EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Euro. Trans. Telecomms. 2006; **17**:505–524
Published online 6 March 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/ett.1093

Communication Networks

Fluid flow analysis of TCP flows in a DiffServ environment

Mario Barbera*, Alfio Lombardo, Giovanni Schembra and C. Andrea Trecarichi

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, University of Catania, V.le A. Doria, 6-95125 Catania, Italy

SUMMARY

The differentiated services (DiffServ) architecture allows IP networks to offer different QoS levels to different users and applications. In this architecture, routers in the core network offer the same per-hop behaviour (PHB) to all packets classified as belonging to the same class at the edge of the network. One of the most important types of PHB is assured forwarding (AF) PHB. Within each AF class, IP packets can be marked with different drop precedence (DP) values, and treated differently in congested DS nodes. To this end, DiffServ nodes in the core network implement active queue management (AQM) mechanisms. The target of this paper is to provide network designers with an accurate fluid-flow analytical model of a DiffServ network, where the RED with in/out and coupled average queues (RIO-C), RED with in/out and decoupled average queues (RIO-DC) and Weighted RED (WRED) AQM techniques are implemented. We address a network simultaneously loaded with both greedy and data-limited TCP flows, and we consider one AF class in which two DPs are defined, one for packets complying with the negotiated profile (IN packets), and another for packets that do not respect it (OUT packets). A token bucket marking mechanism is modelled for this purpose. The proposed model is applied to a complex network topology. Comparison between model and simulation results demonstrates that the model is able to capture both transient and steady-state network behaviour with a high degree of accuracy, even when not all network routers implement the same AQM technique. These characteristics make our modelling approach suitable to address the issue of network parameter optimisation. As an example, the link capacity dimensioning problem in a DiffServ domain by means of an iterative optimisation algorithm is presented. Copyright © 2006 AEIT.

1. INTRODUCTION

The development of the Internet, in both infrastructures and applications, has determined the need to provide users with quality of service (QoS) guarantees.

The first solution proposed in the literature was the integrated services (IntServ) architecture, which offers QoS guarantees for each flow, but is not scalable and requires complex changes in the Internet architecture. These reasons led the IETF to consider simpler alternatives to service differentiation. A promising approach is the differentiated services (DiffServ) architecture, proposed in Reference [1]. This architecture enables IP networks to offer different QoS levels to different users and applications, locating network intelligence mainly at the edge routers, thus relieving core routers from complex tasks. When packets arrive at the edge of a DiffServ domain, a profile meter measures the traffic streams against the negotiated profiles, assigns a drop precedence (DP) to packets according to the measurement results, and stores the DP in the DiffServ code point (DSCP) of the packets [2]. Then packets are forwarded to core routers, whose task is just to offer the same per-hop behaviour (PHB) to all the packets marked with the same DSCP value. Core routers do not need to maintain per-flow state, since they discriminate between packets exclusively on the basis of the DSCP. In Reference [3] assured forwarding (AF) PHB was proposed to provide individual or aggregate flows with guarantees in

* Correspondence to: Mario Barbera, Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, University of Catania, V.le A. Doria, I-95125 Catania, Italy. E-mail: mbarbera@diit.unict.it.

Received 6 October 2004
Revised 18 January 2005
Revised 5 July 2005

terms of throughput and burstiness, according to negotiated profiles.

When congestion occurs, the DP determines the relative importance of a packet within the AF class: a congested DiffServ node tries to protect packets with a lower drop precedence value from being lost by preferably discarding packets with a higher drop precedence value. As suggested in Reference [3], this can be achieved by employing active queue management (AQM) mechanisms in the core routers and configuring the dropping algorithm control parameters independently for each DP. For this reason the random early detection (RED) [4] AQM technique was extended to the case of two or more DPs, resulting in RED with in/out (RIO) [5], with its two variants RIO with coupled average queues (RIO-C) and RIO with decoupled average queues (RIO-DC) [6], and weighted RED (WRED) [7].

At the same time, great effort has recently been made to calculate the end-to-end performance of TCP flows in a DiffServ network by means of analytical approaches. In Reference [8–10] for example, expressions for the steady-state throughput of TCP sources in a DiffServ environment are derived, only taking into account long-lived TCP flows loading a bottleneck link. In Reference [11] the model proposed in Reference [12] for a non-DiffServ-compliant network is extended to a network supporting AF PHB with two DPs by means of a generic AQM technique. This approach uses a fluid-flow approximation to model traffic and queue behaviour in order to keep the model complexity low for any buffer queue dimension or network topology. However, in Reference [11] only a steady-state analysis is performed, with the aim of studying the stability conditions of the network; consequently the marking process is only modelled with two parameters that denote the fraction of fluid belonging to the two DPs. Further, Reference [11] does not consider data-limited TCP flows, but again only greedy sources. The use of greedy sources is obviously an approximation that is very often far from reality; a significant amount of Internet connections, in fact, concern the Web environment, where small files are transferred [13]. For this reason both slow-start and congestion avoidance mechanisms were modelled in Reference [14] in order to capture the behaviour of more realistic scenarios. In Reference [14] several cases are presented in which the comparison with simulation shows the loss of accuracy when the slow-start mechanism is not considered. A different approach is proposed in Reference [15] where an analytical framework is defined to study TCP performance based on Markov-modulated fluid models (MMFM). However, this framework presents scalability problems and is not suitable for DiffServ networks due to difficulties in extending it to cases in which routers implement AQM techniques.

The target of this paper is to provide an accurate fluid-flow analytical model of a DiffServ network simultaneously loaded by greedy and data-limited TCP flows. We consider the case of one AF class in which two DPs are defined, one for packets complying with the negotiated profile (IN packets), and another for packets that do not respect it (OUT packets). Let us note that in this scenario best-effort traffic can easily be taken into account by treating the related packets as OUT packets, that is, non-compliant packets. At the edge of the network, flows are grouped into traffic aggregates, each independently policed by a token bucket. In the core routers RIO-C, RIO-DC and WRED algorithms are modelled. Unlike most previous proposals, the model can be applied to any complex network topology. In addition, it is able to capture cases in which not all network routers implement the same AQM technique, some implementing RIO-C, others RIO-DC and others again WRED.

Comparison between model results and those obtained using the ns-2 simulator [16] demonstrates that the proposed model is able to capture not only steady-state network behaviour but also transient phenomena with a high degree of accuracy. In addition, the scalability of the proposed analytical approach with respect to both the number of traffic flows considered and the network topology complexity makes it very suitable to address the issue of network parameter optimisation. Our analytical approach, in fact, provides results in much less time than simulation, above all when the number of flows and/or link capacities considered increase. Furthermore, while the use of simulation results is very difficult because their inherent randomness, the model results can easily be analysed in order to find the solution to the optimisation problem.

The rest of the paper is organised as follows. In Section 2, we present the analytical model. In Section 3, we assess our model by applying it to a network scenario and comparing our results with those obtained using the ns-2 simulator. Section 4 shows how the proposed model can be used to approach a DiffServ design problem, that is, optimisation of the link capacities in a DiffServ network. Finally, we present our conclusions in Section 5.

2. MODEL AND ANALYSIS

The target of this section is to derive a fluid-flow model of a DiffServ domain in which an AF-PHB is defined with

two DPs for TCP traffic. We will perform an analysis of the average behaviour of the network and sources.

We will consider both greedy sources and sources that have to transmit a finite-size file. In the rest of the paper we will indicate the second type of sources as *data-limited* sources.

We assume that:

1. the TCP layer for each source receives data from applications all at once, so we consider data-limited sources as always having data to transmit until the end of the file;
2. the file size that each data-limited source has to transmit is known *a priori*;
3. the instant at which each source starts to transmit is known.

Let us note that according to these hypotheses, if there were only drop-tail routers in the network, the system would be completely deterministic. When, on the other hand, DiffServ routers adopt AQM techniques to differentiate services, stochastic behaviour is induced in the system by the drop probability in the network routers.

In order to capture this behaviour, the model used will be based on the queuing system fluid-flow approach. According to this approach, the average behaviour of the whole system (network and sources) is derived assuming that each AQM buffer in the network has a deterministic behaviour equal to its average behaviour. In other words, we will consider that, if $\lambda(t)$ is the packet arrival rate at a generic AQM buffer, and $p(t)$ is its drop probability at the same time instant, $\lambda(t)(1-p(t))$ will be the 'actual' rate of packets queued in the buffer, while $\lambda(t)p(t)$ will be the rate of lost packets.

The fluid model we derive is a set of differential equations that describes the average temporal evolution of all the processes characterising the whole system.

The system considered in this paper comprises a set of DiffServ-compliant nodes making up a DiffServ domain, loaded by N TCP flows labelled as $i = 1, \ldots, N$. Let L be the set of router output buffers which store packets before transmitting them on the associated unidirectional output links. A generic link l has a transmission capacity of C_l bytes per second, and a constant propagation delay of d_l seconds. Further, we indicate the queue length of the generic buffer $l \in L$ at the time $t \geq 0$ as $q_l(t)$ (expressed in bytes).

In order to make TCP modelling independent of the AQM techniques used in the routers, we separate the TCP behaviour model from the Diffserv network model. More specifically, we derive the TCP source and receiver models assuming knowledge of some quantities coming from the network model and vice versa. The quantities exchanged between the TCP model and the network model are listed in Table 1. Actually, the TCP receiver model is very simple and it is only introduced for the sake of clearness, while the TCP source model and the network model are represented by two sets of differential equations that constitute the model of the whole system.

In the following sections, we first introduce the TCP source and TCP receiver behaviour models (Subsections 2.1 and 2.2), and then we describe the DiffServ router network model (Subsection 2.3). Finally, Subsection 2.4 discusses issues relating to the numerical solution of the model.

2.1. TCP behaviour modelling

Let $W_i(t)$ denote the congestion window of the flow i at the time $t \geq 0$. In our framework we will not consider

Table 1. Quantities exchanged between TCP model and network model.

	To TCP source model	To TCP receiver model	To network model
From TCP source model			—Packet emission rate for each source (or group of sources)
From TCP receiver model			—ACK emission rate for each receiver (or group of receivers) Rate of loss indications from each receiver (or group of receivers)
From network model	—Queue length for each buffer —arrival rate of ACKs —Arrival rate of loss indications	—Arrival rate of packets Packet loss rate	

the end-to-end flow control algorithm, assuming TCP throughput to be bounded by the congestion control algorithm only. For this reason $W_i(t)$ also represents the transmission window of the ith TCP flow.

Furthermore, let $T_i(t)$ be the value of the threshold separating the slow-start range and the congestion avoidance range for the congestion window of the ith TCP flow at the instant t.

Since we are interested in analysing not only the behaviour of greedy sources, but also that of sources which have to transmit finite amounts of data, we also need to consider the process $D_i(t)$ representing the number of bytes successfully sent by the ith TCP flow from its start to the time t.

In order to characterise our model completely, we need to derive the mean values of the processes $W_i(t)$, $T_i(t)$ and $D_i(t)$.

