The Impact of Jitter on Traffic Flow Optimization in Communication Networks

Hamza Dahmouni, André Girard, Mohamed Ouzineb, and Brunilde Sansò

***Abstract*—Current network planning and design methods use the average delay, packet loss and throughput as metrics to optimize the network cost and performance. New multimedia applications, on the other hand, also have critical jitter requirements that are not taken into account by these methods. Here, we explore the impact on the network performance of adding these jitter constraints. We use a fast jitter calculation model to solve the optimal routing problem for flows subject to jitter or delay constraints. We find that the optimal routing is very different for the two kinds of flows: They should be routed on different paths, the jitter-constrained flows should not be split on multiple paths while the opposite conclusion is true for delay-constrained flows.**

***Index Terms*—Delay, jitter model, optimization, design, traffic engineering, IP network planning, QoS, multimedia services.**

I. Introduction

NETWORK operators must provide services in the most cost-effective way while dealing with the fast changes of technologies and increasing traffic growth, especially the dramatic increase of multimedia applications. For many real-time applications, such as video conferencing or games, the playback buffer cannot be large because of the fast response needed by the application. In these cases, buffer over- and underflow must be avoided and packet jitter can have a greater impact on the quality than the delay or loss. For these services, controlling packet jitter is an essential part of QoS. In past and recent work on network design, the average network delay and throughput have usually been used as metrics [1], [2] and there is little work where jitter is taken into account in the models. This raises the first question

> *How can we design a network routing at all if we want to take into account both delay and jitter?*

Once we have answered this question by providing a computationally feasible optimization model, we can look at a second issue

> *How much does jitter change the traffic flow allocation and in what conditions does this happen?*

Manuscript received May 7, 2011; revised January 16, 2012. The associate editor coordinating the review of this paper and approving it for publication was R. Stadler.

H. Dahmouni is with the Institut National des Postes et Télécommunications, 2, Avenue Allal El Fassi, Madinat Al Irfane, 10000, Rabat Maroc (e-mail: dahmouni@inpt.ac.ma).

A. Girard is with INRS-EMT and GERAD 800, de la Gauchetière O Suite 6900, Montreal, Qc, Canada H5A 1K6 (e-mail: andre.girard@gerad.ca).

M. Ouzineb is with the Institut National de Statistique et d'Economie Appliquée and GERAD, B.P.:6217 Rabat-Instituts, Rabat, Maroc (e-mail: Mohamed.Ouzineb@gerad.ca).

B. Sansò is with Ecole Polytechnique de Montréal and GERAD Electrical Engineering Department, CP 6079 succ Centre-Ville, Montreal, Qc, Canada H3C 3A9 (e-mail: brunilde.sanso@polymtl.ca).

Digital Object Identifier 10.1109/TNSM.2012.051712.110148

The results of this work show that there is a complex relationship between jitter and delay and that planners and operators will eventually have to carefully manage the assignment of traffic to routes paying close attention to which measure, delay or jitter, is the most important.

In order to gain some insight on this relationship, we first describe in section II the network model that we have used and discuss some issues that arise from this choice. Section III contains a brief literature review of related work on jitter calculation for ATM networks and some of our own work in the area as well as recent trends in routing and traffic engineering. The section also contain a specific list of the contributions of this paper. We then describe in section IV a summary of the fast jitter model that we derived for Poisson traffic and the approximation for the end-to-end jitter. The optimization model is presented in section V. We then show in section VI some numerical results for the optimal routing of a single flow. Similar results for two flows are presented in section VII. Finally, section VIII concludes the paper.

II. The Network Model

We want to get some *qualitative* insight into the effect of jitter on network design. For this reason, we use a simple extension of the classical work of [3]. This is a network of queues where we want to compute the path flows that will optimize a *performance* function such as the average end-to-end delay or jitter. We also allow the possibility of having upper bound constraints on these functions to guarantee a minimum amount of QoS. In this paper, we call this flow allocation a *routing*. We are mostly interested in the impact that the jitter can have on the optimal routing in a network.

A. Computation Requirements

Routing optimization problems are often modelled as large nonlinear multi-commodity flow programs where the nonlinearity comes from the delay or jitter functions which can appear either as objectives or as a set of QoS constraints. In this context, the performance functions have to be computed a very large number of times for the calculation of the gradients and also during the line search. Obviously, the first requirement of the performance functions is simplicity and fast evaluation. Accuracy is also needed but it is not very useful to have a very accurate model if it requires such a large computation time that it will make it impossible to solve the optimization problem in a reasonable time.

This need for speed is a strong requirement that severely limits the kind of traffic that can be analyzed. As discussed in section III, we have not found any model with realistic traffic

processes that can be calculated within the times required for optimization. Right now, the only thing that seems to be fast enough is a Poisson model so that in all that follows, we assume that the network is made up of M/M/1 queues. We also assume that the propagation delay is negligible so that the end-to-end packet delay is only the queuing delay.

B. The Poisson Model

It should be clear that we do not claim that Poisson traffic is an accurate model for real IP traffic. We *do* claim that a Poisson model can provide meaningful insight to *compare* the effect of some parameter on network performance. If we want to compare the effect of jitter on routing, the actual values of the jitter may not be very accurate but there is a *qualitative* difference in the routing of traffic depending on whether the constraint is delay or jitter. This is due to the different shape of the two performance functions and we believe that the differences we observe in the routing should also be present with other traffic models.

Another area where a Poisson model may be useful is related to the modelling of network traffic by hierarchical MMPP processes [4] where the packet process is represented by a Poisson process during the ON period of a session. An analytic model such as the one that we are presenting here could be used within a decomposition technique to compute the jitter of the hierarchical MMPP.

Finally, we have looked at actual measurements in the access network of a large ISP. We have found that traffic generated by several HTTP sources is a major component of the total traffic and that the downstream, upstream or total traffic can be very accurately approximated by an exponential distribution. This shows that the Poisson assumption *is* realistic in *some* kinds of access networks.

III. Related Work

A. ATM Jitter

There has been much work during the 1990's on the estimation of the cell delay jitter of a tagged stream in ATM networks. Most of the time, the jitter is calculated for discrete time processes and First Come First Served (FCFS) queuing. The results are based on queuing systems and assume that the tagged stream is originally periodic or a general renewal process. The jitter steady state distribution is derived in [5] for a periodic traffic stream by assuming a Markovian structure for the cell delay process. The authors of [6], [7] used generating functions to estimate the end-to-end jitter of a general renewal stream in heavy and light traffic. A similar approach based on discrete time queuing has been used in [8] to calculate end-to-end delay and jitter in packet networks.

The jitter probability density function of a renewal stream multiplexed with uncorrelated background traffic is derived in [9], [10]. An analytical approximation for the delay jitter first-order and second-order statistics incurred by a periodic traffic is proposed in [11]. The authors of [12] provide a complete characterization of the jitter process when the tagged stream and the background traffic are constant bit rate. A simple analytical approximation for the delay jitter incurred by a periodic stream multiplexed with a background traffic and governed by a general renewal processes is described in [13]. Finally, a somewhat different approach is described in [14] where a real-time algorithm is proposed to shape the traffic in such a way that the jitter remains within some prescribed bounds. This kind of technique is not suitable for use in a design algorithm.

There have been some proposals for delay jitter models in DiffServ networks. An extension of [12] was proposed in [15] to evaluate the per-class jitter. The authors of [16] provided some analysis of the delay jitter by means of event driven simulations where Expected Forwarding (EF) flows are represented by renewal periodic ON-OFF flows.

Most of these methods concentrate on the analysis of the jitter incurred by a tagged *periodic* cell stream going through nodes of an ATM network where the service time has a *deterministic* distribution. Furthermore, the computation time is large which makes them unsuitable as a component of a network optimization tool. As such, they are not suitable for use in design algorithms for IP networks.

B. Our Previous Work on Jitter Evaluation

We presented in [17] a formula for the jitter of Poisson traffic in a single queue that can be quickly calculated. This formula provides an analytical expression for the delay-jitter as a function of the traffic load, bandwidth and latency. Furthermore, we have showed that the jitter incurred in a network node is bounded by the packet average transit delay and the packet average service time.

We then extended this model in [18] to calculate an approximate expression for the end-to-end jitter along a path in a tandem queueing network. This takes into account the correlation between processes at tandem queues which have a significant impact on the end-to-end jitter. This work has produced some important insights for network design procedures. Typically, we found that the jitter is more important at the first multiplexing node and decreases as the correlation between successive packets increases. We also showed that under some conditions, jitter can decrease as the load increases, in strong contrast with other QoS measures such as delay or loss. We also found that the jitter on a path depends on where the more congested links occur on that path.

We validated these results in [19] where we analyzed and measured the jitter on Pre-WiMax technology for real-time applications in realistic environments. We confirmed that the jitter has a behavior opposite that of the delay and also showed that the jitter and packet loss behave the same way in both a static or mobile environment. Furthermore, the qualitative behavior of the jitter, throughput and delay predicted by the analytical model was confirmed by live measurements at various speeds.

C. Routing and Traffic Engineering

There is a large body of work [20], [21], [22], [23], [24] devoted to the planning of traditional communication networks such as ATM and MPLS. There has also been much work on the design and evaluation of traffic-aware routing [25]. Some work [26], [27] is focused on traffic engineering problems where the objective is to determine a set of paths and link loads

optimizing the total network delay while the network topology and the link capacities are known. Other work deals with on-demand models [28], [29], [30] where the objective is to maximize the network capacity in the presence of new traffic requests. In general, the average network delay and throughput have usually been used as metrics [1], [2] to optimize the network cost and performance.

