

# Integration of Streaming and Elastic Traffic: A Fixed Point Approach

H. HASSAN, O. BRUN  
LAAS-CNRS

University of Toulouse

7 av du colonel ROCHE, 31077 Toulouse, France  
{hassan,brun}@laas.fr

J.M. GARCIA, D. GAUCHARD  
QoS Design

6 av Marcel DORET, 31500 Toulouse, France  
{jmgarcia,dgauchard}@qosdesign.com

## ABSTRACT

We present a fixed point approach to evaluate the quality of service of streaming traffic multiplexed with elastic traffic in multi-service networks. First, we handle elastic traffic and streaming traffic separately, and then we derive a general fixed point formulation integrating both types of traffic in best effort networks. Then, we extend the application of this formulation to multi-service networks where priorities and bandwidth sharing schemes can be applied to different flows. Our approach is mainly oriented towards very large scale networks where traditional simulation techniques are not scalable, and where a large number of flows have to be evaluated in reasonable time. We assess the accuracy of our approach by means of event-driven simulations.

## Categories and Subject Descriptors

C.4 [Computer System Organization]: Performance of Systems

## General Terms

Performance

## Keywords

Fixed point, streaming traffic, elastic traffic, quality of service.

## 1. INTRODUCTION

The evolution of the Internet from data transport network to multi-service network promoted the notion of quality of service (QoS). Although network operators can propose a wide range of multimedia services today, the success of any new deployed service is tightly related to the QoS under which it is perceived by end users. From a packet network point of view, operators have to deal with packets belonging to different applications subject to different time constraints. Different scheduling mechanisms were introduced to offer priority to time sensitive flows, and different traffic engineering approaches were explored to guarantee an acceptable level of QoS usually denoted as Service Level Agreement (SLA). Besides, a wide range of signaling and transport protocols were introduced to contribute making data

networks a reliable transmission medium for different kinds of applications.

However, with this wide arsenal of approaches and methodologies, network operators are interested in only one thing, will the SLA be respected, and under which conditions? The best way to answer this question is to have in hand efficient and reliable performance evaluation tools. In this context, stochastic modeling and simulation techniques are suitable to provide network operators with practical tools to evaluate the performance of deployed services in different contexts. However, in the large scale networks context, tools should also be efficient and scalable, in order to achieve performance evaluation in reasonable time, especially when it concerns a big number of traffics. As a consequence, a tradeoff between exact simulation techniques and approximate but yet accurate estimation approaches is mandatory. That is why planning tools that are capable of giving primary indications about performances with a low computational cost are of great use nowadays. The goal of this paper is to present a light and scalable approach to evaluate the performance of multimedia traffics in packet networks.

Given the large variety of Internet applications, we distinguish between two broad categories according to temporal constraints: real-time applications, often associated with the User Datagram Protocol (UDP) and known as streaming applications, and data applications associated with Transmission Control Protocol (TCP) and known as elastic applications.

The major difference between these two categories is in the packet transmission profile regarding the network congestion conditions. Indeed, the packet rate of streaming applications is independent of the network state and is only determined by the source, while the packet rate of elastic applications varies according to the network congestion level. Hence, the major concern in network performance evaluation is to accurately model elastic and streaming traffic, and then propose simple approaches to evaluate its performances. In this paper, we handle this problem using a fixed point approach.

The paper is organized as follows: in the next section, we present relevant related work. In section 3 we present the network model, and in section 4 and 5 we consider networks with elastic traffic and streaming traffic separately. Then in section 6 we formulate a general fixed point approach for the integration of elastic and streaming traffic. Finally, in section 7 we extend our approach to multi-service networks and we conclude by some remarks and guidelines for future work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*Conference SIMUTools, March 03 – 07, 2008, Marseille, France.*

Copyright 2008 ACM ISBN 978-963-9799-20-2...\$5.00.

## 2. RELATED WORK

Network simulation has gained the interest of researchers a long time ago. However, the most common and accurate method is still the event-driven technique. Event-driven simulations are widely used in all engineering fields and are based on the reconstitution of the system behaviour using scheduled events. In the context of modern telecommunication networks, event-driven simulations are based on packets, as packets are the basic elements in the network transmission mechanism. Many tools are developed around this technique. We can cite, for example, the Network Simulator (NS) [7] as one of the major packet simulation tools used by the network community. However, the event driven technology at the packet level has big limitations. Indeed, the number of simulated events is proportional to the number of generated packets in the network, and this may lead quickly to huge and extremely time consuming scenarios.

Hopefully, the network community proposed different solutions to this problem based on different techniques. We retain two important trends: efficient simulation techniques, e.g. differential traffic theory, and analytical approximations.

The differential traffic theory proposed in [9] defines a general fluid modelling framework which allows studying both transient and stationary states of network resources. Extended with the Hybrid simulation technique, event-driven simulations can be combined with differential analytical models in order to simulate complex traffics. Differential and hybrid simulation techniques represent an efficient alternative to event-driven simulations as they have fairly low computation cost.

On the other hand, analytical approximations concern mainly analytical traffic models that provide approximations of packet arrivals at the source level using mathematical models, such as Markov based models e.g. [17] [18] [19]. Generally, those models are decoupled from the network topology as they only consider packet arrivals. This is particularly inadequate for TCP traffic as traffic profile is tightly related to the network congestion conditions. Therefore, several studies focused particularly on the TCP traffic (see [4] [13] [14] [15] [16]). They propose in general stochastic models for the TCP congestion mechanism using assumptions concerning the packet loss process, the ACK reception profile or the dynamic transmission rate of connections.

