Transport of TCP/IP Traffic over Assured Forwarding IP-Differentiated Services

Paolo Giacomazzi, Luigi Musumeci, and Giacomo Verticale,
Politecnico di Milano, Italy

Abstract

The Internet is currently facing a twofold challenge: to increase network capacity in order to accommodate a steadily increasing number of users, and to guarantee the Quality of Service for existing applications as well as for new multimedia applications requiring real-time network response. In order to meet these requirements, the Internet Engineering Task Force is currently defining the Differentiated Service architecture, which should offer a simple and scalable platform to guarantee differentiated Quality of Service in the Internet. In the Differentiated Services domain, the Assured Forwarding service is designed to provide data applications with acceptable performance, overcoming the limits of the best-effort service currently offered by the Internet. Since data applications mostly rely on the TCP transport protocol, it is important to examine the interaction between the TCP and the assured forwarding congestion avoidance and control mechanisms. The main purpose of this article is to shed light on this interaction and to show that, in the current Differentiated Services framework, poor performance of TCP traffic flows can result from the existing mismatch between the assured forwarding traffic conditioning procedures and the TCP congestion management. We propose a new adaptive packet marking policy to deal with congestion situations that may occur in the network. We show that with this policy the provisioned rate for TCP flows can be achieved.

Currently, the Internet is in a phase of rapid evolution from both a quantitative and a qualitative point of view. While the Internet's traffic volume continuously increases, it becomes more challenging to provide different levels of Quality of Service (QoS) for applications with specific service requirements.

The Internet Engineering Task Force (IETF) has defined two different frameworks to support the Quality of Service of Internet traffic: The Integrated Services (IntServ) [1, 2], and the Differentiated Services (DiffServ) frameworks [3–6]. In the Integrated Services model, resources are allocated to individual user flows, possibly leading to scalability problems. In this respect, the DiffServ model is simpler as it focuses on traffic aggregates, that is, large sets of flows with similar service requirements. Thus, the DiffServ approach removes the need for per-flow resource reservation.

Moreover, in the DiffServ architecture, traffic control functions are mostly performed by border routers, while simpler functions are implemented by internal routers. In particular, border routers are responsible for ensuring that individual traffic flows conform to the traffic profile specified by the network provider. They are also in charge of the classification function that groups individual traffic flows into a small number of traffic classes according to the similarity of their Quality of Service requirements. In this way, internal routers can manage traffic classes as opposed to a large number of individual traffic flows. As the number of traffic flows increases, internal routers are not overloaded by the packet processing burden, which is mainly limited to packet forwarding. In turn, this approach makes the network scalable.

All traffic entering a DiffServ network is *classified* and then *conditioned* to comply with profile requirements. Traffic conditioning is performed according to shaping and/or policing techniques.

The DiffServ service class is specified in the DiffServ field of each IP packet. In particular, the eight-bit type of service (ToS) field in the IPv4 header is replaced by the DiffServ field. Six bits of the DiffServ field constitute the differentiated services code point (DSCP) [7, 8], which identifies a processing action, called per hop behavior (PHB) [6, 9], performed by routers on all incoming packets. DiffServ classes are also specified in the IPv6 packet header [4].

The DiffServ technique requires that a service level agreement [6] is established between network subscribers and service providers. The packet classification and conditioning functions are ruled by the traffic conditioning agreement, which is part of the service-level agreement.

The packet classification function is based on information in the packet header, such as source and destination addresses, port numbers, and protocol types. In addition, packet classifi-

Work partially supported by the Italian "Ministero per l'universita' e la ricerca scientifica e tecnologica" (MURST) 40 percent funds.

0890-8044/03/$17.00 © 2003 IEEE

cation can be based on information stored inside routers. The conditioning operation includes metering, marking, shaping, and dropping (Fig. 1).

After packet classification, metering functions are performed to verify that the incoming traffic satisfies profile requirements. Then the marker module shown in Fig. 1 assigns to each packet a differentiated services code point, based on the results of classification and metering functions. The marker can also modify the existing value of the differentiated services code point. This can occur when packets are transmitted across two Internet domains ruled by different administrations.

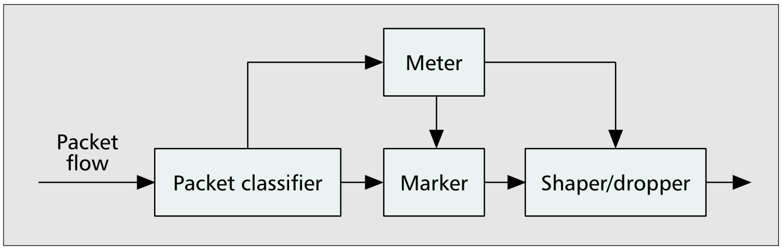


Figure 1. *Traffic conditioning operation in a DiffServ border router.*

When a packet violates the traffic profile, it may be dropped or forwarded with a lower service priority level (policing action). The shaping module in Fig. 1 holds packets in a queue until they can be forwarded in compliance with the associated profile.

The Internet Engineering Task Force has defined three types of forwarding differentiated services: Expedited Forwarding (EF) [10], Assured Forwarding (AF) [11], and Best-Effort (BE). Expedited Forwarding provides minimal delay, jitter, and packet loss, and it guarantees the required bandwidth. Packets that violate traffic profile requirements are dropped. The Expedited Forwarding service is suitable for delay-sensitive applications such as voice and video. The Assured Forwarding service classifies IP packets into four traffic classes and three levels of drop precedence. In case of congestion, high-drop-precedence packets are more likely to be dropped than low-drop-precedence packets.

The implementation of the Assured Forwarding service requires an active queue management algorithm capable of solving possible long-term congestion problems within each Assured Forwarding class by dropping packets, while handling short-term congestion problems by queuing packets. Packets must be dropped gradually, based on a smooth congestion indication, in order to avoid dramatic congestion situations in the network.

In the Assured Forwarding service, the packet-dropping operation can be performed according to a well known technique called RIO [12]. RIO is a simple active queue management algorithm. The basis of the RIO (RED for In and Out) [12] technique is the RED (Random Early Detection) mechanism [13] that drops packets randomly as soon as congestion arises. In a RIO router, all packets forwarded to the same output line are buffered in a single queue. In this queue, two sets of RED thresholds are maintained, for in-profile and for out-of-profile packets, respectively. Two separate average buffer occupancy calculations are performed, for in-profile packets and for all packets in the queue. The dropping probability of in-profile packets only depends on their total number in the buffer, while the dropping probability of out-of-profile packets depends on the total number of packets in the buffer. The RIO scheme is particularly appealing, as it uses a single FIFO (First In First Out) queue and relies on the discarding operation that can be performed by border routers and internal routers.

In a typical application based on the Assured Forwarding Per Hop Behavior [14], it is expected that IP packets are forwarded with high probability as long as they satisfy profile requirements. Packet sources are also permitted to exceed their profile requirements. In this case, excess traffic is forwarded with lower probability. Moreover, it is important to note that the network must not reorder packets of the same flow, neither in-profile nor out-of-profile.

It is likely that the bulk of the Assured Forwarding flows will be generated by applications relying on the TCP transport protocol, such as Web-browsing and e-commerce. The TCP continuously increments bandwidth occupation by repeatedly increasing the data transmission rate and monitoring the behavior of the network. As soon as the network starts dropping packets, the TCP reduces its transmission rate.

With the Assured Forwarding service, this feature of TCP can lead to poor performance. If a user is allowed to send packets exceeding profile requirements, these packets will be classified as out of profile by border routers. A possible subsequent action taken by the network is to forward these out-of-profile packets as Best-Effort packets. In case of network congestion, Best-Effort packets can experience significant losses, which, in turn, trigger a dramatic reduction of the transmission rate at the TCP level. As a consequence, the performance of a TCP flow transported with the Assured Forwarding service is mainly determined by its out-of-profile component. Even if the network has sufficient bandwidth for in-profile packets, the losses experienced by out-of-profile packets downgrade the overall performance of the TCP flow.

