Fair per-flow multi-step scheduler in a new Internet DiffServ node architecture

Paolo Dini[1], Guido Fraietta[2], Dario Pompili[3]

[1]paodini@infocom.uniroma1.it, [2]guifra@inwind.it, [3]pompili@dis.uniroma1.it

Computer Science and INFOCOM Department, University of Rome "La Sapienza", ITALY
Via Eudossiana,18, 00184 Rome

The authors are listed in alphabetical order

Abstract: This document focuses on DiffServ Quality of Service Approach and in particular on Assured Forwarding Service. A new scheduling architecture is proposed in order to avoid some drawbacks carried by the standard DiffServ Approach as, for example, the lack of granularity in data traffic policing which may lead to degradation of quality of service for all data flows within the same class even if only one data flow generates excess traffic. This drawback is avoided in the proposed multi-step scheduler, since a mechanism that provides fairness among different data flows belonging to the same class is studied, without hardening too much processing and system complexity and with no overhead introduction.

I. INTRODUCTION

The major benefit of the DiffServ (DS) approach is its practicality and scalability, due to aggregation of different packet streams (data flows) with the same required service. The main consequence of this concept is that traffic signalling can be almost completely cut off if the communication end-points are in the same DiffServ Domain, otherwise it has to be performed only at the inter-domain links. This can be achieved because in this approach Quality of Service provision is guaranteed aggregating different data flows with the same quality requirements, thus achieving scalability especially in the core network, where it is difficult to maintain separate information because of the large amount of different data flows.

A drawback of the "classical" DiffServ approach, described in [1] and [2], is the lack of granularity in data traffic policing, which may lead to degradation of quality of service for all data flows within the same class even if only one data flow generates excess traffic.

This drawback is avoided in the proposed scheduling architecture, since a mechanism that provides fairness among different data flows belonging to the same class is studied. The classical requirements of DiffServ Assured Service are respected as well, i.e.:

1. Three Assured Forwarding classes have been taken into account, as suggested in [3] associated with "Olympic Services" (Gold, Silver and Bronze). Packets assigned to these three classes are marked with particular codepoints in a specific Class of Service (COS) field of the IP header. In this way packets belonging to the Gold class have greater probability for timely forwarding than packets assigned to the Silver class and the Bronze one; the same holds for Silver packets, which have a greater probability for timely forwarding than those belonging to the Bronze class.
   Packets within each class are further differentiated by giving them a low, medium or high drop precedence.
   The Bronze, Silver and Gold service classes in the network are respectively mapped to the Assured Forwarding (AF) classes 1, 2 and 3. Similarly, low, medium and high drop precedence are mapped to AF drop precedence levels 1, 2 or 3. Furthermore, in the following of the document different drop precedence are associated with colors, i.e. low, medium and high drop precedence will be respectively mapped in the green, yellow and red color.

2. In the proposed system a minimum amount of forwarding resources (buffer space and bandwidth) is provided to each implemented AF class. Each class is served in order to achieve the configured service rate (bandwidth) over both small and large time scales, as required in [3].

The main improvements introduced in the proposed scheduling architecture are:

- **Fairness within each class.** It means that each data flow is separately colored and if it respects a certain medium rate (that depends on the Service Level Agreement for the class and on the number of currently active flows of that class) it is not demoted to higher drop precedence.

- **Selective discard.** If the aggregated rate exceeds the Service Level Agreement rate of its class, only data flows that are already been demoted to red color may be discarded. Those flows have exceeded the medium rate currently available for each active flow and thus the drop of these packets will not lead to lower performances for the well-behaving flows.

The paper structure is described hereafter.

First of all, a brief description of the traffic conditioners (markers and shapers) used in the proposed architecture is provided. Then, the Earliest Deadline First (EDF) scheduling policy is described, both in its classic behaviour and in the enhanced version studied in this paper. The scheduling architecture is then deeply described and in the last section simulation results, based on a network simulator, OPNET (OPtimum NETwork performance) will be shown.