Besides the above mentioned techniques some recent papers [20] [1] [21] have addressed the problem of predicting elastic traffic behaviour using a fixed point approach. The basic idea is to formulate non-linear equalities with complementarity conditions governing TCP throughput. Then by fixed point iterations a feasible point satisfying both network constraints e.g. link capacities and complementarity conditions can be found. Particularly, the fixed point formulation proposed in [1] is well adapted to the generic network calculus problem, as bottlenecks are discovered automatically as in a global optimization problem. However, to the best of our knowledge tailoring the fixed point approach to include both elastic and streaming traffic has not been addressed before. We note finally that there are some recent works that handled the problem of evaluating the performances of streaming traffic when multiplexed with elastic traffic. We cite, for example, the remarkable work of Bonald and Proutière [22] on evaluating the performance bounds when integrating streaming and elastic traffic, and Delcoigne et al [23] in modelling the

integration of streaming and data traffic. The major difference between our approach and those mentioned above is that we try to solve a global optimization problem based on TCP non-linear throughput equations and  $M^N/D^N/1/K$  queue approximation for streaming traffic. Besides, we provide practical and easy to implement formulas to estimate end-to-end loss rate, end-to-end delay and end-to-end jitter for streaming and elastic flows.

## 3. NETWORK MODEL

We consider a network  $W$  formed of  $N$  nodes. Each node corresponds to a router with different interfaces. Router interface is modeled by a deterministic service queue with limited buffer capacity  $K$  and constant packet service rate  $\mu$  corresponding to the capacity of the output link. All flows are transported in the same service class. In other words no differentiation between elastic and streaming packets is done. This is the general case as most of today networks are still functioning on a best effort basis due to the difficulty of marking priority packets or defining appropriate service classes on a multi-provider path. Besides, the generalization of the best effort case to a priority or a processor sharing paradigms (GPS) is straightforward as we will see later in the multi-service network case.

A group  $G$ , of elastic and streaming flows, transits across the network  $W$ . The originating node of each flow  $f^i \in G$  is denoted as  $O^i$  and the destination node is denoted as  $D^i$ . The route followed by each flow  $f^i$  is denoted as  $R^i = \{n_1^i, n_2^i, \dots, n_{d(i)}^i\}$  where  $n_j^i$  represents a node crossed by the  $f_i$  flow and  $d(i)$  is the index of the destination node  $D^i$ . We refer also to the portion of route from the originating node  $O^i$  to the intermediate node  $n_m^i$  by  $r^i(m) = \{n_1^i, n_2^i, \dots, n_m^i\}$ .

Based on the previous definitions, we will present in the next sections a fixed point formulation to evaluate network performance in presence of elastic (TCP) flows alone, streaming (UDP) flows alone and finally the integration of elastic and streaming flows.

## 4. ELASTIC TRAFFIC

In this section we consider a group of flows  $G$  composed of only TCP persistent connections. The goal is to estimate flow rate and end-to-end loss rate for each flow in the group  $G$ . First, we recall the fixed point formulation proposed in [1] for large bandwidth-delay product networks, and then we provide a simple heuristic to get faster convergence. Finally, we extend the fixed point formulation to small bandwidth-delay product networks by enhancing RTT estimation.

### 4.1 Fixed-point formulation

Let  $P^i = (p_1^i, p_2^i, \dots, p_{d(i)}^i)$  be the vector of loss probabilities across route  $R^i$ .  $p_j^i$  corresponds to the packet loss probability at node  $n_j^i$  on the route  $R^i$ . The packet loss probability on each node can be estimated using the following iterative formula [1]:

$$p_j^i(k+1) = \text{Pr}_{[0,1]} \left\{ p_j^i(k) - \alpha \mu_j^{\text{res}} \right\} \quad (1)$$

$$\text{Where } \mu_j^{\text{res}} = \mu_j - \sum_{z \in G} \prod_{n \in \mathcal{R}^z(j)} (1 - p_n^z(k)) * T^z(P^z(k))$$

The function  $\text{Pr}_{[0,1]} \{x\}$  is the projection on the interval  $[0, 1]$  and  $\alpha$  is a positive constant related to the convergence speed of the algorithm. We note that the choice of  $\alpha$  has an important impact on the convergence of the algorithm.

$T^z$  is the flow throughput as a function of the end-to-end packet loss rate ( $L^z$ ). In literature, we find many expressions to evaluate  $T^z$  in function of  $L^z$ , more details can be found in [2] [3]. We give hereafter the square root formula [2] which is known to be suitable for a small number of timeout events:

$$T^z(L^z) = \frac{MSS^z}{RTT^z} \min \left\{ \sqrt{\frac{c^z}{L^z}}, W_{\max}^z \right\} \quad (2)$$

$$L^z = \sum_{n \in \mathcal{R}^z} p_n^z \prod_{h \in \mathcal{R}^z(n) \setminus n} (1 - p_h^z) = 1 - \prod_{n \in \mathcal{R}^z} (1 - p_n^z) \quad (3)$$

Where  $W_{\max}^z$  is the maximum receiver window size,  $RTT^z$  is the round-trip time on the considered route and  $c^z$  is a constant equals to 2 if delayed ACKs are used and the loss events are assumed to be exponentially distributed [4].

## 4.2 Choice of $\alpha$ and the convergence speed

The fixed point formulation presented in (1) updates the value of loss rate at node  $n_j^i$  according to the evolution of the bandwidth consumed by TCP. Starting from an initial loss rate value if the throughput value estimated by (2) exceeds the available resources at node  $n_j^i$ , the value of  $p_j^i$  is increased, otherwise it is decreased. When the value of  $p_j^i$  exceeds 1, the projection function forces it to 1 and when it becomes negative it forces it to 0. As the throughput function (2) is continuous and inversely proportional to the square root of the loss rate (see (2)), the estimated throughput oscillates around the target value. Particularly, the magnitude of oscillations is tightly related to the choice of the parameter  $\alpha$  in equation (1). Hence, if the value of  $\alpha$  is too big, the loss rate value  $p_j^i$  oscillates between 0 and 1, and if the value of  $\alpha$  is too small the algorithm takes a long time before it converges. In Table 1, we show the number of iterations required by the algorithm as a function of two values of  $\alpha$  for three simple scenarios (with only two nodes for each one).