To this end, let us first calculate the expression of $R_i(t)$ representing the round-trip time (RTT) for the generic ith TCP flow. The RTT for a generic flow i is the sum of the queuing times in all the buffers along its path and the propagation times associated with the output links of these buffers. Therefore, if we indicate the set of buffers passed through by the packets belonging to the flow i and by their corresponding ACKs as L_i, we obtain:

$$R_i(t) = \sum_{l \in L_i} \left(d_l + \frac{q_l(t)}{C_l} \right) \quad i = 1, 2, \ldots, N \tag{1}$$

We will now derive the relationship that describes the additive-increase multiplicative-decrease (AIMD) behaviour of the TCP window size. We can write the variation of the window size $W_i(t)$ as the sum of two contributions: the first term, $A_i(t)$, corresponds to the additive-increase part, the second, $B_i(t)$, corresponds to the multiplicative-decrease part:

$$\frac{\mathrm{d}W_i(t)}{\mathrm{d}t} = A_i(t) + B_i(t) \qquad i = 1, 2, \ldots, N \tag{2}$$

First we will calculate the additive-increase term $A_i(t)$. To this end let us note that, for each ACK packet reaching a generic TCP source, the congestion window size $W_i(t)$ increases by one packet during the slow-start phase, while it increases by $1/W_i(t)$ packets during the congestion avoidance phase. So, if we indicate the arrival rate of ACKs for the flow i as $\lambda_i^{(A)}(t)$ (expressed in bytes/s), we can write:

$$A_i(t) = \begin{cases} \frac{S_i^{(P)}}{S^{(A)}} \cdot \lambda_i^{(A)}(t) & \text{if} \quad W_i(t) < T_i(t) \\ \frac{S_i^{(P)}}{S^{(A)}} \cdot \frac{\lambda_i^{(A)}(t)}{W_i(t)} & \text{if} \quad W_i(t) \geq T_i(t) \end{cases} \quad i = 1, 2, \ldots, N \tag{3}$$

where $S_i^{(P)}$ is the size (in bytes) of the generic data packet belonging to the flow i, while $S^{(A)}$ is the ACK size (in bytes).

Derivation of the term $B_i(t)$ in Equation (2) depends on the TCP version of the sources, because of the different algorithms adopted to calculate the new congestion window value when a loss is detected. In the rest of the paper we will refer to one of the most common TCP versions, that is, the New-Reno TCP [17], but our derivation can easily be extended to the other TCP versions. Because a New-Reno TCP source behaves differently according to whether it detects a packet loss by receiving a triple duplicate ACK (TD loss) or because a timeout (TO loss) expires, we need to distinguish between the two different loss causes.

To this end we need to consider the rate at which a generic TCP source is notified of packet losses occurring in the network. We assume that information about losses travels through the network with the packets sent out by the generic TCP source along the same path. So in Subsection 2.2 we will consider the network as also being passed through by N ghost flows, each carrying loss indications relating to a TCP source; let us denote the rate of loss indications for the ith TCP source at the time instant t as $\zeta_i(t)$ (expressed in packets/s).

To distinguish between the losses suffered by the flow i being detected as TD losses or TO losses we also need to consider the number of ACKs $N_i^{(A)}(t-\tau, t)$ received by the generic TCP source during a time interval τ that ends at the time instant t. This number can be calculated as:

$$N_i^{(A)}(t-\tau, t) = \frac{1}{S^{(A)}} \int_{t-\tau}^{t} \lambda_i^{(A)}(\nu)\mathrm{d}\nu \quad i = 1, 2, \ldots, N \ (t > \tau) \tag{4}$$

Now, let us assume that a loss has happened ($\zeta_i(t) > 0$). At the generic instant t, if the number of ACKs received in the last time interval equal to the retransmission timeout (RTO) is less than 3, that is $N_i^{(A)}(t - RTO, t) < 3$, losses are detected at the instant t as TO losses, and therefore the loss rate at the instant $t - \mathrm{RTO}$, $\zeta_i(t - \mathrm{RTO})$, is a TO loss rate, henceforward indicated as $\gamma_i^{(\mathrm{TO})}(t)$. If, on the contrary, there exists a time interval $[t-\tau, t]$ with a duration τ

less than RTO, where the number of ACKs received is equal to 3, that is $N_i^{(A)}(t-\tau,t)=3$, then losses are detected as TD losses, and therefore the loss rate at the instant $t-\tau$, $\zeta_i(t-\tau)$, is a TD loss rate, henceforward indicated as $\gamma_i^{(\mathrm{TD})}(t)$.

So, assuming that the retransmission timeout can be approximated by $4\cdot R_i(t)$ as in Reference [18], if we indicate the rate of TO losses and TD losses detected by the source i as $\gamma_i^{(\mathrm{TO})}(t)$ and $\gamma_i^{(\mathrm{TD})}(t)$ respectively, we have:

$$\gamma_i^{(\mathrm{TO})}(t)=\begin{cases}\zeta_i(t-4\cdot R_i(t)) & \text{if} \quad N_i^{(\mathrm{A})}(t-4\cdot R_i(t),t)<3\\ 0 & \text{elsewhere}\end{cases}$$
$$i=1,2,\ldots,N \tag{5}$$

$$\gamma_i^{(\mathrm{TD})}(t)=\begin{cases}\zeta_i(t-\tau) \text{ with } \tau: N_i^{(A)}(t-\tau,t)=3 & \text{if } \tau<4\cdot R_i(t)\\ 0 & \text{elsewhere}\end{cases}$$
$$i=1,2,\ldots,N \tag{6}$$

A generic New-Reno TCP source halves its congestion window when a TD loss occurs, while it sets its congestion window to one packet size when a timeout expires. Consequently, the variation of the congestion window $W_i(t)$ is equal to $-(W_i(t)/2)$ when a TD loss is detected, while it is equal to $(S_i^{(\mathrm{P})}-W_i(t))$ when a TO loss is detected. From these considerations we obtain the final expression of $B_i(t)$:

$$B_i(t)=-\frac{W_i(t)}{2}\gamma_i^{(\mathrm{TD})}(t)+\left(S_i^{(\mathrm{P})}-W_i(t)\right)\cdot\gamma_i^{(\mathrm{TO})}(t)$$
$$i=1,2,\ldots,N \tag{7}$$

Now we will derive the equation that regulates the behaviour of the threshold $T_i(t)$, separating the slow-start window range and the congestion avoidance window range. This threshold is set to half the congestion window every time a loss is detected; so its variation is equal to zero when there is no loss indication, and to $1/2W_i(t)-T_i(t)$ otherwise. Considering Equations (5) and (6) we have:

$$\frac{\mathrm{d}T_i(t)}{\mathrm{d}t}=\left(\frac{W_i(t)}{2}-T_i(t)\right)\cdot\left(\gamma_i^{(\mathrm{TD})}(t)+\gamma_i^{(\mathrm{TO})}(t)\right)$$
$$i=1,2,\ldots,N \tag{8}$$

Finally, we derive the relationship to calculate the number of bytes $D_i(t)$, which have been successfully sent by the source i from its start until the time t. Let us note that when $D_i(t)$ is equal to the size of the file to be sent by the source i, this source has concluded its transmission.

The variation in the number of bytes successfully sent by the generic source i is given by the arrival rate of ACKs at the source $\lambda_i^{(A)}(t)$, that is:

$$\frac{\mathrm{d}D_i(t)}{\mathrm{d}t}=\frac{S_i^{(\mathrm{P})}}{S^{(\mathrm{A})}}\lambda_i^{(A)}(t)\quad i=1,2,\ldots,N \tag{9}$$

Up to now we have derived a set of $3\cdot N$ differential equations that describe the average behaviour $(W_i(t),T_i(t),D_i(t))$ of N TCP sources (in particular using the New-Reno version) in a generic network. It is important to point out that we can reduce the number of equations, which describe the sources by grouping the flows having the same average behaviour. More specifically, a group is made up of all flows following the same path in the network, having the same packet size and starting at time instants belonging to an interval that is 3–4 times shorter than their average RTT.

In this way we can divide the N flows into K groups $(K\leqslant N)$. A generic group k contains n_k flows, where $k=1,2,\ldots,K$, with the condition $n_1+n_2+\cdots+n_K=N$. Since we are also considering data-limited sources, the N sources may not all be active at the same time, because some of them may have concluded their transmission, or not yet started to transmit. Let $a_k(t)$ be the number of active sources in the group k at the instant t $(a_k(t)\leq n_k)$.

If we indicate the arrival rate of ACKs and the rate of lost packets for the kth group of flows as $\psi_k^{(\mathrm{A})}(t)$ and $\Omega_k(t)$ respectively, we can write:

$$\lambda_i^{(\mathrm{A})}(t)=\frac{\psi_k^{(\mathrm{A})}(t)}{a_k(t)}\quad i=1,2,\ldots,N \tag{10}$$

$$\zeta_i(t)=\frac{\Omega_k(t)}{a_k(t)}\quad i=1,2,\ldots,N \tag{11}$$

$\psi_k^{(\mathrm{A})}(t)$ and $\Omega_k(t)$ will be derived in Subsection 2.2.

For each group of flows we will consider a function $f_k(s)$ that represents the number of sources in the group k that have to transmit a file of a size less than or equal to s, expressed in bytes. Consequently, if we indicate the number of bytes successfully sent by a generic source belonging to the group k as $D_k(t)$, the number of sources belonging to group k which are active at time t is:

$$a_k(t)=n_k-f_k(D_k(t))\quad k=1,2,\ldots,K \tag{12}$$

Equations (2),(8) and (9) are the set of characteristic equations of TCP sources. Given the buffer queue length

$q_l(t)$ for $l = 1,2,\ldots,L$, the arrival rate of ACKs $\psi_k^{(A)}(t)$ and the arrival rate of loss indications $\Omega_k(t)$ for $k = 1,2,\ldots,K$ (see Table 1), it is possible to derive the total emission rate $\mathrm{Th}_k(t)$ at the generic time instant t for each group:

$$\mathrm{Th}_k(t) = a_k(t) \cdot \frac{W_k(t)}{R_k(t)} \quad \text{for } k = 1, 2, \ldots, K \qquad (13)$$

where $W_k(t)$ and $R_k(t)$ respectively represent the window size and the RTT of a generic TCP source belonging to group k.

On the other hand, the processes $\mathrm{Th}_k(t)$, for $k = 1,2,\ldots,K$, are input variables for the network model (see Table 1).

2.2. TCP receiver model

In our framework we assume that the generic TCP receiver sends one ACK packet for each received packet. As we will show, the TCP receiver model is made up of two simple relationships. Furthermore, we do not need to consider them for each receiver, but for each group of receivers, that is the set of receivers of packets sent by sources belonging to the same group.

Let us indicate the total arrival rate (expressed in bytes/s) at the generic group k of receivers as $\lambda_k^{(P)}(t)$. The emission rate of ACKs $\mu_k^{(A)}(t)$ (expressed in bytes/s) can be calculated simply considering the data packet size $S_k^{(P)}$ and the ACK packet size $S^{(A)}$ as scaling factors:

$$\mu_k^{(A)}(t) = \frac{S^{(A)}}{S_k^{(P)}} \cdot \lambda_k^{(P)}(t) \quad \text{for } k = 1, 2, \ldots, K \qquad (14)$$

In a similar way, if we indicate the input packet loss rate for the group of receivers as $\nu_k^{(\mathrm{LOSS})}(t)$, we can derive the output rate of loss indications $r_k^{(\mathrm{LOSS})}(t)$ as follows:

$$r_k^{(\mathrm{LOSS})}(t) = \frac{S^{(A)}}{S_k^{(P)}} \cdot \nu_k^{(\mathrm{LOSS})}(t) \quad \text{for } k = 1, 2, \ldots, K \quad (15)$$

The emission rate of ACKs, $\mu_k^{(A)}(t)$, and the output rate of loss indications, $r_k^{(\mathrm{LOSS})}(t)$, are the input variables for the network model (see Table 1).

