A Fluid-Based Model of Time-Limited TCP Flows in a DiffServ Environment

Mario Barbera, Alfio Lombardo, Giovanni Schembra, C. Andrea Trecarichi
Dipartimento di Ingegneria Informatica e delle Telecomunicazioni - University of Catania
V.le A. Doria, 6 - 95125 Catania - ITALY
E-mail: (mbarbera, lombardo, schembra, ctreca)@diit.unict.it
Tel. +39 095 7382375; Fax. +39 095 338280

***Abstract* — The Differentiated Services (DiffServ - DS) 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 Behavior (PHB) to all the packets classified as belonging to the same class at the edge of the network. One of the most important standard types of PHB is the 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, DS nodes in the core network implement Active Queue Management (AQM) mechanisms. The challenge in this context is appropriate configuration of the AQM control parameters. For this reason, it is important to provide designers with an accurate analytical framework to calculate the end-to-end performance of TCP flows in a DiffServ network. The target of this paper is to provide an accurate fluid-flow analytical model of a DiffServ network where the RIO-C, RIO-DC and WRED AQM techniques are implemented. We address a network simultaneously loaded with both short-lived and long-lived 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). The proposed model can be applied to any complex network topology, not just to a bottleneck link. In addition, it is able to capture cases in which not all network routers implement the same AQM technique.**

Keywords: DiffServ Architecture, TCP, RIO, WRED, Fluid-flow Analysis.

I. 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 - DS) architecture, proposed in [1]. This architecture allows 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 DS 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 DSCP of the packets [2]; then packets are forwarded to core routers, whose task is just to offer the same Per-Hop Behavior (PHB) to all the packets marked with the same DS codepoint (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 [3] Assured Forwarding (AF) PHB was proposed to provide individual or aggregate flows with guarantees in terms of throughput and burstiness, according to negotiated profiles.

In the event of congestion, the drop precedence determines the relative importance of a packet within the AF class: a congested DS 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 [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 has been extended to the case of two or more DPs, that is, RIO (RED with In/Out) [5], with its two variants RIO-C (RIO with Coupled average queues) and RIO-DC (RIO with Decoupled average queues) [6], and WRED (Weighted RED) [7] have been defined.

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 [8], [9], [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 [11] the authors extend the model proposed in [12] for a network of non DS-compliant AQM routers, to a DiffServ network supporting AF PHB with two DPs. This approach uses a fluid-flow approximation to model traffic and queue behavior in order to maintain a low level of model complexity for any buffer queue dimension, without limitations on the network topology or number of TCP flows to be modeled. However, in [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 modeled with two parameters that denote the fraction of fluid belonging to the two DPs. Further, [11] does not consider short-lived 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 the Slow Start mechanism, which is neglected in [11], has to be modeled as well as Congestion Avoidance in order to capture the behavior of more realistic scenarios.

In this paper 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). Moreover best-effort traffic is taken into account, marking the related packets as OUT and treating them like non-compliant packets [14]. The target of the paper is to provide an accurate fluid-flow analytical model of a DiffServ network simultaneously loaded with short-lived and long-lived TCP flows. At the edge of the network, these flows are grouped into traffic aggregates, each independently policed by a token bucket. For the sake of generality, a token bucket is applied to an aggregate flow, a single flow being a special case of an aggregate flow. In the core routers RIO-C, RIO-DC and WRED algorithms are modeled. Unlike most previous proposals, the model can be applied to any complex network topology, not just to a bottleneck link. 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.

The rest of the paper is organized as follows. In Section II, we present the analytical model. In Section III, we consider an application of our model to a network scenario and compare our results with those obtained using the ns-2 simulator [15]. Finally, we present our conclusions in Section IV.

II. MODEL AND ANALYSIS

After a description of the assumptions made in Section II.A, in order to make TCP modeling independent of the AQM techniques used in the routers, in Section II.B we will introduce the TCP behavior model, and in Section II.C we will describe the model of the network of DiffServ routers.

A. Preliminary considerations and definitions

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 approximate analysis of the average behavior 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:

- 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;
- the file size that each data-limited source has to transmit is known *a priori*;
- the instant at which each source starts to transmit is known.

According to these hypotheses, if there were only drop-tail routers in the network, the system would be completely deterministic. In more general terms, when Diffserv routers adopt AQM techniques to differentiate between services, the stochastic behavior of the system is only induced by the drop probability in the network routers.

Since we are interested in analyzing the average behavior of the network, 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. In other words, we derive an approximation of the average behavior of the whole system (network and sources) assuming that each AQM buffer in the network has a deterministic behavior equal to its average behavior.

B. TCP behavior modeling

Let us consider a set of DS-compliant routers making up a DiffServ domain and let L be the set of all the 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 packets 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 $q_l(t)$.

Let us consider a workload of N TCP flows labeled as $i = 1,\ldots,N$, and let $W_i(t)$ denote the congestion window process of the flow *i*. In our framework we will not consider the end-to-end flow control algorithm, assuming TCP throughput to be bounded by the congestion control algorithm alone. For this reason $W_i(t)$ also represents the transmission window of the *i*-th 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 *i*-th TCP flow at the instant *t*.

Since we are interested in analyzing not only the behavior 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 packets successfully sent by the *i*-th TCP flow from its start to the time *t*.

In order to characterize 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 *i*-th 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 delays 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* as L_i, we obtain:

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

Now we derive the relationship that describes the additive-increase multiplicative-decrease (AIMD) behavior 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:

$$\frac{dW_i(t)}{dt} = A_i(t) + B_i(t) \qquad i = 1,2,...,N \tag{2}$$

First we will calculate the additive-increase term $A_i(t)$. To this end let us note that the congestion window size $W_i(t)$ of each ACK packet reaching a generic TCP source 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 $\lambda_i^{(A)}(t)$, we can write:

$$A_i(t) = \begin{cases} \lambda_i^{(A)}(t) & \text{if} \quad W_i(t) < T_i(t) \\ \dfrac{\lambda_i^{(A)}(t)}{W_i(t)} & \text{if} \quad W_i(t) \geq T_i(t) \end{cases} \qquad i = 1,2,...,N \tag{3}$$

The derivation of the term $B_i(t)$ in (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 could be extended to the other TCP versions. Because a New-Reno TCP source behaves differently if 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 Section II.C 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 *i*-th TCP source at the time instant *t* as $\zeta_i(t)$.