**Table 1. Influence of  $\alpha$  on the convergence speed**

$\alpha$	BW Mbps	Delay ms	Number of iterations	Convergence
10e-10	1	1	$\infty$	No
10e-10	5	1	$\infty$	No
10e-10	10	10	8310	Yes

$\alpha$	BW Mbps	Delay ms	Number of iterations	Convergence
10e-13	1	1	391561	Yes
10e-13	5	1	127037	Yes
10e-13	10	10	40282	Yes

Two main problems appear: first we can not use the same value of  $\alpha$  for all bandwidth-delay product scenarios and second, the algorithm may converge very slowly if a very small value of  $\alpha$  is used. The convergence problem of fixed point algorithms is a wide and complicated subject (interested reader may refer to [10] for details). However, in our case a simple solution may be used to get faster convergence based on the evolution of the loss rate.

Starting with an initial big value of  $\alpha$  we test the evolution of the calculated loss rate. If the loss rate exceeds 1 or less than 0 then we divide  $\alpha$  by two, moreover if the difference between old and new loss rate is increasing we divide  $\alpha$  by two. Indeed, in these two cases the algorithm can not converge as the value of  $\alpha$  is too big. This adaptive way in choosing  $\alpha$  makes the algorithm executes faster and the initial value of  $\alpha$  has no impact on the convergence speed. When applying this adaptive method, the algorithm converges in less than 100 iterations in average on a wide range of scenarios and bandwidth-delay product values.

## 4.3 Relaxing the large bandwidth-delay product assumption

In large bandwidth-delay product networks, the value of RTT of static routes is almost constant as packet service delay in router interfaces is negligible compared with the fixed link delay. However, packet service delay becomes an important factor in the RTT value when bandwidth-delay product is too small. For example on 1 Mbps link, the TCP packet service delay is  $1040*8/(1024*1024) = 0.0079 \text{ sec} = 7.9 \text{ ms}$  for MSS value of 984 bytes. When we consider 1 ms link delay, for example, the packet service delay is about eight times more important than the fixed link delay. As a consequence, the average number of packets waiting to be served along the route constitutes the most important contribution to the RTT value. As this number could be variable, the value of RTT will fluctuate. On the other hand, on a 100 Mbps link, the TCP packet service delay is only 0.079 ms and its value may be negligible even if the link delay is about 1 ms.

In order to relax the large bandwidth-delay product assumption used in [1], and make the proposed fixed point formulation valid for all bandwidth-delay product networks, we will consider the packet service delay contribution in the RTT value. Thus, we need to evaluate the average number of packet waiting to be served along the route followed by a flow.

TCP is designed to utilize the available resources in the network. Considering this, a link could be either underutilized or a bottleneck. Normally, underutilized links will have a small average queue length as they have little packet loss rate, while bottleneck links will have an important average queue length. In other words, the average number of packets waiting to be served

along the route is approximately the average queue tail of the queue with the highest loss rate along the route.

Let  $S^i = \{s_1^i, s_2^i, \dots, s_{d(i)-1}^i\}$  be the fixed link delay vector associated with route  $R^i = \{n_1^i, n_2^i, \dots, n_{d(i)}^i\}$ , where  $s_j^i$  is the fixed link delay between node  $n_j^i$  and node  $n_{j+1}^i$ . The RTT value along route  $R_j^i$  can then be estimated by:

$$RTT^i = 2 * \sum_{s \in S^i} s_j^i + \beta * K_{j,\max} * \mu_{j,\max} \quad (4)$$

Where  $K_{j,\max}$  and  $\mu_{j,\max}$  are the buffer size and packet service rate of the highest loss rate queue along the route.  $\beta$  is a positive constant. Practically, the value of  $\beta$  is found to be suitable between 0.9 and 0.95.

In the next section we will evaluate the performance of the fixed point approximation using the dynamic RTT estimation and we will compare it with the original formulation without including packet service time in RTT estimation.

#### 4.4 Simulations

In this section we compare the analytical evaluation of TCP flows with even-driven simulations undertaken in DHS. DHS is a distributed Hybrid Simulator developed by QoS Design [8] available within the Nest software package. We use DHS in the event-driven mode.

Using a cross-like topology showed on Figure 1, with two TCP flows (F1 and F2) originating from node 1 to node 5 and 3, and two other TCP flows (F3 and F4) originating from node 4 to node 5 and 3, we perform different simulations using different bandwidth-delay product values. The number of connections is the same for all flows  $N1 = N2 = N3 = N4 = 3$ .

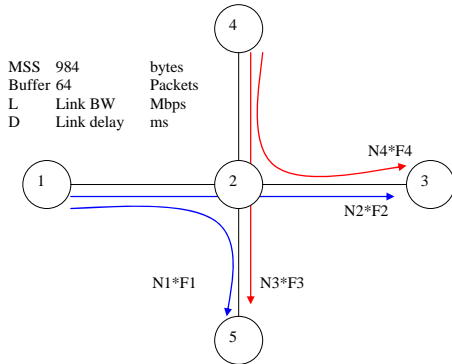


Figure 1. Cross network topology

In Table 2, we present the results of output rate and end-to-end loss estimation for flow F1 using both analytical (with static RTT) and event-driven simulations. Similar results were obtained for other flows. It is important to note that the relevant parameter for TCP performance is the output rate. However, the good estimation of end-to-end loss is essential to evaluate the bandwidth consumed by TCP, and this is of great importance when multiplexing with streaming traffic.

Table 2. Performance evaluation of flow F1 (static RTT)

BW Mbps	Delay ms	Throughput Mbps		e-t-e Loss %	
		Event	Analytic	Event	Analytic
1	1	0.51	0.2	0.077	97.5
5	1	2.48	1.55	0.58	82.2
10	10	4.95	4.81	0.32	2.59
10	20	4.95	4.98	0.17	0.62
100	40	9.21	9	0	0

It is clear that the fixed point approximation does not work for small bandwidth-delay product networks when RTT value is static. The obtained loss rate by fixed point formulas is very high (97.5%) for a link of 1 Mbps bandwidth with 1 ms of delay compared to event-driven simulation. However, it performs very well under large bandwidth-delay products offering very good estimation of TCP throughput.

In Table 3, we present the results of output rate and end-to-end loss estimation for flow F1 using enhanced fixed point approximation (with dynamic RTT estimation) and event-driven simulations.