More recently, given the increasing difference in the QoS needs of current services, there have been new proposals for routing and Traffic Engineering in IP and MPLS networks. The type of streams that are multiplexed in MPLS tunnels are optimized in [31] to minimize a measure of traffic distortion, while guaranteeing the appropriate QoS for the different traffic streams. In pure IP routing, on the other hand, the concept of a multi-topology routing has been proposed in [32], [33] where multiple OSPF topologies are overlaid in the IP network. This architecture can be used to route traffics with different requirements, as it was proposed in [34], where two OSPF topologies are exploited for improved performance in normal and failure conditions.

While this work provides an implementation framework for routing and traffic engineering, it has not dealt specifically with the behavior of traffic when the trade-off of jitter versus delay is optimized in routing and traffic engineering models. This is precisely the objective of this article.

D. Contributions

The main results presented here can be summarized as follows. The first one has already been presented in [17] and is repeated here for the sake of completeness. We also present simulation results for items 2 and 3 below.

1) For a single queue, jitter does *not* go to zero at low load but tends to the service time.
2) On a path, the end-to-end jitter tends to the sum of the individual jitters at low load.
3) On a path, increasing the load *decreases* the end-to-end jitter due to correlation between the queues
4) In a network, flows with a jitter constraint should be routed on a single path while flows with a delay constraint should be split between paths
5) In a network, it may be impossible to meet a reasonable set of delay and jitter constraints by rearranging the flows

An important goal of traffic engineering tool is to find the optimal routing of packets. It is known that routing traffic flows over a single path can induce traffic imbalances leading to poor network utilization [35]. Splitting traffic flows across a large number of paths can improve the overall network utilization but it cannot ensure an optimal network performance [36]. For these reasons, we need an optimization model to get the best compromise between these two extreme solutions. In the present work, we design an optimization algorithm of IP networks that takes into account the end-to-end jitter constraints. We address, in particular, the problem of determining the number of paths required to carry the traffic demand and meet the end-to-end QoS requirements. From this, we can deduce the load on each link which provides information about the network elements that are close to or exceed their usage limit and can cause congestion. The model can also be used to determine whether a set of delay and jitter constraints can be met simply by routing the flows or whether some other action is needed, such as increasing the capacity of links or even adding new links.

TABLE I
NOTATIONS FOR THE JITTER MODEL

Variable	Comment
λ	The total arrival rate $\lambda = \sum_{m=1}^{K} \lambda_m$
μ	The total service rate $\mu = \sum_{m=1}^{K} \mu_m$
η	The sojourn time rate
λ_k	The arrival rate of the tagged flow
τ_k	The mean inter-arrival time of the tagged flow $= 1/\lambda_k$
λ_0	The background arrival rate $\lambda_0 = \lambda - \lambda_k$
T_j	The delay of packet j in the queue
ρ	The total traffic load given by $\rho = \lambda/\mu$

IV. Analytical Jitter Model

We now recall the main results of [17] that provide a simple formula for the end-to-end delay jitter that is fast enough to be used in a network planning and design tool. Our definition of jitter is that of the Internet Engineering Task Force (IETF) as proposed in [37]. This requires the specification of two measurement points and the jitter is the difference in time taken by two consecutive packets to go from one to the other.

A. Single Queue

We first consider a single queue with infinite buffer and a First Come First Serve (FCFS) discipline. There are K streams of packets arriving to this queue, each with possibly different parameters for the inter-arrival λ_m and service time μ_m distributions. We are interested in a particular stream k called the *tagged* stream. The measurement points are the entry to the buffer and the exit from the server. A set of variables used in this work are given in Table I.

First, suppose that $\lambda_k \ll \lambda_0$. In this case, two packets of stream k are separated by a large number of packets from the other streams so that we can assume that T_{j+1} and T_j are two independent random variables with a negative exponential distribution with parameter $\eta = \mu - \lambda$. We can then use a well known property of exponential distributions [38] and the jitter is given by

$$J = E\left[|T_{j+1} - T_j|\right] = \frac{1}{\eta} \tag{1}$$

and if $\lambda \approx 0$, then the jitter $J \approx 1/\mu$ is nearly equal to the service time. It is easy to see that this result is correct if we consider the case where $\lambda_0 = 0$. At low load, the transit time of a packet i is essentially the service time S_i, which is exponential. But because packets arrive rarely, the queue will be empty at most arrival times. Given that an empty queue is a regeneration point, we see that the random variables T_i and T_{i-1} are independent. The jitter is then given by $J = |S_i - S_{i-1}|$ and we know that the distribution of the difference of two independent exponential variables is a symmetric Laplace distribution. The distribution of the absolute value is thus exponential with parameter μ which explains why the mean is the holding time.

Next consider the case where $\lambda_k \approx \lambda$. In this case, we can neglect the presence of the other streams and consider that we have a queue with only one flow. Its jitter is given by

$$J = E\left[|T_{j+1} - T_j|\right] = \frac{1}{\mu}. \tag{2}$$

Finally, we consider the case where the arrival rate of the tagged flow takes intermediate values between the two extreme points. We have shown in [17] that the end-to-end jitter J^k of a tagged flow k produced by a single node can be approximated to a very good accuracy by

$$J^k \approx \frac{1}{\eta} f(\tau_k, \eta) \tag{3}$$

where

$$f(\tau_k, \eta) = 1 - e^{-\eta\tau_k}\left(\eta\tau_k + e^{-\eta\tau_k}\right). \tag{4}$$

B. Path Jitter

We now consider the case where the tagged flow goes through N tandem nodes. The two measurement points are the entry of the first buffer and the exit from the last server on the path. All the parameters of section IV-A are now indexed with a superscript n to indicate the node. For instance, we let $\lambda_k^{(n)}$ be the arrival rate of packets of flow k, $\lambda_0^{(n)}$ the arrival rate of the background traffic and $\mu^{(n)}$ the service rate at node n. We then have $\rho^{(n)} = \lambda^{(n)}/\mu^{(n)}$ and $\eta^{(n)} = \mu^{(n)} - \lambda_0^{(n)}$.

Let $T_j^{(n)}$ be the delay of packet j through node n. The path jitter of the tagged flow k is given by

$$J_k^{(N)} = E\left[\left|\sum_{n=1}^{N} T_{j+1}^{(n)} - T_j^{(n)}\right|\right]. \tag{5}$$

In order to gain some insight on the behavior of path jitter, we have simulated a linear network of 5 links. There is no background traffic and all the links have the same rate. We increase the value of λ_k progressively and we plot on Fig. 1 the value of the jitter normalized to the holding time at each node. The 95% confidence interval is about 2% of the values and does not show up on the plot. We have one curve for the measurements made at each node: The curve labelled 1 corresponds to the jitter at node 1, etc. up to the curve labelled 5 for the jitter at node 5. The most striking feature of these curves is that contrary to intuition, the end-to-end jitter *decreases* as the load increases. This effect is very weak at the first node, as can be seen from the lower curve on the plot, and gets more important as we observe the jitter further away on the path. We can also see on the plot that jitter itself increases as the number of nodes increases and that this effect is more important at low load.

A second important feature is that path jitter does *not* go to zero at low load. As the load decreases, packets will arrive less often but the *difference* may be, and in fact is, still quite large. In fact, we can see that for low load, the path jitter is close to the sum of the jitter at each node in the path. This can be explained by the fact that for small traffic, the queues are almost independent and the absolute value of the sum in (5) is well approximated by the sum of the absolute values. At high load, on the other hand, the queues become strongly coupled and we see that the path jitter is close to the jitter at a single node. Clearly, the correlation between the queues has a strong impact and the absolute value of the sum of jitters cannot be approximated by the sum of absolute values.

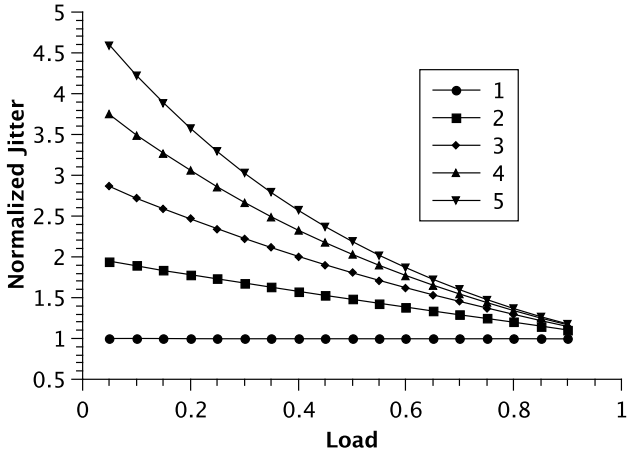

Fig. 1. Simulation of a 5-link path.

