# Worst-Case Analysis for Deterministic Allocation in a Differentiated Services Network

Marco Listanti (*), Fabio Ricciato (*), Stefano Salsano (**), Luca Veltri (**)

(*) INFOCOM Dept., University of Roma "La Sapienza"
(**) CoRiTeL - Consorzio di Ricerca sulle Telecomunicazioni

## Abstract

The Differentiated Service architecture is proposed as a scalable approach to QoS for IP networks. Therefore it is based on the aggregate (per class) scheduling of packets, but it aims at providing QoS to single flows. In particular the Expedited Forwarding (EF) Per Hop Behavior (PHB) and the related Premium Service have been defined in order to provide determinist QoS guarantees to IP flows: zero loss and very low delay and jitter. Hence a possible approach to characterize and to dimension a network using the EF PHB is the worst case analysis. In this work we propose a worst case analysis which provides bounds to the queuing delay for a class of network topologies. Our results are compared with similar available results, which provide "looser" bounds. The comparison with simulation results shows that the bound is not enough tight to be of practical use. We justify the reason for this behavior and indicate directions for further improvements.

## 1. Introduction

The Differentiated Services (Diffserv) model has been proposed within the IETF [1] to be a scalable solution to provide end-to-end Quality of Service (QoS). It defines a set of packet forwarding behaviors (called per-hop behaviors or PHBs) and provides a sort of service differentiation for large aggregates of traffic on the basis of few classes. With the exception of policing and shaping at network boundaries, the only actions needed to be handled in the forwarding path are the packet classification into one or few queues and the packet scheduling according to proper rules.

Among the proposed PHBs, the EF PHB ([3]) should provide no loss, low latency and jitter and could be used to build a low delay, assured bandwidth end-to-end service. Such a service appears to the endpoints like a point-to-point connection or a "Virtual Leased Line" (VLL). This service has also been described and defined as Premium service ([2]).

According to [2], premium traffic should be limited and shaped to a contracted peak-rate and packets should move through the network with almost no queuing delay. To build such a service, two components are required:

- nodes must be configured so that the aggregate flows have a minimum service rate independent of the dynamic state of the node;
- the input flows should be limited and shaped to a contracted peak-rate at the network boundaries so that the aggregate arrival rate at any node is always less than a configured service rate.

While the first point can be solved by the guarantee of an appropriate per-hop-behavior (like the EF PHB), the latter requires additional functions like flow admission and set-up procedures and network dimensioning criteria to allocate network resources and to configure boundary nodes.

It is not clear which criteria can be used to correctly determinate the bandwidth that should be allocated to accommodate all individual flows in order to provide the requiring end-to-end QoS. It is a common belief ([4]) that in order to provide a Premium service only a small percentage of the total network capacity should be allocated, but how much this percentage must be low to keep the queuing delay bounded is not clear yet. Therefore, the open point is to define practical criteria to allocate network resources, i.e. bandwidth and buffers.

An approach commonly adopted consists in the study of the end-to-end network performance in order to determine upper bounds for packet delay on the basis of a worst case analysis. Unfortunately, up to now definitive results in this direction are not yet available.

The goals of this paper are:

- to briefly review the more recent contributions about worst case analysis of packet delay in a Diffserv network, considering the applicability limitations;
- to propose a novel approach aiming at partially overcoming such limitations;
- to compare the worst case analytical results with those arising from simulation of a network topology reproducing an actual IP backbone.

It is to be observed that, although our approach leads to results closer to those arising from practical cases, it still actually provides an "upper bound" of the worst case of packet delay. In some topologies such a bound can be much higher than the actual worst case, so it would lead to a great inefficiency in bandwidth usage if it were used as an allocation rule. In the final part of the paper we discuss the reason of this discrepancy and we give some guidelines to remove such an approximation.

In Sec. 2, the definition of the EF PHB and the relationship between the EF PHB and the "Premium service" is presented and the analytical approaches available in literature for the evaluation of the worst case delay are discussed. In Sec. 3 and 4 our approach is presented and results are discussed. Finally, Sec. 5 deals with the guidelines for future research and investigations in the field of worst case analysis.