Table 3. Performance evaluation of flow F1 (dynamic RTT)

BW Mbps	Delay ms	Throughput Mbps		e-t-e Loss %	
		Event	Analytic	Event	Analytic
1	1	0.51	0.5	0.077	2
5	1	2.48	2.52	0.58	1.88
10	10	4.95	4.87	0.32	0.61
10	20	4.95	5.0	0.17	0.25
100	40	9.21	9	0	0

The dynamic evaluation of the RTT value (using (4)) leads to better estimation of end-to-end loss and very good approximation of TCP throughput for all bandwidth-delay products: large and small.

## 5. STREAMING TRAFFIC

We note that the fixed point formulation in the previous section evaluates the loss rate as a fixed point of non-linear equations representing TCP flow throughput. The existence of a fixed point for this system is due in part to the fact the TCP adapts its behavior to the network conditions. In other words, TCP performs auto regulation to attain stationary regime. However, in the case of UDP flows, packet rate at the originating node is independent of the network state. In order to attain a stationary regime the

system load  $\rho = \frac{\text{Input Rate}}{\text{Service Rate}}$  should be bounded by 1.

Under the assumption that  $\rho \leq 1$  we evaluate the performance of UDP flows using queuing theory approximations. We need to evaluate the loss rate and throughput at each node in function of

input rate and traffic characteristics (packet size distribution and packet inter-arrival distribution). This leads us to consider a G/G/1/K queuing system which is not easy to solve in the general case. Based on the fact that our UDP traffic represents streaming applications (Voice over IP and Video applications) we will simplify our problem by considering exponential packet inter-arrivals. In fact, this approximation holds when the number of multiplexed connections is large enough. Reader may refer to [6] for an exhaustive analysis of the traffic resulting from the superposition of VoIP and Video applications.

Assuming that packet inter-arrivals are exponentially distributed, the system input is resumed to multi-rate exponential flows with discrete packet sizes. Thus, the considered queuing system becomes a special case that we denote  $M^N/D^N/1/K$ , where  $K$  is the buffer size in packets,  $N$  is the number of different discrete packet sizes referred to as packet classes. It is just like if we transformed our input flow to  $N$  flows with exponential packet arrivals and constant packet sizes. As the router interface has fixed capacity, the resulting service law is discrete with  $N$  points corresponding to the  $N$  packet classes.

In the following sections we present the analytical solution of the  $M^N/D^N/1/K$  queue in order to evaluate the packet loss rate and the average packet number in the queue (which will be used to evaluate the delay). Then we will formulate our fixed point approach in order to get the average throughput for each flow. However, the average throughput, packet loss and delay are not sufficient to characterize completely the performance of streaming flows. We need also to include jitter estimation. For this we will recall important results about jitter estimation presented by Brun et al [5] and provide its application in our case.

### 5.1 Analytical solution of $M^N/D^N/1/K$

Let  $\lambda = \sum \lambda_n$  be the total arrival rate to the system. Since we assume Poisson arrival processes, the probability that an arriving customer of class  $n$  is  $p_n = \lambda_n / \lambda$ . Let  $T = \sum p_n T_n$  be the average service time.

Let  $\alpha_n$  be the probability of  $n$  arrivals during a customer service. Since arrivals are Poisson distributed with rate  $\lambda_n$  for class  $n$  and since service duration is a constant  $T_n$  for class  $n$ , we have:

$$\alpha_n = \sum_{i=1}^N p_i \frac{(\gamma_i)^n}{n} e^{-\gamma_i} \text{ where } \gamma_i = \lambda T_i, i = 1 \dots N$$

We denote the probability that  $j$  customers are left behind the  $m^{\text{th}}$  departure as  $q_j$  (with  $q_0 = 1$ ), then:

$$q_j = \frac{1}{\alpha_0} \left[ q_{j-1} - \alpha_{j-1} - \sum_{i=1}^{j-1} \alpha_i q_{j-i} \right], j = 1 \dots K-1$$

After normalization:

$$q_j = \frac{q_j}{\sum_{k=0}^{K-1} q_k}$$

Finally, the probability distribution function of the number of customers in the system is given by:

$$P_j = \frac{q_j}{\rho + q_0}, j < K$$

$$P_K = 1 - \frac{1}{\rho + q_0}$$
(5)

The mean number of customers of the  $M^N/D^N/1/K$  queue is given by:

$$X_K = \sum_{i=1}^K i P_i$$
(6)

### 5.2 Fixed-point formulation

Let  $F^N = \{f^1, f^2, \dots, f^N\}$  be the input flow vector with  $PS^N = \{ps^1, ps^2, \dots, ps^N\}$  the corresponding packet size (class) vector, and let  $P^1 = (p_1^1, p_2^1, \dots, p_{d(1)}^1)$  be the vector of loss probabilities.  $p_j^i$  corresponds to the packet loss probability at node  $n_j^i$  of flow  $f^i$ . The value of  $p_j^i$  is calculated for each flow  $f^i$  at each node  $n_j^i$  using the analytical solution of the corresponding  $M^N/D^N/1/K$  queue. Then the throughput  $T^i$  at each node  $n_j^i$  is given by the following iterative expression:

$$T_{output}^i(k+1) = T_{input}^i(k) (1 - p_j^i(k))$$
(7)

Using (7) we evaluate the average flow throughput at each node and the resulting value is propagated along the route followed by the flow. The output rate at the last node gives the average flow rate, and allows estimating end-to-end loss rate. After convergence the delay incurred by the flow at each node is estimated using the average number of customers given by (6). End-to-end delay is obtained by adding this value to the fixed link delay along the route.