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 easily be calculated as:

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

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, i.e. $N_i^{(A)}(t-RTO,t) < 3$, losses are detected at the instant *t* as TO_losses, and therefore the loss rate at the instant $t-RTO$, $\zeta_i(t-RTO)$, is a TO_loss rate, henceforward indicated as $\gamma_i^{(TO)}(t)$. 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, i.e. $N_i^{(A)}(t-\tau,t) = 3$, 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 $\gamma_i^{(TD)}(t)$.

So, assuming that the retransmission timeout can be approximated by $4 \cdot R_i(t)$ as in [16], we have:

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

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

A generic New-Reno TCP source halves its congestion window when a TD_loss occurs, while it sets its congestion window to one when a timeout expires. Consequently, the variation of the congestion window $W_i(t)$ is equal to $-\left(W_i(t)/2\right)$ when a TD_loss is detected, while it is equal to $\left(1-W_i(t)\right)$ when a TO_loss is detected. From these considerations we obtain the final expression of $B_i(t)$:

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

Now we will derive the equation that regulates the behavior 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 $\frac{1}{2}W_i(t) - T_i(t)$ otherwise. Considering (5) and (6) we have:

$$\frac{dT_i(t)}{dt} = \left(\frac{W_i(t)}{2} - T_i(t)\right)\cdot\left(\gamma_i^{(TD)}(t) + \gamma_i^{(TO)}(t)\right) \qquad i = 1,2,...,N \tag{8}$$

Finally we derive the relationship to calculate the number of packets $D_i(t)$ 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 source *i*, this source has ended its transmission.

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

$$\frac{dD_i(t)}{dt} = \lambda_i^{(A)}(t) \qquad i = 1,2,...,N \tag{9}$$

Up to now we have derived a set of $3 \cdot N$ differential equations that describe the average behavior $\left(W_i(t), T_i(t), D_i(t)\right)$ 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 describing the sources by grouping the flows having the same average behavior. More specifically, a group is constituted by the set of all flows belonging to the same traffic aggregate, following the same path in the network, and starting at time instants within an interval that is 3÷4 times shorter than their average RTT.

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

If we indicate the arrival rate of ACKs and the rate of lost packets for the *k*-th group of flows as $\lambda_k^{(A)}(t)$ and $\zeta_k(t)$, respectively, we can write:

$$\lambda_i^{(A)}(t) = \frac{\lambda_k^{(A)}(t)}{a_k(t)} \qquad i = 1,2,...,N \tag{10}$$

$$\zeta_i(t) = \frac{\zeta_k(t)}{a_k(t)} \qquad i = 1,2,...,N \tag{11}$$

$\lambda_k^{(A)}(t)$ and $\zeta_k(t)$ will be derived in Section II.C.

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 packets. Consequently, if we indicate the number of packets 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:

$$a_k(t) = n_k - f_k\left(D_k(t)\right) \qquad k = 1,2,...,K \tag{12}$$

For the next derivations it is necessary to calculate the total emission rate at the generic time instant *t* for each group, that is:

$$Th_k(t) = a_k(t) \cdot \frac{W_k(t)}{R_k(t)} \qquad \text{for } k = 1,2,...,K \tag{13}$$

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

C. DiffServ network modeling

As we have already said, we are considering a DiffServ architecture in which Assured Forwarding PHB with two drop precedence (DP) values is defined. We consider *M* traffic aggregates to be present, a traffic aggregate being a collection of one or more groups of flows with the same service profile. As a consequence, two flows belonging to the same group cannot belong to two different traffic aggregates.

A profile meter marks a packet as IN if it is "in-profile", or OUT (i.e. "out-of-profile") otherwise. The profile meters reside at the edge router. We assume the service profile to be completely defined by the two parameters *CIR* (Committed Information Rate) and *CBS* (Committed Burst Size), and a token bucket for each aggregate is used as the policer that classifies packets at edge routers as IN or OUT ($CIR = CBS = 0$ defines the service profile of a best effort traffic aggregate). As is well known, a token bucket can be seen as a virtual buffer of size *CBS* filled with a rate of *CIR* tokens per second. A token represents the right to transmit one packet, assuming the size of all data packets to be constant for the sake of simplicity. For this reason, CIR will be expressed in packets per second and CBS in packets. 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 *M* traffic aggregates, we will have *M* token buckets in the network.

Let $\psi_m(t)$ be the arrival rate of the *m*-th traffic aggregate at the input of the *m*-th token bucket ($m = 1,2,\ldots M$). Because $\psi_m(t)$ collects the emission rates of the sources belonging to the *m*-th traffic aggregate, if we indicate the set of group of flows belonging to the traffic aggregate *m* as G_m, we obtain:

$$\psi_m(t) = \sum_{k \in G_m} Th_k(t) \qquad m = 1,2,\ldots M \tag{14}$$

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 the associated Committed Information Rate and Committed Burst Size. Furthermore, we indicate the output rate of IN and OUT

packets at the output of the token bucket m as $\varphi_m^{(IN)}(t)$ and $\varphi_m^{(OUT)}(t)$, 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 CIR_m 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:

$$\varphi_m^{(IN)}(t) = \begin{cases} \psi_m(t) & if \;\; V_m(t) > 0 \\ \min\left(CIR_m, \psi_m(t)\right) & if \;\; V_m(t) = 0 \end{cases} \tag{15}$$

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

$$\varphi_m^{(OUT)}(t) = \psi_m(t) - \varphi_m^{(IN)}(t) \tag{16}$$

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

$$\frac{dV_m(t)}{dt} = CIR_m - \varphi_m^{(IN)}(t) \qquad \text{with the constraint} \qquad V_m(t) \le CBS_m \tag{17}$$

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

- $\Lambda_l^{(IN)}(t)$ and $\Lambda_l^{(OUT)}(t)$: the total arrival rates (in packets per second) of IN and OUT packets at a generic buffer l at time $t \ge 0$, respectively;
- $\lambda_{k,l}^{(IN)}(t)$ and $\lambda_{k,l}^{(OUT)}(t)$: the average arrival rates (in packets per second) at the buffer l of IN and OUT packets respectively, belonging to the group k at time $t \ge 0$;
- $\mu_{k,l}^{(IN)}(t)$ and $\mu_{k,l}^{(OUT)}(t)$: the average output rates (in packets per second) from the buffer l of IN and OUT packets respectively, belonging to the group k at time $t \ge 0$;
- $p_l^{(IN)}(t)$ and $p_l^{(OUT)}(t)$: the discard probability functions applied to IN and OUT packets respectively, at the generic buffer $l \in L$ at time $t \ge 0$;
- $p_l(t)$: the discard probability function applied to a generic packet at the generic buffer $l \in L$ at time $t \ge 0$;
- $\gamma_{k,l}(t)$: the loss rate (in packets per second) for sources belonging to the group k at the output of the buffer l.