## 2. EF PHB definition and state of art of "worst case delay" analysis

The EF PHB is defined as a forwarding treatment where the departure rate of the aggregate EF packets must equal or

This work has been partly supported by the European Commission under projects ELISA (ACTS AC310) and AQUILA (IST-1999-10077)

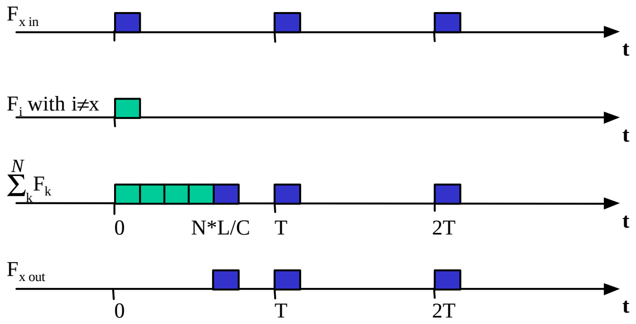

*Fig. 1 - Flow $F_x$ characterization on the 1-link.*

exceed a configurable rate (i.e. a fraction of the link bandwidth). This constraint must apply within a very short time interval, equal to the time it takes to send an MTU packet at the configured rate. Therefore if the link is shared with other aggregates, the configured rate cannot exceed the 50% of the output link bandwidth.

The RFC 2598 [3] describes a Premium service, which aims to provide no loss and negligible delay and jitter. It is composed of two logical components: 1) the EF PHB; 2) traffic conditioning at network edge (and resource allocation).

The idea for traffic conditioning at network edge is to control the input traffic so that “at every transit node, the aggregate maximum arrival rate is less than the aggregate minimum departure rate”. This condition should guarantee that no queuing is needed at interior nodes. The problem is that the concept expressed in the previous definition is very loose. The RFC 2598 does not provide additional information on allocation strategies and on buffer dimensioning.

One should answer the following two questions: i) how much traffic can be admitted at network edge and at a generic network link? and ii) which is the needed buffer size in a Diffserv router?

The responses to the previous questions should take into account the sequence of “hops” that are crossed by the traffic, because, due to the aggregate scheduling, the traffic characteristics of the input flows are modified by the multiplexing with other flows on the output links on each crossed hop. We refer to this problem as “multi-stage” analysis.

We give a practical example to clarify the degradation of the traffic characteristic of a flow in a the multiplexing process. Assume that a set of input flows is policed (i.e. by a token bucket) at the peak rate. They of course are “allowed” not to emit packets at a given time. We denote this flow as Peak Rate Regulated (PRR) flows. Let us consider (see Fig. 1) a set of PRR flows entering a first stage multiplexer, we want to compare the arrival pattern of a given flow $F_x$ on its input link with the departure pattern of the same flow on the exit link. We assume that $N$ flows, coming from different input links, are multiplexed on the output link. $N$ packets can arrive in the same time, so that they will be queued in the buffer for transmission on the output link (generating a burst of $N$ packets). At this point, it can happen that $N$-$1$ flows become silent and only one flow ($F_x$) emits a packet at its peak rate. Therefore a packet of the flow $F_x$ experiences a “large” delay while the following packet of the same flow experiences no delay and it results in a in a narrowing of the inter-departure time of packets of the same flow. One can evaluate the worst case departure pattern of a given flow taking into account the maximum delay in the multiplexer buffer. The worst-case combination of these patterns at the next multiplexing stages must be taken into account to evaluate the maximum packet bursts the buffer requirements and consequently the maximum delays.

The problem of the multistage analysis is dealt with in some recent contributions. In [5] a discussion on buffer requirements is given and the maximum burst of packets that can appear on a generic link is evaluated. The maximum burst length in a network link outgoing from a router is simply evaluated by adding the maximum burst lengths appearing in the input links. This procedure is iterated for any stage. Assuming a fan-in (i.e. number of input lines) equal to $i$ and assuming a perfectly regulated traffic on the input lines (burst length=1), the burst length $b$ at stage $h$ will be given by: $b(i,h)=i^h$. Considering that such a burst is composed of packets belonging to different flows, the burst length would be limited by the number $N$ of different flows that share a link. This procedure for aggregating the burst lengths is not general, as it does not take into account the worst case combination of arrival patterns that can happen when a subset of the flows sharing a link is extracted at the next multiplexing stage.

A more general evaluation of delay bounds for aggregate traffic applicable to EF PHB is provided in [6], where some mechanisms leading to the increase of burst length (and delay) are described. As pointed out in [7], the bound found by [6] applies to a particular class of topologies, satisfying the so called "monotonic degree" constraint. This class of networks is derived on the basis of the definition of *degree* of a link as the maximum number of hops that have been crossed by the flows sharing the link. A network satisfies the monotonic degree constraint if the set of flows that share a $j$ degree link has only crossed links with a degree lower that $j$.