These results suggest a simple approximation of (5) that takes into account the correlation of tandem queues. We assume that the jitter at some node n on the path is given by

$$J_k^{(n)} = \sum_{i=1}^{n} \frac{1}{\eta^{(i)}} K^{(i)} f\left((\tau_k^{(i)}, \eta^{(i)}\right) \tag{6}$$

where $0 \leq K \leq 1$ is a reduction factor that has to be estimated. We make the further assumption that this reduction is related to the auto-correlation function E of the queue length $q_i^{(n)}$ at the time packet i arrives at node n by the relation

$$K(\lambda_k^{(n)}, \eta^{(n)}) = 1 - E\left[q_i^{(n)} q_{i+1}^{(n)}\right]. \tag{7}$$

After some algebra, we get the auto-correlation function for packets of flow k by [17]

$$E\left[q_i^{(n)} q_{i+1}^{(n)}\right] = \frac{1}{\left(L^{(n)}\right)^2 + \left(R^{(n)}\right)^2} \times \left(\left(L^{(n)}\right)^2 + \frac{\lambda_k^{(n)}\left(R^{(n)}\right)^2}{\lambda_k^{(n)} + \mu/\left(R^{(n)}\right)^2}\right) \tag{8}$$

where the mean queue length and square length are given by

$$L^{(n)} = \frac{\rho^{(n)}}{1-\rho^{(n)}} \quad \text{and} \quad \left(R^{(n)}\right)^2 = \frac{\rho^{(n)}}{(1-\rho^{(n)})^2}$$

Setting $K^{(1)} = 1$, we can then replace (8) and (7) in (6) to get the approximate value for the path jitter.

We have compared the model with the simulation results in Fig. 2 where we plot the jitter measured in the simulation and the calculated values as a function of load. The jitter values are normalized to the average holding time.

The important result is that the model reproduces the decreasing behavior of jitter with increasing load. This is the main feature that we want to capture since we expect that this will strongly influence the network design. We can also see that it overestimates the jitter at low load. This is not surprising since $E(q)$ goes to 0 as $\rho \to 0$ so that the

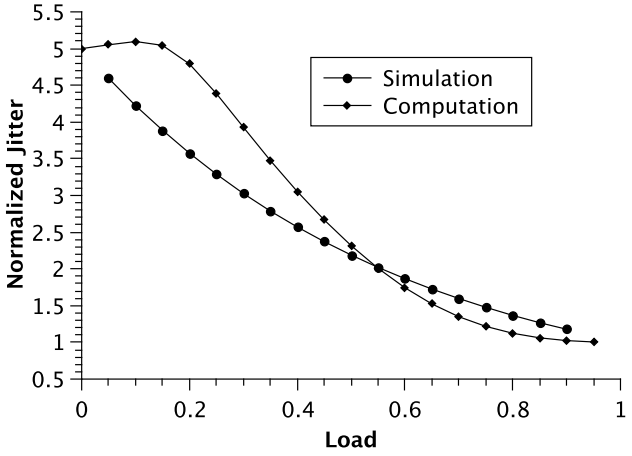

Fig. 2. Accuracy of end-to-end models.

jitter values simply add up on the path. Overestimation is a desirable feature of an approximation since it will produce a conservative design where the actual jitter will be lower that what has been calculated. There is a slight under-estimation at high load but the accuracy is reasonably good for our purpose here which is to gain qualitative insight on the effect of jitter.

V. Routing Optimization Model

To investigate the effect of jitter on network design, we consider the classical routing optimization problem where each user can use a given number of paths to send its traffic over a fixed network structure. Jitter can be taken into account in two different ways: if an application needs strict QoS requirements, we set an upper bound on the amount of jitter that is allowed. On the other hand, if the application would benefit from a lower jitter but does not have strict requirements, it is enough to add a term for the jitter in the objective function. The model that we present in this section takes into account both cases.

We assume that multiple flows between a pair of edge nodes can send different amounts of traffic on different paths between these nodes. Our goal in this study is to show the difference between the solutions produced by an optimization based on the delay as opposed to one based on jitter.

A. Optimization Model

The routing optimization problem is generally modelled as a nonlinear multi-commodity flow program. These can be defined either with a path or a link formulation based on the choice of the independent variables. In this work, we have used both formulations depending on the solution method. For simplicity, we state the problem using the path formulation. The link formulation can be derived by standard methods.

In the path formulation, the independent variables are $X_p^{o,d}$, the amount of flow of commodity (o,d) on path p where a commodity is generally defined by an origin-destination pair (o,d). If d_u is the average packet delay on link u, then the average delay for commodity (o,d) on a path p is given by

$$D_p^{o,d} = \sum_u A_{p,u}^{o,d} d_u$$

Next, we define the weighted average delay for commodity (o,d) as

$$D^{o,d} = \frac{1}{\lambda^{o,d}} \sum_p X_p^{o,d} D_p^{o,d}$$

We also define the total weighted delay as

$$D = \frac{1}{\gamma} \sum_{o,d} \lambda^{o,d} D^{o,d}$$

where $\gamma = \sum_{o,d} \lambda^{o,d}$. These definitions are all based on the fact that the total delay over a path is the sum of the delay on the links in the path. We can give similar definitions for the jitter provided that we have this additive property as well. This is the case for traffic model (6). In that case, we get similar relations for the path, commodity and network jitter

$$J_p^{o,d} = \sum_u A_{p,u}^{o,d} J_u$$

$$J^{o,d} = \frac{1}{\lambda^{o,d}} \sum_p X_p^{o,d} J_p^{o,d}$$

$$J = \frac{1}{\gamma} \sum_{o,d} \lambda^{o,d} J^{o,d}$$

For the M/M/1 queue, the link average delay and jitter are given by

$$d_u = \frac{1}{C_u - X_u} \quad \text{and} \quad J_u = \frac{f(X_u^{o,d}, \eta_u)}{\eta_u}$$

where

$$X_u = \sum_{o,d} X_u^{o,d} \tag{9}$$

$$X_u^{o,d} = \sum_p X_p^{o,d} A_{u,p}^{o,d} \tag{10}$$

Our objective is to show the trade-off between jitter and delay when optimizing routing. This is examined not only through the objective function, but also by the effect of the jitter and delay constraints. For this, we propose the general routing problem which is to find a set of path flows $X_p^{o,d}$ that will minimize a convex linear combination of the total delay and jitter subject to commodity constraints on these values

$$\min_X F(X_u^{o,d}) = (1-\alpha)D + \alpha J \tag{11}$$

$$\sum_p X_p^{o,d} = \lambda^{o,d} \tag{12}$$

$$J^{o,d} \leq \overline{J}^{o,d} \tag{13}$$

$$D^{o,d} \leq \overline{D}^{o,d} \tag{14}$$

$$X_p^{o,d} \geq 0. \tag{15}$$

The objective function (11) represents the trade-off between delay and jitter: It is represented by the parameter α. Equation (12) guarantees that all the traffic entering the network reaches its destination. Eqs. (13) and (14) are the jitter and delay upper bounds. Finally, (15) insure that the flow variables are always positive.

To summarize, the known parameters used for input to the problem are given in Table II. The decision variables are given in Table III and the intermediate variables that are derived either from the parameters or the decision variables are defined in Table IV.

TABLE II
PARAMETERS USED IN THE OPTIMIZATION PROBLEM

Variable	Comment
N	Number of nodes
L	Number of links
(o,d)	Origin/Destination pair
$\lambda^{o,d}$	The arrival rate of traffic for node pair (o,d)
$A_{u,p}^{o,d}$	The link-path incidence matrix. An element is equal to 1 if link u is in path p between pair (o,d) and zero otherwise.
γ	The total packet arrival rate in the network
C_u	The transmission rate, or capacity, of link u
$\overline{D}^{o,d}$	Upper bound on the average delay for (o,d) packets
$\overline{J}^{o,d}$	Upper bound on the jitter for (o,d) packets
α	Weight parameter for the jitter in the objective function

TABLE III
DECISION VARIABLES

Variable	Comment
$X_p^{o,d}$	the amount of (o,d) traffic on path $p \in P^{o,d}$

B. Solution Methods

We have seen that the jitter is a complex function of the flow variables. In order to understand its effect on network optimization, consider the regular network of Fig. 3 where we allow only two paths between a single pair of (o,d). The total traffic between the nodes is given by λ. Because of (12), the routing problem then reduces to splitting this traffic on the two paths so that the objective (11) is minimized. Let $0 \leq r \leq 1$ be the fraction of λ that is sent on path 1 so that

$$X_1^{o,d} = r\lambda \quad \text{and} \quad X_2^{o,d} = (1-r)\lambda.$$

We have plotted the jitter and the delay as a function of r on Fig. 4. It is clear that while the delay shows the expected convex form, the jitter appears to be concave. It is known that network delay is a convex function of a flow rate [39], [40]. On the other hand, a formal proof of concavity appears difficult for the jitter but this result shows clearly that the routing problem with jitter is definitely not convex.

It is expected that computing an optimum solution using some global optimization method will not be possible for networks of realistic size. Instead, we have used two different techniques to get good solutions that stand a reasonable chance of being close to optimal.

The first one is to use a standard nonlinear solver, in the present case, Minos [41]. Solvers of this kind can only find local optima and they can be used to get solutions of non-convex problems by computing local solutions by starting the algorithm with a large number of different initial solutions.

TABLE IV
AUXILIARY VARIABLES

Variable	Comment
$X_u^{o,d}$	The amount of (o,d) traffic carried on link u
X_u	The total traffic carried on link u
$D^{o,d}$	The weighted average packet delay per (o,d)
d_u	The packet average transmission delay at link u. We assume that $d_u = d_u(X_u)$.
D	The network weighted average delay over all (o,d) pairs
$J^{o,d}$	The weighted average jitter per (o,d)
J_u	The total traffic carried on link u
J	The network weighted average jitter over all (o,d) pairs

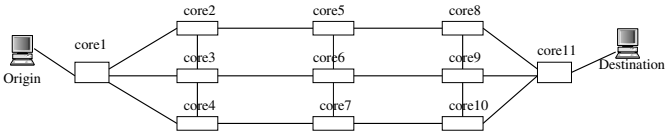

Fig. 3. Regular network II.

