabilities vs. the bottleneck capacity.

Of course, what has been said about Fig. 8 is strictly related to the marking probability estimated by ARED, p_r, and the two marking probabilities calculated as in Section III-B, p_1 and p_2. Their average values are shown in Fig. 9, where we can observe that they coincide when $C > 18$ Mbit/s, that is, when all the sources are always satisfied. Instead, in the range of lower values of C, there is a higher gap between the $E[p_1(t)]$ and $E[p_2(t)]$ curves. With C ranging from 12 to 18 Mbit/s, a difference between the two marking probabilities is still present, and the $E[p_2(t)]$ curve presents a higher slope than the $E[p_1(t)]$ curve, due to what we have already said about the same capacity range in Fig. 8.

Fig. 10 shows the average emission rate for the two classes of sources. We can note that it is very similar to Fig. 8, given that the $SPSNR$ is strictly related, and roughly proportional, to the received bandwidth due to the DARED marking probability computation algorithm, on the basis of the requests received from the sources.

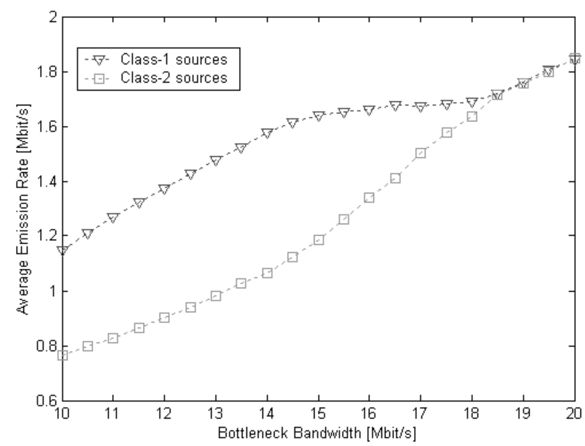


Fig. 10. Average Emission Rate vs. the bottleneck capacity.

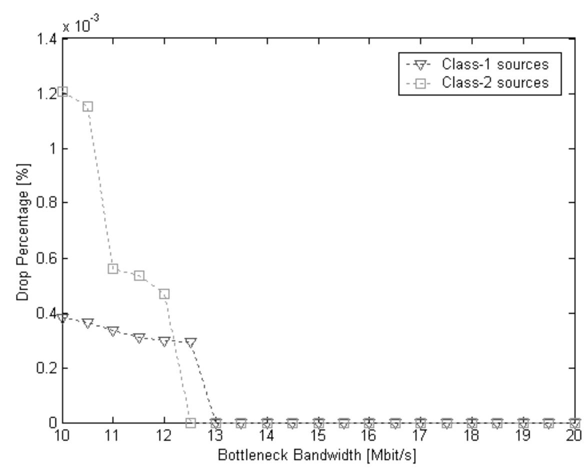


Fig. 11. Loss performance vs. the bottleneck capacity.

The loss performance of the new AQM technique we propose, in terms of the percentage of packets sent by the sources and discarded by the network routers, is shown in Fig.

TABLE I

SOURCES ACTIVITIES

Time	0 s	200 s	400 s	600 s	800 s	1000 s	1200 s	1400 s	1600 s
Class-1 sources	2	4	6	6	6	4	2	2	2
Class-2 sources	2	2	2	4	6	6	6	4	2

11. With the extension of the ECN in the whole range of the estimated average queue, the feedback to prevent congestion is performed without the early discarding of packets; so, as expected, the losses for the DARED algorithms are very close to zero, except for a negligible percentage in the lower bottleneck capacity range, when a small number of packets may be discarded due to buffer saturation.

B. Transient analysis

We will now analyze the transient behavior of the proposed architecture. To this end, we will vary the number N of video sources for each QoS class during the simulation time, according to the Table I, maintaining a fixed link capacity of $C = 10$ Mbit/s. The $SPSNR$ admissible ranges $A_1^{(SPSNR)}$ and $A_2^{(SPSNR)}$ are the same as in Section IV-A. The following processes are considered in our analysis: the source bandwidth process, the $SPSNR$ process, and the average marking probability process at the router level. We do not present any plots regarding the loss-percentage process, because we have already shown that the system architecture we propose is completely loss-less.