There are many points of contact between this paper and [6]; the goal of both papers is to provide a bound on the delay on one hop and to use this delay to bound the increase of flow burstiness on the output link. While [6] bounds the delay in a generic n-hop node of the network given the maximum load on any link, we provide a methodology to evaluate the maximum delay at each single node given the network topology, the link line rates, the set of input flows, the actual packet size. The main difference is that we also take into account the finite speed of input lines. This allows us to give a tighter upper bound to the maximum delay.

## 3. Worst case analysis

Our aim is to evaluate the minimum buffer requirements for EF in a cascade of multiplexer ensuring deterministic zero loss. We assume that each EF flow arrives at the 1st node (ingress stage) from a separate input line. At the output link of each stage a certain number of flows is directed to the successive multiplexer while the others exit the cascade. Fixed EF packet size ($L$ bits) is assumed. We consider a homogeneous scenario in the sense that, for the generic $k$

stage, all the EF input flows are assumed to have the same characteristics. The generic EF source flow is only characterized by its peak emission rate, as any statistical characteristic is left out of consideration in a worst case analysis. We further assume that at each node EF packets are served with non-preemptive priority over non-EF ones. $MTU_{nonEF}$ denotes the maximum size of non-EF packets.

As for the first stage, it is easy to show that the maximum EF buffer occupancy is originated by the superposition of all synchronized (zero-phased) flows. In this case the maximum EF buffer occupancy at the first stage is given by:

$$B_{max,1} = N_1{}^{x}L \qquad \textit{eq. 1}$$

where $N_1$ is the number of input flows.

At the successive stages the evaluation of the worst case EF delay is somewhat more complex, as the packet clumping phenomenon introduced by previous multiplexing stages has to be taken into account. The minimum inter-arrival time between successive packets of the same flow at the $k$ stage can decrease with $k$, i.e. the instantaneous peak rate get larger. This can lead to an increase of buffer occupancy (and queuing delay) for EF. As strict priority scheduling is enforced between EF and non-EF packets, the maximum queuing delay for EF packets at stage 1 is given by:

$$D_{\max 1} = \frac{B_{\max 1} - L}{C_1} + \frac{MTU_{nonEF}}{C_1} \cong \left(B_{\max 1} + MTU_{nonEF}\right)/C_1 \qquad \textit{eq. 2}$$

That is the time spent to send all previous EF packets plus the time need for a non-EF packet that started being served before the first EF packet arrived.

Once the maximum delay at the 1st stage $D_{max,1}$ has been evaluated from $B_{max,1}$ by means of eq. 2, we can take it into account in the characterization of the worst case arrival pattern at the successive stage in order to evaluate the maximum buffer occupancy $B_{max,2}$ and the maximum queuing delay $D_{max,2}$ occurring at the 2nd stage. The parameters of the 2nd stage multiplexer (fan-in, link capacity) must be considered as well. The procedure can be iterated for the successive stages: at the generic $k$ stage the sum of maximum delays of previous stages is used to evaluate $B_{max,k}$ and $D_{max,k}$.

This procedure is represented by the formulas of eq. 3, which express $B_{max,k}$ and $D_{max,k}$ as functions $\Re_B(\cdot)$ and $\Re_D(\cdot)$ of the maximum delays at previous stages and of the parameters of the $k$ multiplexer. The evaluation of $\Re_B(\cdot)$ and $\Re_D(\cdot)$ is the goal of the following section.

$$\begin{aligned} B_{\max,k} &= \Re_B\left(\sum_{i=1}^{k-1} D_{\max,i}\,, \text{parameters of k - mux}\right) \\ D_{\max,k} &= \Re_D\left(\sum_{i=1}^{k-1} D_{\max,i}\,, \text{parameters of k - mux}\right) \end{aligned} \qquad \textit{eq. 3}$$

### *3.1. The analytical model*

Consider the multiplexer model depicted in Fig. 2. Before entering the multiplexer each single source flow crosses a module introducing a variable delay on its packets. Such a delay is assumed to be generally distributed and deterministically bounded by $D$ (sec). These modules are representative of the network section before the multiplexer, in other words they take into account the jitter introduced by previous stages. $D$ is then the maximum delay budget the flows can cumulate in the $k$-$1$ previous stages. Fig. 2 also shows the worst-case arrival pattern at the multiplexer, i.e. the one that originates the maximum buffer occupancy. It is given by the zero-phased superposition of all "per-flow worst patterns". The single per-flow worst pattern is built using the maximum delay budget $D$ to gather as many packets as possible in front of the burst. Given such a worst pattern and given the parameters of the multiplexer (fan-in, link capacity), our objective is now to find the minimum buffer size required to have deterministic zero loss, i.e. the maximum buffer occupancy at the multiplexer. Let us now present the adopted notation:

- $L$: the packet size (bits);
- $C$: the output link capacity;
- $P$: the source peak emission rate;
- $T=L/P$: the minimum inter-packet time of the source;
- $N$: the number of input flows;
- $M$: the number of input lines;
- $C_{in,tot}$: total input capacity, i.e. $M$ times the single input line capacity;
- $B_{max}$: a bound to the max buffer occupancy at the mux;
- $D_{max}$: a bound to the max queuing delay at the mux.

It is assumed that $C > N\cdot\frac{L}{T} = N\cdot P$, which is the condition for peak rate allocation. Let us further define the following quantities that characterize the "per-flow worst-case arrival pattern" at the multiplexer:

- $h = \left\lfloor \frac{D}{T} \right\rfloor + 1$: max number of packets in the front burst;
- $g = T\cdot h - D$: time lag between the front burst and the successive arrival.

The meaning of $h$ and $g$ is evidenced in Fig. 2.

We will initially assume that each single flow enters the multiplexer from separate input lines ("single line per flow" condition). This case is equivalent to consider infinite capacity input lines ($C_{in,tot}$= ¥). By composing all the single per-flow worst arrival patterns under the single-line-per-flow condition we derive the worst case arrival curve $v=v_{\yen}$ depicted in Fig. 3.

Successively, the effect of the finite capacity of the input lines is taken into account, and the worst case arrival curve is modified accordingly. The curve $v_{\yen}$ as evaluated in the single-line-per-flow condition still represents an upper bound for the arrival curve when $C_{in,tot}$ is finite, and drives the analysis in this last case.

Fig. 3 shows the arrival ($v$) and service ($s$) curves vs. time for the multiplexer of Fig. 2 in the single line per flow condition. The amount of EF bits which enter ($v$) and leave

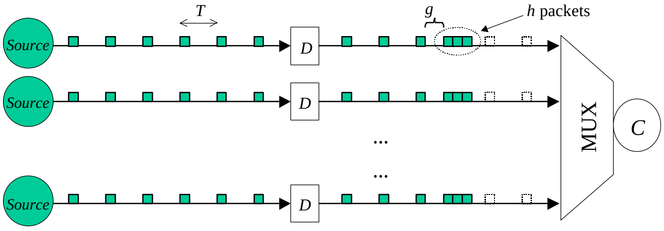

*Fig. 2 – Conceptual scheme for the analytical model.*

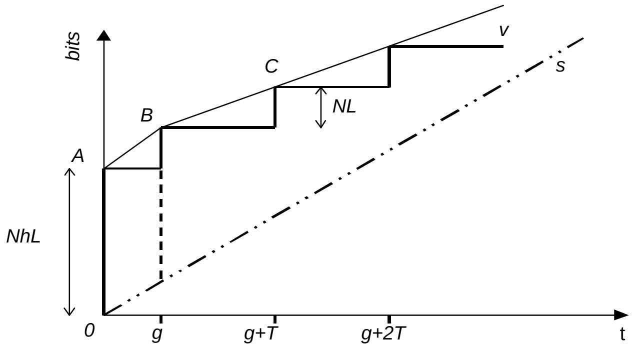

*Fig. 3 - Arrival/departure curves for the 'single line per flow' case ($C_{in}$ = ¥).*

(*s*) the multiplexer are in abscissa, while time is in ordinate. At any time the instantaneous buffer size is given by the vertical distance *v*(t)-*s*(t), while the maximum delay is given by the maximum horizontal distance between *v*(t) and *s*(t). In the case depicted in the figure the EF service curve starts a time *t=0* (i.e. *s(t): $C \times t^+$*), which means that no non-EF packets are being transmitted when the first bundle of EF packets arrive. To be rigorous, the EF service curve in the worst case should start at time $t=MTU_{nonEF}/C$ in order to account for the conflict with a MTU sized non-EF packet. This would add complexity to the analysis, as the service curve becomes concave. In order to simplify the analysis, we consider an EF service curve starting at *t=0* as in Fig. 3, and we account for the effect of the conflict with a MTU sized non-EF packet by the additive terms $MTU_{nonEF}$ and $MTU_{nonEF}/C$ in the final expressions of $B_{max}$ and $D_{max}$ respectively. In other words we will consider:

$$\begin{aligned} B_{\max} &= B^*_{\max} + MTU_{nonEF} \\ D_{\max} &= D^*_{\max} + MTU_{nonEF}\ /C \end{aligned} \qquad \textit{eq. 4}$$

with $B^*_{max}$: and $D^*_{max}$ evaluated considering the EF service curve *s(t): $C \times t^+$*. This simplification will not impact the considerations we are going to derive from the analysis.