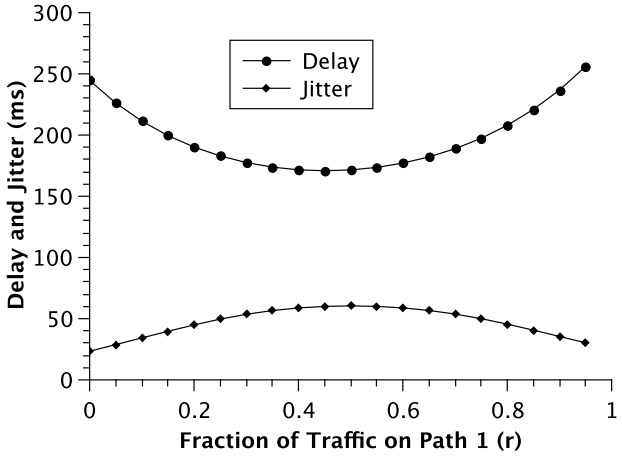

Fig. 4. Jitter and delay as a function of flow.

Another technique is to use some meta-heuristic that can also give good approximations to the global optimum. Here, we chose to solve it with a fast tabu search heuristic that gives very good results. This is described in [42] and consists of two parts. First, we divide the search space into several disjoint subsets. In each subspace selected we then use Tabu Search [43] to find a good solution in these subsets. This algorithm combines the strong points of the two parts. In the first part, we explore the more important regions in the total search space. In the second part, the role of TS is to intensify the search of the global optimum in the regions which seem to be more interesting.

In the first step, the algorithm generates randomly an initial population of solutions. Increasing the size of the population allows better solutions, but slows down the convergence of general algorithm. Each individual in the population is represented as string of finite integer numbers and represents only one region in the space search. At each iteration, one individual is randomly selected and we use TS to produce a new solution. We then decode the new solution and discard it from the population. Equivalent solutions are also eliminated. If the population contains only one solution, we then regenerate randomly a new set of solutions.

C. Robustness and Accuracy of the TS Algorithm

The first question that arises when one uses a meta-heuristic is how good are the solutions that it produces. For the routing problem, we can show that in some cases, the TS heuristic is in fact optimal. Consider the case where there is no jitter. We know that the delay is a convex function of the flow so that the routing problem is convex and in that case, Minos will find a global optimum. For this, we have solved the optimal routing problem for the network of Fig. 3 and also for the

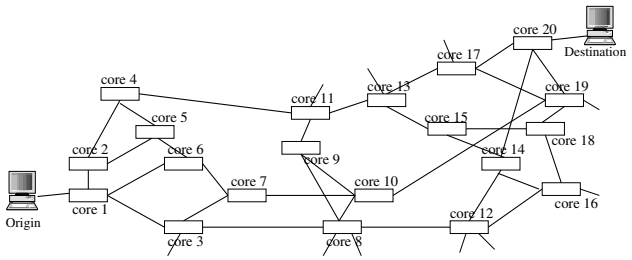

Fig. 5. Large network: ARPA topology.

TABLE V
SOLUTIONS PRODUCED BY TS AND MINOS MINIMIZING THE DELAY ONLY

	TS		Minos	
Network	*Delay*	*Jitter*	*Delay*	*Jitter*
Regular Network II without QoS const.	82.1572	36.7966	**82.1572**	36.7966
Regular Network II with delay const.	82.1572	36.7966	**82.1572**	36.7966
ARPA Network without QoS const.	95.2348	45.841	**94.548**	45.1467
ARPA Network with delay const.	95.2348	45.841	**94.548**	45.1467

classical topology of the ARPA network shown in Fig. 5. All links have a capacity of 128 and the traffic between nodes 1 and 11 is 81. The delay constraint is set at 130.

We have compared in Table V the results produced by TS and Minos both for the case without and with delay constraints. It is quite clear that Tabu Search can find solutions that are very close to the optimal.

When the jitter is included in the objective function, the optimization problem is no longer convex. We can evaluate the quality of the results produced by TS by comparing them with the best solution produced by Minos out of 100 runs using different random initial solutions. The objective is the value of the jitter and we have set the constraint values to $\overline{D}^{o,d} = 120$ ms and $\overline{J}^{o,d} = 20$ ms for the regular network. For the ARPA network, the constraint values have been set to $\overline{D}^{o,d} = 130$ ms and $\overline{J}^{o,d} = 30$ ms.

Table VI shows the comparison results between TS and Minos. Here again, we see that TS gives solutions that are very close to the ones produced by Minos which is a good indication that we may be producing very good, if not provably optimal solutions.

TABLE VI
THE COMPARISON RESULTS BETWEEN TS AND NONLINEAR MINOS MINIMIZING THE JITTER ONLY

	TS		Minos	
Network	*Delay*	*Jitter*	*Delay*	*Jitter*
Regular Network II without QoS const.	82.1572	**36.7966**	82.1572	36.7966
Regular Network II with QoS const.	119.837	**15.9035**	127.7	13.75
ARPA Network without QoS const.	148.936	**14.9388**	148.936	14.9388
ARPA Network with QoS const.	128.393	**20.9328**	130	20.36

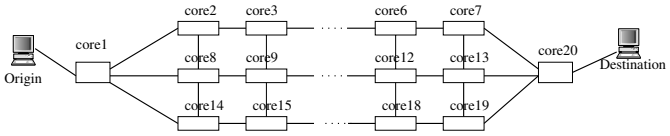

Fig. 6. Regular network III.

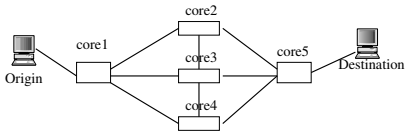

Fig. 7. Regular network I.

VI. SOFT QOS REQUIREMENTS

In this section, we investigate the impact of jitter on the routing when we need to provide only soft QoS constraints. In the present case, we would like to have a delay of about 150 ms and a jitter of about 30 ms more or less but we do not insist that these limits be strictly enforced. To mark the difference with a truly constrained problem, we use the term *target* to denote the desired performance values. The corresponding flow optimization problem is given by dropping constraints (13–14) from problem (11–15). In general, this problem is not convex except when $\alpha = 0$. In all that follows, the term "optimal" solutions should be understood as meaning "the best solution found by Minos or the tabu algorithm" starting with 100 different initial solutions chosen randomly

We use four architectures representing different topologies and sizes, and a number of scenarios with different values given to the weight α of the jitter term in the objective function. We can then examine how the optimal routing changes in each scenario and get some insight on the impact of jitter on routing.

A. Regular Networks, Single Flow

The conflicting impact of delay and jitter can be easily seen for so-called *regular* networks. They have the general structure shown on Fig. 6 with a varying number of intermediate stages. The network of Fig. 3 belongs to this class as well as the smaller network of Fig. 7. We have also used a fourth regular network with 10 intermediate stages, called Network IV, which is not shown here to conserve space. All links have the same capacity $C_u = 128$, and we consider a single (o, d) flow from node 1 to the node further to the right in each case. Unless otherwise noted, we optimize the routing for three values of $\lambda_k = 10, 51, 96$ corresponding to loads $\rho = 0.08, 0.4, 0.75$.

1) Optimal Flow Allocation: The first result has to do with the very different flow allocations depending on the objective function. First, note that all solutions route traffic on the min-hop paths between the two end nodes and that the cross links like (2,8), (8,14) etc., are never used. A typical set of results is presented in Table VII for the network of Fig. 3 with $\lambda_k = 81$ which corresponding to $\rho = 0.60$ and we solve problem (11–15) for increasing values of α, that is, we give increasing importance to the jitter in the objective.

In this table, we show the fraction of the total traffic routed on the three paths $p_1 = (1, 2, 5, 8, 11)$, $p_2 = (1, 3, 6, 9, 11)$

TABLE VII
TRAFFIC ROUTING AS A FUNCTION OF α

α	Traffic split		
	p_1	p_2	p_3
0.0	0.333	0.333	0.333
0.1	0.333	0.333	0.333
0.2	0.333	0.333	0.333
0.3	0.333	0.333	0.333
0.4	0.333	0.333	0.333
0.5	0.471	0.451	0.078
0.6	0.510	0.490	0.000
0.7	0.844	0.078	0.078
0.8	0.980	0.020	0.000
0.9	1.000	0.000	0.000
1.0	1.000	0.000	0.000

and $p_3 = (1, 4, 7, 10, 11)$. For low values of α, when we are minimizing delay, we get the familiar result that traffic should be split among paths with equal marginal delay. In the present case, since the only traffic is from 1 to 11, the three min-hop paths have equal marginal delay and the solution is to split the traffic equally among these paths. For high values of α, on the other hand, the situation is quite the opposite and we see that all traffic is sent on a single path. This is due to the fact that the path jitter is reduced when the load increases so that it is better to concentrate traffic on a single path to take advantage of the packet correlations.