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 is:

$$\frac{dq_l(t)}{dt} = \begin{cases} -C_l + \left(1 - p_l(t)\right) \cdot \\ \cdot \left(\Lambda_l^{(IN)}(t) + \Lambda_l^{(OUT)}(t)\right) & \text{if } \; \bar{q}_l(t) > 0 \\ \left[-C_l + \left(1 - p_l(t)\right) \cdot \right. \\ \left. \cdot \left(\Lambda_l^{(IN)}(t) + \Lambda_l^{(OUT)}(t)\right) \right]^+ & \text{if } \; \bar{q}_l(t) = 0 \end{cases} \qquad \forall l \in L \tag{18}$$

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

The next relationship that we want to derive is the one that exists in the buffer l between the arrival rate and the emission rate of IN packets belonging to the generic group k. Because an analogous equation exists between the arrival rate and the emission rate of OUT packets, we will substitute the IN (or OUT) apex with X. This notation will be frequently adopted in the rest of the paper, so when we use the (X) apex, the reader can replace it with either (IN) or (OUT).

The total number of X packets belonging to the 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 up to the instant t, minus the total number of X packets lost in the same buffer, that is:

$$\begin{aligned} &\int_0^{t + \frac{q_l(t)}{C_l}} \mu_{k,l}^{(X)}(\nu)\, d\nu = \\ &= \int_0^t \lambda_{k,l}^{(X)}(\nu)\, d\nu - \int_0^t \lambda_{k,l}^{(X)}(\nu)\, p_l^{(X)}(\nu)\, d\nu \end{aligned} \tag{19}$$

Deriving both sides of (19) we obtain:

$$\mu_{k,l}^{(X)}\left(t + \frac{q_l(t)}{C_l}\right) = \frac{\lambda_{k,l}^{(X)}(t) \cdot \left(1 - p_l^{(X)}(t)\right)}{1 + \frac{1}{C_l}\frac{dq_l(t)}{dt}} \qquad \begin{matrix} k = 1,2,\ldots K \\ \forall l \in L \end{matrix} \tag{20}$$

Note that, from (18), the denominator in the right side of (20) can be equal to zero only if no packets enter the buffer at time instant t. Consequently, in this case (20) becomes an indeterminate form 0/0, that can be easily solved separately, and its value is 0.

To calculate the average arrival rate $\lambda_{k,l}^{(X)}(t)$ we have to consider the ordered set of buffers $H_k = \left\{h_1^{(k)}, h_2^{(k)}, \ldots, h_J^{(k)}\right\}$ in the path followed by packets belonging to the group k. With this notation we have:

$$\lambda_{k,h_j^{(k)}}^{(X)}(t)=\begin{cases} Th_k(t)\dfrac{\varphi_m^{(X)}(t)}{\psi_m(t)} & \text{if } j=1 \quad (\forall k\in G_m) \\ \mu_{k,h_{j-1}^{(k)}}^{(X)}\left(t-d_{h_{j-1}^{(k)}}\right) & \text{if } j>1 \end{cases} \qquad k=1,2,\dots K \tag{21}$$

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 F_l, we obtain:

$$\Lambda_l^{(X)}(t)=\sum_{k\in F_l}\lambda_{k,l}^{(X)}(t) \qquad \forall l\in L \tag{22}$$

We also note that the arrival rate of ACKs $\lambda_k^{(A)}(t)$ introduced in Section II.B can easily be calculated as:

$$\lambda_k^{(A)}(t)=\mu_{k,h_j^{(k)}}^{(IN)}\left(t-d_{h_{j-1}^{(k)}}\right)+\mu_{k,h_j^{(k)}}^{(OUT)}\left(t-d_{h_{j-1}^{(k)}}\right) \qquad k=1,2,\dots K \tag{23}$$

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

$$\gamma_{k,h_j^{(k)}}\left(t+\frac{q_l(t)}{C_l}\right)=\begin{cases} \lambda_{k,h_j^{(k)}}^{(IN)}(t)\cdot p_l^{(IN)}(t)+ & \\ +\lambda_{k,h_j^{(k)}}^{(OUT)}(t)\cdot p_l^{(OUT)}(t) & \text{if } j=1 \\ \gamma_{k,h_{j-1}^{(k)}}\left(t-d_{h_{j-1}^{(k)}}\right)+ & \\ +\lambda_{k,h_j^{(k)}}^{(IN)}(t)\cdot p_l^{(IN)}(t)+ & \text{if } j>1 \\ +\lambda_{k,h_j^{(k)}}^{(OUT)}(t)\cdot p_l^{(OUT)}(t) & \end{cases} \qquad k=1,2,\dots K \tag{24}$$

Consequently, the rate of packet losses $\zeta_k(t)$ for the *k*-th group of flows, introduced in Section II.B will be:

$$\zeta_k(t)=\gamma_{k,h_j^{(k)}}\left(t-d_{h_j^{(k)}}\right) \qquad k=1,2,\dots K \tag{25}$$

The last relationship that we need to derive concerns $p_l(t)$, representing the discard probability in the generic buffer *l* at time $t\ge 0$. Applying the theorem of total probability, we can calculate $p_l(t)$ as the probability $p_l^{(IN)}(t)$ that an IN packet is dropped provided that an IN packet is arriving, multiplied by the probability that an IN packet is arriving, plus the probability $p_l^{(OUT)}(t)$ that an OUT packet is dropped provided that an OUT packet is arriving, multiplied by the probability that an OUT packet is arriving, that is:

$$p_l(t)=p_l^{(IN)}(t)\cdot\frac{\Lambda_l^{(IN)}(t)}{\Lambda_l^{(IN)}(t)+\Lambda_l^{(OUT)}(t)}+ +p_l^{(OUT)}(t)\cdot\frac{\Lambda_l^{(OUT)}(t)}{\Lambda_l^{(IN)}(t)+\Lambda_l^{(OUT)}(t)} \tag{26}$$