$B^*_{max}$: and $D^*_{max}$ are related by eq. 5, so in the next section we focus on the evaluation of $B^*_{max}$ only:

$$D^*_{\max} = \frac{B^*_{\max} - L}{C} \cong B^*_{\max} \big/ C \qquad \textit{eq. 5}$$

### *3.2. Bounds for the maximum queue size and queuing delay at a generic stage*

Returning to Fig. 3, at *t*=0 a number of $N \times h$ packets arrive instantaneously at the multiplexer, followed at time *g* by *N* more arrivals (point B). At time *t*=0 packets begin to be served at rate *C* (service curve *s(t)*). It can be shown from the figure that the maximum buffer occupancy can occur at time $t_1=0$ or $t_2=g$ according to the value of the output capacity *C*:

$$B^*_{\max} = \max\{N \cdot h \cdot L, N \cdot (h+1) \cdot L - C \cdot g\} \qquad \textit{eq. 6}$$

Further on we will modify the worst arrival curve *v* to take into account that the *N* input packet flows enter the multiplexer from *M<N* input lines, with a total input capacity $C_{in,tot}$ which is finite. Note that the maximum vertical step that can be found in the arrival curve *v* is given by $M \times L$, as *M* is the maximum number of packets that can instantaneously arrive at the multiplexer from the *M* input lines. To be rigorous the arrival curve should appear like the dotted-line curve in Fig. 4, as packets are assumed to arrive to the multiplexer once they are completely stored in the buffer, i.e. the packet arrival time is concentrated in the last bit arrival instant. The finite input capacity implies that there is a minimum interval ($=M \times L/C_{in,tot}$) between successive input packets. In the following analysis for sake of simplicity the arrival envelope evidenced in Fig. 4 by continuos line will be used in place of the exact curve. This will lead to a fair over-estimate of the maximum buffer occupancy $B_{max}$, so that the symbol ≅ will be used in place of equality.

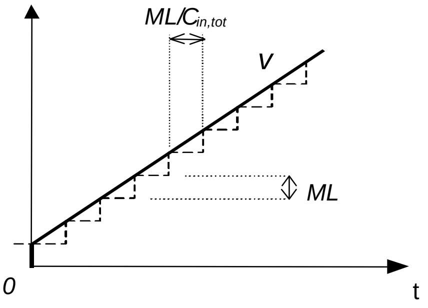

*Fig. 4 - Exact arrival curve and its approximating envelope.*

In order to achieve a precise evaluation of the maximum buffer occupancy at the multiplexer two different cases must be considered according to the input capacity $C_{in,tot}$.

*Case 1:* $C_{in,tot} \le C$.

This case is trivial: the maximum buffer occupancy reduces to a number of packets equal to the number of input lines, i.e.

$$B^*_{\max} = M \cdot L \qquad \textit{eq. 7}$$

*Case 2:* $C_{in,tot} > C$ ;

In this case, depicted in Fig. 5, the maximum buffer capacity can hold at time $t_1^*$ or $t_2^*$, depending on the value of *C*. $t_1^*$ and $t_2^*$ represent respectively the first and second intersection time between the curves *v(t)* and *v*¥*(t)* and can be derived from the following equations:

$$M \cdot L + C_{in,tot} \cdot t_1^* = N \cdot L \cdot h + N \cdot L \cdot \left\lfloor \frac{t_1^* + T - g}{T} \right\rfloor \qquad \textit{eq. 8}$$

$$\begin{aligned} &M \cdot L + C_{in,tot} \cdot (t_2^* - t_x) = N \cdot L \\ &\textit{with } t_x = \max\left\{ g + T \cdot \left\lfloor \frac{t_1^* - g}{T} \right\rfloor, t_1^* + \frac{M \cdot L}{C_{in,tot}} \right\} \end{aligned} \qquad \textit{eq. 9}$$