This article assesses the impact of the Assured Forwarding Per Hop Behavior on TCP flows during congestion periods. The article proposes a new packet marking technique as a solution to TCP performance degradation when non-conformant packets are transmitted with high drop precedence. The proposed marking technique is adaptive, that is, the amount of excess bandwidth allocated by border routers to each TCP flow is variable in order to prevent TCP packet losses caused by excess low-priority traffic in the network.

This adaptive technique requires a congestion signaling procedure from internal routers to border routers. The article shows that a simple congestion signaling technique, named Congestion Signaling Algorithm (CSA), can guarantee the required rate of TCP traffic flows. In particular, it is possible to make border routers aware of congestion occurring in downstream internal routers, in order to adjust the percentage of non-conformant packets entering the network for each Assured Forwarding flow.

The article is organized as follows. The related work in this area is reviewed. We describe the models of network devices and traffic conditioning adopted in the article. The packet marking technique is explained and the Congestion Signaling Algorithm used by internal routers and border routers is described. Simulation results are then presented and, finally, conclusions are drawn.

Related Work

The sensitivity of TCP to congestion with the Assured Forwarding service has been previously addressed in the literature. In [15] the performance of TCP flows over the Assured Forwarding service is discussed. The main contribution of this research effort is a detailed experimental study of the main factors that impact the throughput of TCP flows in a RIO-based DiffServ network. The article shows that in an over-provisioned network all target rates are achieved, but unfair shares of excess bandwidth are obtained. However, as the network approaches an under-provisioned state, not all target rates can be achieved.

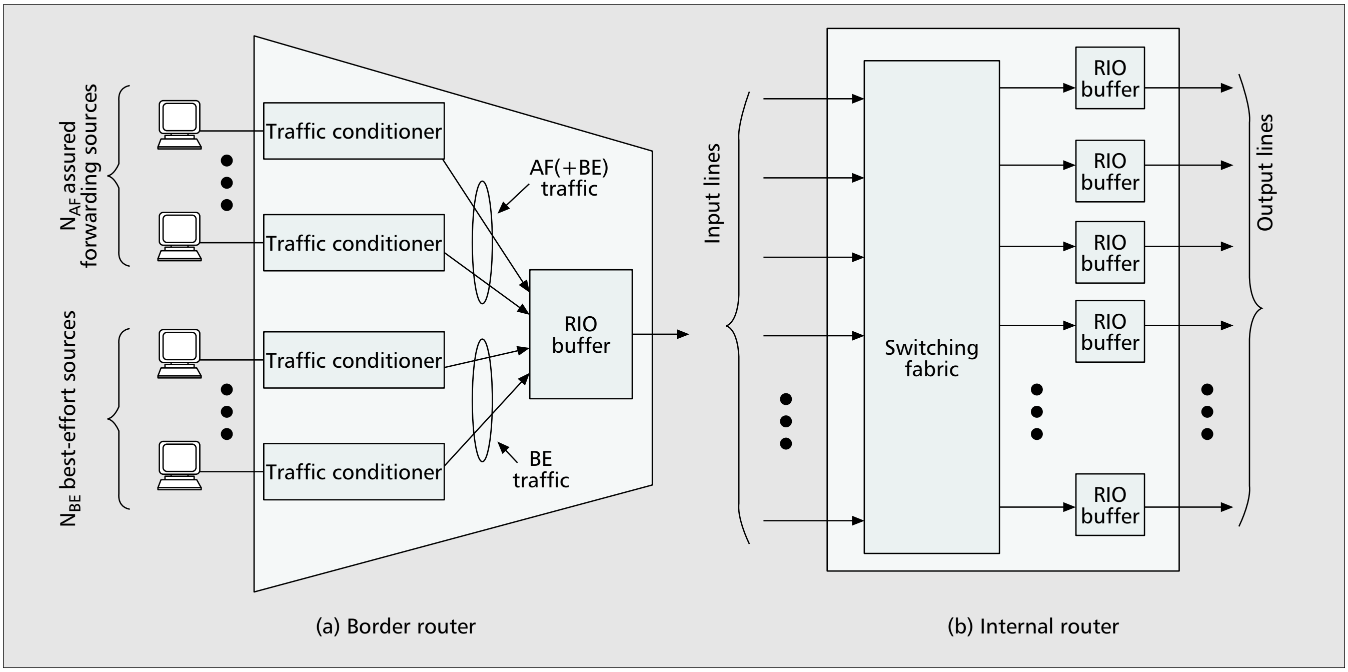


Figure 2. *a) The border router; b) the internal router.*

In [16] a set of experimental measures is presented. The main result is that the differentiation among the transmission rates of TCP flows can be achieved. However, it is difficult to provide the required rates with a good approximation.

In [17] it is shown that through a suitable choice of RED parameters, two TCP and two UDP streams can be managed in compliance with the required transmission rates. In the article's simulations, the token bucket depth is greater than or at least equal to the TCP window. In this way, TCP dynamics are not modeled, as discussed by the authors.

In [18] the effect of the number of drop precedences in the Assured Forwarding service is examined, with a focus on the impact of packet dropping on congestion-sensitive traffic streams, such as TCP streams. The article shows how without any adjustment mechanism most of the excess network bandwidth is used by congestion-insensitive flows. Therefore, the network should improve the allocation of excess network bandwidth to excess packets of congestion-sensitive flows. With the Assured Forwarding service, multiple drop precedence levels can be successfully applied only when the network has sufficient bandwidth. If the network operates close to its maximum capacity or if it is already in a congestion status, three levels of drop precedence are redundant, as there is little excess bandwidth to be shared among traffic flows.

In [19] the performance of traffic flows in a DiffServ network with several buffer management schemes is analyzed. Two versions of multi-level RED are proposed to meet the requirements of the Assured Forwarding Per Hop Behavior. It is shown that the RIO buffer management technique has higher performance than the MRED (Multiple RED) and WRED (Weighted RED) techniques.

In [20] the authors propose an innovative technique to guarantee weighted fairness to individual flows, called "Scalable Core with Aggregation Level LabEling applied to Weighted Fair bandwidth Sharing" (SCALE-WFS). This technique does not require a per-flow management effort from internal routers. Simulation results show that SCALE-WFS is suitable for heterogeneous sources (TCP/UDP), which can also have different round-trip times, profile rates, and bandwidth provisioning on one or multiple congested links. The article shows that SCALE-WFS is effective, scalable, and robust for weighted fairness.

A Measurement-based Connection-Oriented Assured Service (MCOAS) for TCP applications is proposed in [21]. The goals of the article are to achieve end-to-end service assurance for TCP applications, high network resource utilization, and high scalability. Connection-oriented Assured Service is enabled with an aggregate resource reservation approach. A simple adaptive dropping-threshold algorithm prevents Best-Effort traffic from overloading the Assured Forwarding traffic at internal routers. Simulation results show that MCOAS can guarantee a high level of end-to-end service assurance for aggregate Assured Forwarding TCP traffic, and, at the same time, a reasonably high throughput for Best-Effort traffic. However, the MCOAS technique does not provide good isolation of Assured Forwarding flows.

In [22] the interaction between short-lived and long-lived TCP flows is studied. The article shows the need for an architecture that allocates short-lived and long-lived TCP flows into separate classes. With a class-based separation, short-lived and long-lived TCP flows are stored by routers in different service queues. This approach improves performance in terms of both predictability and fairness with respect to traditional shared-queueing systems with tail-drop and Random Early Drop (RED) policies.

The research presented in [23–27] proposes several changes in TCP congestion and flow control mechanisms to improve TCP's ability to cope with service unfairness in DiffServ networks. The overall contribution of this research is to improve the performance of TCP traffic flows with the Assured Forwarding service. However, throughput is not guaranteed with sufficient precision.

In [28] a new packet marker is proposed to enhance the performance and fairness of TCP flows. The article shows that the performance of TCP flows unfortunately still depends on TCP dynamics. Enhanced packet marking techniques are proposed in [29] and [30].

An important issue that has not been previously considered is the impact on TCP performance of excess traffic that Assured Forwarding sources are permitted to transmit. Excess traffic is usually forwarded with a higher drop precedence and, in turn, it can penalize the performance of TCP flows during congestion periods. The goal of this article is to complement previous studies, addressing the impact of Best-Effort traffic, with the analysis of excess traffic from Assured Forwarding sources as an additional cause for low TCP performance.