The way to calculate the discard probabilities $p_l^{(IN)}(t)$ and $p_l^{(OUT)}(t)$ depends on the buffer management technique adopted in the routers. As an application, we will present the expressions of $p_l^{(IN)}(t)$ and $p_l^{(OUT)}(t)$ for three different AQM mechanisms that provide service differentiation, i.e. RIO-C, RIO-DC and WRED. These mechanisms are often globally denoted as MRED (Multi-level RED) algorithms, because all of them are based on the same AQM algorithm, i.e. RED [4], and apply multiple sets of RED parameters to packets having different levels of DP in the same queue. In other words, an MRED router works as a RED router, whose discarding function depends on the type of packet arriving (IN or OUT). Moreover, different MRED algorithms calculate dropping probabilities using different measurement variables. Because all these techniques descend from RED, it is useful to recall the relationship, first derived in [12] and then upgraded in [18], that describes the time variation of the average queue length $m(t)$ estimated by this algorithm:

$$\frac{dm_l(t)}{dt}=\begin{cases} \ln(1-\alpha_l)\cdot\left(m_l(t)-q_l(t)\right)\cdot & \text{if } q_l(t)>0 \\ \cdot\left(\Lambda_l^{(IN)}(t)+\Lambda_l^{(OUT)}(t)\right) & \\ \ln(1-\alpha_l)\cdot C_l\cdot m_l(t) & \text{if } q_l(t)=0 \end{cases} \qquad \forall l\in L \tag{27}$$

where α_l represents the weight in the EWMA filter to calculate the average queue length estimated by RED.

1) Equations for RIO-C

RIO-C stands for *RED with In/Out and Coupled average queues*, and represents the traditional RIO algorithm. As already stated, RIO-C uses two different RED discarding functions, one for IN and the other for OUT packets. The first is based on the estimated average length $m^{(IN)}(t)$ of a virtual queue comprising the ordered sequence of IN packets queued in the buffer, denoted as *IN packet virtual queue*. The discarding function relating to OUT packets is, on the contrary,

based on the estimated average length, $m(t)$, of the whole buffer queue. If we assume the buffer size to be sufficient to avoid losses due to overflow, the discard probability functions $p_l^{(IN)}(t)$ and $p_l^{(OUT)}(t)$, applied by the RIO-C router if the arriving packet is marked as IN or OUT respectively, can be derived directly from the RIO-C algorithm as:

$$p_l^{(IN)}(t) = \begin{cases} 0 & \text{if } m_l^{(IN)}(t) < t_{\min_l}^{(IN)} \\ \dfrac{m_l^{(IN)}(t) - t_{\min_l}^{(IN)}}{t_{\max_l}^{(IN)} - t_{\min_l}^{(IN)}} p_{\max_l}^{(IN)} & \text{if } t_{\min_l}^{(IN)} \le m_l^{(IN)}(t) \le t_{\max_l}^{(IN)} \\ 1 & \text{if } m_l^{(IN)}(t) > t_{\max_l}^{(IN)} \end{cases} \quad (28)$$

$$p_l^{(OUT)}(t) = \begin{cases} 0 & \text{if } m_l(t) < t_{\min_l}^{(OUT)} \\ \dfrac{m_l(t) - t_{\min_l}^{(OUT)}}{t_{\max_l}^{(OUT)} - t_{\min_l}^{(OUT)}} p_{\max_l}^{(OUT)} & \text{if } t_{\min_l}^{(OUT)} \le m_l(t) \le t_{\max_l}^{(OUT)} \\ 1 & \text{if } m_l(t) > t_{\max_l}^{(OUT)} \end{cases} \quad (29)$$

where $t_{\min}^{(IN)}$, $t_{\max}^{(IN)}$, $p_{\max}^{(IN)}$, $t_{\min}^{(OUT)}$, $t_{\max}^{(OUT)}$, $p_{\max}^{(OUT)}$ are the RIO-C parameters[1].

The relationship for the calculus of $m_l^{(IN)}(t)$ can be directly derived from (27):

$$\frac{dm_l^{(IN)}(t)}{dt} = \begin{cases} \ln(1-\alpha_l) \cdot \Lambda_l^{(IN)}(t) \cdot \\ \cdot \left(m_l^{(IN)}(t) - q_l^{(IN)}(t) \right) & \text{if } q_l^{(IN)}(t) > 0 \\ \ln(1-\alpha_l) \cdot C_l \cdot m_l^{(IN)}(t) & \text{if } q_l^{(IN)}(t) = 0 \end{cases} \quad (30)$$

$$\forall l \in L$$

while the variation of the length of the IN packet virtual queue can be approximated by the following equation:

$$\frac{d\overline{q}_l^{(IN)}(t)}{dt} = \begin{cases} -\dfrac{q_l^{(IN)}(t)}{q_l(t)} C_l + \\ + \left(1 - p_l^{(IN)}(t)\right) \cdot \Lambda_l^{(IN)}(t) & \text{if } q_l(t) > 0 \\ \left(1 - p_l^{(IN)}(t)\right) \cdot \Lambda_l^{(IN)}(t) & \text{if } q_l(t) = 0 \end{cases} \quad (31)$$

$$\forall l \in L$$

[1] Note that the definition of $p_l^{(IN)}(t)$ and $p_l^{(OUT)}(t)$ can easily be extended to the gentle version of RIO [19].

2) Equations for RIO-DC

RIO-DC stands for *RED with In/Out and Decoupled average queues*. In the case of RIO-DC, the discard function relating to IN packets is based on the estimated average length, $m^{(IN)}(t)$, of the *IN packet virtual queue,* while that relating to OUT packets is based on the estimated average length, $m^{(OUT)}(t)$, of the *OUT packet virtual queue*, i.e. the ordered sequence of OUT packets queued in the buffer. With the router buffer size hypothesis adopted in Section II.C.1, we have:

$$p_l^{(IN)}(t) = \begin{cases} 0 & \text{if } m_l^{(IN)}(t) < t_{\min_l}^{(IN)} \\ \dfrac{m_l^{(IN)}(t) - t_{\min_l}^{(IN)}}{t_{\max_l}^{(IN)} - t_{\min_l}^{(IN)}} p_{\max_l}^{(IN)} & \text{if } t_{\min_l}^{(IN)} \le m_l^{(IN)}(t) \le t_{\max_l}^{(IN)} \\ 1 & \text{if } m_l^{(IN)}(t) > t_{\max_l}^{(IN)} \end{cases} \quad (32)$$

$$p_l^{(OUT)}(t) = \begin{cases} 0 & \text{if } m_l^{(OUT)}(t) < t_{\min_l}^{(OUT)} \\ \dfrac{m_l^{(OUT)}(t) - t_{\min_l}^{(OUT)}}{t_{\max_l}^{(OUT)} - t_{\min_l}^{(OUT)}} p_{\max_l}^{(OUT)} & \text{if } t_{\min_l}^{(OUT)} \le m_l^{(OUT)}(t) \le t_{\max_l}^{(OUT)} \\ 1 & \text{if } m_l^{(OUT)}(t) > t_{\max_l}^{(OUT)} \end{cases} \quad (33)$$