II. TRAFFIC CONDITIONERS: MARKERS AND SHAPERS

Traffic conditioners performs metering, shaping, policing and/or re-marking to ensure that traffic entering the DiffServ (DS) domain conforms to the rules specified in the Traffic Conditioning Agreement (TCA), in accordance with the domain service provisioning policy.

A traffic conditioner may contain the following elements: meter, marker, shaper and dropper. A traffic stream is selected by a classifier, which steers packets to a logical instance of a traffic conditioner. A meter is used to measure the traffic stream against a traffic profile. The state of the meter with respect to a particular packet (e.g., whether it is in- or out-of-profile) may be used to affect a marking, dropping, or shaping action. When packets exit the traffic conditioner of a DS boundary node, the DS codepoint of each packet must be set to an appropriate value. Traffic meters measure the temporal properties of the stream of packets selected by a classifier against a traffic profile specified in a TCA. A meter passes state information to other conditioning functions to trigger a particular action for each packet which is either in- or out-of-profile. Packet markers set the DS field of a packet to a particular codepoint, adding the marked packet to a particular DS behavior aggregate. The marker may be configured to mark all packets which are steered to it to a single codepoint, or may be configured to mark a packet to one of a set of codepoints used to select a Per Hop Behavior (PHB) in a PHB group, according to the state of a meter. When the marker changes the codepoint in a packet it is said to have "re-marked" the packet. Shapers delay some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. A shaper usually has a finite-size buffer, and packets may be discarded if there is not sufficient buffer space to hold the delayed packets. Droppers discard some or all packets in a traffic stream in order to bring the stream into compliance with a traffic profile. This process is know as "policing" the stream and can be implemented as a special case of a shaper by setting the shaper buffer size to zero (or a few) packets or by discarding non-compliant packets somewhere else in the scheduling system, as will be done in the proposed DiffServ node scheduling architecture.

The traffic originating from the source domain across a boundary may be marked by the traffic sources directly or by intermediate nodes before leaving the source domain. This is referred to as initial marking or "pre-marking" and this is the case that will be considered in the following with respect to the three “Olympic” Assured Service Classes, the Gold, the Silver and the Bronze one. One of the main advantage of marking packets close to the traffic source is that a traffic source can more easily take applications’ preferences into account when deciding which packets should receive better forwarding treatment.

A key element in the proposed architecture is the single rate Three Color Marker (srTCM). The single rate Three Color Marker [6] meters an IP packet stream and marks its packets green, yellow or red. The marking process exploits three traffic parameters, a Committed Information Rate (CIR) and two associated burst sizes, a Committed Burst Size (CBS) and an Excess Burst Size (EBS). A packet is marked green if it does not exceed the CBS, yellow if it does exceed the CBS, but not the EBS, and red otherwise. The srTCM is based on two leaky buckets fed by the same rate (CIR) but with different bucket sizes, respectively the CBS and the EBS. This marker is mainly useful for ingress policing of a service, where only the length, not the peak rate, of the burst determines service eligibility.

The Meter meters each packet and passes the packet and the metering result to the Marker. The Meter operates in one of two modes. In the Color-Blind mode, the Meter assumes that the packet stream is uncolored, while in the Color-Aware mode the Meter assumes that some preceding entity has pre-colored the incoming packet stream so that each packet is either green, yellow or red. The Marker (re)colors an IP packet according to the results of the Meter.

Rate Adaptive Shapers (RAS) [4] can be used in combination with the single rate Three Color Markers. These RAS improve the performance of TCP when a Three Color Marker (TCM) is used at the ingress of a DiffServ network by reducing the burstiness of the traffic. With TCP traffic, this reduction of the burstiness is accompanied by a reduction of the number of marked packets and by an improved TCP good-put. The RAS can be used at the ingress of DiffServ networks providing the Assured Forwarding Per Hop Behavior (AF PHB). By reducing the burstiness of the traffic, the adaptive shapers increase the percentage of packets marked as green by the TCM and thus the overall good-put of the users attached to such a shaper. Such Rate Adaptive Shapers will probably be useful at the edge of the network.