We have obtained similar results for all cases at different loads: they are not presented in order to conserve space. When the load is not very low, optimizing for the delay produces solutions with split traffic flows while optimizing for jitter produces solutions on a single path.

2) Low Load: With this in mind, we can now examine table VIII showing the results of the optimization for different loads and jitter weights. As we can see, for the three networks at low load, the jitter becomes nearly equal to the average delay. This can be seen for a single queue from (1) where $1/\eta$ is in fact the average time in system and for low λ, $J \approx 1/\mu$. We have also seen from Fig. 1 that at low load, the path jitter is simply the sum of the jitter at the queues on the paths which reduces to the sum of the service times. In this case, the best solution for jitter is simply to send traffic on a min-hop path to the destination.

The optimal solution for the delay is also straightforward. At low load, there is no queuing so that the path delay is simply the sum of the service times at the queues on the path. Here too, the optimal solution is to send the flow on any one of the min-hop paths to the destination. An important conclusion is that

At low load, delay and jitter optimization produce the same solution

This explains the numerical values that we get for $\rho = 0.08$. It is clear that the optimal value of delay will be the same as that for jitter, that this is independent of α and depends only on the topology of the network. In the present case, we get $J \approx 7.8$ on each link. In the first three networks, for the case $\alpha = 1$, the flow goes on paths with 4, 6 and 9 links which corresponds to the jitter values of $4 \times 7.8 = 31$, $6 \times 7.8 = 47$ and $9 \times 7.8 = 70$ at low load. When we optimize for delay, on the other hand, we see that we get the same value for the delay as for the jitter as expected.

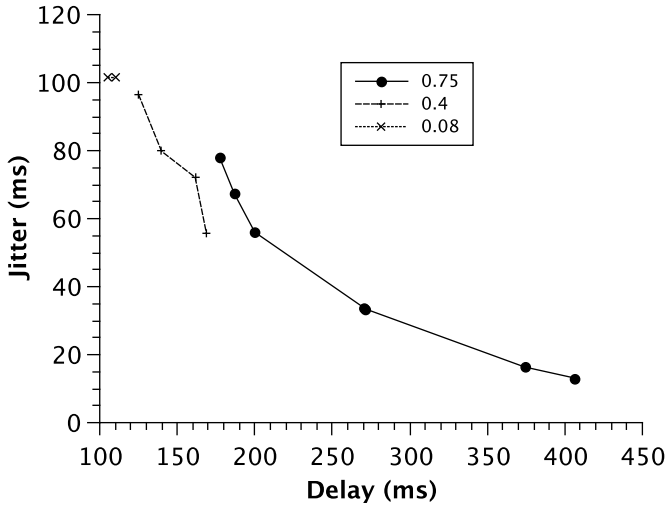

Fig. 8. Undominated solutions, network IV.

3) Higher Loads: As the load gets higher, the situation changes dramatically because the path jitter is now less than the sum of the link jitters on the path and it decreases with the load. The delay, on the other hand, increases with the load so that at higher loads, jitter and delay optimization work against each other. We see from the table that optimizing for delay yields low delay, as expected, and somewhat smaller jitter than for the low load case. The main difference is when we optimize for jitter, where we can decrease the optimal value substantially from what it was at low load. This, however, comes at the cost of a large increase of the delay, in the order of 50%. In other words, delay and jitter work at cross-purposes and it seems that improving one can only be done at the expense of the other. A second conclusion is then

When the load is significant, jitter and delay cannot be reduced together

We also present in Table VIII some results for the intermediate value $\alpha = 0.5$. We see that this has the desired effect of producing solutions with delays and jitter between the two extreme values obtained for $\alpha = 1$ and $\alpha = 0$. Consider the case where $\rho = 0.4$ and $\alpha = 0.5$. Comparing the three networks, we can see that it gets increasingly harder to provide both jitter and delay targets as the network gets larger. For network III, with $\alpha = 0.5$, we get an average jitter of 51.6 which is rather high when compared with a desired value of about 30. We can still reduce the jitter by choosing $\alpha = 1$ but we wee that in this case, the delay is increasing significantly. This particular solution is still acceptable but we see that for Network IV with 10 intermediate stages, it is not possible to provide the required levels of performance for both the delay and the jitter at the same time. This is mostly due to the fact that we cannot decrease the jitter near its target value of 30 even if we are willing to pay the cost with a delay of almost 170, quite above its target value of 150.

The main conclusion that we can draw from this is that providing good jitter is fundamentally different from providing good values for other performance measures such as delay or loss. In this case, improving delay, for instance, will also

TABLE VIII
ROUTING OPTIMIZATION FOR REGULAR NETWORKS, SINGLE FLOW, NO CONSTRAINTS

Network Topology	Traffic Load (ρ)	Delay (ms)			Jitter (ms)		
		$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
Regular Network I	0.08	32.9919	32.9919	33.8983	31.2407	31.2407	31.2237
	0.40	43.992	46.6794	51.9481	27.1919	24.1668	19.7721
	0.75	83.3333	83.4053	125	20.9773	20.9037	09.1396
Regular Network II	0.08	49.0347	49.0347	50.8475	46.8651	46.8651	46.8312
	0.40	62.0101	67.3849	77.9221	42.5847	36.5345	27.7451
	0.75	104.167	104.311	187.5	33.6997	33.5525	10.0245
Regular Network III	0.08	73.8597	73.8597	73.8597	70.297	70.29	70.29
	0.40	92.9961	102.517	116.883	64.9644	51.6036	39.7047
	0.75	143.593	150	281.25	49.1401	38.6475	11.351
Regular Network IV	0.40	125	139.8	168.8	96.5	79.8	55.7

improve loss and vice-versa. This is no longer true with jitter where improving delay can, and generally will, degrade jitter and vice-versa. In other words, another conclusion is that

> *the optimal routing problem really becomes a multi-criteria problem with two objectives conflicting with each other.*

Here, there is no "best" solution but only a series of compromises between delay and jitter, some of which may be satisfactory, or maybe none at all.

This is most clearly shown by a Pareto surface in the jitter-delay plane, as shown on Fig. 8 for Network IV. On this figure, we plot, for 3 values of load, the set of undominated solutions to the routing problem when α ranges from 0 to 1. A solution s_1 with values of jitter and delay J_1, D_1 is said to be dominated by some other solution s_2 with values J_2, D_2 if $J_2 < J_1$ and $D_2 < D_1$. In other words, there is no reason to use s_1 since s_2 does better both in terms of jitter and delay. An undominated solution is a solution that is not dominated by any other solution. We clearly see that the jitter target of 30 ms cannot be attained at any load. At low load, the surface is trivial since there are only two optimal solutions which are basically determined by the length of the min-hop path. At middle load $\rho = 0.4$, we see that some tradeoff is possible but that there is no solution that will meet both targets. The high load solution for $\rho = 0.75$ has a large tradeoff region but again the target values clearly fall below the boundary. In other words, for this network, it is not possible to find a routing that will meet the target for both the delay and the jitter at the same time and this, at all loads.

B. Regular Networks, 2 Flows

Minimum delay is often the most important objective function for many real-time IP applications such as VoIP, VoD, etc. For this reason, we choose an objective function that minimizes the average network delay without constraints on the QoS parameters

We can see the routing on Fig. 9 where the traffic is split on 3 paths. Each of the two flows is split on different paths as expected when optimizing a multi-commodity flow with a convex function. This is similar to the results we had obtained with a single flow.

In some cases, jitter is a more important measure of quality of service than the network delay. In this case, we consider multipath routing to minimize the average network jitter by setting $\alpha = 1$ in the objective function. The optimal routing

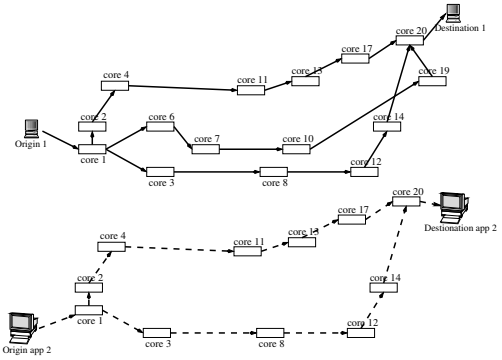

Fig. 9. Delay objective function ($\alpha = 0$) without constraints.

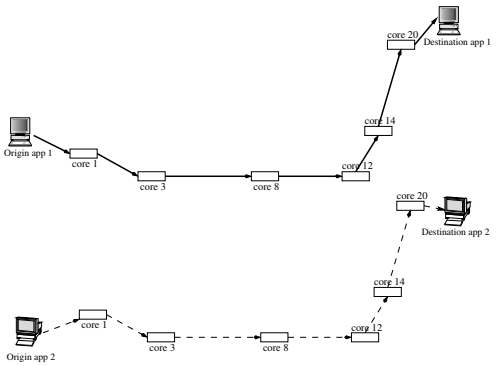

Fig. 10. Jitter objective function ($\alpha = 1$) without constraints.

is shown on Fig. 10 where we see that the routing is very different from the case of delay minimization: Each flow is routed on a single path, again a result consistent with optimizing a multi-commodity flow with a concave function.

C. ARPA Network, Background Flows

We can check that this separation of traffic on different paths occurs with more complex traffic patterns using the ARPA network of Fig. 11 with two traffic flows with $\lambda = 20$ both from node 1 to node 20. In this network, we have added what we call *background traffic* to the links. For a given link i, j, this is traffic that originates at node i and is completely carried on the link where it exits at node j. This simulate the presence of flows from other origin-destination pairs. The size of this background traffic is represented by the thickness of the links in the figure. This traffic ranges from 64 to 102 with an average load of 35%.