where $t_{\min}^{(IN)}$, $t_{\max}^{(IN)}$, $p_{\max}^{(IN)}$, $t_{\min}^{(OUT)}$, $t_{\max}^{(OUT)}$, $p_{\max}^{(OUT)}$ are the RIO-DC parameters, $m_l^{(IN)}(t)$ is calculated as in (30), while $m_l^{(OUT)}(t)$ is the average estimated length of the OUT packet virtual queue, and can easily be calculated as:

$$m_l^{(OUT)}(t) = m_l(t) - m_l^{(IN)}(t) \qquad \forall l \in L \quad (34)$$

3) Equations for WRED

WRED stands for *Weighted-RED*. In this MRED algorithm the discard functions relating to both IN and OUT packets are based on the estimated average length, $m(t)$, of the whole buffer queue. In a WRED router the discard probability functions $p_l^{(IN)}(t)$ and $p_l^{(OUT)}(t)$ are:

$$p_l^{(IN)}(t) = \begin{cases} 0 & \text{if } m_l(t) < t_{\min_l}^{(IN)} \\ \dfrac{m_l(t) - t_{\min_l}^{(IN)}}{t_{\max_l}^{(IN)} - t_{\min_l}^{(IN)}} p_{\max_l}^{(IN)} & \text{if } t_{\min_l}^{(IN)} \le m_l(t) \le t_{\max_l}^{(IN)} \\ 1 & \text{if } m_l(t) > t_{\max_l}^{(IN)} \end{cases} \quad (35)$$

$$p_l^{(OUT)}(t)=\begin{cases}0 & \text{if } m_l(t)<t_{\min_l}^{(OUT)}\\ \dfrac{m_l(t)-t_{\min_l}^{(OUT)}}{t_{\max_l}^{(OUT)}-t_{\min_l}^{(OUT)}}\,p_{\max_l}^{(OUT)} & \text{if } t_{\min_l}^{(OUT)}\le m_l(t)\le t_{\max_l}^{(OUT)}\\ 1 & \text{if } m_l(t)>t_{\max_l}^{(OUT)}\end{cases} \quad (36)$$

III. NUMERICAL RESULTS

In order to demonstrate the accuracy of the model proposed in this paper, in this section we will compare the results obtained with our model with those achieved by the ns-2 simulator. We will apply our model to the network topology presented in Fig. 1.

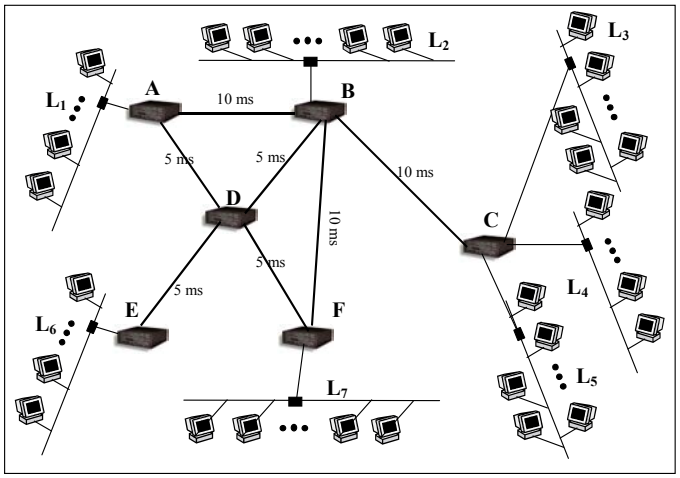


Figure 1. Network topology

We consider six MRED routers named A, B, C, D, E and F with one or more LANs directly attached to each of them. We will study three scenarios that differ as regards the MRED mechanism adopted. For the sake of simplicity we assume that in each scenario all the routers adopt the same MRED mechanism for all the queues located on the links going out of it. In particular we will refer to Scenario-1, Scenario-2 and Scenario-3 when all the routers have RIO-C, RIO-DC and WRED buffers, respectively. The other configuration parameters for each MRED router are shown in Table I.

TABLE I. 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 is loaded by 6 traffic aggregates, following the paths listed in the second column in Table II. The third and forth columns in the table give the traffic profile used for each traffic aggregate, expressed in terms of CIR and CBS.

TABLE II. DESCRIPTION OF THE TRAFFIC AGGREGATES CONSIDERED.

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

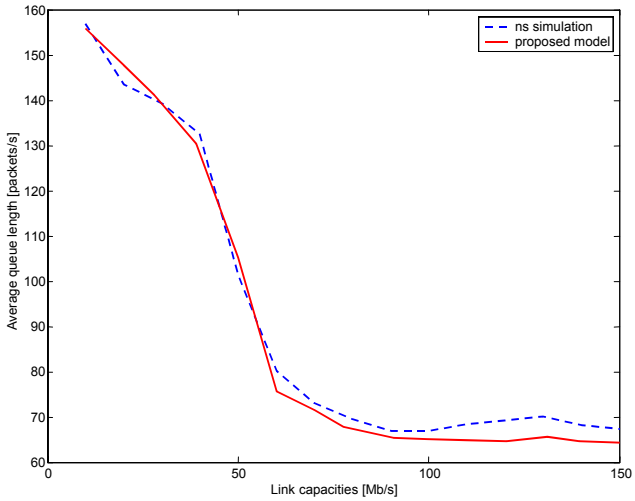
Each traffic aggregate is made up of two groups of flows: a group of greedy sources and a group of data-limited sources. We assume that the data-limited sources belonging to the same traffic aggregate start to transmit along each path at approximately the same instant, and have a file of the same size to transmit, even though these are not restrictive hypotheses. Table III gives detailed information about all the groups of flows.

TABLE III. INFORMATION ABOUT GROUPS OF FLOWS.