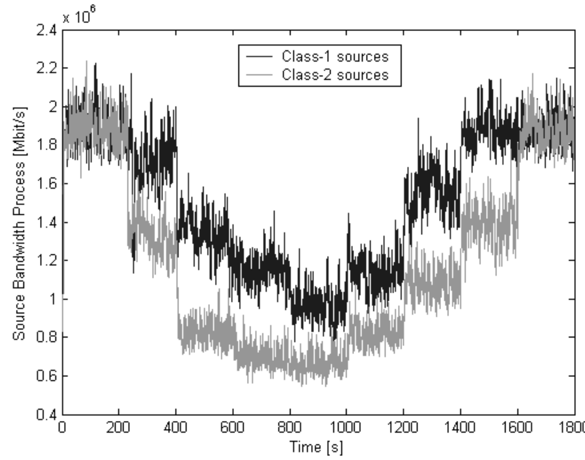


Fig. 12. Source Bandwidth Process during simulation time.

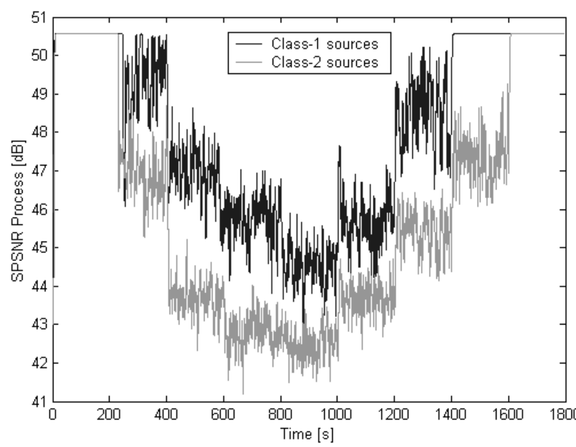


Fig. 13. SPSNR Process during simulation time.

Figs. 12 and 13 show how the network resources are distributed for the two classes of sources, and how this distribution affects the quality parameter. As expected for what has been said in the previous section, we can note that the behaviors plotted in the above figures are very similar. The marking probability processes are shown in Fig. 14, which shows the temporal behavior of the marking probability calculated according to ARED algorithm, $p_r(t)$, and the probabilities applied to the class-1 and class-2 flows, $p_1(t)$ and $p_2(t)$, respectively. We can note the difference between the probability $p_1(t)$ and $p_2(t)$, and their symmetry around $p_r(t)$.

During the first time interval, [0, 200] s, the dynamic marking probability computation algorithm is not applied, due to the small number of sources with respect to the available resources. During the interval [200,400] s, there is a difference in the encoding quality of the two classes (see Fig. 13), due to the requests sent by class-1 sources during the transient period immediately after the start of the new sessions (around the time $t = 220$ s, when the $SPSNR$ goes below the admissible range for these sources). At $t = 400$ s, two new class-1 sources become active, so that the reduction of the available bandwidth for every source, and the requests due to their QoS requirements, produce a new redistribution in favor of the class-1 sources, which are more numerous and more demanding. The same behavior can be noted during the intervals $[600, 800]$ s and $[800, 1000]$ s, but with lower $SPSNR$ values. Moreover, it is interesting to note that on the right-half side of the figures, from t=1000 s to t=1200 s, the bandwidth and $SPSNR$ processes have the same behavior as the symmetric left-half side. However, there is a lower gap due to a majority of class-2 sources as compared to class-1 ones, with the result that the $SPSNR$ the latter sources are provided with is more far away from their lower bound of 47 dB. On the contrary, from $t = 1200$ s to $t = 1400$ s, another increment in the bandwidth and $SPSNR$ gap occurs, given that the class-2 sources are always satisfied, while the class-1 sources still continue to send requests. Finally, from $t = 1400$ s to $t = 1600$ s, a different quality distribution remains due to the previous difference between $p_1(t)$ and $p_2(t)$, which are no more varied because all the sources are satisfied, and therefore

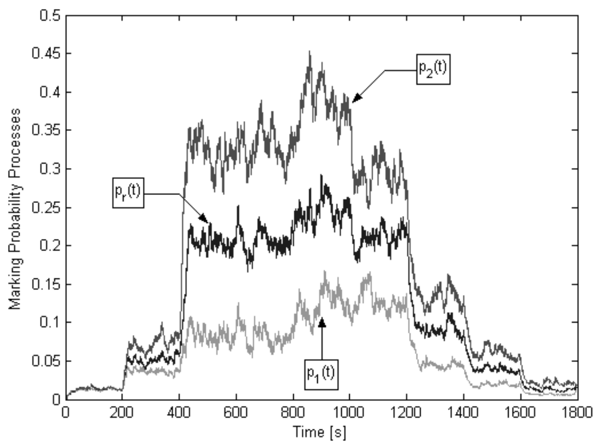


Fig. 14. Marking Probability processes at the router site.

do not send anymore requests.