### 5.3 Jitter approximation

In this section, we recall the jitter estimation procedure proposed in [5] and provide its application for exponential flows with constant packet sizes. Let  $V$  be a generic service time and  $U / \rho$  be a generic inter-arrival time with traffic intensity  $\rho$ . Using the square coefficient of variations (SCV)  $c_a^2$  for the arrival process and  $c_s^2$  for the service time process, we define the constant  $\eta$  [24]:

$$\eta = \frac{2(1-\rho)}{c_a^2 + c_s^2} [1 - (1-\rho)\eta^*]$$

$$\text{Where } \eta^* = \frac{[2V_3 - 3c_s^2(c_s^2 + 2)] - [2u_3 - 3c_a^2(c_a^2 + 2)]}{3(c_a^2 + c_s^2)}$$

And the constant  $\alpha$  as:

$$\alpha = \frac{\eta\rho}{1-\rho} \left[ \frac{c_a^2 + c_s^2}{2} - (1-\rho) \left( \frac{2\frac{u_3}{3} + (c_a^2 + 1)(c_s^2 - 1)}{2(c_a^2 + c_s^2)} - c_a^2 \right) \right]$$

$u_i$  (resp.  $v_i$ ) is the moment of order  $i$  of the arrival process (resp. the service process).

The end-to-end jitter incurred by a flow  $f^i$  multiplexed with background traffic characterized by the two coefficients  $\alpha$  and  $\eta$  is estimated by the expected absolute value of the sum of inter-packet delay variations introduced by each node along the path between the source and the destination. Considering that the average inter-arrival time of flow  $f^i$  is  $T^i$ , Brun et al [5] provide a simple analytical formula for estimating end-to-end jitter denoted as  $J_{[1..n]}(T)$ . The expression is as follows:

$$J_{[1..n]}(T) \approx \sum_{j=1}^n K_j^n J_j(T) \quad (8)$$

Where  $J_j(T)$  is the jitter introduced by node  $n_j^i$  and given by:

$$J_j(T) = \frac{\phi_j}{\eta_j} \left[ 1 - \frac{1}{2} (1 + \eta_j T_j) e^{-\eta_j T_j} \right]$$

with  $T_j = T - \frac{ps^i}{\mu_j}$  and  $\phi_i = 2\alpha_i \left( 1 - \frac{\alpha_i}{2} \right)$

And  $K_j^n$  is a weighing factor expressed as:

$$K_j^n = \prod_{i=1, i \neq j}^n \left( 1 - \phi_i \frac{\eta_j^2}{\eta_j^2 - \eta_i^2} \right)$$

The simplicity of the above formula is that it only depends on the first three moments of arrival and service process.

Considering the  $M^N/D^N/1/K$  queuing system, the first three moments of the Poisson arrival process are given by  $u_1 = \frac{1}{\lambda}$ ,

$u_2 = \frac{2}{\lambda^2}$  and  $u_3 = \frac{6}{\lambda^3}$ . Besides, the probability distribution function of service time is given by:

$$s(x) = \sum_{n=1}^N p_n u(x - ps_n)$$

$ps_n, n = 1 \dots N$  are packet sizes and  $p_n, n = 1 \dots N$  are corresponding weighing factors.

Just like delay, the jitter estimation procedure is invoked when convergence is obtained for loss rate and flow throughput.

## 5.4 Simulations

Using the same topology of Figure 1 we substitute TCP flows by VoIP flows of G729 type. Each VoIP connection is modeled by an ON-OFF process with the ON period corresponding to the talkspurt and the OFF period corresponding to the silence. The duration of ON and OFF periods is exponentially distributed with average ON period of 0.352 seconds and average OFF period of

0.65 seconds (see [11]). The G729 codec produces packets of 70 bytes during the ON period at a constant packet rate of 33.3 packets/sec. The number of VoIP connection per flow is the same  $N1 = N2 = N3 = N4 = 50$ .

We evaluate the end-to-end loss rate, delay and jitter per VoIP connection for each flow using the fixed point approximation and we compare it with event-driven simulations. In Table 4, we present the results for flow F1. Similar results were obtained for other flows.

**Table 4. Performance evaluation of VoIP traffic (F1)**

$\rho$	e-t-e Loss %		e-t-e delay ms		e-t-e jitter ms	
	Event	Ana	Event	Ana	Event	Ana
0.3	0.0	0.0	2.8	2.78	0.2	0.18
0.5	0.0	0.0	2.93	2.94	0.33	0.34
0.7	0.0	0.0	4.72	5	1.35	1.47
0.97	0.22	0.24	36.2	38.1	7.27	21.6

We can see from these results that the fixed point approach based on the  $M^N/D^N/1/K$  queue analytical solution provides excellent estimates for end-to-end loss and end-to-end delay under low to high traffic intensities. End-to-end jitter is also remarkably well estimated in light to medium traffic intensities. However, it presents some shortcomings at very high traffic intensities. Indeed, high traffic intensity violates the assumptions used by the authors in [5] when evaluating jitter.

## 6. INTEGRATION OF ELASTIC AND STREAMING TRAFFIC

After have been studying elastic and streaming traffic separately we handle in this section the problem of integrating both traffics. The main difference between elastic and streaming traffic is that elastic traffic adapts its input rate according to the network congestion state while streaming traffic maintains the same input rate whatever the congestion state on the network.

The integration of both elastic and streaming traffic results in forcing elastic traffic to adapt its input rate according to the residual bandwidth left by streaming traffic. However, estimating this residual bandwidth is not trivial. Indeed, TCP tries always to discover and utilize the maximum available resources and this will increase the observed loss rate at router interface. On the other hand, increasing the loss rate at router interface makes TCP connection reduce its input rate until a stability point is found.

The fixed point formulation used for TCP only networks can thus be modified to include the bandwidth consumed by streaming flows. The loss rate evaluated in equation (1) depends via the constant  $\alpha$  on the difference between available link capacity and the consumed bandwidth by TCP. Thus, we can add to this difference term, the bandwidth consumed by streaming flows, which is the fixed input rate of streaming flows minus the dropped packets. It is, therefore, important to evaluate quite well the loss rate incurred by streaming flows in presence of elastic traffic. In order to keep an analytically tractable solution we use the  $M^N/D^N/1/K$  approximation described before. Of course,

we are aware of the fact that TCP packet arrivals can not be approximated by Poisson arrivals. However, in the case of slow access links in comparison with the bottleneck capacity, this assumption is acceptable (see [12]).