From eq. 8 and eq. 9 the maximum buffer occupancy is evaluated as follows:

$$B^*_{\max} = \max\{B(t_1^*), B(t_2^*)\} \qquad \textit{eq. 10}$$

with:

$$\begin{aligned} B(t_1^*) &\cong M \cdot L + (C_{in,tot} - C) \cdot t_1^* \\ B(t_2^*) &\cong B(t_1^*) + N \cdot L - C \cdot (t_2^* - t_1^*) \end{aligned} \qquad \textit{eq. 11}$$

Note that this case includes the 'single line per flow' case as a limit, because $t_1^* \to 0$ and $t_2^* \to g$ for $C_{in,tot} \to \infty$.

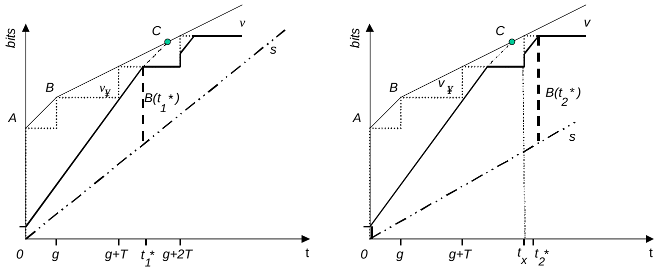

*Fig. 5 - Arrival/departure curves for case $C<C_{in,tot}<$¥.*

Using some algebra, eq. 6, 7, 10, 11 can be covered by the following eq. 12, where $[x]^+ = \max(x,0)$. The eq. 8 and eq. 9 are still needed to evaluate $t_1^*$ and $t_2^*$.

$$B_{\max}^* = \begin{cases} M\cdot L & C_{in,tot} \le C \\ M\cdot L + (C_{in,tot} - C)\cdot t_1^* + \left[N\cdot L - C\cdot(t_2^* - t_1^*)\right]^+ & C < C_{in,tot} < \infty \quad \textit{eq. 12} \\ N\cdot h\cdot L + [N\cdot L - C\cdot g]^+ & C_{in,tot} = \infty \end{cases}$$

The eq. 4, eq. 5 and eq. 12 exactly define the functions $\Re_B(\cdot)$ and $\Re_D(\cdot)$ introduced in eq. 3.

## 4. Numerical Results

The worst case delay bound found in the previous section has been computed for a sample network scenario. Such a bound has been compared with the bound found in [6, eq.34]. It has also been compared with simulation results in order to investigate the distance between the analytical bound and the actual system behavior. The considered topology is depicted in Fig. 6. It is a three-levels hierarchical symmetric network and is aimed at representing a structured Diffserv domain with both an external and a core section.

Traffic sources are connected by dedicated links to the Edge Routers (ER), which in turns are connected to their respective Core Router (CR) by links of capacity $C_1$ = 10 Mbit/s. The capacity of core links is $C_2$ = 30 Mbit/s. The considered source flows have a peak rate $P$ = 64 Kbit/s and packet size $L$ = 576 bytes. The size of the queues is infinite. The considered traffic matrix is symmetric and originate homogeneous EF traffic load on each link. Each traffic flow enters the network from its ingress ER and crosses 5 multiplexing stages (1 ER + 4 CR) before getting to its egress ER.