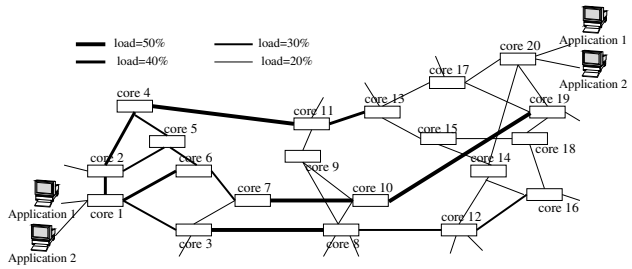

Fig. 11. ARPA network with background traffic load.

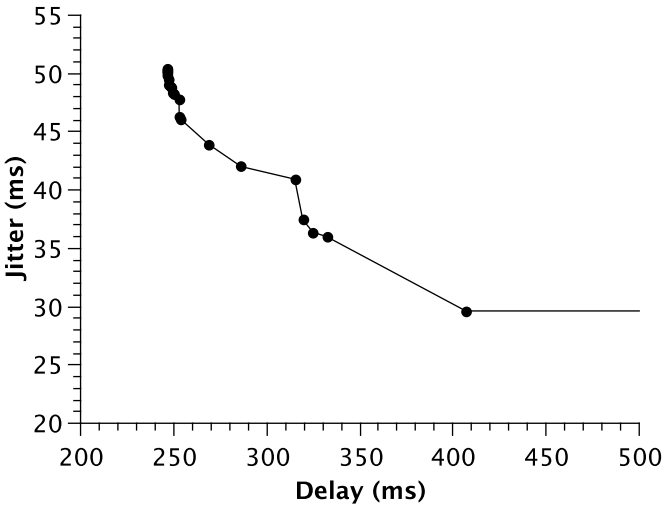

Fig. 12. Undominated solutions, ARPA networks with background traffic.

The Pareto region for this network is shown on Fig. 12. The region extends far away to the right where there exist 3 solutions close to each other. The one with the smallest jitter is at coordinates (7983.74, 24.3069). This is obviously for $\alpha = 1$ and the optimization algorithm has been able to find a very good solution as far as jitter is concerned, which is what was asked for, but at the cost of a very large delay, which is fine since we have decided that delay does not matter. Looking at this solution, we see that all the (s_2, d_2) and about 92% of the (s_1, d_i) flows have been sent on the same path (1, 3, 8, 12, 14, 20) to reduce jitter as much as possible. The total traffic on the links is never below 100 and on link (1, 3), it is 127.87 which by itself accounts for about 1 second of the large end-to-end delay.

D. ARPA Network with Converging Flows

We now consider a somewhat more complex flow pattern to understand better the interaction between jitter and delay. This is done for the ARPA network with $C = 128$. We set up a converging flow pattern where we have two sources for each of nodes 1 to 7, each with $\lambda = 10$, and the destination of all these flows is node 20. This mimics the case where a number of applications are trying to access a server or where a node is used as gateway to some other network. It also forces some interactions between the flows near the gateway but leaves a lot of room in the left part of the network since on these links, the traffic is much smaller than the capacity. We also assume

TABLE IX
END-TO-END QOS, ARPA NETWORK, NO CONSTRAINTS, $\alpha = 0$

Flow		Delay	Jitter
s1	d1	71.663	56.3749
s2	d2	71.663	56.3749
s3	d3	71.663	56.3749
s4	d4	71.663	56.3749
s5	d5	62.4037	48.5036
s6	d6	62.4037	48.5036
s7	d7	62.4037	48.5036
s8	d8	62.4037	48.5036
s9	d9	81.6897	58.0956
s10	d10	81.6897	58.0956
s11	d11	72.4304	50.2243
s12	d12	72.4304	50.2243
s13	d13	61.0668	42.0024
s14	d14	61.0668	42.0024

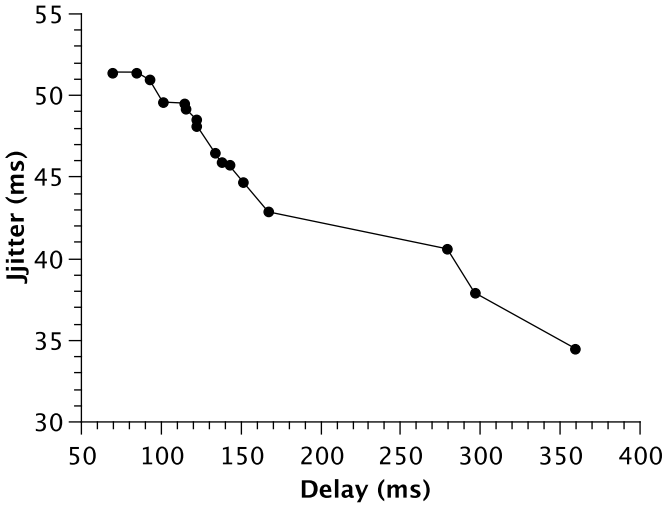

Fig. 13. Undominated solutions, ARPA network, converging flows.

that all flows have targets of 150 ms for the delay and 30 ms for the jitter. Again, these are realistic values that are often proposed for real-time services. What we want is to find a routing that will meet these requirements.

The first thing one might want to ask is whether the routing optimized for the delay only could meet the jitter requirement. The results of Table IX clearly show that this is not enough: The delay is much smaller than the requirement for each flow but the jitter is not very good with values as high as 56 for some flows. Clearly, we need to do something about the jitter. The next easiest solution is to set α to some value greater than 0 in the hope of reducing the jitter. We can then compute a number of solutions for different values of α ranging from 0 to 1. The results have been summarized in Fig. 13 where we have plotted the undominated solutions in the (delay, jitter) plane.

The figure shows clearly the tradeoff between jitter and delay. More importantly, however, is the fact that it is very unlikely that we can reach both targets at the same time. We can get some insight into the structure of the solution from table X where we give the total link flow on all the links that carry some flow either in the delay- or jitter-optimized solution. We can see that the optimization algorithm tries to

TABLE X
LINK FLOWS, ARPA NETWORK, CONVERGING FLOWS

Link		$\alpha = 0$	$\alpha = 1$
n1	n3	20	20
n2	n5	0	20
n2	n4	20	0
n3	n8	40	0
n3	n7	0	40
n4	n11	40	20
n5	n6	20	40
n6	n7	40	60
n7	n10	60	120
n8	n12	40	0
n10	n19	60	120
n11	n13	40	20
n12	n14	40	0
n13	n17	40	20
n14	n20	40	0
n17	n19	40	0
n17	n20	40	20
n19	n20	60	120

TABLE XI
QOS VALUES, ARPA NETWORK, CONVERGING FLOWS, $\alpha = 1$

Flow		Delay	Jitter
s1	d1	412.572	36.008
s2	d2	412.572	36.008
s3	d3	427.278	44.8034
s4	d4	427.278	44.8034
s5	d5	403.313	28.1367
s6	d6	403.313	28.1367
s7	d7	53.9862	47.1013
s8	d8	53.9862	47.1013
s9	d9	418.019	36.9321
s10	d10	418.019	36.9321
s11	d11	406.655	28.7103
s12	d12	406.655	28.7103
s13	d13	391.949	19.9149
s14	d14	391.949	19.9149

concentrate as much traffic as possible in order to reduce jitter, especially on the links from node 7 to the sink. This of course increases the load, and hence the delay, on these links with the ensuing poor delay. The actual QoS values for all the flows are shown on table XI where we can see that some flows get a jitter better than the target but with a very large delay value.

The results of Table XI illustrate very well how jitter and delay work against each other. This can be seen from Table XII which shows the actual routing for each flow. The first two columns contain the source and destination numbers and the remaining columns show the nodes in the path used to route the flows. First consider flows 7 and 8 which both have a relatively low delay and somewhat high jitter in Table XI. We can see that these flows are the only ones routed on the path $(4, 11, 13, 17, 20)$ so that all the links carry only a flow of 20 and the delay is correspondingly low. Note however that the load on this path is quite low and as we have seen, jitter tends to be higher when the load is low, as is the case here. This explains why the jitter values are higher than all the others in Table XI.

We can also see from Table XII that the algorithm tried to concentrate all the other flows as much as possible. They are all routed to node 7 from which they all use the partial path $(7, 10, 19, 20)$. This produces a high load on the links and reduces the jitter for these flows, in some cases well below the 30 ms target. Note however that this comes at the expense of a high delay of about 400 ms for all these flows. In other words, we can concentrate traffic to reduce jitter but this will cause an increase in delay, or spread out traffic, which will reduce delay but increase jitter. Depending on the flow patterns, it may not be possible to do both at the same time, as is the case here and something else needs to be done beyond adjusting the flows, like increasing the link capacity or opening new links.