The article proposes a RIO-based mechanism together with a Congestion Signaling Algorithm (CSA), implemented in both internal routers and border routers to guarantee the target throughput of TCP flows. The Congestion Signaling Algorithm dynamically adjusts the amount of excess packets accepted by border routers. In this way, the performance degradation due to excess packets during congestion periods is significantly lower. Target throughput is obtained without a need for overprovisioning. Fairness improvements are also observed and could represent an interesting subject for further research.

Modeling Network Devices and Traffic Conditioning Procedures

In this section, we describe the models of network devices and of traffic conditioning procedures for performance analyses. We also propose a new packet marking technique and discuss the Congestion Signaling Algorithm that we have designed to guarantee the provisioned rate of TCP traffic flows.

We assume that each Assured Forwarding Per Hop Behavior is specified by its average rate *r* (kb/s). All in-profile packets must be delivered with a low loss probability. At the access link, users can transmit at an average rate higher than *r*. Out-of-profile packets, that is, packets exceeding *r*, are handled as Best-Effort traffic.

Modeling Border and Internal Routers

Both Assured Forwarding and Best-Effort users can access border routers directly (Fig. 2a). The number of Assured Forwarding and Best-Effort users is N_{AF} and N_{BE}, respectively. In border routers, traffic conditioning functions, including the packet dropping operation, are performed by the traffic conditioner (Fig. 1). We have a conditioner for each input line and a common buffer (RIO buffer), controlled by a RIO packet discarding procedure.

The RIO packet discarding procedure regulates network congestion by selectively discarding in-profile and out-of-profile packets. In particular, when buffer occupancy is high, out-of-profile packets are dropped with higher probability than in-profile packets.

The traffic conditioner can modify the per hop behavior of non-conformant packets by downgrading part of the Assured Forwarding traffic to Best Effort. Therefore, each traffic conditioner receiving Assured Forwarding traffic can feed the RIO buffer with both Assured Forwarding traffic and Best-Effort traffic.

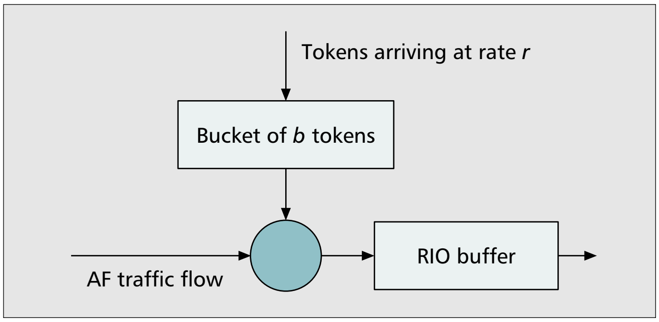


Figure 3. *Token bucket traffic filter.*

The RIO buffer discards packets based on their per hop behavior. The packet dropping probability increases with the congestion level and is higher for the Best-Effort traffic. Internal routers (Fig. 2b) do not perform complex traffic conditioning functions. Packets enter internal routers from input lines and are routed to the appropriate output lines on the basis of their destination. Each output line is provided with a RIO buffer performing the same selective packet discarding procedures adopted by border routers.

The Token Bucket — The traffic conditioning function is performed by a token bucket traffic filter (Fig. 3), provided with a bucket of *b* tokens. For the sake of simplicity, we assume that packets have constant length *L* (bytes). The token bucket is filled at a $10^3r/(8L)$ (tokens/s) rate, or alternatively, at an *r* (kb/s) rate. When an Assured Forwarding packet enters the filter, if at least one token is available in the bucket, the packet is served without downgrading its per hop behavior and one token is removed from the bucket. Otherwise, if there are no tokens in the bucket, the packet is served, but its per hop behavior is downgraded.

Therefore, the token bucket splits the traffic flow into a *conformant flow* (in-profile packets) at an average *r* (kb/s) rate, and a *non-conformant flow* (out-of-profile packets), whose per hop behavior is downgraded.

The RIO Buffer — The RIO technique applies the RED mechanism [13] to both in-profile (IN) and out-of-profile (OUT) packets, which are queued in the same buffer. Therefore, RIO

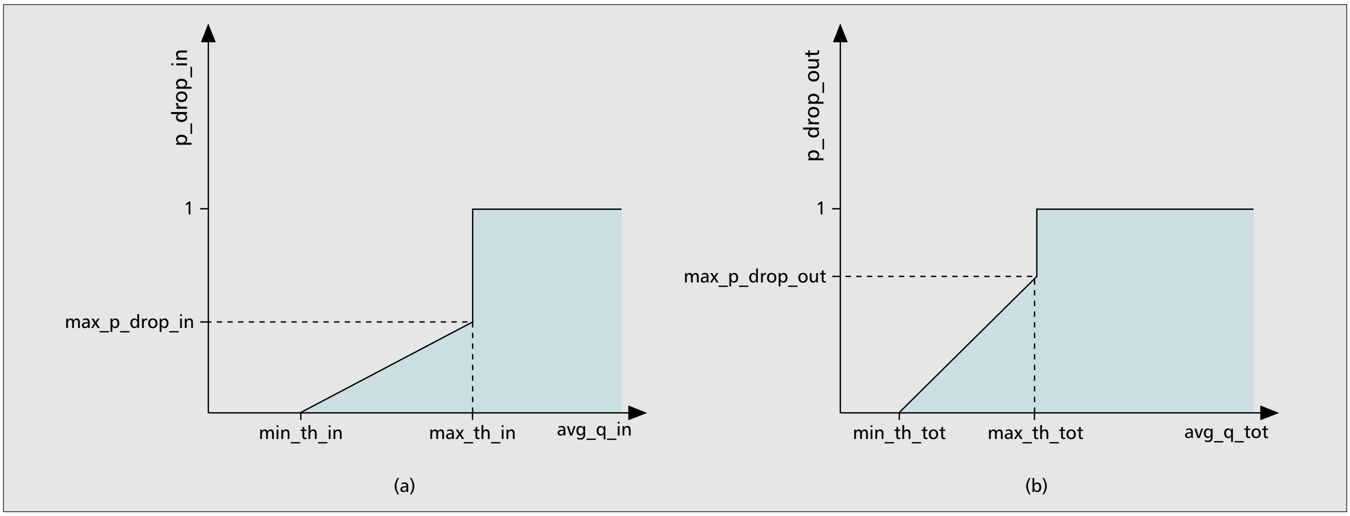
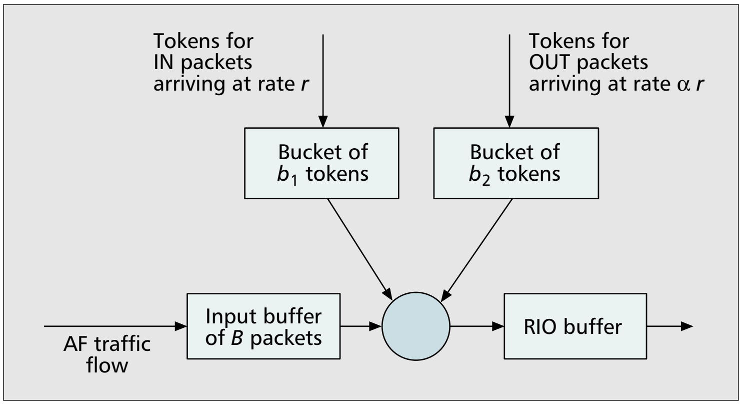


Figure 4. *Dropping probability of IN and OUT packets in RIO.*



■ Figure 5. *Enhanced token bucket.*

is based on two concurrent estimations of the average queue length for the queue of IN packets, *avg_q_in*, and for the queue of IN and OUT packets, *avg_q_tot* [12].

The dropping probability of IN packets, *p_drop_in*, only depends on the number of IN packets in the buffer, while the dropping probability of OUT packets, *p_drop_out*, depends on the total number of packets in the buffer (both IN and OUT). Furthermore, the RIO scheme needs a proper setting of minimum and maximum thresholds for IN packets, *min_th_in* and *max_th_in*, and for OUT packets, *min_th_out* and *max_th_out*.