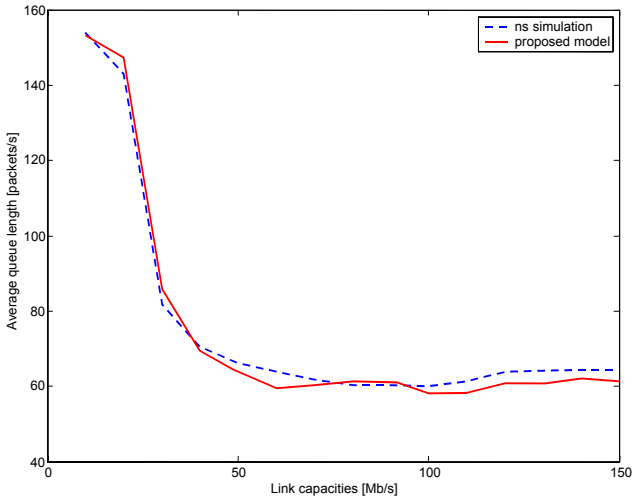
Index of group (k)	Number of flows	Traffic Aggregate	File size [packets]	Starting time [s]
1	100	A_1	greedy	0
2	40	A_1	50	20
3	100	A_2	greedy	0
4	40	A_2	30	60
5	50	A_3	greedy	0
6	20	A_3	100	70
7	100	A_4	greedy	0
8	20	A_4	50	50
9	50	A_5	greedy	0
10	40	A_5	1000	40
11	50	A_6	greedy	0
12	20	A_6	500	30

The first analysis of the network shown in Fig. 1 was carried out assuming that all the links had the same capacity, and considering a variation range between 10 Mb/s and 150 Mb/s, which correspond, if the packets are considered to have a fixed size of 1000 bytes, to a capacity of 1250 packets/s and 18750 packets/s, respectively. The results given by our model are shown in Figs. 2, 3 and 4 where they are compared with those obtained by the ns-2 simulator.

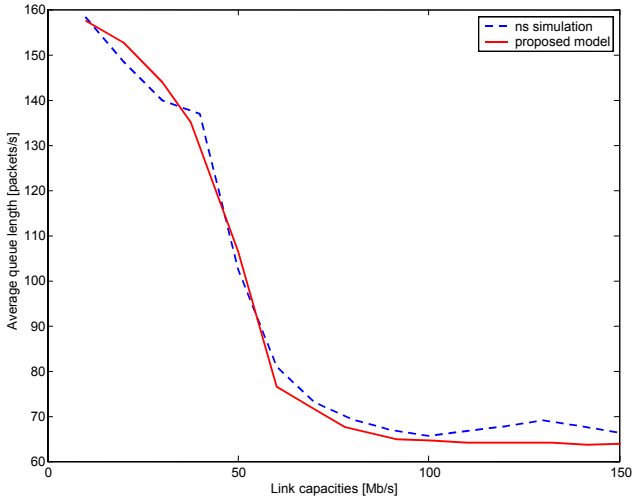
Fig. 2 shows the average queue lengths only in the two buffers representing bottlenecks for the network, as the analysis using both ns-2 and our model shows that the queues in the



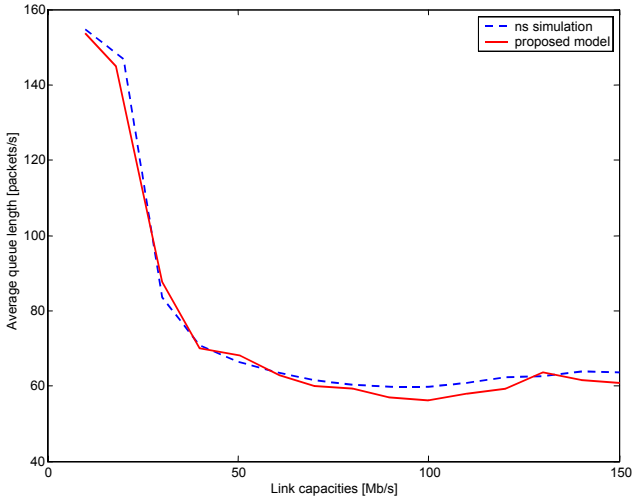
(a) Average queue length of the router B output buffer towards router C – scenario 1



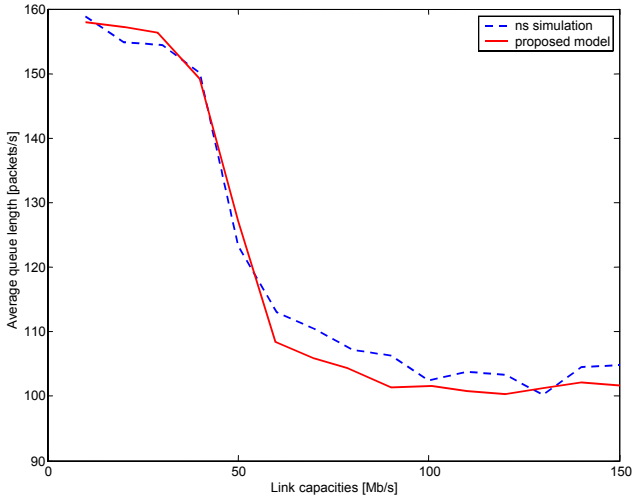
(b) Average queue length of the router D output buffer towards router F – scenario 1



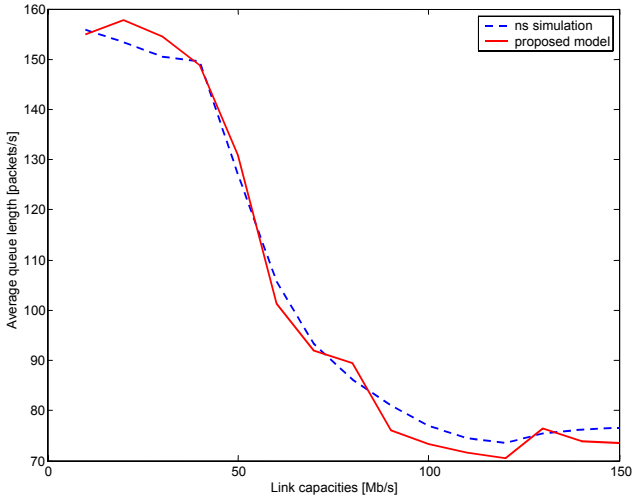
(c) Average queue length of the router B output buffer towards router C – scenario 2



(d) Average queue length of the router D output buffer towards router F – scenario 2

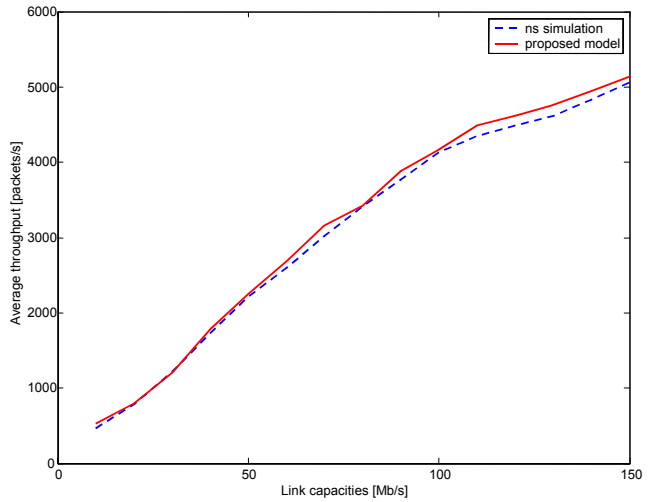


(e) Average queue length of the router B output buffer towards router C – scenario 3