TABLE XII
OPTIMAL ROUTING, ARPA NETWORK WITH CONVERGING FLOWS, $\alpha = 1$

Flows	Nodes on Path						
1,2	1	3		7	10	19	20
3,4	2	5	6	7	10	19	20
5,6	3			7	10	19	20
7,8	4	11	13	17	20		
9,10	5		6	7	10	19	20
11,12			6	7	10	19	20

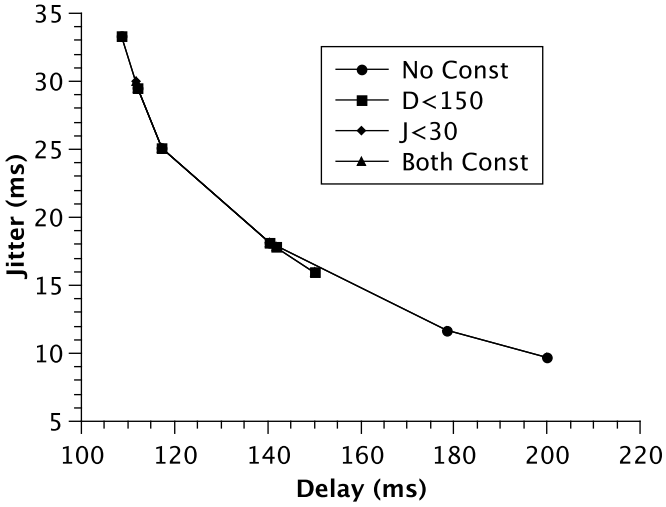

Fig. 14. Delay and jitter variation, regular network II with constraints.

VII. HARD QOS REQUIREMENTS

The discussion of section VI shows that, up to a certain point, it is possible to improve the overall QoS by optimizing flows. This produces solutions where flows get a delay and a jitter that are not too far from some desired target. In some cases, however, this is not enough since there may be situations where a strict guarantee is needed. This can happen in the context of a SLA that specifies a service guarantee or for some applications, such as on-line games, that become unusable if the delay exceeds a certain value. These cases can be taken into account by adding hard constraints like (13–14) to the optimization model to represent the end-to-end delay and jitter requirements.

A. Regular Network with One Flow

We examine the effect of constraints on the delay and jitter on the optimal solution for Network II with one flow of $\lambda = 98$. We use the objective function (11) subject to the flow conservation equations (12). We have three different scenarios depending on the constraints that are activated. In the first one, we impose a 150 ms upper bound on the delay but drop the

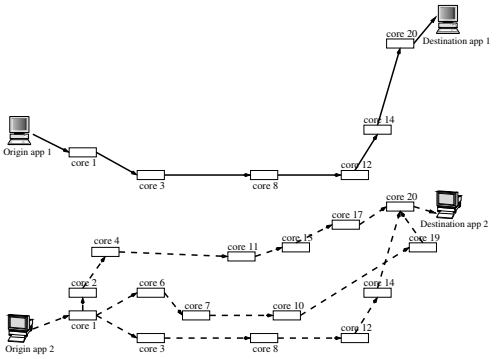

Fig. 15. Delay objective function $\alpha = 0$ with $D_2 \leq \overline{D}$ and $J_1 \leq \overline{J}$.

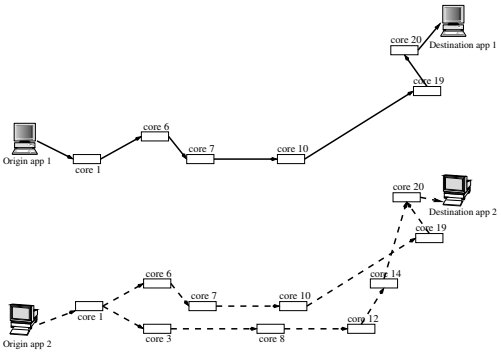

Fig. 16. Jitter objective function ($\alpha = 1$) with $D_2 \leq \overline{D}$ and $J_1 \leq \overline{J}$.

jitter constraint (13). The second one is the opposite where we drop the delay constraint and impose a jitter constraint of 30 ms. In the third case, we impose both constraints. In each case, we vary the jitter weight α and optimize the routing.

We have plotted on Fig. 14 the set of undominated solutions for the three cases in addition to the case where there are no constraints at all. The curve with the label "No Const" corresponds to the case without constraints. It extends to the whole range of values from $D = 109$ to 200. The other curves lie on the no-constraint curve but only for the portion allowed by the constraints.

B. ARPA Network with 2 Flows

We now examine the actual routing for a somewhat more complex scenario. We use the ARPA network with background loads of Fig. 11 with a traffic of 81 for each flow. We want to see how the flows are routed under different scenarios of constraints and objectives. We know that using a delay or a jitter objective will produce very different routings when there are no constraints. In the case where we have a constraint, say jitter, that is different from the objective, say delay, we want to see what kind of routing will be produced by the optimization.

First, we set $\alpha = 0$ and impose a jitter upper bound of 30 ms on flow 1. We find that the routing of flow 1 is on a single path, as expected from the jitter constraint, and not on multiple paths as expected from the objective. Next, we set $\alpha = 1$ and impose a delay upper bound of 130 ms on flow 2. In this case, the routing of flow 2 is split on 3 paths as a consequence of the delay constraint.

Finally, we consider two objective functions that minimize either the delay, $\alpha = 0$ or the jitter, $\alpha = 1$. We put both upper bounds as above on the jitter of flow 1 and the delay of flow 2. This models a situation where a user has two applications running at the same time, each with hard QoS constraints but of a different kind.

The results of Fig. 15 show the routing when we optimize for delay and Fig. 16 for jitter. Here too the routing is determined mostly by the constraints rather than the objective function. The delay-constrained flow is split on 3 paths while the jitter-constrained one is carried on a single path. Still, there is *some* effect of the objective function as can be seen on Fig. 16 where the delay-constrained flow is carried on only 2 paths as opposed to 3 when we are optimizing for delay.

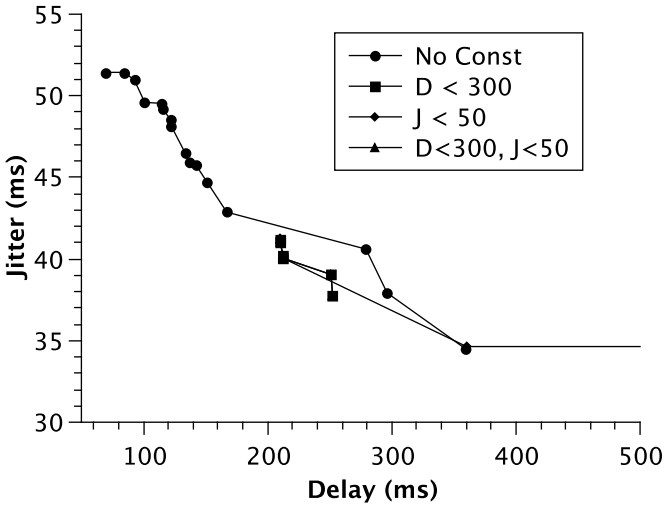

Fig. 17. Converging flows with constraints.

Finally, we examine the effect of constraints on the solutions for the ARPA network with converging flows. Here we assume that at each node, one of the two sources may have a delay constraint of 300 ms and the other may have a jitter constraint of 50 ms. We show in Fig. 17 the set of undominated solutions for the case when none of the constraints are enforced, when only the delay constraints are imposed, when only the jitter constraints are present and finally, for both sets of constraints at the same time.

First note that the curves corresponding to the constrained problems are *below* the curve without constraints. This is due to the fact that the two solution techniques we use here do not guarantee a global optimal solution since the problem is not convex. The curves simply indicate that the algorithm missed some solutions in the unconstrained case which were found when the constraints were imposed. In effect, the points on the lower curves should also be part of the unconstrained curve since they also represent undominated solutions of the unconstrained problem.

With this in mind, we can see that for this network, imposing the constraints severely restricts the range of undominated solutions, as expected, but that within that range, the objective is about the same as with an unconstrained problem.

VIII. Conclusion

We have given a short summary of a model for computing path jitter for a network of M/M/1 queues. We have then proposed an optimization problem for optimizing the routing of packets where the objective can be weighted towards delay or jitter and where the jitter and delay QoS constraints can be imposed on each flow separately. We have then investigated how the presence of jitter can impact the optimal routing on regular networks with a limited number of flows.

The main finding is that routing for jitter is qualitatively different from routing for delay. In the first case, flows are concentrated on a single path while they are split among 2 or 3 paths when optimizing for delay. This is related to the concavity of the jitter and the convexity of delay as a function of flow. This difference would be less striking if a single-path constraint were added to the delay optimization problem since both types of flows will be on a single path. The question is then whether the two paths would be the same since they are computed with different metrics.

We have also found that when two applications are present with different QoS requirements, the optimal solution is to route them *separately* depending on the QoS that is imposed, We have also seen that the constraint has a stronger impact on the solution than the objective.

The limitations of this work are mainly due to the assumptions made on the type of traffic. In that respect, it would be interesting to investigate by simulation the jitter of non-Poisson traffic sources assigned to routes calculated with the M/M/1 assumption for different types of trade-offs in the objective function. Our educated guess is that the differences for jitter and delay will be even more striking in that case.

This work has shown that unless networks are seriously over-dimensioned, jitter optimization can no longer be ignored or taken for granted in network planning and traffic engineering. This is increasingly important for wireless where bandwidth is severely limited by the available spectrum. Our findings suggest that network planners and operators will have to pay a much closer attention to the way individual traffic streams are assigned to routes, discriminating not only on the *type* of traffic, but also on the more important performance measure for that type, making the case for more intelligent heterogeneous routing.

Acknowledgement

This work has been supported by the canadian National Science and Engineering Research Council CRD grant CRDPJ 335934-06 and STPG grant 365205-08.