## 6.1 Fixed point formulation

The group of flows  $G$  is composed now of elastic and streaming flows. Equation (1) is modified to take into consideration the bandwidth consumed by streaming flows as follows:

$$p_j^i(k+1) = \Pr[0, 1] \left\{ p_j^i(k) - \alpha (\mu_j - BW_{TCP} - BW_{UDP}) \right\} \quad (9)$$

Where  $BW_{TCP}$  is the bandwidth consumed by elastic flows (TCP), given by:

$$BW_{TCP} = \sum_{z \in G(TCP)} \left( \prod_{n \in r^z(j)} (1 - p_n^z(k)) \right) * T^z(P_{TCP}^z(k))$$

And  $BW_{UDP}$  is the bandwidth consumed by streaming flows (UDP), given by:

$$BW_{UDP} = \sum_{z \in G(UDP)} \left( \prod_{n \in r^z(j)} (1 - p_n^z(k)) \right) * T^z(P_{UDP}^z(k))$$

$G(TCP)$  (resp.  $G(UDP)$ ) is the subset of elastic flows (resp. streaming flows) in the global set of flows  $G$ .

Using this formulation we distinguish between two loss rates evaluated at each node. The loss rate observed by elastic flows (denoted as  $P_{TCP}^z$ ) given by equation (9), and the loss rate observed by streaming flows (denoted as  $P_{UDP}^z$ ), evaluated using the analytical solution of the  $M^N/D^N/1/K$  queue. Of course,  $P_{UDP}^z$  is evaluated by approximating elastic flows with exponential flows of constant packet size equal to the TCP packet size.

Practically the fixed point algorithm operates as follows:

---

### *Elastic and streaming fixed point algorithm*

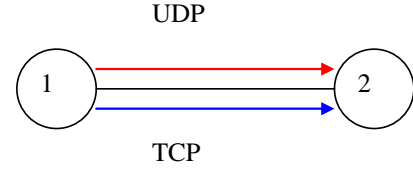
---

- 1: **Initialize**  $P_{TCP}^z$  and  $P_{UDP}^z$
  - 2: **While** (!(UDPconv) && (TCPconv))
  - 3:     Evaluate  $T^z(P_{TCP}^z(k))$  using (2)
  - 4:     Evaluate  $P_{UDP}^z(k)$  using (5)
  - 5:     Evaluate  $P_{TCP}^z$  using (9)
  - 6:     Compute UDPconv and TCPconv
  - 7: **End while**
- 

UDPconv (resp. TCPconv) is the convergence condition for UDP (resp. TCP) flows. In the next section we evaluate the proposed algorithm in different network scenarios.

## 6.2 Simulations

Two scenarios will be used to validate our approach. The first is a simple two nodes topology (Figure 2), and the second is the cross like topology of Figure 1. TCP flows are persistent TCP connections with fixed packet sizes, and UDP flows are exponential of 500 Kbps rate, with constant packet size of 512 bytes.



MSS: 984 Bytes     UDP packets: 512 Bytes  
 Buffer: 64 Packets     UDP rate: 500 Kbps  
 BW: 5 Mbps     Delay: 1 ms

**Figure 2. Two nodes network topology**

Using the two nodes topology, we propagate 5 UDP flows and 5 TCP flows. The UDP traffic load is 50% of the total load. In Table 5, we present the fixed point evaluation results against even-driven simulation. UDP flows are characterized by loss rate, delay and jitter while TCP flows are characterized by loss rate, delay and average throughput.

**Table 5. Evaluation of UDP and TCP flows (simple)**

UDP	e-t-e Loss %		e-t-e delay ms		e-t-e jitter ms	
	Event	Ana	Event	Ana	Event	Ana
	3.04	3.98	55.5	54.2	2.05	3.99
TCP	e-t-e Loss %		e-t-e delay ms		Throughput Mbps	
	Event	Ana	Event	Ana	Event	Ana
	4.77	4.7	55.7	54.9	2.64	2.63

Results we obtain are very satisfactory concerning TCP flows. The achieved throughput by TCP is very well estimated. On the other hand, the jitter incurred by UDP flows is significantly above the results obtained by event-driven simulations. Jitter values can be explained by the high load observed in queues. Indeed, TCP connection adapts its rate to fully utilize the available resources. Consequently, the simulated queues will be always very highly loaded, making it very hard to estimate jitter incurred by streaming flows as explained before.

Now we use the cross like topology with 4 Mbps and 1 ms delay for all links. Four flows are propagated throughout this network. Flow F1 originating from node 1 to node 5, flow F2 originating from node 1 to node 3, flow F3 originating from node 4 to node 5, and flow F4 originating from node 4 to node 3. Each of these flows is composed of one TCP connection and two UDP connections of the same characteristics as used before. In Table 6 we present the fixed point evaluation results against even-driven simulations.

**Table 6. Evaluation of UDP and TCP flows (cross)**

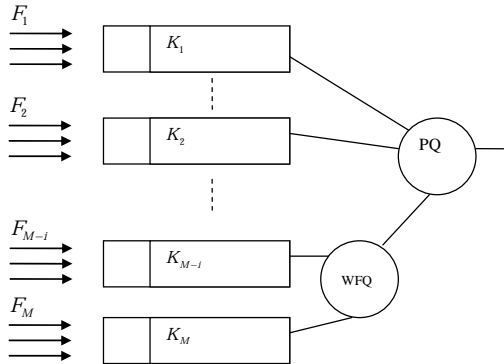
UDP	e-t-e Loss %		e-t-e delay ms		e-t-e jitter ms	
	Event	Ana	Event	Ana	Event	Ana
	0.82	1.95	83.91	88.6	3.12	5.2
TCP	e-t-e Loss %		e-t-e delay ms		Throughput Mbps	
	Event	Ana	Event	Ana	Event	Ana
	1.02	0.85	82.8	90.4	1.01	1.02

Once again we obtain good results for TCP flows. While QoS of UDP flows is less accurate.

It is clear that the high load in queues influences jitter estimation. Besides, approximating TCP packet arrivals by Poisson arrivals is not always suitable. However, the proposed approach represents a very efficient way to give a primary indication about the performance of network flows, especially in the context of very large scale networks with a high number of interacting flows.