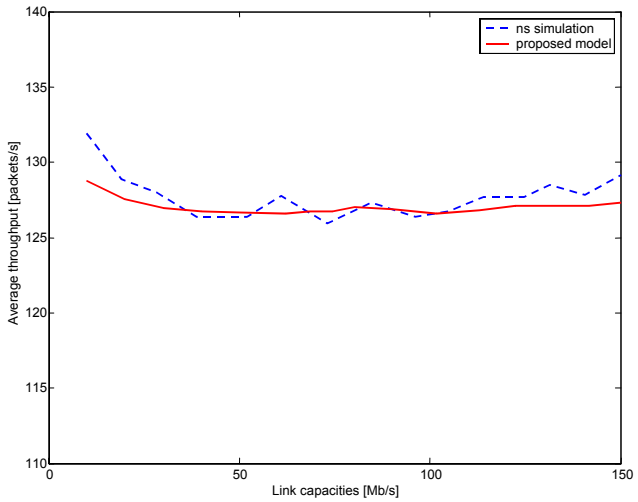


(f) Average queue length of the router D output buffer towards router F – scenario 3

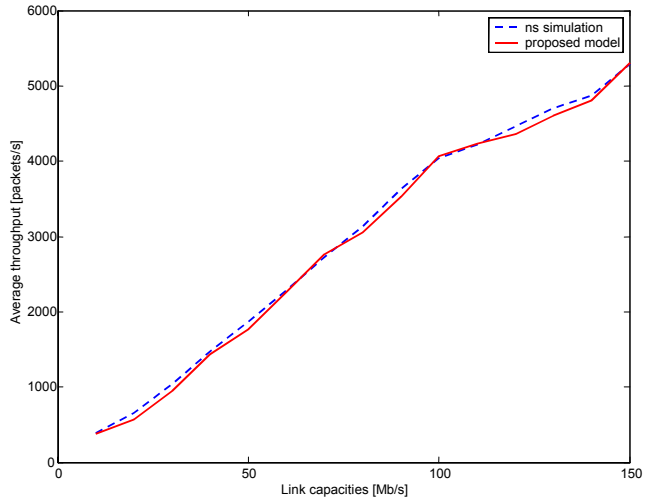
Figure 2. Average queue length : comparison between simulation and proposed model



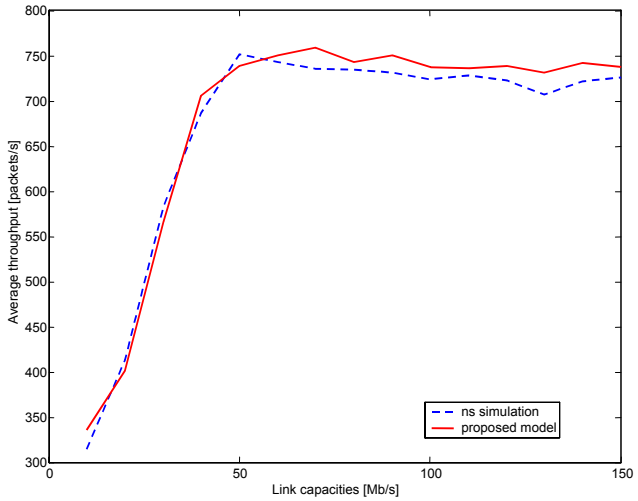
(a) Average throughput for greedy sources belonging to traffic aggegate A_1 – scenario-1



(b) Average throughput for data-limited sources belonging to traffic aggregate A_1 – scenario-1



(c) Average throughput for greedy sources belonging to traffic aggegate A_5 – scenario-1



(d) Average throughput for data-limited sources belonging to traffic aggregate A_5 – scenario-1

Figure 3. Average throughput: comparison between simulation and proposed model

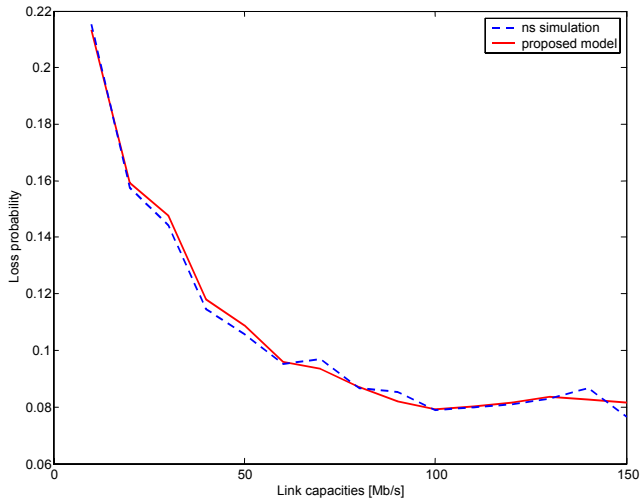
other buffers are practically empty throughout the observation period. As can be seen in Fig. 2, our model captures the average queue length in both buffers and in the three scenarios quite well. We have not plotted the results for the three scenarios in the same graph because the three curves would almost overlap and the figures would be unreadable. This means that in the network topology and for the TCP traffic load that we have considered (Table III) network performance does not depend on the AQM mechanism. However, useful information can be extracted from Fig. 2: in all three scenarios the average queue length tends to become stable if the link capacity is increased. This means that, by increasing the link capacity, there are no more improvements in the average RTT, but only in the throughput assigned to the sources.

Fig. 3 compares the average throughput for some traffic aggregates in the three different scenarios, while Fig. 4 shows the average loss probability suffered by a generic TCP source belonging to these traffic aggregates. We only plot the results obtained for Scenario 1 (WRED routers) because the other two scenarios gave similar results.

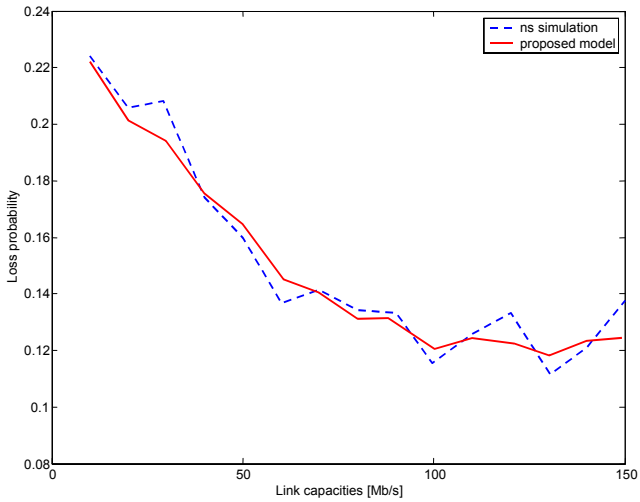
These results also demonstrate the accuracy of our fluid model in predicting the average throughput and loss probability of TCP sources in the network.