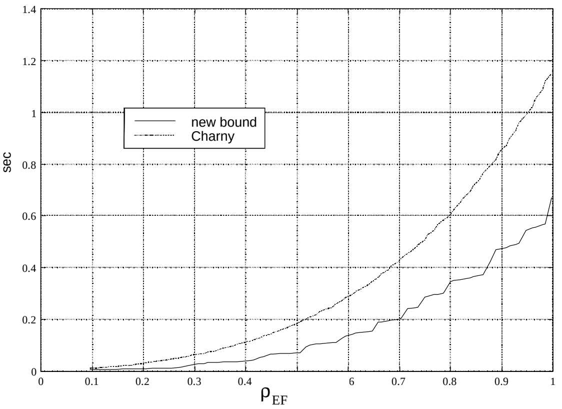

*Fig. 7 - Analytical delay bound at 5th stage.*

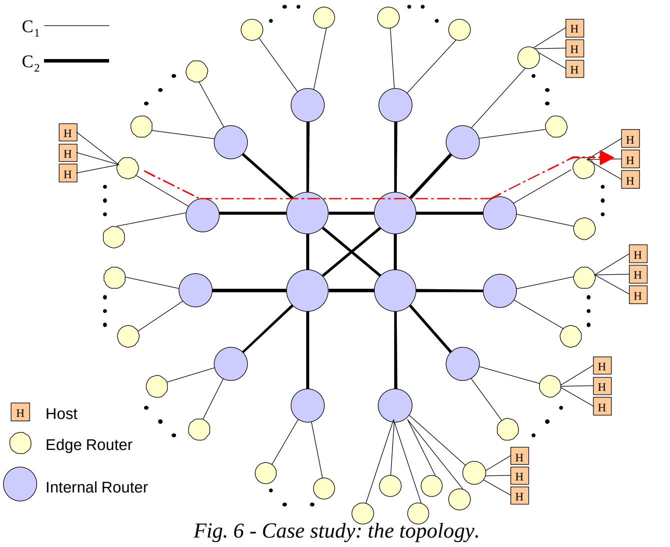

*Fig. 6 - Case study: the topology.*

Simulations have been run by Network Simulator [8] in order to compare the analytical bounds with the ”actual” system behavior. In the simulations markovian on-off sources with activity $a$ = 0.9 and active/idle average time of $T_{on}$ = 1.35 sec and $T_{off}$ = 0.15 sec have been considered.

Fig. 7 shows our analytical bound (*continuos line*) for EF queuing delay vs. EF load for the 5th multiplexing stage, i.e. at the last Core Router along the flow paths. The ratio between the EF peak rate and the link capacity ($r_{EF}=N{\times}P/C$) is the same on both external and internal links. It is varied by varying the number of flows N on each link. In Fig. 7 the bound given in [6, eq. 34] is also depicted (*dotted line*). It is evident that both curves sharply increase with the load. As expected, our bound remains below the Charny’s one. In facts at each stage we take into account the effect of the finite capacity of the input links.

In Fig. 8 our analytical bound for queuing delay has been compared with the 99.99 percentile and the maximum observed in simulation, over a number of packets ≅$10^7$. It is evident that the experienced delay is much less than the analytical bound in the whole considered load range. In particular the empirical delay slowly increases with the load.

We observe that even if our bound is a good refinement of

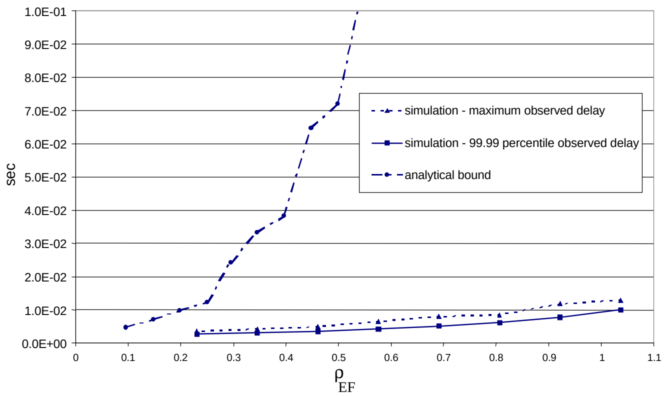

*Fig. 8 - Queuing delay at 5th stage: analytical bound and simulation results.*

the Charny’s one, it is still very conservative. Therefore, it is open whether these worst case bounds are suitable to be used as the basis of an effective allocation approach. In the next section we discuss this issue and we propose directions for further improvements of worst case bounds.

## 5. Final considerations

The goal of this section is to discuss the reasons why the experimental worst case is so far from the theoretical upper bound derived here. Two elements have to be taken into account

- The first element is based the consideration that the probability of the worst case could be so low that the worst case event can be neither achieved nor approximated with finite simulations; we are of course aware of this, this is an intrinsic limitation of the worst case approach and represents the price that we have to pay to have the zero loss guarantee.fgh
- The second element is that the evaluated bounds could be too “loose” failing to model some important aspects; we want to consider more in detail this aspect.

The described analyses, the one presented in this paper as well as the one proposed in [6], use the delay on one node to independently bound the burstiness of each single flow on the output link. The goal is to evaluate a sort of worst case arrival pattern, which is needed to evaluate the worst case delay at the next multiplexing stage. If a bundle of *n* flows on a given link are directed to a specific output link of the next multiplexing stage, the worst case arrival pattern is basically assumed as the “superposition” of the worst case arrival patterns of the *n* flows considered independently (as if they were transmitted over different links). In a realistic scenario, several flows from each input line are multiplexed in one output line. The worst case arrival pattern of such bundle of input flows is quite different (let say lower) of the "sum" of the worst case arrival patterns of a single flows.