New packets arriving at the RIO buffer are handled according to the following procedure:

1) If the packet is IN (Fig. 4a):
- If *avg_q_in* ≤ *min_th_in*, the packet is accepted.
- If *min_th_in* < *avg_q_in* ≤ *max_th_in*, the packet is dropped with probability *p_drop_in* (*p_drop_in* grows linearly from zero to *max_p_drop_in*).
- If *avg_q_in* > *max_th_in*, the packet is dropped.

2) If the packet is OUT (Fig. 4b):
- If *avg_q_tot* ≤ *min_th_out*, the packet is accepted.
- If *min_th_out* < *avg_q_tot* ≤ *max_th_out*, the packet is dropped with probability *p_drop_out* (*p_drop_out* grows linearly from zero to *max_p_drop_out*).
- If *avg_q_tot* > *max_th_out*, the packet is dropped.

Control of the Percentage of Non-conformant Packets at the Token Bucket

As it will be shown later, a poor performance in the transport of TCP traffic over the Assured Forwarding service is observed if the percentage of downgraded Assured Forwarding packets is not kept low. In particular, when the number of Assured Forwarding traffic sources is high, a large number of non-conformant packets can be offered to border routers. This traffic is forwarded through the RIO buffer with a higher packet loss probability. This can significantly reduce the throughput of TCP connections. In fact, packet losses trigger the slow start and congestion avoidance procedures of TCP and, in turn, these procedures slow down TCP connections [31]. Thus, even if conformant packets are forwarded with a lower loss probability, the higher loss probability of non-conformant packets has a negative effect on the overall performance of TCP flows. Therefore, if the percentage of non-conformant packets is not kept low, the provisioned throughput of Assured Forwarding flows cannot be guaranteed.

To control throughput, we monitor the congestion status of the RIO buffer at border routers. We implement the following control procedures. When the RIO buffer is not congested, the token bucket filter can increase the percentage of non-conformant traffic in order to exploit the bandwidth available on the output line. Conversely, when the RIO buffer is congested, the token bucket filter must reduce the percentage of non-conformant traffic. In this way, Assured Forwarding packets, which are mostly conformant, are discarded with a lower probability, and therefore the performance of the Assured Forwarding traffic is downgraded to a lower degree during congestion periods in the RIO buffer.

In order to implement a token bucket filter that can automatically regulate the percentage of non-conformant packets, we propose an enhanced version of the token bucket (Fig. 5). The traffic filter is provided with an input buffer and a second bucket of tokens to handle non-conformant packets. The bucket of conformant tokens has depth equal to b_1 and is filled at rate r, while the bucket of non-conformant tokens has depth b_2 and is filled at rate αr. By properly setting the α factor, the percentage of non-conformant packets can be differently limited for each Assured Forwarding flow in the network.

When an Assured Forwarding packet enters the filter, one of the following events can occur: if the input buffer is full, the packet is dropped; otherwise, it is queued in the input buffer. Each packet in the input buffer is served according to the following three-step procedure:

1) If the conformant bucket has at least one token, it is served as conformant and a token is removed from the bucket.
2) If the conformant bucket is empty, but the out-of-profile bucket has at least one token, then the packet is served with per hop behavior downgraded to Best Effort and a token is removed from the corresponding bucket.
3) If both buckets are empty, the packet is left in the input buffer.

In-profile and out-of-profile accepted packets are forwarded to the RIO buffer. Note that for TCP applications the queuing delay is not critical.

The α parameter represents the percentage of non-conformant packets transmitted by the token bucket. We propose a Congestion Signaling Algorithm (CSA) that adjusts the value of the α parameter when the RIO buffer is congested.

The RIO buffer can be either non-congested (when *avg_q_tot* < *min_th_out*) or congested (when *avg_q_tot* > *min_th_out*). Let us assume that the arrival of the first OUT packet during a congestion period occurs at time t_0. Two timers, T_1 and T_2 (with $T_1 < T_2$), are started and a packet counter, P, is set to 1.[1] P is a modulo-8 counter and it returns to the 0 value at the arrival of the 8-th OUT packet. Let us assume that the 8-th arrival occurs at time t. One of the following two conditions holds:

Case 1: $(t - t_0) > T_1$, that is, timer T_1 expires before the arrival of the 8-th OUT packet. In this case, the α parameter is decreased by $\Delta\alpha$, in such a way that $\alpha = \max\{\alpha - \Delta\alpha, min\_\alpha\}$. $\Delta\alpha$ is the decrease/increase step and min_α is the minimum value allowed for α. In particular, α can be decreased at most by $\Delta\alpha$ every T_1 seconds; thus, the maximum decreasing rate of α evaluates to $\Delta\alpha/T_1$. After α has been decreased, both T_1 and T_2 are restarted.

Case 2: $(t - t_0) < T_1$, that is, the 8-th packet arrives before the expiration of timer T_1. In this case, the value of α is not decreased to prevent an overly rapid reduction of α.

In any case, when T_2 expires (that is, when the α parameter has not been decreased for T_2 consecutive seconds), if the

[1] *The following procedures and timers are taken from the ISUP congestion management procedures related to congestion notification from the signaling MTP-2 buffers (see ITU Q.704 [32] and ITU Q.764 [33]) and adapted to the management of RIO buffers. In particular, the T_1 and T_2 timers act as the T_{29} and T_{30} timers in ISUP, respectively.*

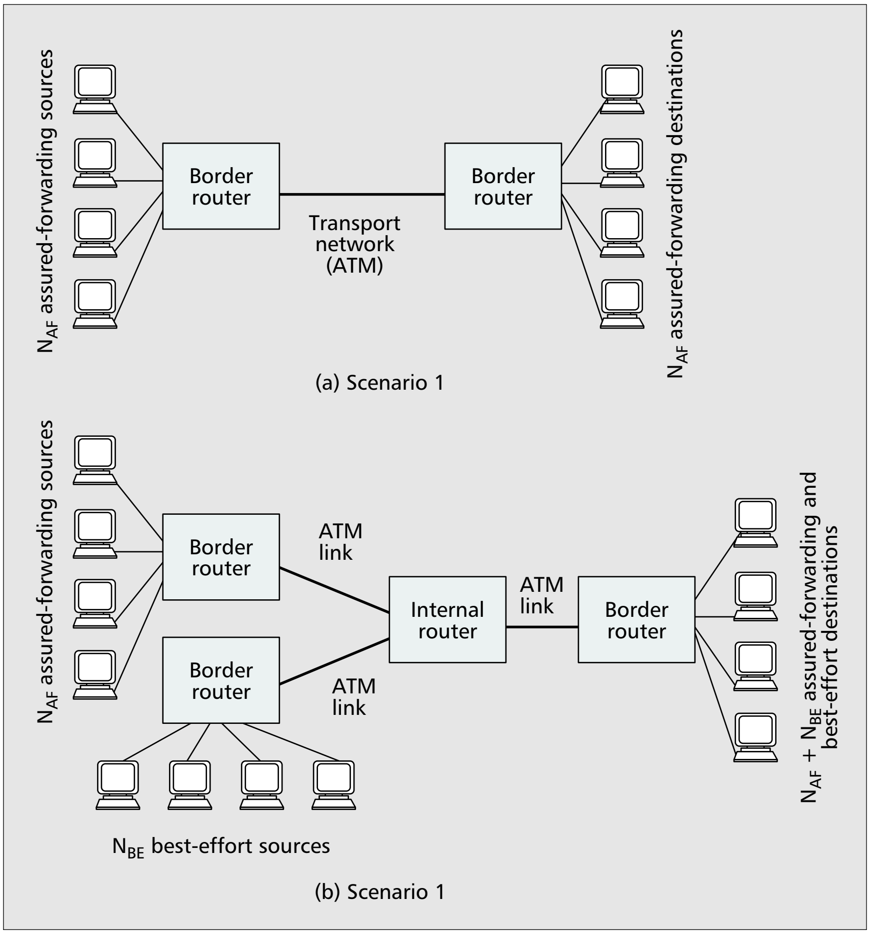


Figure 6. *a) Scenario 1 and b) Scenario 2.*

RIO buffer is not congested, the value of α is automatically increased by $\Delta\alpha$ and timer T_2 is restarted. In this way, the maximum increasing rate of the α parameter after a congestion period is set to $\Delta\alpha/T_2$.