References

[1] T. Hoang, "Planning and optimization of multi-service computer networks," in *Proc. 2007 Communications and Networking Simulation Symposium*.

[2] P. Barreto and P. Carvalho, "Network planning optimization for multimedia networks," in *Proc. 2008 IEEE International Symposium on Network Computing and Applications*, pp. 60–67.

[3] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: an approach to store-and-forward communication network design," *Networks*, vol. 3, pp. 97–133, 1973.

[4] L. Muscariello, M. Mellia, M. Meo, M. Ajmone Marsan, and R. Lo Cigno, "An MMPP-based hierarchical model of internet traffic," in *Proc. 2004 IEEE International Conference on Communications*, pp. 2143–2147.

[5] J. Roberts and F. Guillemin, "Jitter in ATM networks and its impact on peak rate enforcement," *IEEE/ACM Trans. Networking*, vol. 16, pp. 35–48, 1992.

[6] W. Matragi, C. Bisdikian, and K. Sohraby, "Jitter calculus in ATM networks: single node case," in *Proc. 1994 IEEE INFOCOM*, vol. 1, pp. 232–241.

[7] W. Matragi, K. Sohraby, and C. Bisdikian, "Jitter calculus in ATM networks: multiple node case," *IEEE/ACM Trans. Networking*, vol. 5, pp. 122–133, 1997.

[8] O. Osterbo, "A discrete time queueing model for end-to-end delay and jitter analysis," in *Proc. 2009 International Teletraffic Congress*.

[9] R. Landry and I. Stavrakakis, "Traffic shaping of a tagged stream in an ATM networks: approximate end-to-end analysis," in *Proc. 1995 IEEE INFOCOM*.

[10] ——, "Study of delay jitter with and without peak rate enforcement," *IEEE/ACM Trans. Networking*, vol. 5, pp. 529–539, 1997.

[11] C. Fulton and S. Li, "Delay jitter first-order and second-order statistical functions of general traffic on high-speed multimedia networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 141–149, 1998.

[12] A. Privalov and K. Sohraby, "Per-stream jitter analysis in CBR ATM multiplexors," *IEEE/ACM Trans. Networking*, vol. 6, pp. 141–149, 1998.

[13] O. Brun, C. Bockstal, and J. Garcia, "Analytical approximation of the jitter incurred by CBR traffics in IP networks," *Telecommun. Systems*, vol. 33, 2006.

[14] Y. Mansour and B. Patt-Shamir, "Jitter control in QoS networks," *IEEE/ACM Trans. Networking*, vol. 9, no. 4, pp. 492–502, Aug. 2001.

[15] J. Chung and H. Soo, "Jitter analysis of homogeneous traffic in differentiated services networks," *IEEE Commun. Lett.*, vol. 7, 2003.

[16] H. Alshaer and J. Elmirghani, "Expedited forwarding end-to-end delay and jitter in DiffServ," *International J. Commun. Syst.*, vol. 21, pp. 815–841, 2008.

[17] H. Dahmouni, A. Girard, and B. Sansò, "Analytical jitter model for IP network planning and design," in *Proc. 2009 Comnet*.

[18] ——, "An analytical model for jitter in IP networks," *Annals of Telecommun.*, vol. 67, pp. 81–90, Jan. 2012. Available: http://www.springerlink.com/content/5046104335872460/

[19] H. Dahmouni, H. Elghazi, D. Bonacci, B. Sansò, and A. Girard, "Improving QoS of all-IP generation of pre-WiMax networks using delay-jitter model," *J. Telecommun.*, vol. 2, no. 2, pp. 99–103, 2010.

[20] B. Sansò and P. Soriano, editors, *Telecommunications Network Planning*, ser. Center for Transportation Research 25th Anniversary Series. Kluwer Academic Publishers, 1998.

[21] M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufman, 2004.

[22] D. Medhi and K. Ramasamy, *Network Routing*. Morgan Kaufman, 2007.

[23] F. Mobiot, B. Sansò, and A. Girard, "Design of reliable IP/GMPLS networks: an integrated approach," *J. Network and Systems Management*, vol. 13, no. 1, pp. 77–97, Mar. 2005.

[24] B. Sansò, A. Girard, and F. Mobiot, "Integrating reliability and quality of service in networks with switched virtual circuits," *Computers & Operations Research*, vol. 32, no. 1, pp. 35–58, Jan. 2005.

[25] A. Sridharan, S. Bhattacharyya, C. Diot, R. Gufin, J. Jetcheva, and N. Taft, "On the impact of aggregation on the performance of traffic aware routing," in *Proc. 2001 International Teletraffic Congress*.

[26] D. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Commun. Mag.*, vol. 37, no. 12, pp. 42–47, Dec. 1999.

[27] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. 2000 IEEE INFOCOM*, vol. 2, pp. 519–528.

[28] G. Apostolopoulos, R. Guérin, S. Kamat, and S. Tripathi, "Quality of service based routing: a performance perspective," in *Proc. 1998 ACM SIGCOMM*.

[29] G. Apostolopoulos, R. Guérin, S. Kamat, and S. Tripathi, "Improving QoS performance under inaccurate link state information," in *Proc, 1999 International Teletraffic Congress*.

[30] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next generation high speed networks," *IEEE Network*, vol. 12, pp. 64–79, 1998.

[31] S. Srivastava, A. V. de Liefvoort, and D. Medhi, "Traffic engineering of MPLS backbone networks in the presence of heterogeneous streams," *Computer Networks*, vol. 53, pp. 2688–2702, 2009.

[32] P. Psenak, S. Mirtorabi, A. Roy, L. Nguyen, and P. Pillay-Esnault, "Multi-topology (MT) routing in OSPF," IETF RFC 4915, 2007.

[33] S. Bae and T. Henderson, "Traffic engineering with OSPF multi-topology routing," in *Proc. 2007 Military Communications Conference*, pp. 1–7.

[34] K.-W. Kwong, R. Guérin, A. Shaikh, and S. Tao, "Balancing performance, robustness and flexibility in routing systems," *IEEE Trans. Network Service Management*, vol. 7, no. 3, pp. 186–199, Sep. 2010.

[35] R. Kuhn and S. M. Mostafavi, "Optimal routing policies," *IEEE Commun. Lett.*, vol. 12, pp. 222–224, 2008.
[36] S. M. Mostafavi, E. Hamadani, and R. Tafazolli, "Delay minimization in multipath routing," in *Proc. 2010 International Wireless Communications and Mobile Computing Conference*.
[37] S. Poretsky, J. Perser, S. Erramilli, and S. Khurana, "Terminology for benchmarking network-layer traffic control mechanisms," IETF RFC 4689, Oct. 2006.
[38] P. S. Puri and H. Rubin, "A characterization based on the absolute difference of two iid random variables," *The Annals of Mathematical Statistics*, vol. 41, no. 6, pp. 2113–2122, Dec. 1970.
[39] D. Bertsekas and R. Gallager, *Data Networks*, 2nd edition. Prentice-Hall, 1992.
[40] M. J. Neely and E. Modianoa, "Convexity in queues with general inputs," *IEEE Trans. Inf. Theory*, vol. 51, no. 2, pp. 706–714, Feb. 2005.
[41] B. A. Murtaugh and M. A. Saunders, "MINOS 5.4 users guide," Department of Operations Research, Stanford University, Stanford, CA 94305 USA, Tech. Rep. SOL 83-20R, Dec. 1993.
[42] M. Ouzineb, H. Dahmouni, B. Sansò, and A. Girard, "Optimizing jitter and delay using heuristic methods," in *Proc. 2010 CIRO*.
[43] F. W. Glover and M. Laguna, *Tabu Search*. Kluwer Academic Publishers, 1998.

Hamza Dahmouni received the PhD degree in Networks and Computer Science from Telecom Bretagne, France in 2007. He obtained a M. Sc in Networks from Paris-IV University in 2003, and a M. Sc in Wireless network design from Telecom SudParis, France in 2002. He is an Associate Professor of Institut National des Postes et Télécommunications (INPT), Rabat, Morocco. In 2004-2007, he worked as research engineer at France Telecom. His research interests are related to traffic engineering and performance evaluation in heterogeneous networks.

André Girard received the Ph.D. degree in physics from the University of Pennsylvania, Philadelphia, in 1971. He is an Honorary Professor with INRS-EMT and an Adjunct Professor with École Polytechnique of Montréal, QC, Canada. His research interests all have to do with the optimization of telecommunication networks, and in particular with performance evaluation, routing, dimensioning, and reliability. He has made numerous theoretical and algorithmic contributions to the design of telephone, ATM, IP and wireless networks.

Mohamed Ouzineb received the Ph.D. and M.Sc degrees in Computer Science and Operations Research; Ph.D. degree from University of Montreal, Canada in 2009; and M.Sc degree from Paris-VI University, France in 2003. He is an Associate Professor of National Institute of Statistics and Applied Economics (INSEA), Rabat, Morocco. His research interests deals with the development of efficient metaheuristics for solving difficult problems related to transportation and telecommunication planning.

Brunilde Sansò is a full professor of electrical engineering at École Polytechnique de Montreal and director of the LORLAB. Her interests are in performance, reliability, design, and optimization of wireless and wireline networks. She is a recipient of several awards, Associate Editor of Telecommunication Systems, and editor of two books on planning and performance.