## 7. MULTISERVICE NETWORKS

Up to this point we were considering only router interfaces with no service differentiation. Elastic and streaming flows are multiplexed together without any predefined priorities or service sharing paradigm. In this section, we consider a general router interface with priority classes and Weighted Fair Queuing (WFQ) classes (see Figure 3). Packet scheduling is achieved as follows: priority classes are handled first and WFQ classes are considered as the last priority. Within this last class the bandwidth is shared according to the General Processor Sharing Paradigm (GPS).



**Figure 3. Multi-service router interface**

Our purpose is to, first, estimate the bandwidth sharing between different classes, and then apply the fixed point approach defined in previous sections on each queue (service class). This will lead us to an integrated framework for fast performance evaluation of multimedia flows in multiservice networks.

### 7.1 Bandwidth sharing

The bandwidth sharing mechanism is two fold: Priority queues (PQ) and Weighted Fair Queuing queues (WFQ). First, the interface capacity is dispatched between PQ queues with WFQ

queues group as the last priority queue, and then the calculated capacity of WFQ group is dispatched between different WFQ queues.

#### 7.1.1 Priority queues

Consider a system of  $M$  priority queues (as illustrated on Figure 3), with  $\mu$  global capacity. As an approximation, the system can be transformed into  $M$  equivalent queues with an equivalent capacity  $\mu_k$  each. The value of  $\mu_k$  is estimated as follows:

$$\mu_k = \left( \mu - \sum_{i=0}^{k-1} r_i \right)^+ \quad (10)$$

where  $r_i$  is the output rate of the  $i^{th}$  queue expressed as follows:

$$r_i = \min(\mu_i, \lambda_i)$$

And  $\lambda_i$  is the input rate of the  $i^{th}$  queue.

#### 7.1.2 WFQ queues

In a WFQ queuing system the bandwidth sharing between different queues is performed according to the associated weights. Consider a system of  $N$  WFQ queues, with  $\mu$  global capacity. The system can be transformed into  $N$  equivalent queues with an equivalent capacity  $\mu_k$  each. The value of  $\mu_k$  is estimated as follows:

$$\mu_k = \alpha_k \mu \quad (11)$$

Where  $\alpha_k, k = 1, \dots, N$  are the associated weights.

However, the effective capacity seen by each queue depends on the system load. Indeed, when high weight queues are lightly loaded, other queues may take benefit of their residual bandwidth. To consider this we differentiate between two types of queues: in-deficit and in-excess queues. In-excess queues (noted  $E$ ) are those whose output rate is inferior than their equivalent capacity  $\mu_k$ :

$$E = \{i \mid r_i < \mu_i\}$$

$r_i$  is the output rate of the considered queue expressed as:

$$r_i = \min(\mu_i, \lambda_i)$$

In-deficit queues are the complementary group  $D$ :

$$D = \{1, \dots, N\} / E$$

Therefore, to better model the bandwidth sharing we evaluate the residual capacity of in-excess queues and redistribute it among in-deficit queues according to the following algorithm.

---

#### Estimation of the equivalent capacity for WFQ queues

---

1: **Evaluate** the residual capacity  $\mu_{res} = \sum_{k \in E} (\mu_k - r_k)$

2: **While** ( $\mu_{res} > 0$  and  $D \neq \emptyset$ )

3:     **For all**  $i \in D$

4:         
$$\mu_i = \mu_i + \frac{\alpha_i}{\sum_{j \in D} \alpha_j} \mu_{res}$$

5:         Update  $\mu_{res}$  and  $D$

6: **End while**

---

## 7.2 Simulations

In order to evaluate the approximation proposed for priority and WFQ queues, we use the same cross topology as in section 6.2 with the same flows. The only difference is that we substitute best effort interfaces with DiffServ interfaces. Two interfaces will be considered in our test: two priority queues interface, and two WFQ queues interface. In both cases we want to give the priority to delay sensitive traffic represented by the Poisson traffic.

In Table 7, we present the results for the priority queues interface with UDP traffic as the high priority one.

**Table 7. Evaluation of priority queues**

UDP	e-t-e Loss %		e-t-e delay ms		e-t-e jitter ms	
	Event	Ana	Event	Ana	Event	Ana
	0	0	6.5	4.88	1.09	0.72
TCP	e-t-e Loss %		e-t-e delay ms		Throughput Mbps	
	Event	Ana	Event	Ana	Event	Ana
	0.28	0.85	190	234	1.04	1.02

Before analyzing the fixed point evaluation results against event-driven simulations, we underline the improvement on streaming traffic QoS parameters when it is given priority. In comparison with the best effort case (see Table 6), we have a loss rate of 0 % instead of 0.82%, end-to-end delay of 6.5 ms instead of 83.91 ms and end-to-end jitter of 1.09 instead of 3.12 ms. Moreover, the TCP traffic does not suffer in terms of achieved throughput, while observed loss rate is smaller (0.28 % instead of 1.02%). Indeed, by giving priority to UDP packets we limit the interaction between streaming and elastic traffic enhancing the overall performances.

We can see that the estimated QoS parameters using the fixed point approach are good in comparison with event-driven simulations.

Now we change priority router interface into two WFQ queues interface, with 50% weight each (streaming and elastic). We repeat the same simulations. Results are given in Table 8.

**Table 8. Evaluation of WFQ queues**

UDP	e-t-e Loss %		e-t-e delay ms		e-t-e jitter ms	
	Event	Ana	Event	Ana	Event	Ana
	0.12	0.06	62.8	53	3.44	3.2
TCP	e-t-e Loss %		e-t-e delay ms		Throughput Mbps	
	Event	Ana	Event	Ana	Event	Ana
	0.24	0.9	188	146	1.02	0.99

Once again, we can see that the proposed fixed point approach gives the right indications about QoS parameters.

We should particularly note that the proposed formulation provides a complete framework for both elastic and streaming flows under best-effort and DiffServ configurations. Besides, we should underline the enormous gain in the evaluation process speed compared with other techniques. Globally, with hundred iterations we are able to predict the network performance in very large scale network configurations quite successfully.