Taking this fact into consideration, we see that the described approach yields an upper bound tight to the worst case if, at each multiplexing stage, only one flow from each input line is taken as input to the next output link. As long as more flows are extracted from each input line to be multiplexed in the output lines the bound becomes looser.

Let us consider a generic node with *M* input lines and let us suppose that *n* flows of each input line are multiplexed onto the same output line; the total number of flows on that output link is then $N=M\cdot n$.

Fixed the value of *N*, the distance between our bound and the real worst case depends on the values of *n* and *M*. When *n=1* (*M=N*), i.e. in the single line per flow assumption, our bound tightly approximates the real worst case, as there is no correlation between input flows. When $1<n<N$, the effect of correlation between the flows coming from the same input line is not accounted for in the analysis. As a result, the bound overestimates the real worst case. When *n=N* (*M=1*), our bound is still a tight approximation. In fact, the effect of the inter-flow correlation is completely “included” in the effect of the finiteness of the input link capacity, which is taken into account in our analysis. This phenomenon is qualitatively shown in Fig. 9.

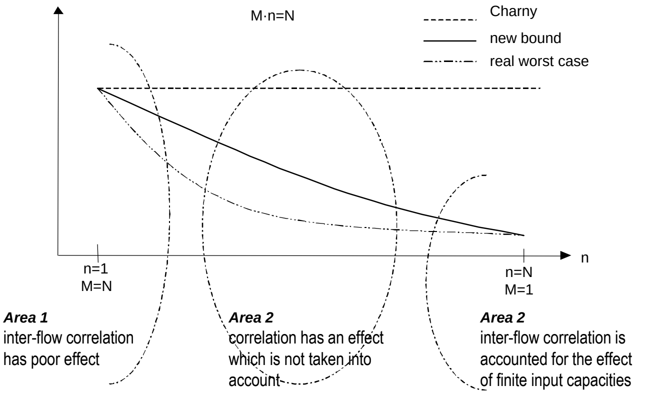

*Fig. 9 - Inter-flow correlation effects.*

The generic quantitative evaluation of the real worst case taking into account the flow correlation is not easy and is a matter of further study. We have quantitative results only for very simple scenarios. For example, assume *M*=4 input links and that *n*=20 flows (out of 40) from each input link are extracted to be multiplexed in the output link (N=80). Furhter assume that the peak rate of each flow, *P* is *64* kbit/s and that $C_{input}$ = 3200 Kbit/s (peak utilization = 0.8), $C_{output}$ = 6400 Kbit/s (peak utilization = 0.8). Applying Charny’s bound, we obtain an upper bound to the maximum buffer size on the output link buffer equal to 144 packets. If we use the methodology described in section 3, we obtain a buffer size of 82 packets. On the other hand, if we take into account the actual worst case arrival pattern of the bundle of 20 flows extracted from each input line, it can be verified that the maximum buffer size on the output link buffer equals to 62 packets. It is possible to make the evaluation of this worst case “by hand”, because we are at the second multiplexing stage and the scenario is homogenous.

## 6. References

[1] D. Black et al. “An Architecture for Differentiated Services”, RFC 2475, Dec. 1998.

[2] K. Nichols, V. Jacobson and L. Zang, “A Two-bit Differentiated Services Architecture for the Internet”, RFC 2638, July 1999.

[3] V. Jacobson, K. Nichols, K. Poduri “An Expedited Forwarding PHB”, RFC 2598, June 1999.

[4] X. Xiao, L. M. Ni “Internet QoS: A Big Picture” IEEE Network, Vol 13, No. 2, March/April 1999.

[5] K. Nichols "An Opinionated View of the Current State of IP DiffServ", Berkeley MIG Seminar, Sep 99, http://bmrc.berkeley.edu/courseware/cs298/fall99/nichols/kmn_ucbmm.pdf

[6] A. Charny "Delay bounds in a network with aggregate scheduling", Draft version 3, 2/99, Cisco, Available from ftp://ftpeng.cisco.com/acharny/aggregate_delay.ps

[7] J.Y. Le Boudec "A proven delay bound for a network with Aggregate Scheduling" EPFL-DCS Technical Report DSC2000/002, http://ica1www.epfl.ch/PS_files/ds2.pdf

[8] Network Simulator 2, http://www-mash.cs.berkeley.edu/ns