The main objective of the shaper is to produce at its output a traffic that is less bursty than the input traffic, but the shaper avoids discarding packets in contrast with classical token bucket based shapers. The shaper itself consists of a tail-drop FIFO queue that is emptied at a variable rate. The shaping rate, i.e. the rate at which the queue is emptied, is a function of the occupancy of the FIFO queue. If the queue occupancy increases, the shaping rate will also increase in order to prevent loss and too large delays through the shaper. The shaping rate is also a function of the average rate of the incoming traffic. The shaper was designed to be used in conjunction with meters such as the srTCM.

III. EARLIEST DEADLINE FIRST

The Earliest Deadline First (EDF) scheduler is a form of dynamic priority scheduler where the priorities for each packet are assigned as it arrives. Specifically, each packet is assigned a deadline which is given by the sum of its arrival time and the delay guarantee associated with the flow the packet belongs to. The EDF scheduler selects the packet with the smallest deadline for transmission on the link and hence the name. The dynamic nature of the priority in the EDF scheduler is evident from the fact that the priority of the packet increases with the amount of time it spends in the system. This ensures that packets with loose delay requirements obtain better service than they would in a static priority scheduler without sacrificing the tight delay guarantees that may be provided to other flows. An advantage of EDF is to minimize the maximum packets' lateness, defined as the difference between the deadline of a packet and the time it is actually transmitted on the link.

EDF has been proven to be an optimal scheduling discipline in the sense that, if a set of tasks is schedulable under any scheduling discipline (i.e., if the packets can be scheduled in such a way that all of their deadlines are met), then the set is also schedulable under EDF.

One of the main attractions of the EDF policy is that it allows the separation of delay and throughput guarantees for a flow. The EDF policy by itself cannot be used to provide efficient end-to-end delay guarantees. In order to achieve that, one could reshape the traffic at each node to a pre-specified envelope before it is made eligible for scheduling. Coupled with traffic shapers the EDF policy can be used to provide efficient end-to-end delay guarantees on a per flow basis. Every time a packet arrives at one of the queues, it is assigned a "deadline" equal to its arrival time plus the maximum queuing delay tolerated by the packets belonging to the particular queue. This queuing delay, "$delta_i$", is a static parameter which in connection services is specified at the connection set-up and, of course, it is very small for the "voice" and "real time video" packets, and larger for WEB and FTP packets.

It is important to underline that the delay constraint is fundamental only for the real time connections; however, it is opportune that the non-delay sensitive applications do not suffer of an *uncontrolled* delay, in order to keep a certain QoS provisioning. It is worth observing that when a QoS queue is empty, it does not obtain the grant from the scheduler, and the bandwidth it does not use is available for the other queues. The EDF policy guarantees that the QoS queues begin transmitting not after their "$delta_i$", if all the incoming traffic is Compliant, so to respect the delay constraints for each queue.

The EDF dynamic system has two states which determine its behavior, the transient state and the steady one. In the steady state the available output bandwidth is partitioned proportionally to the incoming rate, while in the transient period (in the initial stage, for example, when all the queues are empty or after a period in which sources have emitted data packets below their average rate), the spare bandwidth is given to sources with tighter delay constrains, i.e. with an inferior static priority value.

In this paper an original use of the EDF scheduler is suggested in order to achieve different discard probabilities for packets with the same delay constraints. As can be seen in Fig. 1 two queues with different lengths (L_{SHORT} and L_{LONG}) are served with an EDF policy with the same static parameter (no matter the value). In this way packets in the longer queue will be guaranteed a lower drop probability than packets in the shorter one. Moreover if their incoming throughput is the same, the longer queue will be assigned a greater output bandwidth, like it is pointed out in Fig. 2.