2.3. DiffServ network modelling

As we have already said, we are considering a DiffServ architecture in which the AF PHB is defined with two DP values. In this paper we assume the service profile of a given traffic aggregate to be completely defined by both its committed average information rate (CIR) and its committed maximum burst size (CBS).[1] In a general case, a traffic aggregate can comprise both forward TCP traffic and reverse ACK traffic. Although our modelling approach is potentially able to capture this occurrence, for the sake of simplicity we will consider $2 \cdot M$ traffic aggregates in our framework: the traffic aggregates 1, 2, ..., M collect forward TCP traffic, while the traffic aggregates $M+1$, $M+2,\ldots,2 \cdot M$ collect the corresponding reverse ACK traffic.

We also assume that reverse ACK traffic has the same traffic profile as the relative forward TCP traffic, that is the service profile assigned to the mth traffic aggregate is the same as that assigned to the $(m+M)$th one.

Packets belonging to the generic traffic aggregate are marked by a token bucket in the network access routers, then they pass through a set of router buffers in which multi-level RED (MRED) AQM algorithms are implemented. In Subsections 2.2.1, 2.2.2 and 2.2.3 the fluid-flow models of token buckets, buffers and MRED algorithms are presented.

2.3.1. Token-bucket model. At the edge routers a token bucket for each aggregate marks packets as IN if they are 'in-profile', or OUT if they are 'out-of-profile'. As is well known, a token bucket can be seen as a virtual buffer of size CBS loaded at a rate of CIR tokens per second. A token represents the right to transmit a fixed amount of bytes, $S^{(T)}$. CIR will be expressed in bytes per second and CBS in bytes. When a packet arrives at an edge router, if the corresponding token bucket is not empty, a token is removed and the incoming packet is marked as IN; otherwise it is marked as OUT. Because we assume the presence of $2 \cdot M$ traffic aggregates, we will have $2 \cdot M$ token buckets in the network.

Let $\Psi_m(t)$ be the arrival rate of the mth traffic aggregate at the input of the mth token bucket $(m = 1, 2, \ldots, M)$. Because $\Psi_m(t)$ is the sum of the emission rates of the sources (or receivers) belonging to the mth traffic aggregate, if we indicate the set of groups of flows belonging to the traffic aggregate m as G_m, we obtain:

$$\Psi_m(t) = \begin{cases} \sum_{k \in G_m} \mathrm{Th}_k(t) & m = 1, 2, \ldots M \\ \sum_{k \in G_m} \mu_k^{(A)}(t) & m = M+1, M+2, \ldots, 2 \cdot M \end{cases} \qquad (16)$$

Let $V_m(t)$ be the length of the virtual buffer of the generic token bucket m, and let CIR_m and CBS_m respectively be

[1] In this scenario CIR = CBS = 0 defines the service profile of a best-effort traffic aggregate, if any.

the associated CIR and CBS. Furthermore, we indicate the output rate of IN and OUT packets at the output of the token bucket m as $\varphi_m^{(\mathrm{IN})}(t)$ and $\varphi_m^{(\mathrm{OUT})}(t)$ respectively. When the virtual token buffer is not empty all the packets are marked as IN and the rate of IN packets will therefore be equal to the rate of incoming packets; when, on the other hand, the virtual buffer is empty, at most $\mathrm{CIR}_m/S^{(\mathrm{T})}$ packets/s are marked as IN, while the remaining packets are marked as OUT.

In other words, the rate of IN packets at the output of the token bucket is given by:

$$\varphi_m^{(\mathrm{IN})}(t) = \begin{cases} \Psi_m(t) & \text{if } V_m(t) > 0 \\ \min(\mathrm{CIR}_m, \Psi_m(t)) & \text{if } V_m(t) = 0 \end{cases} \tag{17}$$

The rate of OUT packets will be the difference between the rate of incoming packets and the rate of IN packets:

$$\varphi_m^{(\mathrm{OUT})}(t) = \Psi_m(t) - \varphi_m^{(\mathrm{IN})}(t) \tag{18}$$

Therefore, $V_m(t)$ can be calculated as follows:

$$\frac{\mathrm{d}V_m(t)}{\mathrm{d}t} = \mathrm{CIR}_m - \varphi_m^{(\mathrm{IN})}(t) \quad \text{with the constraint} \quad V_m(t) \leqslant \mathrm{CBS}_m \tag{19}$$

2.3.2. Buffer model. The token bucket output rates of IN and OUT packets, $\varphi_m^{(\mathrm{IN})}(t)$ and $\varphi_m^{(\mathrm{OUT})}(t)$ respectively, represent the input traffic for the first buffer in the set of buffers passed through by packets belonging to the mth traffic aggregate. Because the temporal variation of the metrics related to IN and OUT packets can be described by the same equations, henceforward, if not explicitly expressed, we will substitute the IN (or the OUT) apex with X. So when we use the (X) apex, the reader can replace it with either (IN) or (OUT).

Before continuing with the description of the buffer model we need to introduce some further notation:

- $\Lambda_l^{(X)}(t)$: total arrival rate (in bytes per second) of X packets at a generic buffer l at time $t \geq 0$;
- $\psi_{k,l}^{(X)}(t)$: average arrival rate (in bytes per second) at the buffer l of X packets, belonging to the group k at time $t \geq 0$;
- $\mu_{k,l}^{(X)}(t)$: average output rate (in bytes per second) from the buffer l of X packets, belonging to the group k at time $t \geq 0$;
- $p_l^{(X)}(t)$: drop probability function applied to X packets, at the generic buffer $l \in L$ at time $t \geq 0$;
- $p_l(t)$: drop probability function applied to a generic packet at the generic buffer $l \in L$ at time $t \geq 0$;
- $\gamma_{k,l}(t)$: loss rate (in bytes per second) for sources belonging to the group k at the output of the buffer l.

To calculate the average arrival rate $\psi_{k,l}^{(X)}(t)$ at the buffer l of X packets belonging to the group k, we have to consider the ordered set of buffers $H_k = \{h_1^{(k)}, h_2^{(k)}, \ldots, h_{\mathrm{R}}^{(k)}, h_{\mathrm{R}+1}^{(k)}, \ldots, h_{\mathrm{S}}^{(k)}\}$ in the path followed by data packets and then by ACKs belonging to group k. With this notation $h_{\mathrm{R}}^{(k)}$ and $h_{\mathrm{S}}^{(k)}$ represent the buffers that send their packets to the routers to which the receivers and sources of group k are directly connected, while $h_{\mathrm{R}+1}^{(k)}$ is the buffer to which the receivers of group k send their ACK packets. Therefore we have:

$$\psi_{k,h_j^{(k)}}^{(X)}(t) = \begin{cases} \mathrm{Th}_k(t)\dfrac{\varphi_m^{(X)}(t)}{\Psi_m(t)} & \text{if } j = 1 (\forall k \in G_m, 1 \leqslant m \leqslant M) \\ \mu_k^{(\mathrm{A})}(t)\dfrac{\varphi_m^{(X)}(t)}{\Psi_m(t)} & \text{if } j = R+1 (\forall k \in G_m, M+1 \leqslant m \leqslant 2 \cdot M) \\ \mu_{k,h_{j-1}^{(k)}}^{(X)}\left(t - d_{h_{j-1}^{(k)}}\right) & \text{if } j \neq 1, R+1 \end{cases} \quad k = 1, 2, \ldots, K \tag{20}$$

where $d_{h_{j-1}^{(k)}}$ is the propagation time along the output link of the buffer $h_{j-1}^{(k)}$.

Consequently, if we indicate the set of groups of flows passing through the buffer l as G_l, we obtain:

$$\Lambda_l^{(X)}(t) = \sum_{k \in G_l} \psi_{k,l}^{(X)}(t) \quad \forall l \in L \tag{21}$$

The equation that regulates variations in the queue length $q_l(t)$ of the generic buffer $l \in L$, derives from the Lindley equation and reads:

$$\frac{\mathrm{d}q_l(t)}{\mathrm{d}t} = \begin{cases} -C_l + (1 - p_l(t)) \cdot \left(\Lambda_l^{(\mathrm{IN})}(t) + \Lambda_l^{(\mathrm{OUT})}(t)\right) & \text{if } q_l(t) > 0 \\ \left[-C_l + (1 - p_l(t)) \cdot \left(\Lambda_l^{(\mathrm{IN})}(t) + \Lambda_l^{(\mathrm{OUT})}(t)\right)\right]^+ & \text{if } q_l(t) = 0 \end{cases} \quad \forall l \in L \tag{22}$$

where $[f(t)]^+$ is equal to $f(t)$ when it is positive, and equal to zero otherwise.

The next relationship that we will derive is the one that exists in the buffer l to link the arrival rate and the emission rate of X packets belonging to the generic group k.

The total number of X packets belonging to group k that will be served by the buffer l up to the time instant $t + q_l(t)/C_l$ is equal to the total number of X packets arriving in the buffer minus the total number of X packets lost in the same buffer up to the instant t, that is:

$$\int_0^{t+\frac{q_l(t)}{C_l}} \mu_{k,l}^{(X)}(\nu)\mathrm{d}\nu = \int_0^t \psi_{k,l}^{(X)}(\nu)\mathrm{d}\nu - \int_0^t \psi_{k,l}^{(X)}(\nu)p_l^{(X)}(\nu)\mathrm{d}\nu \quad (23)$$

Applying the derivative to both sides of Equation (23) we obtain:

$$\mu_{k,l}^{(X)}\left(t+\frac{q_l(t)}{C_l}\right) = \frac{\psi_{k,l}^{(X)}(t)\cdot\left(1-p_l^{(X)}(t)\right)}{F_l(t)} \quad k=1,2,\ldots,K;\forall l\in L \quad (24)$$

where:

$$F_l(t) = 1 + \frac{1}{C_l}\frac{\mathrm{d}q_l(t)}{\mathrm{d}t} \quad \forall l \in L \quad (25)$$

The time argument $q_l(t)/C_l$ on the left-hand side of Equation (24) represents the delay introduced by the buffer. Equation (24) allows us to calculate the emission rate of X packets belonging to group k, by dividing the corresponding actual arrival rate $\psi_{k,l}^{(X)}(t)\cdot(1-p_l^{(X)}(t))$ by a factor $F_l(t)$.

Let us note that, when the queue length does not change, it follows from Equation (25) that $F_l(t)$ is equal to 1 and, as expected, Equation (24) states that the emission rate, after the delay introduced in the buffer, will be exactly equal to the actual arrival rate. Likewise, when the queue length increases ($F_l(t) > 1$ in Equation (25)), the emission rate after the delay $q_l(t)/C_l$ will be less than the actual arrival rate, as expected given that the total actual arrival rate (by all the groups) at the buffer l is in this case greater than its link capacity C_l. Finally, when the queue length decreases ($F_l(t) > 1$ in Equation (25)), from Equation (25) the emission rate is greater than the total actual arrival rate. This was also expected because in this case the total output rate is equal to C_l.