The token bucket functions can be implemented with software procedures. Therefore, the implementation of the enhanced token bucket does not require additional circuits.

Remote Control of the Percentage of Non-conformant Packets

In non-congested border routers, Assured Forwarding flows can enter the network with a high percentage of non-conformant packets. If these traffic flows enter a congested RIO buffer of an internal router, they are likely to experience high packet loss due to their non-conformant component. This packet loss will in turn significantly downgrade TCP performance. Therefore, it is important to regulate the percentage of non-conformant packets of border routers not only with local information about the congestion status of the RIO buffer, but also with remote information about the congestion status in RIO buffers of internal routers.

This allows border routers to fine tune the value of α on the basis of the congestion status of internal routers. Congestion information can be signaled by internal routers through the IP protocol, for example, by means of Internet Control Message Protocol (ICMP) packets [34]. In the short term, the identification of border routers may raise critical implementation issues. However, in future network scenarios where DiffServ and the MultiProtocol Label Switching (MPLS) techniques coexist, border routers involved in an Assured Forwarding aggregate will be easy to identify, as end-to-end routes will be explicitly defined.

Performance Analysis

The main objective of this section is to analyze through simulation the transport performance of TCP traffic flows over the Assured Forwarding service. We will show that the basic Assured Forwarding service has poor performance without congestion notification. By adopting the enhanced token bucket and the Congestion Signaling Algorithm described in the previous section, the provisioned rate of TCP traffic flows can be guaranteed.

Analyses are performed in two reference scenarios. Results obtained for more complex scenarios are not reported here as they confirm the observations that can be drawn from reference scenarios. In the first scenario (Fig. 6a), N_{AF} Assured Forwarding traffic sources connected to a border router send traffic to an equal number of remote destinations connected to a second border router. No internal IP router is involved in the transport of these traffic flows.

The transport network connecting routers is based on Asynchronous Transfer Mode (ATM) technology. In particular, permanent ATM constant bit rate (CBR) connections are supposed to be implemented as it is quite common among telecommunication companies to have an ATM backbone network used for both data and voice traffic. A point-to-point link is easily set up within an ATM transport network by establishing a permanent ATM virtual connection between two IP routers and by transporting IP over ATM by means of the ATM Adaptation Layer 5 protocol.

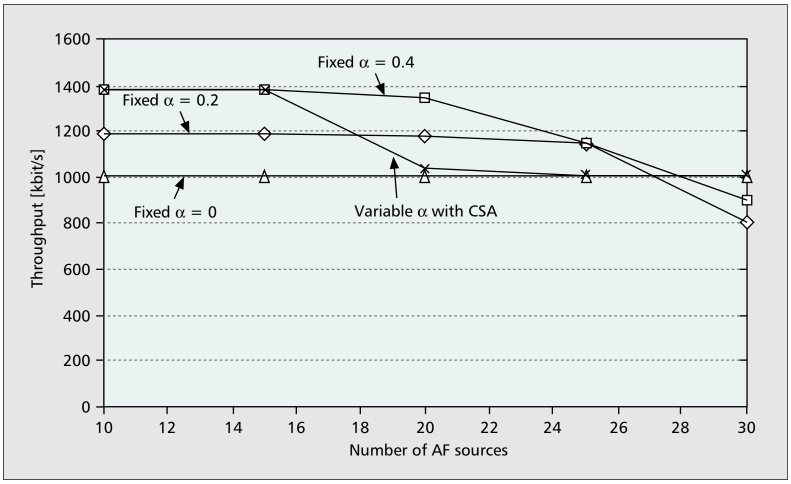
All sources and destinations are connected to border routers with dedicated IP point-to-point links and adopt the TCP/IP protocol stack. In particular, each traffic source establishes a TCP connection with a remote destination before starting the data transmission phase.

In the second scenario (Fig. 6b), N_{AF} Assured Forwarding sources send data to N_{AF} Assured Forwarding remote destinations, and N_{BE} Best-Effort sources send traffic to N_{BE} Best-Effort remote destinations. Assured Forwarding and Best-Effort destinations are connected to the same border router. Similarly, Assured Forwarding and Best-Effort traffic flows are combined by the internal IP router and share the same ATM backbone connection.

General Settings of Parameters

In both scenarios, transmission channels are ideal, that is, no transmission errors occur. The link connecting any source/destination to the IP border router has a 2 Mb/s capacity, according to the standard E1 PDH (Plesiochronous Digital Hierarchy) system. ATM CBR permanent connections between routers inside the DiffServ domain have 34 Mb/s peak cell rate, according to the standard E3 PDH system [35].

We consider IP packets with length L equal to 576 bytes. Each IP packet generates 12 ATM cells, with an ATM overhead evaluating to $12 \cdot 5 = 60$ bytes. Therefore, the net capacity of ATM CBR connections at the IP level is equal to



■ Figure 7. *Throughput versus number of Assured Forwarding sources in scenario 1.*

34 · 576/(576 + 60) = 30.792 Mb/s. For Assured Forwarding flows, this capacity can be allocated by IP routers in such a way that the following acceptance constraint is satisfied:

$$\sum_{i=1}^{n} r_i < 30.792\ Mb/s,$$

where n is the cardinality of a generic set of IP Assured Forwarding flows and r_i is the token bucket rate of the i-th flow. Each TCP source has been implemented according to the specifications discussed in [31]. Additional TCP settings are: 64 kbytes maximum window size and 500 ms timer granularity. All TCP sources are supposed always to have data to transmit when their transmission window is open.

The settings of parameters for the Assured Forwarding per hop behaviors considered in this study are shown in Table 1 (the per hop behavior for the Best-Effort traffic is named BE_PHB). The settings of parameters for the RIO buffer and for the CSA algorithm are also reported in Table 1.

Packets arrive at the input buffer at a 2 Mb/s access rate. The capacity of the input buffer strictly depends on this access rate and on the depth of the conformant bucket b_1. The value of b_1 determines the length of conformant packet bursts in the output flow. With several simulation experiments, we have determined that an input buffer with capacity B = 100 packets is appropriate when the access rate evaluates to 2 Mb/s and b_1 ranges from 10 tokens to 50 tokens. It has been verified that performance is not influenced by the value of b_1 which, in the following, has been set to 10. Moreover, shown in Appendix C, performance is not significantly sensitive to input buffer capacity, B.

The depth of the non-conformant bucket, b_2, regulates the length of the bursts of non-conformant packets entering the network. By choosing b_2 = 1, we prevent excess traffic from producing bursts of packets. Performance is not significantly sensitive to b_2, as shown in Appendix B.

For an appropriate and predictable operation of the Congestion Signaling Algorithm, it is important that system performance does not significantly depend on the settings of timers T_1 and T_2. This issue is discussed in Appendix A.

Forty million packets were generated for each simulation.