Two drop policies may be implemented to discard packets: the LAST ARRIVED PACKET DROP and the RANDOM DROP. If an incoming packet is discarded when it finds its queue full (this is the LAST ARRIVED PACKET DROP policy depicted in Fig.2), the total bandwidth will be divided between the short and the long queue in the following way:

$$R_{SHORT}(t)=\frac{L_{SHORT}}{L_{SHORT}+L_{LONG}}R_{TOT}(t)\,; \qquad (1)$$

$$R_{LONG}(t)=\frac{L_{LONG}}{L_{SHORT}+L_{LONG}}R_{TOT}(t)\,; \qquad (2)$$

being $$R_{TOT}(t)=R_{SHORT}(t)+R_{LONG}(t). \qquad (3)$$

If, on the other hand, a RANDOM DROP policy is chosen between packets already in the queue, it will be possible to assign a greater bandwidth to the service with the longer queue. In the two following simulation scenarios the two queues, kept full by continuous packets' arrivals, are served with a constant R_{TOT} =100000 bps. Both the two drop policy scenarios have queues with length equals to $L_{LONG}=0.6*(L_{LONG}+L_{SHORT})$ and $L_{SHORT}=0.4*(L_{LONG}+L_{SHORT})$; with these values it is possible to see in Fig. 2 that in the LAST ARRIVED PACKET DROP scenario the total bandwidth R_{TOT} is shared by the two queues proportionally to their length while in the RANDOM DROP scenario the long queue has a greater amount of bandwidth than in the previous case. This result shows how it is possible to decrease the packets' discard probability simply by changing the drop policy in the queue temporally affected by congestion, without modifying the queues' lengths.

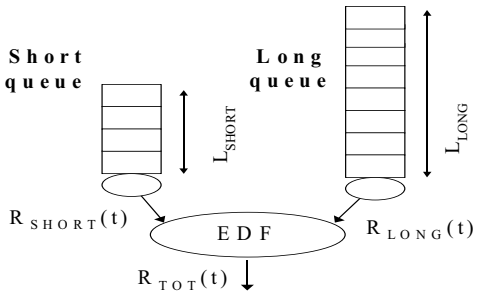

Fig. 1. Two queues with different lengths are served with an EDF policy with the same static parameter in order to achieve different discard probabilities for packets with the same delay constraints.

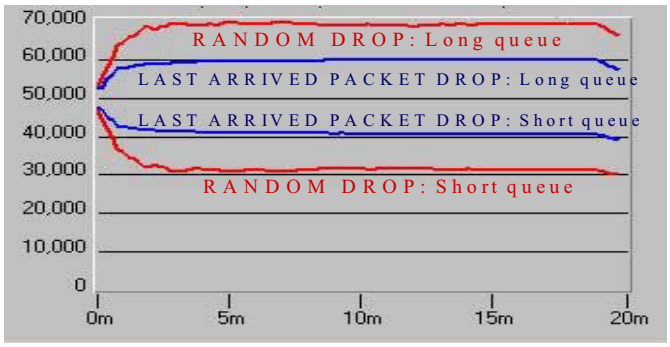

Fig. 2. Assigned bandwidth to two different length queues served with an EDF policy with the same static parameter. Two drop-scenarios are shown: RANDOM DROP and LAST ARRIVED PACKET DROP.

IV. SCHEDULING ARCHITECTURE

An original scheduling architecture is proposed in this section in order to avoid the drawbacks carried by the standard DiffServ Approach without hardening too much processing and system complexity. The main disadvantage of the "classical" DiffServ approach is the lack of granularity in data traffic policing, which may lead to degradation of quality of service for all data flows within the same class even if only one data flow generates excess traffic.

The main improvements introduced with the proposed multi-step classifier scheduler are **Fairness within each class** and **Selective discard.**

As far as concern the first innovative point, it means that each data flow, belonging to one of the Olympic Class, is separately colored and if it respects a certain medium rate (that depends on the class Service Level Agreement and on the number of currently active flows of that class) it is not demoted to higher drop precedence. The second point means that if the aggregated rate exceeds the Service Level Agreement rate of its class, only data flows that are already been demoted to red color may be discarded; those flows have exceeded the medium rate currently available for each active flow and thus the discard of red packets will not lead to lower performances for the well-behaving flows.