V. CONCLUSIONS AND FUTURE WORKS

The paper deals with the problem of real-time multimedia traffic transmission on the current best-effort Internet. A transmission system architecture for multimedia applications is proposed, with two main tasks: first, to deal with the different QoS requirements of different applications, given that this problem is not taken into account by the current best-effort Internet; secondly, to guarantee prevention of network congestion. In order to achieve these aims, the system architecture is made up of rate-quality adaptive sources and network routers implementing a new AQM technique called DARED. By considering a scenario where rate controlled MPEG video sources are used as rate-quality adaptive multimedia sources, the strength of the proposed architecture in matching QoS requirements at the application level has been numerically demonstrated. To this end, a particular case study has been analyzed, assuming the Smoothed Peak Signal-to-Noise Ratio ($SPSNR$) as the main QoS parameter at the application level, and differentiating the applications on the basis of the minimum $SPSNR$ required. Starting from the system architecture proposed, and the principles behind our idea, a first task for the future is to analyze more heterogeneous network scenarios. As an example, some applications may be more sensitive to end-to-end delay than to throughput ($SPSNR$), or differently loss-sensitive, specifying different targets for all these QoS parameters. Analysis of the system architecture proposed with more complex network topologies is also an issue to be addressed in future works.

REFERENCES

[1] R. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Patridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet," *RFC2309*, April 1998.

[2] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, August 1993.

[3] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review*, 24.

[4] S. Floyd, "Optimum functions for computing the drop probability". Email available at http://www.icir.org/floyd/REDfunc.txt, October 1997.

[5] S. Floyd, "Recommendation on using the *gentle* variant of RED". http://www.icir.org/floyd/red/gentle.html, Mar 2000.

[6] W. Feng, D. Kandlur, D. Saha, and K. Shin, "A Self Configuring Red Gateway," *Infocom*, Mar 1999.

[7] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An Algorithm for Increasing the Robustness of RED's Active Queue Management," August 1, 2001. http://www.icir.org/floyd/papers/adaptiveRed.pdf.

[8] S. Floyd, M. Handley, J. Padhye, J. Widmer, "Equation-based congestion control for unicast applications," *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, August 28 - September 1, 2000.

[9] R. Rejaie, M. Handley, and D. Estrin, "RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the internet," *in Proc. of INFOCOM*, New York, NY, March 1999.

[10] F. Licandro, G. Schembra, "A Rate/Quality Controlled MPEG Video Transmission System in a TCP-Friendly Internet Scenario," *Proceedings of PV2002*, Pittsburgh, PA, April 24-26, 2002.

[11] F. Licandro, A. Lombardo, G. Schembra, "A TFRC-Based Real-time Video Transmission System: Perceived Quality and Fairness Analysis," *Proceedings of IWDC '02*, Capri, Italy, September 8-11, 2002.

[12] Test Model 5, ISO-IEC/JTC1/SC29/WG11, April 1993.

[13] A. Lombardo, G. Schembra, "TSpec Enforcement for MPEG Video Transmission over the Next Generation Internet: an Analytical Framework," *Computer Networks and ISDN Systems*, Vol. 37, No. 5, November 2001.

[14] W. Luo, M. El Zarki, "Quality Control for VBR Video over ATM Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 6, August 1997.

[15] R Rejae, M. Handley, D. Estrin, "Architectural considerations for playback of quality adaptive video over the Internet," Proceedings of the *IEEE Internationals Conference on Networks* (*ICON' 00*), 2000.

[16] "Network simulator - ns (version 2)," available at: *http://www.isi.edu/nsnam/ns/*.

[17] "Video MPEG," available at: *http://www.diit.unict.it/arti/NS2-modules.htm*.