As further proof of the correctness of Equation (24), substituting Equation (22) in Equations (24) and (25) for the case in which $q_l(t) > 0$, it can easily be obtained that:

$$\sum_{k\in F_l}\mu_{k,l}^{(\mathrm{IN})}\left(t+\frac{q_l(t)}{C_l}\right) + \sum_{k\in F_l}\mu_{k,l}^{(\mathrm{OUT})}\left(t+\frac{q_l(t)}{C_l}\right) = C_l \quad \forall l\in L \quad (26)$$

The above equation states that the total output rate of the generic buffer l is always equal to its service capacity when the buffer is not empty.

Note that if no packets enter the buffer at the time instant t, $\psi_{k,l}^{(X)}(t) = 0$ and $\mathrm{d}q_l(t)/\mathrm{d}t = -C_l$. In this case, although Equation (24) is an indeterminate form 0/0, it can easily be argued that $\mu_{k,l}^{(X)}(t + q_l(t)/C_l) = 0$.

Through Equation (24) we can immediately derive the arrival rate of ACKs at the sources belonging to group k, $\psi_k^{(\mathrm{A})}(t)$, and the total packet arrival rate at the receivers belonging to the same group, $\lambda_k^{(\mathrm{P})}(t)$. In fact, we have:

$$\psi_k^{(A)}(t) = \mu_{k,h_S^{(k)}}^{(\mathrm{IN})}\left(t-d_{h_S^{(k)}}\right) + \mu_{k,h_S^{(k)}}^{(\mathrm{OUT})}\left(t-d_{h_S^{(k)}}\right) \quad k=1,2,\ldots,K \quad (27)$$

$$\lambda_k^{(\mathrm{P})}(t) = \mu_{k,h_R^{(k)}}^{(\mathrm{IN})}\left(t-d_{h_R^{(k)}}\right) + \mu_{k,h_R^{(k)}}^{(\mathrm{OUT})}\left(t-d_{h_R^{(k)}}\right) \quad k=1,2,\ldots,K \quad (28)$$

where $d_{h_S^{(k)}}$ ($d_{h_R^{(k)}}$) is the propagation delay relating to the output link of the buffer $h_S^{(k)}$ ($h_R^{(k)}$) from which the sources (receivers) belonging to group k receive their ACK (data) packets.

These two quantities are provided to both the TCP source model and the TCP receiver model (see Table 1).

Now we will derive the rate $\gamma_{k,l}(t)$ of packet losses suffered by the group of flows k at the output of the generic buffer l. We assume that the packet loss rate at the output of the buffer l at the time instant $t + q_l(t)/C_l$ is equal to the packet loss rate at the input of the same buffer at the time instant t, that is:

$$\gamma_{k,h_j^{(k)}}\left(t+\frac{q_{h_j^{(k)}}(t)}{C_{h_j^{(k)}}}\right) = \begin{cases} \psi_{k,h_j^{(k)}}^{(\mathrm{IN})}(t)\cdot Zp_l^{(\mathrm{IN})}(t) + \psi_{k,h_j^{(k)}}^{(\mathrm{OUT})}(t)\cdot p_l^{(\mathrm{OUT})}(t) & \text{if } j=1 \text{ and } j=R+1 \\ \gamma_{k,h_{j-1}^{(k)}}\left(t-d_{h_{j-1}^{(k)}}\right) + \psi_{k,h_j^{(k)}}^{(\mathrm{IN})}(t)\cdot p_l^{(\mathrm{IN})}(t) + \psi_{k,h_j^{(k)}}^{(\mathrm{OUT})}(t)\cdot p_l^{(\mathrm{OUT})}(t) & \text{if } j>1 \end{cases} \quad k=1,2,\ldots,K \quad (29)$$

Consequently, the rate of packet losses $\Omega_k(t)$ (in packets/s) for the kth group of flows introduced in Subsection 2.1 will be:

$$\Omega_k(t) = \frac{1}{S^{(\mathrm{A})}} \cdot \gamma_{k,h_S^{(k)}}\left(t - d_{h_S^{(k)}}\right) \quad k = 1, 2, \ldots K \tag{30}$$

while the input packet loss rate (in bytes/s) for the receiver group k, $\nu_k^{(\mathrm{LOSS})}(t)$ is:

$$\nu_k^{(\mathrm{LOSS})}(t) = \gamma_{k,h_R^{(k)}}\left(t - d_{h_R^{(k)}}\right) \quad k = 1, 2, \ldots K \tag{31}$$

measurement variables. First let us recall the relationship that describes the time variation of the average queue length $m(t)$ estimated by RED [12, 14]:

$$\frac{\mathrm{d}m_l(t)}{\mathrm{d}t} = \frac{1}{S^{(\mathrm{P})}} \begin{cases} \ln(1-\alpha_l) \cdot (m_l(t) - q_l(t)) \cdot \left(\Lambda_l^{(\mathrm{IN})}(t) + \Lambda_l^{(\mathrm{OUT})}(t)\right) & \text{if } q_l(t) > 0 \\ \ln(1-\alpha_l) \cdot C_l \cdot m_l(t) & \text{if } q_l(t) = 0 \end{cases} \quad \forall l \in L \tag{33}$$

where α_l represents the weight in the EWMA filter used to estimate the average queue length.

Let $m^{(\mathrm{IN})}(t)$ be the estimated average length of the *IN packet virtual queue*, that is the virtual queue collecting the ordered sequence of IN packets queued in the buffer, denoted as $q_l^{(\mathrm{IN})}(t)$.
The relationship for the calculus of $m_l^{(\mathrm{IN})}(t)$ can be directly derived from Equation (33):

$$\frac{\mathrm{d}m_l^{(\mathrm{IN})}(t)}{\mathrm{d}t} = \frac{1}{S^{(P)}} \begin{cases} \ln(1-\alpha_l) \cdot \left(m_l^{(\mathrm{IN})}(t) - q_l^{(\mathrm{IN})}(t)\right) \cdot \Lambda_l^{(\mathrm{IN})}(t) & \text{if } q_l^{(\mathrm{IN})}(t) > 0 \\ \ln(1-\alpha_l) \cdot C_l \cdot m_l^{(\mathrm{IN})}(t) & \text{if } q_l^{(\mathrm{IN})}(t) = 0 \end{cases} \quad \forall l \in L \tag{34}$$

The last relationship that we need to derive concerns $p_l(t)$, representing the drop probability in the generic buffer l at time $t \geq 0$. Applying the theorem of total probability, we can calculate $p_l(t)$ as follows:

$$p_l(t) = p_l^{(\mathrm{IN})}(t) \cdot \frac{\Lambda_l^{(\mathrm{IN})}(t)}{\Lambda_l^{(\mathrm{IN})}(t) + \Lambda_l^{(\mathrm{OUT})}(t)} + p_l^{(\mathrm{OUT})}(t) \cdot \frac{\Lambda_l^{(\mathrm{OUT})}(t)}{\Lambda_l^{(\mathrm{IN})}(t) + \Lambda_l^{(\mathrm{OUT})}(t)} \tag{32}$$

where the length of the IN packet virtual queue can be derived by the following equation:

$$\frac{\mathrm{d}q_l^{(\mathrm{IN})}(t)}{\mathrm{d}t} = \begin{cases} -\frac{q_l^{(\mathrm{IN})}(t)}{q_l(t)} C_l + \left(1 - p_l^{(\mathrm{IN})}(t)\right) \cdot \Lambda_l^{(\mathrm{IN})}(t) & \text{if } q_l(t) > 0 \\ \left(1 - p_l^{(\mathrm{IN})}(t)\right) \cdot \Lambda_l^{(\mathrm{IN})}(t) & \text{if } q_l(t) = 0 \end{cases} \quad \forall l \in L \tag{35}$$

2.3.3. MRED algorithms

The way to calculate the drop probability functions $p_l^{(\mathrm{IN})}(t)$ and $p_l^{(\mathrm{OUT})}(t)$ depends on the buffer management technique adopted in the routers. As an application, we will present the expressions of $p_l^{(\mathrm{IN})}(t)$ and $p_l^{(\mathrm{OUT})}(t)$ for three different AQM mechanisms that provide service differentiation, that is RIO-C, RIO-DC and WRED. These mechanisms are often globally denoted as MRED algorithms because all of them are based on the same AQM algorithm, that is RED [4], and apply multiple sets of RED parameters to packets having different priority levels, such as IN and OUT packets. However, different MRED algorithms calculate dropping probabilities using different

Moreover, let $m_l^{(\mathrm{OUT})}(t)$ be the average estimated length of the OUT packet virtual queue; it can easily be calculated as:

$$m_l^{(\mathrm{OUT})}(t) = m_l(t) - m_l^{(\mathrm{IN})}(t) \quad \forall l \in L \tag{36}$$

All MRED algorithms use two different RED discarding functions, $p_l^{(\mathrm{IN})}(t)$ for IN packets and $p_l^{(\mathrm{OUT})}(t)$ for OUT packets respectively. If we assume the buffer size to be correctly designed, that is the AQM discarding function works in such a way that any overflow is prevented, the drop

probability functions $p_l^{(\mathrm{IN})}(t)$ and $p_l^{(\mathrm{OUT})}(t)$ can be derived as:

$$p_l^{(\mathrm{IN})}(t) = \begin{cases} 0 & \text{if } M_l^{(\mathrm{IN})}(t) < t_{\min_l}^{(\mathrm{IN})} \\ \dfrac{M_l^{(\mathrm{IN})}(t) - t_{\min_l}^{(\mathrm{IN})}}{t_{\max_l}^{(\mathrm{IN})} - t_{\min_l}^{(\mathrm{IN})}} p_{\max_l}^{(\mathrm{IN})} & \text{if } t_{\min_l}^{(\mathrm{IN})} \leqslant M_l^{(\mathrm{IN})}(t) \leqslant t_{\max_l}^{(\mathrm{IN})} \\ 1 & \text{if } M_l^{(\mathrm{IN})}(t) > t_{\max_l}^{(\mathrm{IN})} \end{cases} \tag{37}$$

$$p_l^{(\mathrm{OUT})}(t) = \begin{cases} 0 & \text{if } M_l^{(\mathrm{OUT})}(t) < t_{\min_l}^{(\mathrm{OUT})} \\ \dfrac{M_l^{(\mathrm{OUT})}(t) - t_{\min_l}^{(\mathrm{OUT})}}{t_{\max_l}^{(\mathrm{OUT})} - t_{\min_l}^{(\mathrm{OUT})}} p_{\max_l}^{(\mathrm{OUT})} & \text{if } t_{\min_l}^{(\mathrm{OUT})} \leqslant M_l^{(\mathrm{OUT})}(t) \leqslant t_{\max_l}^{(\mathrm{OUT})} \\ 1 & \text{if } M_l^{(\mathrm{OUT})}(t) > t_{\max_l}^{(\mathrm{OUT})} \end{cases} \tag{38}$$

where $t_{\min}^{(\mathrm{IN})}, t_{\max}^{(\mathrm{IN})}, p_{\max}^{(\mathrm{IN})}, t_{\min}^{(\mathrm{OUT})}, t_{\max}^{(\mathrm{OUT})}, p_{\max}^{(\mathrm{OUT})}$ are the MRED parameters, and $M_l^{(\mathrm{IN})}(t), M_l^{(\mathrm{OUT})}(t)$ are the estimated average lengths considered by the buffer l in order to derive $p_l^{(\mathrm{IN})}(t)$ and $p_l^{(\mathrm{OUT})}(t)$ respectively. They depend on which MRED algorithm is adopted (RIO-C, RIO-DC or WRED).