As expected, the throughput of the sources grows with link capacity, while the loss probability decreases. We note that the flows belonging to the same traffic aggregate, both greedy and data-limited sources, experience approximately the same loss probability, because they follow the same path in the network.

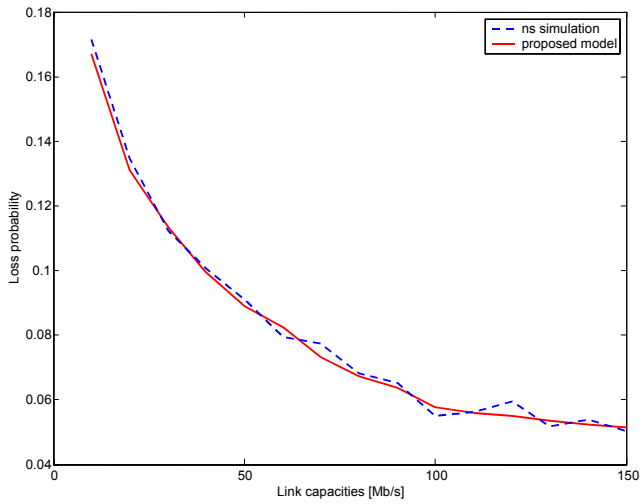
For the sake of conciseness, we have not presented the results for the other traffic aggregates, but we found as good as a match as the one that we present in Figs. 3 and 4.



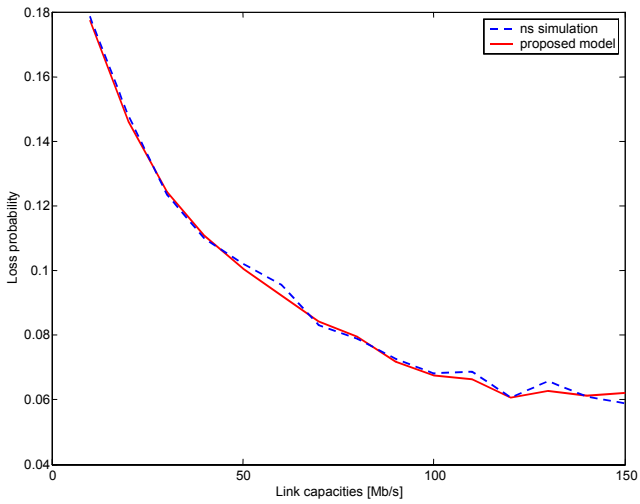
(a) Average loss probability for a greedy source belonging to the traffic aggegate A_1 – scenario-1



(b) Average loss probability for a data-limited source belonging to the traffic aggregate A_1 – scenario-1



(c) Average loss probability for a greedy source belonging to the traffic aggegate A_5 – scenario-1



(d) Average loss probability for a data-limited source belonging to the traffic aggregate A_5 – scenario-1

Figure 4. Average loss probability: comparison between simulation and proposed model

IV. CONCLUSIONS

In this paper we have constructed an accurate fluid model for TCP sources which are not necessarily greedy, also taking the Slow-Start phase into consideration. These characteristics permit us to study the effects of short connections on network performance. We have considered TCP flows in a DiffServ network where the nodes adopt AQM mechanisms to guarantee service differentiation and have compared the results given by the model with those obtained via simulation, obtaining a good match. 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. For the topology in Fig. 1 the computation time is less than 3 minutes using a normal 1-GHz PC Pentium III, while a ns-simulation takes about 15 minutes (to obtain average values for all the metrics considered, we ran 30 ns-simulations for each case). For this reason the proposed model could be used to find the optimal network parameter configuration for different traffic conditions.

REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, "An architecture for differentiated services". *RFC 2475*, December 1998.

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

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

[4] S. Floyd, V. Jacobson. "Random Early Detection Gateways for Congestion Avoidance". IEEE/ACM Transactions on Networking (1993).

[5] D.D. Clark and W. Fang, “Explicit allocation of best effort packet delivery service”, IEEE/ACM Transactions on Networking, August 1998.

[6] N. Seddigh, B. Nandy, P. Pieda, J. Hadi Salim, A. Chapman, “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] A. Abouzeid, S. Roy, “Modeling random early detection in a differentiated services network”, Computer Networks (Elsevier), 40(4): 537-556, November 2002.

[9] N. Malouch, Z. Liu, “On steady state analysis of TCP in networks with Differentiated Services”, Proceedings of Seventeenth International Teletraffic Congress, ITC'17, December 2001.

[10] I. Yeom, A. Reddy, “Modeling TCP behavior in a differentiated-services network”, IEEE/ACM Transactions on Networking, February 2001.

[11] Y. Chait, C. Hollot, V. Misra, D. Towsley, and H. Zhang, "Providing throughput differentiation for TCP flows using adaptive two color marking and multi-level AQM," IEEE INFOCOM 2002, New York, NY, 23-27 June 2002.

[12] V. Misra, W. Gong, D. Towsley, “Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED”. SIGCOMM ’00 (August 2000).

[13] L. Guo, I. Matta, “The War Between Mice and Elephants”, ICNP 2001, Riverside, CA, November 2001

[14] J. Ibanez, K. Nichols, “Preliminary Simulation Evaluation of an assured service”, INTERNET DRAFT draft-ibanez-diffserv-assured-eval-00.txt, August 1998.

[15] The network simulator – ns-2. LBL, URL: http://www.isi.edu/nsnam/ns/

[16] S. Floyd, M. Handley, J. Padhye, J. Widmer, “Equation Based Congestion Control for Unicast Applications”, SIGCOMM 2000, August 2000

[17] S. Floyd, T. Henderson “The NewReno Modification to TCP’s Fast Recovery Algorithm”. RFC 2582 (April 1999).

[18] M.Barbera, A. Lombardo, G. Schembra “A Fluid-Based Model of Time-Limited TCP Flows”, appearing on “Computer Networks”

[19] S. Floyd., “Recommendation on using the “gentle_” variant of RED”. http://www.aciri.org/floyd/red/gentle.html (March 2000)