Settings of parameters for the AF PHBs				
Type of PHB	Average rate, *r* (kb/s)	Depth of input buffer, *B* (packets)	Depth of in-profile bucket, b_1 (tokens)	Depth of out-of-profile bucket, b_2 (tokens)
AF_PHB_1	1600	100	10	1
AF_PHB_2	1000	100	10	1
AF_PHB_3	400	100	10	1

Settings of parameters for the RIO buffer	
min_th_in	70 packets
max_th_in	120 packets
max_p_drop_in	0.01
min_th_out	10 packets
max_th_out	60 packets
max_p_drop_out	0.2

Settings of parameters for the CSA	
Timer T_1	30 ms
Timer T_2	90 ms
min_α	0.04
max_α	0.4
Δα	0.02

■ Table 1. *Settings of parameters for the AF_PHBs, for the RIO buffer, and for the CSA algorithm.*

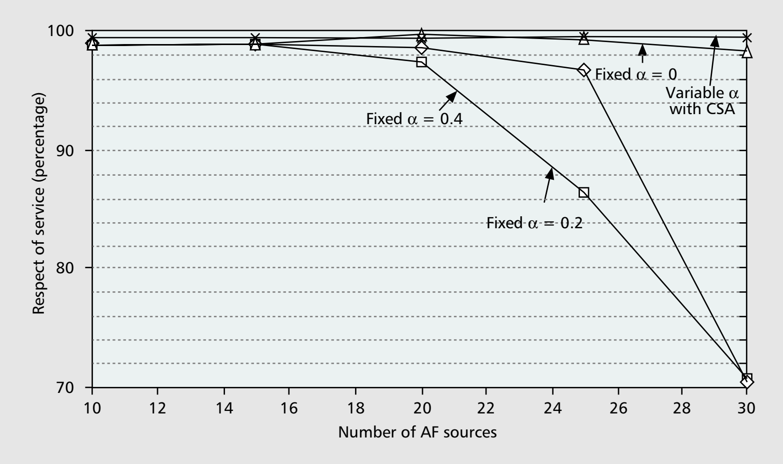


Figure 8. *Respect of service versus number of AF sources in scenario 1.*

Performance Parameters

Throughput is measured in kb/s and is defined as the ratio of the total number of bits transmitted by a source during a simulation to total simulation time.

The Respect of Service (RoS) is defined as the ratio of the actual rate of conformant packets exiting the meter to the provisioned token bucket rate *r*. Obviously, RoS ranges from zero to one. Values of RoS close to one indicate that the Assured Forwarding flow is receiving a bandwidth close to *r*, while small values of RoS correspond to Assured Forwarding flows receiving a bandwidth lower than *r*.

Packet loss is measured at multiple interfaces: P_{inb} is the packet loss at the token bucket input buffer; $P_{nc,i}$ is the loss of non conformant packets at the *i*-th IP router; $P_{c,i}$ is the loss of conformant packets at the *i*-th IP router; $P_{be,i}$ is the loss of Best-Effort packets at the *i*-th IP router.

Scenario 1

In scenario 1 (Fig. 6a), which considers a set of N_{AF} homogeneous Assured Forwarding sources operating with AF_PHB_2, throughput (Fig. 7) and Respect of Service (Fig. 8) are plotted for different values of the α parameter. In Figs. 7 and 8, α is either fixed or dynamically adjusted by the Congestion Signaling Algorithm.

When α = 0, throughput is constant and equal to the provisioned 1000 kb/s. Moreover, Respect of Service is high, greater than 99 percent. The value of Respect of Service is high because the network does not allocate more bandwidth than the provisioned 1000 kb/s. The provisioned rate is strictly guaranteed, but spare capacity, available when the number of sources is small, is not used. Throughput is regulated by packet losses in the input buffer. Table 2 reports the values of packet loss observed in the input buffer and in the RIO buffer.

When α = 0.2, throughput is higher than the provisioned 1000 kb/s when the number of sources is small. Throughput is adjusted by packet loss in the input buffer (Table 2). When the number of sources is high, throughput can decrease below the provisioned 1000 kb/s. From Fig. 8, it can be noted that in this case Respect of Service dramatically decreases to 70 percent with 30 Assured Forwarding sources. The network tries to allocate more bandwidth than the provisioned 1000 kb/s, but this leads to heavy packet loss in the RIO buffer, as shown in Table 2.

The same phenomena occur for α = 0.4. In particular, a higher degradation of Respect of Service can be observed as the number of sources grows (Fig. 8).

We can conclude that when the α parameter is fixed and greater than 0, it is possible to achieve a per-connection throughput greater than the provisioned 1000 kb/s when the ATM link is not overloaded (i.e., when the number of sources is not too large). However, throughput and Respect of Service can significantly decrease as the number of sources grows and provisioned rates are not guaranteed.

A different behavior is observed when the α parameter is adjusted by the Congestion Signaling Algorithm. In this case, throughput is always greater than the provisioned 1000 kb/s and Respect of Service is always greater than 98 percent, that is, satisfactory. As the number of sources grows and congestion arises, packet losses are also observed in the RIO buffer (Table 2). This reduces the throughput of TCP connections and, in turn, the packet loss of the input buffer. However, as throughput decreases, Respect of Service does not decrease. This shows that the provisioned throughput of IN packets is guaranteed. As a drawback, in some cases it is not possible to allocate all of the

N_{AF}	Throughput (kb/s)	RoS (%)	P_{inb}	$P_{nc,1}$	$P_{c,1}$
Fixed α = 0.4					
10	1381	98.7	3.7e-04	0	0
15	1381	98.7	3.7e-04	0	0
20	1346	97.3	3.7e-05	4.51e-03	0
25	1157	86.5	0	8.43e-03	8.0e-07
30	907	70.4	0	1.20e-02	1.4e-04
Fixed α = 0.2					
10	1188	99.09	4.1e-04	0	0
15	1188	99.09	4.1e-04	0	0
20	1181	98.78	8.0e-05	2.59e-03	0
25	1148	96.77	0	5.77e-03	7.84e-07
30	811	70.07	0	1.01e-02	1.30e-04
Fixed α = 0					
10	994.97	99.49	4.7e-04	0	0
15	994.97	99.49	4.7e-04	0	0
20	994.97	99.49	4.7e-04	0	0
25	995.37	99.53	4.4e-04	0	9.0e-07
30	994.44	99.44	2.0e-04	0	1.2e-04
Variable α with Congestion Signaling Algorithm					
10	1381.32	98.70	3.7e-04	0	0
15	1381.31	98.70	3.7e-04	0	0
20	1040.13	99.71	2.6e-04	3.5e-05	0
25	1013.19	99.31	9.05e-05	3.5e-04	8.86e-07
30	1004.72	98.42	1.64e-05	9.2e-04	1.60e-04

Table 2. *Throughput, respect of service, and packet loss in the input buffer and in the RIO buffer.*

excess bandwidth to connections, as shown in Fig. 7 for 20 sources.

Scenario 2

In Scenario 2 (Fig. 6b), sources are served both with Assured Forwarding and with Best-Effort per hop behaviors. Assured Forwarding and Best-Effort flows are generated in different locations and multiplexed by internal routers. We investigate four different cases with increasing load:

- $N_{AF} = 10, N_{BE} = 10$
- $N_{AF} = 25, N_{BE} = 10$
- $N_{AF} = 10, N_{BE} = 40$
- $N_{AF} = 25, N_{BE} = 40$

Border routers always perform the Congestion Signaling Algorithm, while internal routers may or may not perform the Congestion Signaling Algorithm. Results are shown in Table 3.

In case (a) of Table 3, the Congestion Signaling Algorithm is only applied by the border router. Load is relatively low and no congestion occurs in border and internal routers. Throughput is greater than the target value in the per hop behavior, Respect of Service is high, and packet loss is acceptable. With the adoption of the Congestion Signaling Algorithm in the internal router, performance improvements are negligible.

In case (b), the Assured Forwarding traffic is heavier and congestion may occur. Without the Congestion Signaling Algorithm in the internal router, the packet loss in the border router, $P_{nc,br}$, is low, but the throughput of AF_PBH_1 is not guaranteed. A high percentage of non-conformant packets can reach the internal router, where a considerable loss of both conformant and non-conformant packets is observed with a consequent reduction of Respect of Service. If the Congestion Signaling Algorithm is activated in the internal router, it is possible to guarantee the provisioned throughput of Assured Forwarding per hop behaviors. Moreover, Respect of Service is greater than 99 percent.

Case (c) is characterized by a small number of Assured Forwarding sources and a high number of Best-Effort sources. Therefore, a significant loss of non-conformant packets is observed in the internal router. Without the Congestion Signaling Algorithm in the internal router, the provisioned throughput of the AF_PHB_1 flow is guaranteed, but Respect of Service is low. The AF_PHB_1 flow is heavily damaged by congestion, as it can carry a high percentage of non-conformant packets. If the Congestion Signaling Algorithm is activated in the internal router, the expected throughput is fully guaranteed for both AF_PHB_1 and AF_PHB_3 with high values of Respect of Service.