RIO-C stands for RED with in/out and coupled average queues, and represents the traditional RIO algorithm. In the RIO-C algorithm, the discarding function relating to IN packets is based on the estimated average length $m^{(\mathrm{IN})}(t)$ of the *IN packet virtual queue*, while that relating to OUT packets is based on the estimated average length, $m(t)$, of the whole buffer queue.

RIO-DC stands for RED with in/out and decoupled average queues. In the case of RIO-DC, the discarding function relating to IN packets is based on the estimated average length, $m^{(\mathrm{IN})}(t)$, of the IN packet virtual queue, while that relating to OUT packets is based on the estimated average length, $m^{(\mathrm{OUT})}(t)$, of the OUT packet virtual queue, that is the ordered sequence of OUT packets queued in the buffer.

WRED stands for weighted-RED. In this MRED algorithm both the discarding functions relating to IN and OUT packets are based on the estimated average length, $m(t)$, of the whole buffer queue.

Therefore, depending on the MRED algorithm adopted by the buffer l, $M_l^{(\mathrm{IN})}(t)$ and $M_l^{(\mathrm{OUT})}(t)$ will be set as follows:

$$M_l^{(\mathrm{IN})}(t) = \begin{cases} m_l^{(\mathrm{IN})}(t) & \text{for RIO} - \text{C} \\ m_l^{(\mathrm{IN})}(t) & \text{for RIO} - \text{DC} \\ m_l(t) & \text{for WRED} \end{cases} \tag{39}$$

$$M_l^{(\mathrm{OUT})}(t) = \begin{cases} m_l(t) & \text{for RIO} - \text{C} \\ m_l^{(\mathrm{OUT})}(t) & \text{for RIO} - \text{DC} \\ m_l(t) & \text{for WRED} \end{cases} \tag{40}$$

2.4. Numerical solution of the model

The tool developed to solve the system of differential equations making up the model is based on the finite differences method. Specifically, we consider a first-order approximation of the temporal derivative:

$$\frac{\mathrm{d}f(t)}{\mathrm{d}t} = \frac{f(t+h) - f(t)}{h} \tag{41}$$

It is called the forward difference because it calculates the derivative in the positive (forward) direction. Equation (41) allows us to calculate the value of $f(t+h)$ when the value of $f(t)$ is known. So, starting from the initial conditions it is possible to iteratively determine the sampled temporal evolution of $f(t)$. The sampling period is h: the smaller h is, the higher the level of accuracy. On the other hand, excessively low values of h determine unacceptable processing times.

The correct choice of h in our framework has to be made by taking into account the link capacity values in the scenario addressed. In fact, if we set h equal to the inverse of the link capacity expressed in packets/s, we obtain a sample for each transmitted packet; therefore the accuracy is similar to that achieved in per-packet simulation. Actually we found that choosing values of h slightly greater than the inverse of the maximum link capacity in the network does not produce any significant variations in the shape of the model solution. This means that the inverse of the maximum link capacity can be considered as a lower threshold for h.

The upper threshold for h is represented, instead, by the minimum propagation delay. In fact, if h is greater than the propagation delay, the model solution is not able to represent the propagation effects on the network variables.

From a large number of tests we deduced that the forward difference used to solve the proposed model constitutes the best trade-off between accuracy and computational complexity.

3. MODEL ASSESSMENT

In order to demonstrate the accuracy of the proposed model, in this section we compare the results with those obtained by the ns-2 simulator. We consider a network consisting of six MRED routers named A, B, C, D, E

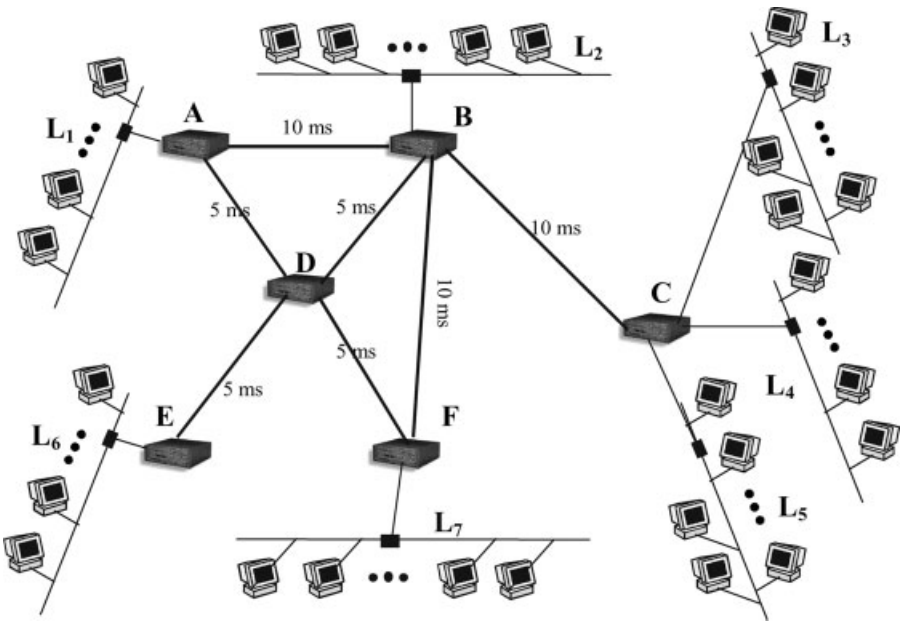

Figure 1. Network topology.

and F, with one or more LANs directly attached to each of them. The network topology is shown in Figure 1.

We study three scenarios which differ from each other according to the MRED mechanism adopted in the routers. For the sake of simplicity in this section we assume that in each scenario all the routers adopt the same MRED mechanism for all the queues located on their output links. We will refer to Scenario 1, Scenario 2 and Scenario 3 when all the routers implement the RIO-C, RIO-DC and WRED techniques in their buffers respectively. The other configuration parameters for each MRED router are shown in Table 2.

Let us assume that the network is loaded by six traffic aggregates, each following the paths listed in the second column in Table 3. The third and fourth columns in the table give the traffic profile used for each traffic aggregate, expressed in terms of CIR and CBS parameters.

Moreover, let us assume each traffic aggregate to be made up of two groups of flows: one generated by greedy TCP sources and the other by data-limited TCP sources. We assume that data-limited sources belonging to the same traffic aggregate start to transmit files of the same size at approximately the same instant, even though this is not a restrictive hypothesis. Table 4 gives detailed information about all the groups of flows.

Table 2. MRED router configurations.

Router	$t_{\min}^{(IN)}$	$t_{\max}^{(IN)}$	$p_{\max}^{(IN)}$	$t_{\min}^{(OUT)}$	$t_{\max}^{(OUT)}$	$p_{\max}^{(OUT)}$	α
A–B–C–D–E–F	100	150	0.1	50	100	0.5	0.0001

The network analysis was carried out assuming that all the links had the same capacity, ranging between 10 and 150 Mb/s. We consider 1000-byte fixed-size packets. The results provided by the proposed model are shown in Figures 2, 3 and 4, where they are compared with those obtained by the ns-2 simulator. All the simulation results were obtained from the average of 30 simulation runs. In the worst case, we had a 5% accuracy with a 95% confidence interval.

The average queue lengths shown in Figure 2 only refer to the two bottleneck buffers in the network; the results obtained by using both ns-2 and our model show, in fact, that the remaining buffers are practically empty throughout the observation period. As can be seen in Figure 2, our model captures the average queue length of both buffers quite well, whatever scenario we address.

Moreover, Figure 2 shows that when the output link capacity is lower than $\Phi = \Sigma_i \mathrm{CIR}_i$, which is the summation of the CIR parameters assigned to the traffic aggregates flowing through the router, all the M-RED mechanisms determine approximately the same average queue length in the router buffers; on the other hand, when the link capacity increases over the threshold Φ, RIO-DC routers (Scenario 2) present higher average queue lengths than RIO-C and WRED routers. When the output link capacity is lower than Φ, in fact, the average throughput

Table 3. Description of traffic aggregates.

Traffic aggregate identifier	Path followed (source–routers–destination)	Committed information rate (CIR) [packets/s]	Committed burst size (CBS) [packets]
A_1	$L_1 - A - B - C - L_3$	2000	150
A_2	$L_1 - A - D - F - L_7$	2000	150
A_3	$L_2 - B - C - L_3$	1000	100
A_4	$L_6 - E - D - B - C - L_4$	2000	150
A_5	$L_6 - E - D - F - L_7$	1000	100
A_6	$L_7 - F - B - C - L_5$	1000	100

Table 4. Information about groups of flows.

Index of group (k)	Number of flows	Traffic aggregate	File size [packets]	Starting time [s]
1	100	A_1	greedy	0
2	40	A_1	50	20
3	100	A_2	greedy	0
4	40	A_2	30	60
5	50	A_3	greedy	0
6	20	A_3	100	70
7	100	A_4	greedy	0
8	20	A_4	50	50
9	50	A_5	greedy	0
10	40	A_5	1000	40
11	50	A_6	greedy	0
12	20	A_6	500	30

of TCP traffic aggregates is low due to the congestion control algorithm; consequently, the token buckets in the edge routers mark almost all the packets as IN packets. So, the M-RED routers provide the same performance as their parameters are the same in all the scenarios addressed.

Of course, good network parameter dimensioning requires the capacity of each link to be greater than the sum of the CIR of the traffic aggregates passing through the link. The above considerations highlight that none of the M-RED mechanisms is able to recover a wrong network dimensioning of link capacities.

When, on the contrary, the link capacity is higher than Φ, the average throughput of TCP traffic aggregates is higher than the assigned CIR, and there are a large number of OUT packets in the network. As the OUT packet discarding probability in RIO-DC routers is driven by the estimated average queue length of OUT packets only, the RIO-DC router discarding probability is lower than that of RIO-C and WRED where, in both cases, the discarding probability is driven by the estimated average queue length of all the packets. Consequently, the average queue length in RIO-DC routers will be higher than in RIO-C and WRED routers. Moreover, as expected, in all the scenarios addressed the average queue length tends to become constant close to the value $(t_{\max}^{(\mathrm{OUT})} + t_{\min}^{(\mathrm{OUT})})/2$ when the link capacity increases. Therefore, there are no more improvements in the average RTT by increasing the link capacity: only the throughput assigned to the sources increases.