To understand how these targets are met it is useful to follow, in a generic DiffServ node implementing the proposed enhanced IP layer scheduling architecture, the arrival of a packet of one of the three Olympic Class (Gold, Silver and Bronze) and its final departure from the IP layer.

First of all it is worth observing (Fig. 3) that each arriving packet, before being forwarded to its IP next hop, has to cross three stages implementing different functionalities.

The first stage, made up of four network elements (one Per-class classifier and one Non-compliant meter associated to each Olympic Class), aims at separating the aggregate data flow into classes and metering individually these class-flows.

The second stage, in charge of implementing the **'Fairness within each class'** function, is composed for each of the three classes of one per-flow classifier, as many srTCMs (single rate Three Color Marker) as the number of active flows and three adders in order to re-aggregate flows with the same color within the same class.

The third stage, the last to be crossed by an out-coming packet, has to achieve the **'Selective discard'** function (with the help of the Non-Compliant per-class meter, as it will be clearer further). For each class three distinct queues are provided to store packets, according to the color they have been marked by the srTCM. These queues differ in their length in order to achieve a decreasing packet discard probability from the lower class (Bronze) to the upper one (Gold), like it has been described in the previous section. The EDF with finite-length queues is the proposed scheduling algorithm in charge of selecting the winner packet among all the ones stored in the head of each of the nine queues, as pointed out in Fig. 3. It is important to stress that the essential policing functionality in charge of avoiding the starvation of Compliant packets, it is no carried out by the leaky bucket in the first stage of the node scheduling architecture. This important node element (called in the figure Non-Compliant meter) does not implement the drop function, task that will be realized in the last stage, but simply meters separately for each class the Non-Compliant bits it is letting pass through. The Non-Compliant bit information will be exploited by the drop entities in the third stage in order to decide how many red packets should be discarded from the red queues. In such a way higher link efficiency is achieved together with a fair per-flow policy discard.

As far as concern the color marking function executed by the srTCMs for each flow in the second stage of the proposed node architecture, it is important to point out that their parameters must have the same values within each Olympic Class. In particular, for each class the CIR is dynamically calculated by dividing the total Class Committed Rate (specified in the Class-SLA) by the number of the current active flows within the class. This adaptive change of the three CIR values according to the dynamic active flow class state is the main mechanism in order to achieve fairness within each class. The proposed multi-step classifier does not introduce any overhead to implement its functionalities.