## 8. CONCLUSION

Performance evaluation of multimedia flows in large scale networks is an important issue in modern telecommunication networks. In this paper, we presented a fixed point approach to evaluate the integration of elastic and streaming flows. The approach was first presented in the context of best effort networks when elastic and streaming flows have the same priority, and then it was extended to multi-service networks with priority and processor sharing service classes.

Using the square root rate formula for TCP, the operating point of TCP flows is determined. And based on the  $M^N/D^N/1/K$  queue system model, performance parameters are estimated for streaming flows. This approximation shows some limitations on streaming flows when integrating elastic and streaming flows in best effort configuration. This is, mainly, due to the fact that TCP flows can not be approximated with Poisson flows in different cases. However, even with this limitation the proposed approach is very promising as it offers a scalable method to evaluate the performance of multimedia flows in multi-service networks. Our future work deals with two points: first, enhancing the approximation of TCP flows by replacing  $M^N/D^N/1/K$  with  $G^N/D^N/1/K$  queuing system. This will allow a better estimation of streaming flows performances in presence of elastic flows. And second, relaxing the non heavy traffic assumption used in jitter evaluation formulas, as this assumption represents a limitation when multiplexing elastic and streaming flows.

## 9. ACKNOWLEDGMENTS

Our thanks to Urtzi Ayesta for discussions about TCP fixed point approaches.

## 10. REFERENCES

- [1] Altman, E., Avrachenkov, K., and Barakat, C., 2002. TCP network calculus: The case of large delay-bandwidth product. In Proceedings of the IEEE INFOCOM. <http://www-sop.inria.fr/mistral/personnel/K.Avrachenkov/pubs/mtcp3a.pdf>.
- [2] Ott T., Kemperman, J., and Mathis, M., The stationary behavior of the ideal TCP congestion avoidance, <ftp://ftp.telcordia.com/pub/tjo/TCP/TCPwindow.ps>
- [3] Padhye, J., Firoiu, V., Towsley, D., and Kurose, J., 1998. Modeling TCP Throughput: a simple Model and its empirical Validation, In Proceedings of the ACM SIGCOMM, 231-242.
- [4] Altman, E., Avrachenkov, K., and Barakat, C., 2000. A stochastic model of TCP/IP with stationary random losses, In Proceedings of the ACM SIGCOMM.
- [5] Brun, O., Bockstal, C., Garcia, J.M., 2006. Analytic approximation of the jitter incurred by CBR traffics in IP networks. Telecommunication Systems. 33, (December 2006) 23-45.
- [6] Hassan, H. 2006. Multimedia traffic Modeling and Performance Analysis in heterogeneous Networks. Doctoral Thesis. LAAS report Number 06842. University of Toulouse III (Paul Sabatier).

- [7] NS-2, Network Simulator (ver. 2). Available at <http://www.isi.edu/nsnam/ns/>
- [8] Nest IP-MPLS, Network Engineering & Simulation Tool. QoS Design Product. <http://www.qosdesign.com>
- [9] Garcia, JM., Gauchard, D., Brun, O., Bacquet, P., Sexton, J., Lawless, E. 2001. Differential Traffic Modelling and Distributed Hybrid Simulation. Réseaux et Systèmes Répartis, Calculateurs Parallèles, Performances des réseaux et systèmes. 13, 6, (2001) 635-664.
- [10] Roland, C., 2005. Méthodes d'Accélération de Convergence en Analyse Numérique et en Statistique. Doctoral Thesis. University of Lille I.
- [11] Deng, S., 1995. Traffic characteristics of packet voice. In Proceedings of the IEEE International Conference on Communications, Vol. 3, 1369-1374.
- [12] Ayesta, U., 2004. Stochastic scheduling and its application to the TCP/IP networks. Doctoral Thesis. University of Nice-Sophia Antipolis-INRIA.
- [13] Kelly, F.P., Maulloo, A., and Tan, D., 1998 . Rate control for communication networks: shadow prices, proportional fairness and stability. Journal of Operations Research Soc, 49, (1998).237-252.
- [14] Mo, J., and Walrand, J., 2000. Fair end-to-end window-based congestion control. IEEE/ACM Transactions on Networking. 8, 5 (Oct 2000).
- [15] Low, S.H., and Lapsley, D.E., 1999. Optimization flow control I: Basic Algorithm and Convergence. IEEE/ACM Transactions on Networking. 7, 6 (Dec 1999).
- [16] Kunniyur, S., and Srikant, R., 2000. End-to-end congestion control schemes: Utility functions, Random losses, ECN marks. In Proceedings of the IEEE INFOCOM.
- [17] Sriram, K., and Whitt, W., 1986. Characterizing superposition arrival processes in packet multiplexers for voice and data. IEEE Journal of Selected Areas in Communications. (Sep 1986). 833-846.
- [18] Shah-Heydari, S., Le-Ngoc, T., 1998. MMPP modeling for aggregated ATM traffic. In Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering. 129-132.
- [19] Klemm, A., Lindemann, C., Lohmann, M., 2003. Modeling IP traffic using the batch markovian arrival process. Performance Evaluation. 54, (2003). 149-173.
- [20] Bu, T., and Towsley, D., 2001. Fixed point approximation for TCP behaviour in an AQM network. ACM SIGMETRICS.
- [21] Baccelli, F., Hong, D., and Liu, Z., 2001. Fixed Point Methods for the Simulation of the Sharing of a Local Loop by a Large Number of Interacting TCP Connections. INRIA research report Number: RR-4154.
- [22] Bonald, T., and Proutière, A., 2004. On performance bounds for the integration of elastic and adaptive streaming flows. SIGMETRICS/Performance (2004).
- [23] Delcoigne, F., Proutière, A., and Regnié, G., 2004. Modelling integration of streaming and data traffic. Performance Evaluation, 55 ,3-4 (2004), 185-209.
- [24] Abate, J., Choudhury, G.L., Whitt, W., 1995. Exponential approximations for tail probabilities in queues, I: waiting times. Operations Research, 43, 5 (1995).