Figure 3 compares the average throughput for two traffic aggregates obtained by ns-simulation and by applying the proposed model. Figure 4 compares the average loss ratio suffered by a generic TCP source belonging to the above traffic aggregates. Both figures demonstrate the accuracy of our fluid model in predicting the average throughput and loss probability of TCP sources in the network. For the sake of conciseness we have not presented the results for the other traffic aggregates, but we found a good match like that achieved in Figures 3 and 4.

The fluid-flow model proposed in this paper is also able to capture the average transient behaviour of the network. As a demonstration, in Figure 5 we have plotted the average queue length value of the two bottleneck routers during a time interval of 100 s. All the routers in the network adopt the WRED mechanism. The link capacity (10 Mbps) was chosen suitably low in such a way as to facilitate comparison with ns-2 simulation results in the plot. The

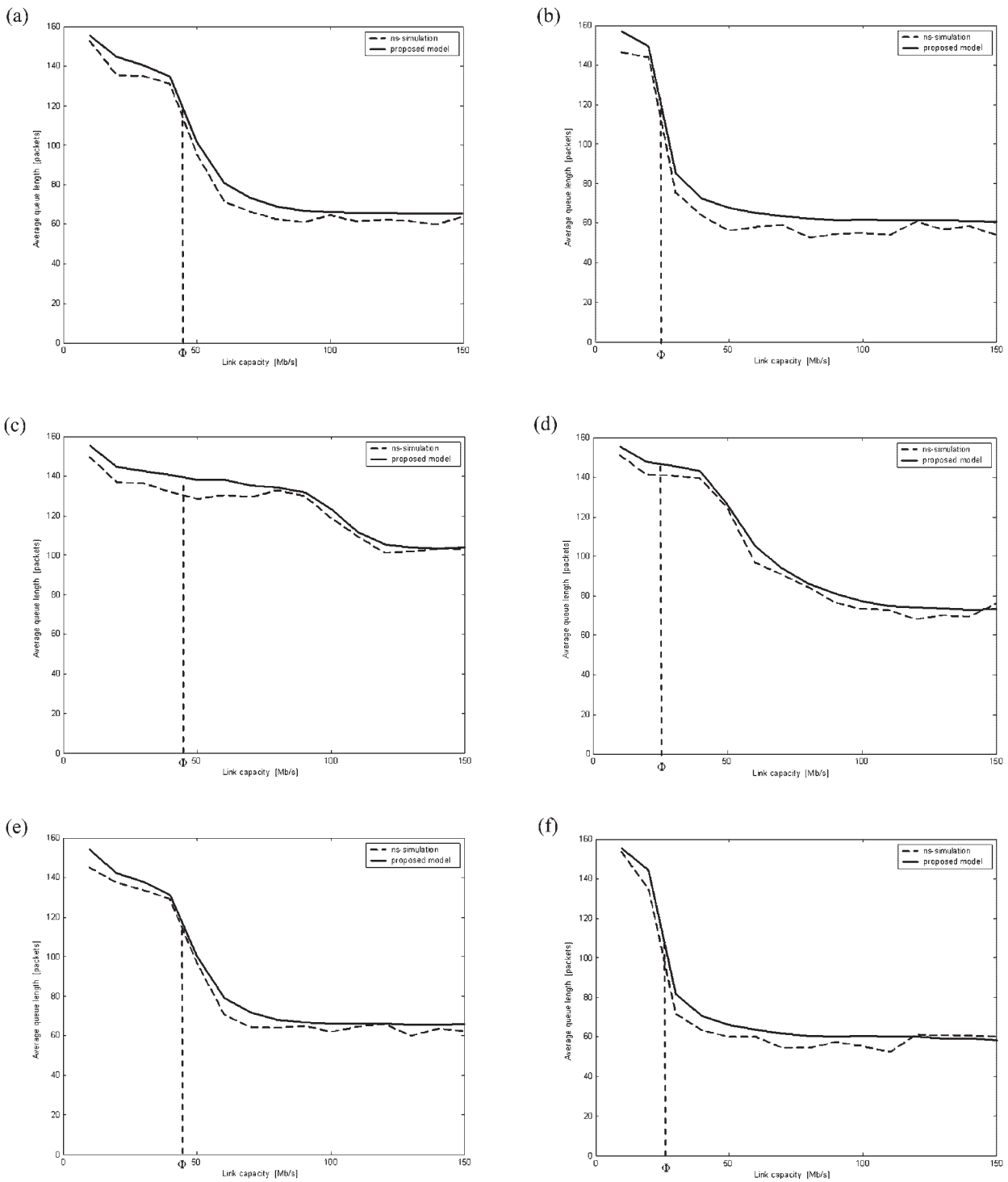

Figure 2. Average queue length: comparison between simulation and proposed model. (a) Average queue length of the router B output buffer towards router C—Scenario 1; (b) average queue length of the router D output buffer towards router F—Scenario 1; (c) average queue length of the router B output buffer towards router C—Scenario 2; (d) average queue length of the router D output buffer towards router F—Scenario 2; (e) average queue length of the router B output buffer towards router C—Scenario 3; (f) average queue length of the router D output buffer towards router F—Scenario 3.

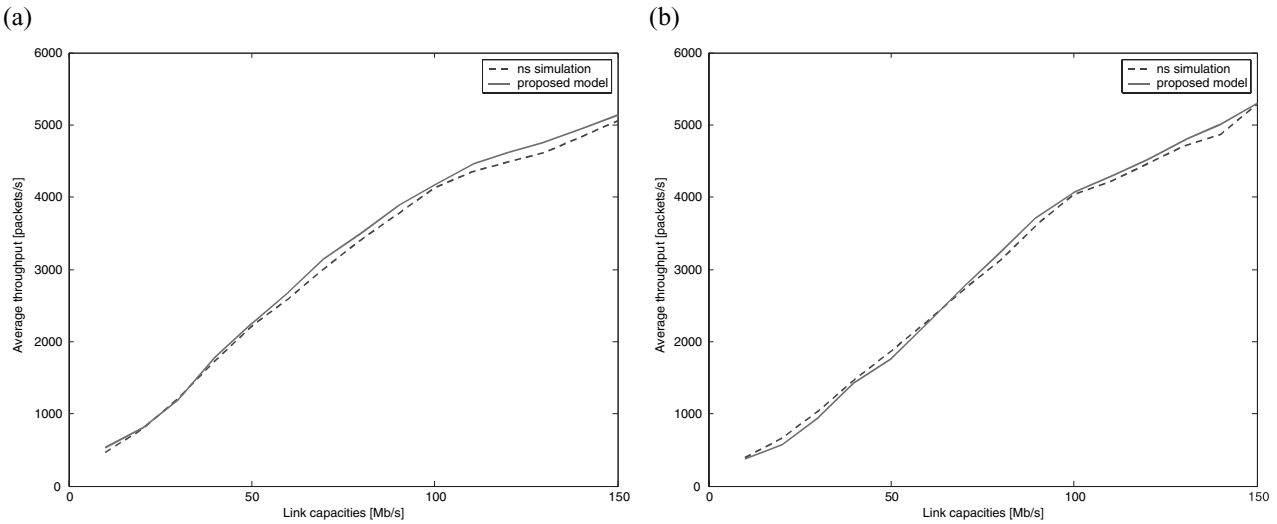

Figure 3. Average throughput: comparison between simulation and proposed model. (a) Average throughput for traffic aggregate A_1—Scenario 1; (b) average throughput for traffic aggregate A_5—Scenario 1.

averaging interval for the simulation results is equal to the sampling period $h = 0.001$.

The average queue length obtained by the fluid model fits the average queue length calculated using the ns-2 simulator quite well in both cases. In particular, the model is able to capture the queue length variations due to activation or deactivation of data-limited TCP sources. For the sake of conciseness we do not show the temporal variation in the average queue length for RIO routers; however, they confirmed the same capacity to capture transient phenomena.

4. A CASE STUDY

As an application of the proposed model, in this section we will address the problem of link capacity dimensioning in a DiffServ domain. We consider a network with the topology presented in Figure 1, and assume that in this domain an AF-PHB (with two DPs) is defined for TCP traffic. Let us note that even when several PHBs exist in the network this is not a restrictive hypothesis because in a DiffServ environment routers usually implement scheduling techniques that assign each PHB a fixed quantity of bandwidth. In the above scenario, we are interested in calculating, for each router, the minimum bandwidth that the schedulers have to assign the AF-PHB relating to TCP traffic such that the QoS requirements are satisfied.

Since the TCP protocol is reliable in packet delivery, we only consider one performance parameter to differentiate the QoS provided by the network, that is, the actual bandwidth available for it. For this reason, for the AF-PHB that we are considering, the QoS requirements will only be expressed in terms of goodput, that is in terms of throughput without considering the packet loss rate.

We assume that the traffic load of the network is the same as that considered in Section 3. For the reader's convenience in Table 5 we reorganise the traffic load information listed in Tables 3 and 4, adding the goodput that we assume to be required by each traffic aggregate in the last column.

Before dimensioning the link capacities we have to choose the token bucket configuration parameters CIR and CBS, with the aim of marking the packets for each aggregate correctly. For each token bucket, the CIR value is obviously equal to the goodput required by the traffic aggregate passing through it. The CBS value, on the other hand, represents the maximum permitted burst size; so for packets to be marked correctly CBS has to be greater than the maximum burst size that an aggregate conforming to its traffic profile could generate. It can easily be argued that this maximum burst size occurs when both the average rate of packets belonging to a traffic aggregate is CIR during an RTT, and the corresponding ACK packets arrive almost at the same time. In this case, in fact, the TCP congestion control algorithm will allow a burst of CIR·RTT in-profile