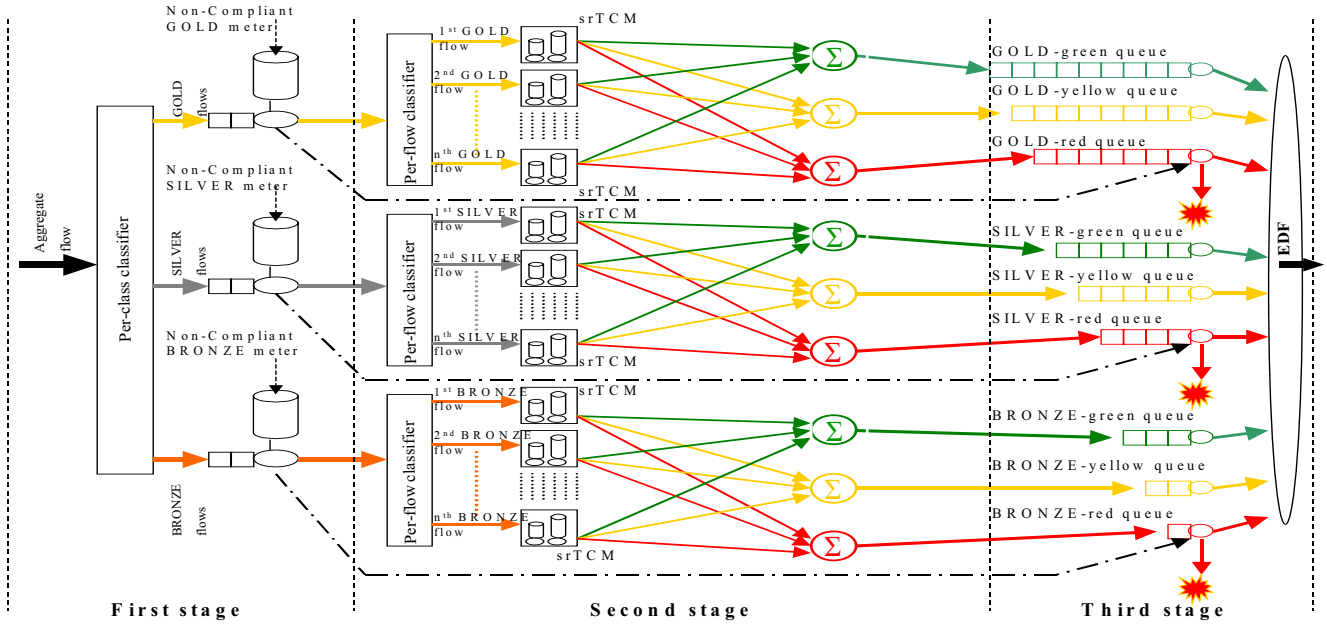

Fig.3. Proposed multi-step classifier scheduler for an Internet DiffServ node

V. SIMULATION RESULTS

OPNET simulation results highlight QoS requirements respect, class fairness and resource optimized utilization. In particular in Fig. 4 it is pointed out, in a logarithmic scale for readers' convenience, the number of discarded bits for each class, when the emitted traffic is the same for the three classes and in heavy congestion state. It is easy to note how lower is the discarded GOLD bits' number with respect to the SILVER and BRONZE one.

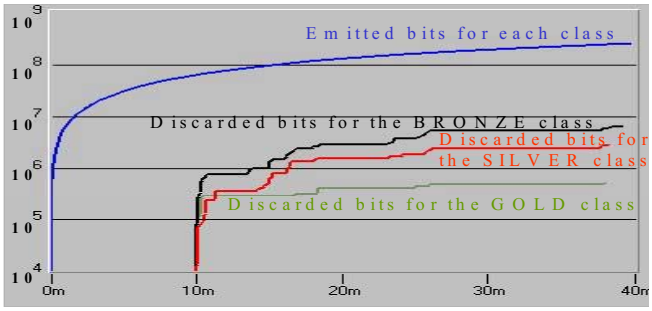

Fig. 4. Discarded bits for GOLD, SILVER and BRONZE class when the emitted traffic is the same for the three classes

VI. CONCLUSIONS

The major benefits of the DiffServ approach, achieved by a scalable Quality of Service provision obtained by aggregating different data flows with the same QoS requirements, have been deeply studied in this paper. The drawbacks of this "classical" DiffServ approach (i.e. the lack of granularity in data traffic policing, which may lead to quality degradation of service for all data flows within the same class even if only one data flow generates excess traffic) have been analyzed as well. The novel scheduler architecture proposed in this paper avoids these drawbacks, since a mechanism providing fairness among different data flows belonging to the same class have been designed.

REFERENCES

[1] IETF RFC 2474, Nichols K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

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

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

[4] De Cnodder S., "Rate Adaptive Shapers for Data Traffic in DiffServ Networks", NetWorld + Interop 2000 Engineers Conference, Las Vegas, Nevada, USA, May 10-11, 2000.

[5] A.Elwalid and D. Mitra, "Traffic Shaping at a Network Node: Theory, Optimum Design, Admission control"

[6] RFC2697, Heinanen J. and R. Guerin, "A Single Rate Three Color Marker", RFC 2697, September 1999.

[7] RFC2698, Heinanen J. and R. Guerin, "A Two Rate Three Color Marker", RFC 2698, September 1999.

[8] RFC2963, O. Bonaventure, S. De Cnodder, "A Rate Adaptive Shaper for Differentiated services", October 2000.