Case A: CSA not applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
5	AF_PHB_1	1945.3	97.5	0	0	0	6.0e-04	0
5	AF_PHB_3	554.7	99.2	2.2e-04	0	0	9.0e-04	0
10	BE_PHB	1216.5		Packet loss in the internal router: 4.5e-03				
Case A: CSA applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
5	AF_PHB_1	1965.2	98.5	0	0	0	6.0e-04	0
5	AF_PHB_3	548.1	99.7	3.5e-04	0	0	7.3e-04	0
10	BE_PHB	1237.5		Packet loss in the internal router: 4.5e-03				
Case B: CSA not applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
13	AF_PHB_1	1574.8	93.2	3.3e-06	5.4e-04	0	1.6e-03	2.8e-06
12	AF_PHB_3	428.4	97.9	4.4e-06	4.6e-05	0	2.5e-03	8.7e-06
10	BE_PHB	297.1		Packet loss in the internal router: 3.9e-02				
Case B: CSA applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
13	AF_PHB_1	1639.2	99.5	1.6e-05	5.4e-05	0	5.6e-04	3.7e-06
12	AF_PHB_3	408.35	99.7	1.8e-05	7.3e-05	0	9.5e-04	2.3e-06
10	BE_PHB	271.1		Packet loss in the internal router: 4.4e-02				
Case C: CSA not applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
5	AF_PHB_1	1683.4	86.8	0	0	0	4.3e-03	0
5	AF_PHB_3	531.2	95.8	3.2e-04	0	0	7.5e-03	0
40	BE_PHB	397.7		Packet loss in the internal router: 2.6e-02				
Case C: CSA applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
5	AF_PHB_1	1631.4	97.6	1.4e-06	0	0	1.2e-03	0
5	AF_PHB_3	410.5	98.9	5.9e-04	0	0	1.0e-03	0
40	BE_PHB	415.7		Packet loss in the internal router: 2.5e-02				
Case D: CSA not applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
13	AF_PHB_1	1330.3	72.4	0	5.2e-05	0	7.3e-03	0
12	AF_PHB_3	456.1	83.9	0	4.6e-05	0	1.5e-02	0
40	BE_PHB	157.7		Packet loss in the internal router: 5.8e-02				
Case D: CSA applied in the IR								
N	PHB	thr (kb/s)	RoS %	P_{inb}	$P_{nc,br}$	$P_{c,br}$	$P_{nc,ir}$	$P_{c,ir}$
13	AF_PHB_1	1623.6	98.7	0	6.9e-05	0	1.7e-03	0
12	AF_PHB_3	408.3	99.8	0	5.0e-05	0	1.9e-03	0
40	BE_PHB	59.1		Packet loss in the internal router: 1.7e-01				

■ Table 3. *Performance of Scenario 2, cases (a), (b), (c), and (d). (br = border router, ir = internal router).*

In case (d), with a large number of both Assured Forwarding and Best-Effort sources, congestion can occur both in border and internal routers. Without the Congestion Signaling Algorithm in the internal router, only the throughput of the AF_PHB_3 flow is guaranteed, that is, the service of different Assured Forwarding flows is unfair. Furthermore, Respect of Service is poor, especially for the AF_PHB_1 flow. On the

Appendix A: Sensitivity of the Congestion Signaling Algorithm to Timers T_1 and T_2

The values of timers T_1 and T_2 may influence the performance of the Congestion Signaling Algorithm. A relationship between timers and performance would raise implementation problems.

Performance is evaluated in scenario 1 with AF_PHB_2 parameters. The following four settings of timers T_1 and T_2 have been considered:

- T_1 = 15 ms, T_2 = 45 ms
- T_1 = 15 ms, T_2= 60 ms
- T_1 = 30 ms, T_2 = 90 ms
- T_1 = 30 ms, T_2 = 120 ms

Simulation results confirm that throughput and Respect of Service, shown in Tables 4 and 5, respectively, do not significantly depend on the values of timers T_1 and T_2. We can conclude that the values of timers are not critical for the implementation of the Congestion Signaling Algorithm.

N_{AF}	Throughput (kb/s) T1 = 15ms, T2 = 45 ms	Throughput (kb/s) T1 = 15ms, T2 = 60 ms	Throughput (kb/s) T1 = 30ms, T2 = 90 ms	Throughput (kb/s) T1 = 30ms, T2 = 120 ms
10	1381.33	1381.32	1381.32	1381.32
15	1381.31	1382.31	1381.31	1381.31
20	1046.41	1042.47	1040.13	1027.03
25	1013.73	1013.88	1013.19	1013.04
30	1007.78	1010.55	1004.72	1006.43

■ Table 4. *Throughput for different settings of timers* T_1 *and* T_2.

N_{AF}	ROS (%) T1 = 15ms, T2 = 45 ms	ROS (%) T1 = 15ms, T2 = 60 ms	ROS (%) T1 = 30ms, T2 = 90 ms	ROS (%) T1 = 30ms, T2 = 120 ms
10	98.70	98.70	98.70	98.70
15	98.70	98.70	98.70	98.70
20	99.69	99.69	99.71	99.74
25	99.35	99.37	99.31	99.29
30	98.77	99.09	98.42	98.64

■ Table 5. *ROS for different settings of timers* T_1 *and* T_2.

Appendix B: Sensitivity of the Congestion Signaling Algorithm to b_2

This Appendix evaluates the sensitivity of system performance to b_2. Throughput (Table 6) and Respect of Service (Table 7) are evaluated with b_2 =1, 2, 3, 4, and 5. Simulation results confirm that throughput and Respect of Service do not significantly depend on the values of b_2.

N_{AF}	thr. (kb/s), $b_2 = 1$	thr. (kb/s), $b_2 = 2$	thr. (kb/s), $b_2 = 3$	thr. (kb/s), $b_2 = 4$	thr. (kb/s), $b_2 = 5$
10	1381.32	1381.36	1381.45	1381.39	1381.43
15	1381.31	1381.36	1381.45	1381.38	1381.43
20	1040.13	1040.03	1040.08	1039.5	1036.57
25	1013.19	1012.04	1011.69	1011.19	1017.04
30	1004.72	1010.72	1010.76	1010.48	1009.69

■ Table 6. *Throughput for different values of* b_2.

N_{AF}	RoS (%), $b_2 = 1$	RoS (%), $b_2 = 2$	RoS (%), $b_2 = 3$	RoS (%), $b_2 = 4$	RoS (%), $b_2 = 5$
10	98.70	98.70	98.71	98.70	98.71
15	98.70	98.70	98.71	98.70	98.71
20	99.71	99.61	99.83	99.83	99.62
25	99.31	99.27	99.17	99.05	99.60
30	98.42	99.17	99.12	99.09	98.97

■ Table 7. *Respect of service for different values of* b_2.

other hand, the adoption of the Congestion Signaling Algorithm guarantees throughput and, at the same time, a high value of Respect of Service.

Conclusions

In this article, the interaction between TCP and the Assured Forwarding service has been examined. Results show that TCP sources may experience poor performance with the Assured Forwarding service.

We have proposed a solution based on the design of an ad-hoc traffic conditioner, implemented with an enhanced token bucket regulating the percentage of non-conformant traffic at each DiffServ border router. A Congestion Signaling Algorithm feeds the enhanced token bucket with information on the congestion status of the RIO buffer at border routers and/or internal routers. In this way, the enhanced token bucket controls the percentage of non-conformant packets on the basis of the congestion status of both border routers (local information) and internal routers (remote information).

We have considered two scenarios and performed simulations with different settings of traffic sources served with the Assured Forwarding and with Best-Effort per hop behaviors. We have shown that our solution can achieve values of Assured Forwarding throughput higher than target values. This holds in different traffic load conditions and regardless of the congestion caused by the Best-Effort traffic at border and internal routers. We have observed values of Respect of Service close to 100 percent during congestion periods. This result confirms that our algorithm guarantees the expected throughput in case of congestion caused by excess Best-Effort traffic. One of the most important features of our algorithm is that it does not require a complex traffic flow management in internal routers, in compliance with the DiffServ architecture,

Appendix C. Sensitivity of the Congestion Signaling Algorithm to B

This Appendix evaluates the sensitivity of system performance to the capacity of the input buffer, B. Throughput (Table 8) and Respect of Service (Table 9) are evaluated with B = 80, 100, 120, 130, and 150. Simulation results confirm that throughput and Respect of Service do not significantly depend on the value of B.