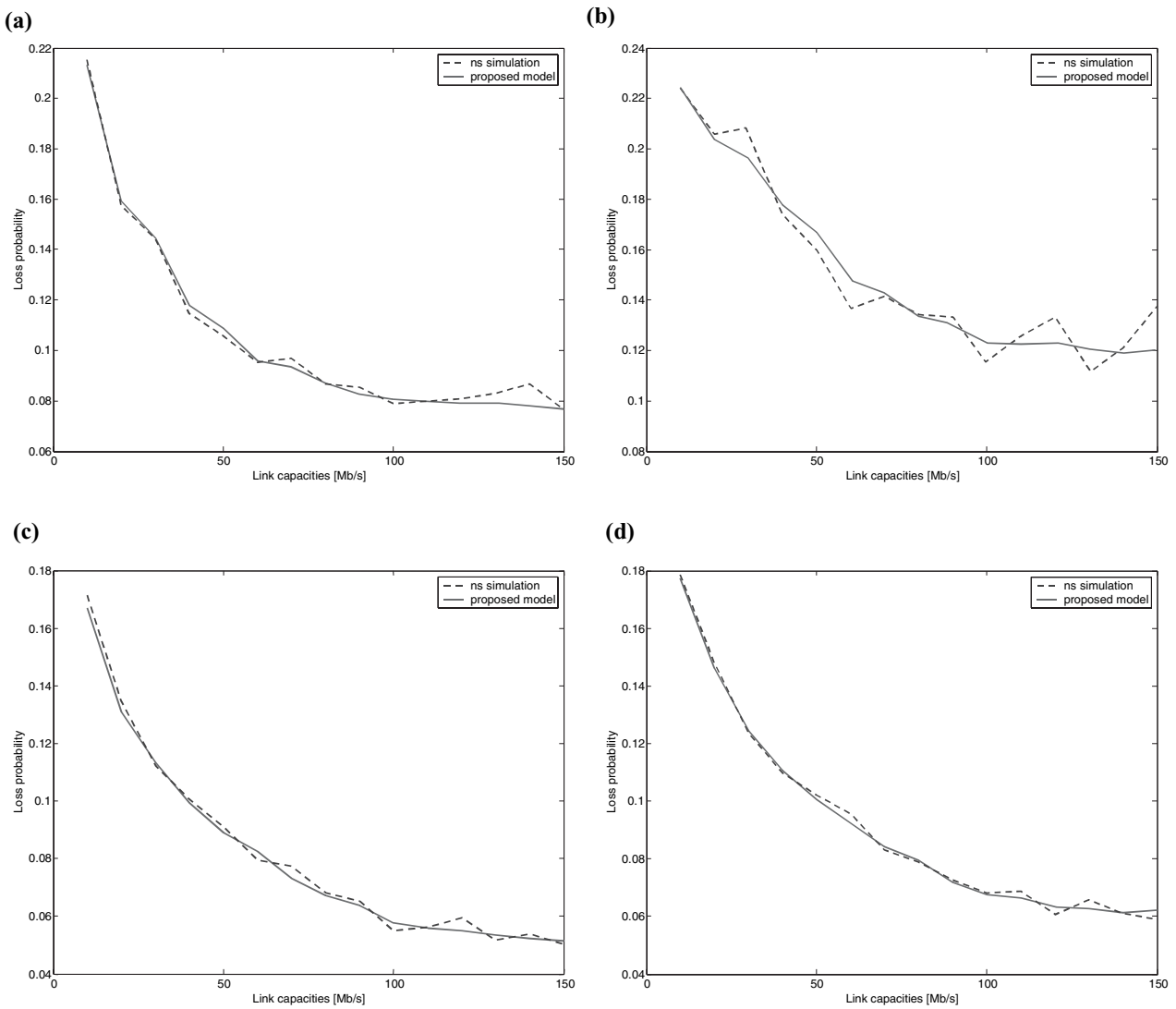

Figure 4. Average loss ratio: comparison between simulation and proposed model. (a) Average loss ratio for a greedy source belonging to the traffic aggregate A_1—Scenario 1; (b) average loss ratio for a data-limited source belonging to the traffic aggregate A_1—Scenario 1; (c) average loss ratio for a greedy source belonging to the traffic aggregate A_5—Scenario 1; (d) average loss ratio for a data-limited source belonging to the traffic aggregate A_5—Scenario 1.

packets to be generated. Let us note that the larger the size of the router buffers (that is the higher the RTT), the lower the probability of the simultaneous arrival of all the ACKs related to 'in flight' packets. For this reason, we will assume the CBS in each token bucket to be $\mathrm{CBS} = \mathrm{CIR} \cdot \mathrm{RTT} \approx \mathrm{CIR} \cdot 4d_{\mathrm{path}}$ where d_{path} is the propagation delay along the path from the source to the receiver. We will see at the end of this section that the RTT estimation used in the above formula is coherent with the RTT value provided by the designed network, that is, $\mathrm{RTT} < 4d_{\mathrm{path}}$. Table 6 lists the set of token bucket parameters that we consider in the network.

In order to solve the problem of link capacity dimensioning we will initially assume that all the routers in the DiffServ network adopt the WRED mechanism. Unlike RIO techniques, in fact, the WRED mechanism has already been adopted in several systems (for example the Cisco 10 000 series). Then, as a second step, we will try to

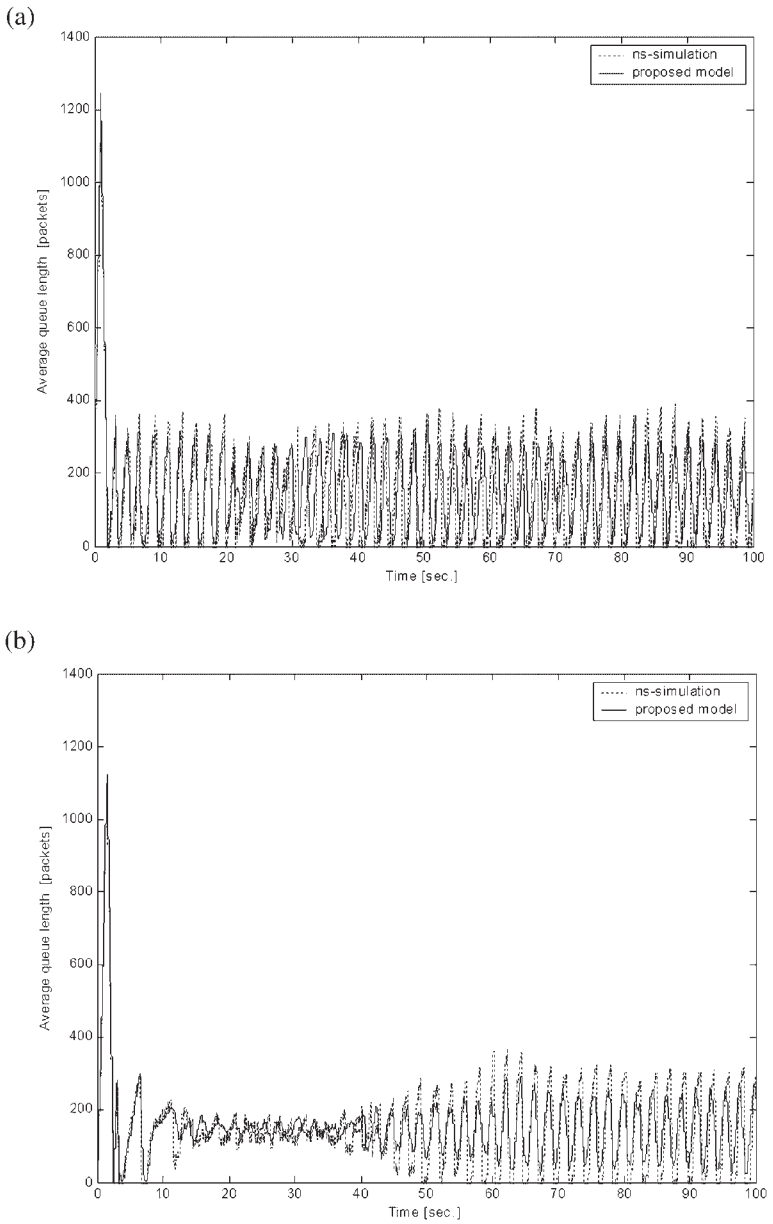

Figure 5. Average queue length temporal behaviour: comparison between simulation and proposed model. (a) Temporal behaviour of the average queue length of the router B output buffer towards router C; (b) temporal behaviour of the average queue length of the router D output buffer towards router F.

further minimise network capacities by substituting some WRED routers with RIO ones. In any case, the parameters listed in Table 2 are used.

In order to calculate the lowest-cost network configuration, that is, the minimum bandwidth to be assigned to the output link of each buffer to provide traffic aggregates with the required QoS, we first consider a configuration in which each buffer in the network routers has a service rate exactly equal to the sum of the goodputs required by the traffic aggregates passing through it (configuration CF_1 in Table 7). We will refer to this configuration as 'exactly provisioned configuration'. It is evident that because of the

Table 5. Information about traffic load.

Traffic aggregate	Path followed (source–routers–destination)	Number of flows	File size [packets]	Starting time [s]	Goodput required [Mb/s]
A_1	$L_1 - A - B - C - L_3$	100	Greedy	0	16
		40	50	20	
A_2	$L_1 - A - D - F - L_7$	100	Greedy	0	16
		40	30	60	
A_3	$L_2 - B - C - L_3$	50	Greedy	0	8
		20	100	70	
A_4	$L_6 - E - D - B - C - L_4$	100	Greedy	0	16
		20	50	50	
A_5	$L_6 - E - D - F - L_7$	50	Greedy	0	8
		40	1000	40	
A_6	$L_7 - F - B - C - L_5$	50	Greedy	0	8
		20	500	30	

Table 6. Token bucket setting.

Traffic aggregate identifier	CIR [Mb/s]	CBS [packets]
A_1	16	150
A_2	16	150
A_3	8	100
A_4	16	150
A_5	8	100
A_6	8	100

AIMD behaviour of the TCP protocol, the 'exactly-provisioned configuration' will not satisfy the QoS requirements. In fact, it only guarantees the required goodputs when there are no OUT packets in the network. For this reason, we will use CF_1 as the threshold configuration. Starting from CF_1, we increase the capacity of the links providing traffic aggregates with a lower QoS than the required one, and decrease the capacity of links providing traffic aggregates with a higher QoS. Of course the capacity of each link has to be greater than the corresponding one in the exactly-provisioned configuration.

Table 7 presents the results obtained, highlighting the unacceptable goodput values (grey cells) for each configuration considered. For the sake of simplicity, we assume that a granularity of 1 Mb/s is permitted in setting the link capacities.

In Table 7 CF_8 is the minimum-cost configuration when the WRED mechanism is used in all the routers because it is the configuration requiring the lowest total link capacity (165 Mb/s) among all the configurations satisfying the QoS requirements. Moreover, we observe that in CF_8 only three links have their minimum value (links $A \rightarrow D$, $D \rightarrow B$ and $E \rightarrow D$), while the link $B \rightarrow C$ requires the higher capacity (51 Mb/s). Let us note that this is the most heavily loaded bottleneck link, as discussed in Section 3. Let us now try to further minimise the other network capacities by substituting some WRED routers with RIO ones. What we expect now is that, in the same conditions, the RIO-C algorithm will perform better than WRED. In fact the RIO-C algorithm protects IN packets more than WRED because RIO-C calculates the IN packet drop probability according to the estimated average queue of the IN packets only; WRED, on the other hand, calculates the IN packet drop probability according to the estimated average queue of all the packets. On the contrary, we expect RIO-DC to perform worse than WRED; RIO-DC, in fact, calculates the OUT packet drop probability according to the estimated average queue of the OUT packets only, whereas WRED calculates the OUT packet drop probability according to the estimated average queue of all the packets, thus providing a more constraining control on OUT packets than RIO DC.

The above considerations are confirmed in Tables 8 and 9, which summarise the results provided by analysis of two scenarios in which the WRED buffer in the most heavily loaded bottleneck link ($B \rightarrow C$) is replaced by a RIO-C and a RIO-DC buffer respectively. In both cases the parameters shown in Table 2 were used.

More specifically, Table 8 shows that RIO-C provides the required QoS by assigning a lower total bandwidth than CF_8 (163 Mb/s with the configuration CF_{15}); Table 9, on the contrary, shows that the RIO-DC results are completely unacceptable when the same bandwidth assignment as CF_8 is considered. In order to reduce as

Table 7. Goodputs achieved by traffic aggregates with different configurations in the presence of WRED routers only.