N_{AF}	thr. (kb/s), $B = 80$	thr. (kb/s), $B = 100$	thr. (kb/s), $B = 120$	thr. (kb/s), $B = 130$	thr. (kb/s), $B = 150$
10	1374.33	1381.32	1393.94	1399.89	1399.89
15	1374.74	1381.31	1393.91	1399.89	1399.89
20	1043.90	1040.13	1045.09	1045.75	1045.75
25	1027.25	1013.19	1026.40	1026.86	1026.86
30	1006.87	1004.72	1004.88	1004.88	1004.88

■ Table 8. *Throughput for different values of* B.

N_{AF}	RoS (%), $B = 80$	RoS (%), $B = 100$	RoS (%), $B =120$	RoS (%), $B = 130$	RoS (%), $B = 150$
10	98.22	98.70	99.62	100	100
15	98.25	98.70	99.62	100	100
20	99.56	99.71	99.72	99.97	99.97
25	99.26	99.31	99.18	99.23	99.23
30	98.30	98.42	98.13	98.13	98.13

■ Table 9. *Respect of service for different values of* B.

which shifts the computational burden toward the border of the network.

Acknowledgments

The authors wish to thank three anonymous reviewers who have provided invaluable comments and suggestions to improve the quality of this article.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview," RFC1633, June 1994.

[2] S. Shenker and J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements," RFC 2215, Sept. 1997.

[3] S. Blake *et al.*, "An Architecture for Differentiated Service," RFC 2475, Dec. 1998.

[4] K. Nichols *et al.*, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec. 1998.

[5] E. Clark, "Differentiated Services," Network Magazine.com, http://www.networkmagazine.com/article/NMG20010823S0016/2

[6] S. Blake *et al.*, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998.

[7] K. Nichols *et al.*, "Definition of the Differentiated Services Field (DiffServ Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec. 1998.

[9] S. Brim, B. Carpenter, and F. le Faucheur, "Per Hop Behavior Identification Codes," RFC 2836, May 2000.

[8] K. Nichols *et al.*, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec. 1998.

[10] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999.

[11] J. Heinanen *et al.*, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[12] D. Clark and W. Fang, "Explicit Allocation of Best-Effort Packet Delivery Service," *IEEE/ACM Trans. Net.*, vol. 6, no. 4, Aug. 1998, pp. 362–73.

[13] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Trans. Net.*, vol. 1, no. 4, Aug. 1993, pp. 397–413.

[14] J. Heinanen *et al.*, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[15] N. Seddigh, B. Nandy, and P. Pieda, "Bandwidth Assurance Issues for TCP Flows in a Differentiated Services Network," *Proc. GLOBECOM '99*, vol. 3, 1999, pp. 1792–98.

[16] J. Harju *et al.*, "Performance Measurements and Analysis of TCP Flows in a Differentiated Services WAN," *Proc. 25th Annual IEEE Conf. Local Comp. Net., 2000 (LCN 2000)*, pp. 296–305.

[17] G. Rogers, M. Minhazuddin, and R. Liu, "Mixing UDP and TCP in a Diffserv Assured Forwarding PHB: A Programmable Networks Scenario," *Proc. 9th IEEE Int'l. Conf. Net.*, 2001, pp. 529–34.

[18] M. Goyal *et al.*, "Effect of Number of Drop Precedences in Assured Forwarding," *Proc. GLOBECOM '99*, vol. 1a , pp. 188–93.

[19] R. Makkar *et al.*, "Empirical Study of Buffer Management Scheme for Diffserv Assured Forwarding PHB," *Proc. 9th Int'l. Conf. Comp. Commun. and Net.*, 2000, pp. 632–37.

[20] A. Sang, H. Zhu, and S.Q. Li, "Weighted Fairness Guarantee for Scalable DiffServ Assured Forwarding," *Proc. IEEE Int'l. Conf. Commun., ICC 2001*, vol. 8, 2001, pp. 2365–69.

[21] X. He and H. Che, "Achieving End-to-End Throughput Guarantee for TCP Flows in a Differentiated Services Network," *Proc. 9th Int'l. Conf. Comp. Commun. and Net.*, 2000, pp. 69–74.

[22] I. Matta and L. Guo, "Differentiated Predictive Fair Service for TCP Flows," *Proc. Int'l. Conf. Network Protocols*, 2000, pp. 49–58.

[23] W. Jie, F. Zhenming, and Y. Jian, "Differentiated Services TCP Algorithm for the Internet," *Electronics Letters*, vol. 35, no. 18, 2 Sept. 1999, pp. 1513–15.

[24] Q. Wang *et al.*, "Fast TCP Flow Control with Differentiated Services," *Proc. 5th Asia-Pacific Conf. Commun.*, 1999, *APCC/OECC '99*, vol. 1, pp. 209–12.

[25] P. Gevros, F. Risso, and P. Kirstein, "Analysis of a Method for Differential TCP Service," *Proc. GLOBECOM '99*, vol. 3, pp. 1699–708.

[26] W. Qian *et al.*, "Differentiated Service fast-TCP Policy for Flow Control and Resource Management," *Proc. WCC — ICCT 2000, Int'l. Conf. Commun. Tech.*, vol. 2, pp. 1645–52.

[27] Y. Zheng, Z. Feng, and J. Wu, "A Modified TCP Algorithm for Service Differentiation and Minimum Bandwidth Guarantee in the Internet," *Proc. Int'l. Conf. Comp. Net. and Mobile Computing*, 2001, pp. 99–104.

[28] A. Feroz, S. Kalyanaraman, and A. Rao, "A TCP-friendly Traffic Marker for IP Differentiated Services," *Proc. 8th Int'l. Wksp. Quality of Service*, IWQOS 2000, pp. 138–47.

[29] J. Hyeong Lee and C. K. Jeong, "Improvement of Fairness Between Assured Service TCP Users in a Differentiated Service Network," *Proc. Joint 4th IEEE Int'l. Conf. ATM (ICATM 2001) and High Speed Intelligent Internet Symp.*, pp. 47–55.

[30] H. Wu *et al.*, "TCP Friendly Fairness in Differentiated Services IP Networks," *Proc. 9th IEEE Int'l. Conf. Networks*, 2001, pp. 81–86.

[31] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, Apr. 1999.

[34] J. Postel, "Internet Control Message Protocol," RFC 0792 Sept.1981.

[35] ITU-T Rec. G.702, "Digital Hierarchy Bit Rates," Blue Book, Geneva, Switzerland, 1988.

[32] ITU-T, "Signalling Network Functions and Messages," recommendation Q.704, July 1996.

[33] ITU-T, "Signalling System no. 7 — ISDN User Part Signalling Procedures," recommendation Q.764, Dec. 1999.

Biographies

PAOLO GIACOMAZZI (giacomaz@elet.polimi.it) received his degree in electrical engineering from the Politecnico di Milano, and the Master in information technology from Cefriel in 1990. From 1992 to 1998 he was an assistant professor at the Politecnico di Milano, where he is currently an associate professor. His research interests include IP Differentiated and Integrated services, UMTS networks, and fourth-generation access networks.

LUIGI MUSUMECI (musumeci@elet.polimi.it) joined the Electronic and Information Department of the Politecnico di Milano in 1991, where he is now an associate professor. He received his degree in electrical engineering from the Politecnico di Milano in 1961. From 1968 to 1986 he was at Italtel, where he was responsible for the design and implementation of Itapac, the Italian packet network. His research interests cover packet data networks, the Internet, and wireless access networks.

GIACOMO VERTICALE (ertical@elet.polimi.it) is a post-doc researcher at Politecnico di Milano. He graduated with a degree in telecommunications engineering in 1998. In 1999 he joined the CEFRIEL research center, where he worked on Voice-over-IP and ADSL technologies. He received his Ph.D. in 2003 from Politecnico di Milano, defending a thesis on the performance of packet transmission in UMTS. His current interests focus on Quality of Service and on ad-hoc networking. He is member of IEEE and ACM.