Config.	Link capacities								Achieved goodputs					
	A → B [WRED]	A → D [WRED]	B → C [WRED]	D → B [WRED]	D → F [WRED]	E → D [WRED]	F → B [WRED]	TOTAL	A_1	A_2	A_3	A_4	A_5	A_6
CF_1	16	16	48	16	24	24	8	152	15.77	15.98	8.55	15.79	8.01	7.87
CF_2	17	17	49	17	25	25	9	159	15.85	16.65	9.22	15.75	8.34	7.97
CF_3	18	17	50	18	26	25	10	164	15.98	16.46	9.37	15.84	8.53	8.16
CF_4	19	17	51	19	27	25	10	168	16.18	16.34	9.62	16.02	8.63	8.24
CF_5	19	17	51	18	27	25	10	167	16.16	16.30	9.59	16.01	8.65	8.21
CF_6	19	17	51	17	27	25	10	166	16.20	16.24	9.78	16.01	8.71	8.20
CF_7	19	17	51	17	27	25	9	165	16.30	16.29	10.01	15.99	8.67	8.15
CF_8	**18**	**17**	**51**	**17**	**27**	**25**	**10**	**165**	**16.02**	**16.24**	**9.94**	**16.01**	**8.72**	**8.20**
CF_9	18	17	51	17	27	25	9	164	16.07	16.28	10.14	15.98	8.66	8.14
CF_{10}	17	17	51	17	27	25	10	164	15.96	16.25	10.18	16.00	8.71	8.24
CF_{11}	18	17	50	17	27	25	10	164	15.95	16.23	9.39	15.91	8.71	8.13
CF_{12}	18	17	51	17	26	25	10	164	16.01	16.44	9.90	15.91	8.54	8.23

Table 8. Goodputs achieved by traffic aggregates with different configurations in the presence of WRED and RIO-C routers.

Config.	Link capacities								Goodputs achieved					
	A → B [WRED]	A → D [WRED]	B → C [RIO-C]	D → B [WRED]	D → F [WRED]	E → D [WRED]	F → B [WRED]	TOTAL	A_1	A_2	A_3	A_4	A_5	A_6
CF_{13}	18	17	51	17	27	25	10	165	16.10	16.26	9.87	16.07	8.68	8.22
CF_{14}	18	17	51	17	27	25	9	164	16.14	16.30	10.06	16.03	8.66	8.16
CF_{15}	**17**	**17**	**51**	**17**	**27**	**25**	**9**	**163**	**16.02**	**16.30**	**10.16**	**16.02**	**8.66**	**8.10**
CF_{16}	17	17	51	17	26	25	9	162	16.06	16.46	10.30	15.99	8.52	8.17
CF_{17}	17	17	50	17	27	25	9	162	15.94	16.27	9.62	15.92	8.66	8.05

Table 9. Goodputs achieved by traffic aggregates with different configurations in the presence of WRED and RIO-DC routers.

Config.	Link capacities								Goodputs achieved					
	A → B [WRED]	A → D [WRED]	B → C [RIO-DC]	D → B [WRED]	D → F [WRED]	E → D [WRED]	F → B [WRED]	TOTAL	A_1	A_2	A_3	A_4	A_5	A_6
CF_{18}	18	17	51	17	27	25	10	165	12.88	16.11	17.02	11.55	8.84	7.27

Table 10. Goodputs achieved by traffic aggregates with different configurations in the presence of WRED and 2 RIO-C routers.

Config.	Link capacities								Goodputs achieved					
	A → B [WRED]	A → D [WRED]	B → C [RIO-C]	D → B [WRED]	D → F [WRED]	E → D [RIO-C]	F → B [WRED]	TOTAL	A_1	A_2	A_3	A_4	A_5	A_6
CF_{19}	17	17	51	17	27	25	9	163	16	16.30	10.16	16.04	8.64	8.13
CF_{20}	**17**	**17**	**51**	**17**	**26**	**25**	**9**	**162**	**16.03**	**16.45**	**10.24**	**16.06**	**8.56**	**8.10**
CF_{21}	17	17	50	17	26	25	9	161	15.94	16.43	9.64	15.95	8.54	8.06
CF_{22}	17	17	51	17	25	25	9	161	16.08	16.66	10.38	15.99	8.34	8.18

Table 11. Goodputs achieved by traffic aggregates with different configurations in the presence of RIO-C routers only.

Config.	Link capacities								Goodputs achieved					
	A → B [RIO-C]	A → D [RIO-C]	B → C [RIO-C]	D → B [RIO-C]	D → F [RIO-C]	E → D [RIO-C]	F → B [RIO-C]	TOTAL	A_1	A_2	A_3	A_4	A_5	A_6
CF_{23}	17	17	51	17	25	25	9	161	16.08	16.65	10.4	15.99	8.34	8.16
CF_{24}	17	17	50	17	26	25	9	161	15.98	16.44	9.66	15.96	8.54	8.07

much as possible the bandwidth that the routers have to assign to traffic belonging to the AF-PHB being considered, we studied both a scenario in which the second bottleneck link (D → F) also adopts the RIO-C algorithm (Table 10) and one in which all the routers are RIO-C routers (Table 11). The results obtained demonstrate that the use of two RIO-C buffers allows us to further reduce the total amount of bandwidth required using configuration CF_{20}, while the use of RIO-C routers in all the network does not produce any further improvements.

Finally, let us note that the assumption we made about the RTT estimation in the computation of CBS, that is, $RTT \approx 4d_{path}$, is confirmed. In CF_{20}, in fact, the link capacities of the two bottleneck links (B → C and D → F) correspond to average queue length of about 100 packets (see Figure 2a,b respectively). Since we considered a packet size of 1000 bytes, the queuing delay is approximately 16 ms in the router B output buffer towards router C and 32 ms in the router D output buffer towards router F. On the other hand, the propagation delay d_{path} from a generic sender to the corresponding receiver in the network is in the range [10, 20] ms, consequently the relationship $RTT < 4d_{path}$ is always true in the scenario addressed.

5. CONCLUSIONS

In this paper we have defined an accurate fluid model of a DiffServ network supporting TCP sources which are not necessarily greedy, also taking the slow-start phase into consideration. Network nodes adopt WRED or RIO AQM mechanisms to guarantee service differentiation. We have compared the results given by the model with those obtained via simulation, obtaining a good match for both transient and steady-state network behaviour. The main characteristic of the proposed analytical approach is its scalability with respect to both the number of traffic flows considered and the complexity of the network topology. For this reason the tool developed to solve the system of differential equations making up the model gives the average values of network and source variables in a much shorter time than simulation. In addition the model results can easily be analysed because they do not present the randomness of the simulation results. These properties make our modelling approach suitable to address the issue of network parameter optimisation. As a case study, the design of link capacity versus the AQM technique used in the network by means of an iterative optimisation algorithm has been discussed.

REFERENCES

1. Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated services. *RFC 2475*, December 1998.
2. Nichols K, Blake S, Baker F, Black D. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. *RFC 2474*, December 1998.
3. Heinanen J, Baker F, Weiss W, Wrocklawski J. Assured Forwarding PHB Group. *RFC 2597*, June 1999.
4. Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking* 1993; **1**(4):397–413.
5. Clark DD, Fang W. Explicit allocation of best effort packet delivery service. *IEEE/ACM Transactions on Networking* 1998; **6**(4):362–373.
6. Seddigh N, Nandy B, Pieda P, Hadi Salim J, Chapman A. An experimental study of Assured services in a DiffServ IP QoS Network. *Proceedings of SPIE symposium, GLOBECOM'99*, Rio De Janeiro, December 99.
7. http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart3/qcwred.htm
8. Abouzeid A, Roy S. Modeling random early detection in a differentiated services network. *Computer Networks (Elsevier)* 2002; **40**(4):537–556.
9. Malouch N, Liu Z. On steady state analysis of TCP in networks with Differentiated Services. Proceedings of Seventeenth International Teletraffic Congress, ITC'17, December 2001.
10. Yeom I, Reddy A. Modeling TCP behavior in a differentiated-services network. *IEEE/ACM Transactions on Networking* 2001; **9**(1):31–46.
11. Chait Y, Hollot C, Misra V, Towsley D, Zhang H. Providing throughput differentiation for TCP flows using adaptive two color marking and multi-level AQM. *IEEE INFOCOM 2002*, New York, NY, June 2002; 23–27.
12. Misra V, Gong W, Towsley D. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. *SIGCOMM'00*, August 2000.

13. Guo L, Matta I. *The War Between Mice and Elephants*. ICNP 2001: Riverside, CA, November 2001.
14. Barbera M, Lombardo A, Schembra G. A fluid-based model of time-limited TCP flows. *Computer Networks (Elsevier)* 2004; **44**(3):275–288.
15. van Foreest N, Mandjes M, Scheinhardt WRW. Analysis of a feedback fluid model for heterogeneous TCP sources. *Stochastic Models* 2003; **19**:299–324.
16. The network simulator—ns-2. LBL, URL: http://www.isi.edu/nsnam/ns/
17. Floyd S, Henderson T. The NewReno modification to TCP's fast recovery algorithm. *RFC 2582* (April 1999).
18. Floyd S, Handley M, Padhye J, Widmer J. Equation Based Congestion Control for Unicast Applications. *SIGCOMM 2000*, August 2000.

AUTHORS' BIOGRAPHIES

Mario Barbera received the degree in electrical engineering from the University of Catania, Italy, in 2001. His final thesis was on fluid-flow analytical models of AQM networks and TCP sources. He received the Ph.D. degree in computer science and telecommunications engineering with a dissertation on modeling and design of next generation IP networks through Fluid-Flow approach in 2004. He is currently a Post-Doc student at the University of Catania.

Alfio Lombardo received his degree in electrical engineering from the University of Catania, Italy, in 1983. Until 1987, he acted as consultant at CREI, the center of the Politecnico di Milano for research on computer networks, where he was involved in the Software Environment for the design of Distributed Open Systems (SEDOS) and Conformance Testing Service-Wide Area Networks (CTS-WAN) CEC projects. He was the Technical Coordinator of the Formal Description Techniques (FDT) COST 11 TER project from 1986 to 1988. In 1988 he joined the University of Catania where he is full professor of Telematics. There he was the leader of the University of Catania team in the European ACTS project DOLMEN (Service Machine Development for an Open Long-term Mobile and Fixed Network Environment) and in the European IST project VESPER (Virtual Home Environment for Service Personalization and Roaming Users). Moreover he has coordinated the University of Catania team in the national projects FIRB-TANGO and COFIN-EURO. His research interests include distributed multimedia applications, multimedia traffic modeling and analysis, Internet2, adaptive video. He is author of about 100 papers on the above subjects.

Giovanni Schembra received the degree in electrical engineering from the University of Catania, Italy, in 1991. Working in the Telecommunications area, he received the master degree from CEFRIEL (Milan—Italy), in 1992. His master's thesis was on the analytical performance evaluation in an ATM network. He received the Ph.D. degree in electronics, computer science and telecommunications engineering with a dissertation on multimedia traffic modeling in a broadband network. He is currently Assistant Professor in Telecommunications at the University of Catania.

Andrea Trecarichi received the degree in electrical engineering from the University of Catania, Italy, in 2002, with a thesis on a fluid-flow model of RIO routers loaded by Markov Modulated Fluid Processes. He has been a collaborator in the Telecommunication Department, University of Catania, for almost two years working basically on fluid-flow models of DiffServ networks in the TANGO FIRB project. Now he works as Design Engineer in ST Microelectronics.

