Routing with neural-based QoS constraints

Christophe Duhamel, Antoine Mahul, Alex Aussem
LIMOS - CNRS UMR 6158
Address : 63173 Aubière Cedex, France.
Phone: +33473405000, fax: +33473405001
Email: {christophe.duhamel, antoine.mahul,alex.aussem}@isima.fr

Abstract

Broadband Integrated Service Digital Networks are intended to allow a flexible integration of a wide spectrum of services, each service having its own traffic characteristics and specific quality of service (QoS) requirements. In this work, we consider the optimal routing problem in a telecommunications network with nonlinear QoS constraints for a set of demands. A lagrangean relaxation on the QoS constraints is performed and the resulting problem is solved by the flow deviation method. The latter requires an accurate approximation of the QoS in each output buffer at each resolution step. This is achieved by substituting the standard M/M/1 analytical formula for a neural network trained by simulation to provide a more realistic evaluation of the QoS. Experiments are being conducted at present and will be reported in due course.

Keywords: MPLS, Routing, Quality of Service, Neural Networks.

1 Introduction

Several techniques have been proposed by the IETF for quality of service QoS provisioning in the Internet. They are intended to allow a flexible integration of a wide spectrum of services (including data, voice and video traffics) with different traffic characteristics and very specific QoS requirements. Yet, this challenge is difficult to meet and congestion is still experienced due to variation in network conditions. The origin of this problem is to be found in the bursty character of the stream of data packets arriving at a buffer.

Multiprotocol Label Switching (MPLS) is being considered as one approach for scalable QoS provisioning. The fundamental idea of MPLS [14] involves assigning short, fixed length labels to the packets at the ingress point of the network. When packets are forwarded within an MPLS domain, the MPLS capable routers, termed as Label Switching Routers (LSRs), only examine the label rather than the IP header, leading to a significant increase in switching performance in terms of delays. Two primary problems of MPLS Traffic Engineering (MPLS-TE) are layout design and flow assignment. Working together with RSVP or DiffServ, MPLS-TE also provides a scalable QoS scheme. Typically, MPLS-TE classifies and groups the incoming packets into different traffic trunks, which are then mapped to so-called Label Switched Paths (LSPs) which can satisfy their QoS requirements with optimized network performance. Explicit routing in MPLS is used in traffic engineering to maximize the operational network performance and to provide Quality of Service (QoS).

In this paper, we focus on the problem of finding optimal routing with QoS constraints for a set of demands in a MPLS backbone. However, classical optimization algorithms, employed in MPLS-TE tasks in order to maximize network performance and balance traffic load, still rely on the assumption that the QoS in an MPLS node obeys the standard M/M/1 formula, which is not true. Switching gives rise to new problems in queueing theory that cannot be solved using the conventional mathematical tools of the queueing theory. We propose to estimate the QoS in an MPLS output interface made up from several queues, one for each class of traffic, by means of neural network.

Neural networks have been succesfully employed for traffic control in the literature: i) to predict the short-term behavior of the traffic [1, 10], ii) to determine the parameters of a source model that best match the statistical descriptors of a real bursty source [7], iii) to estimate the packet loss rate in a multiplexer to perform Call Admission Control (CAC) [10, 13], and bandwidth allocation [6, 12].

This work is supported by the French RNRT-OPIUM project ("Optimisation de la Planification des Infrastructures des reseaUx Mobiles") devoted to MPLS traffic engineering in mobile networks.

2 Problem Description

Let $G = (N, A, c)$ be a directed network, where N is the set of nodes (routers) and A is the set of arcs (connections). A capacity (bandwidth) c_a is associated to each arc $a \in A$. Let $\mathcal{K} = \{(o_i, d_i, q_i, s_i), i = 1 \dots K\}$ be the set of commodities (services) to be routed on G. For each commodity $k \in \mathcal{K}$, o_k is the origin node, d_k is the destination node, q_k and s_k are respectively the requested bandwidth and the class of service (CoS). The bandwidth requirement q_k is considered to be fixed, e.g. traffic is not subject to variations. $\mathcal{S} = \{s_i,\ i = 1 \dots S\}$ is the set of classes of service.
The are several distinct QoS criteria. In this paper, we restrict ourselves to the end-to-end delay defined, for each commodity, as the maximum delay incurred by a packet between the arrival time at d_k and the departure time at o_k. The problem is to route each commodity on the network in order to minimize the overall load on the network, defined as the overall sum of the arc loads, while respecting the all QoS constraints. Note that a somewhat different function could have been used, namely the maximal congestion on the network (i.e. the utilization factor on the most saturated arc) as discussed in [5]. This is left for future work.

3 Mathematical Formulation

The classical mathematical formulation for routing problems in telecommunications relies on the node-arc model. This model perfectly suits the local constraints, e.g. those based on the arcs. However, it is not adequate for the expression of global constraints such as the QoS constraints, since they are dealing with paths rather than with arcs. Therefore the arc-path model is prefered to the arc-node.
Let $\mathcal{P}_k = \left\{p_i^k, i = 1 \dots P_k\right\}$ be the set of all paths for commodity $k \in \mathcal{K}$. Thus, $\mathcal{P} = \cup_{k \in \mathcal{K}} \mathcal{P}_k$ denodes the set of all the paths. Each path $p \in \mathcal{P}$ can be expressed as a set of arcs. Let y_p^k be the decision variable to use path $p \in \mathcal{P}_k$ for commodity k. Let x_p^k be the percent of bandwidth of commodity k assigned to path $p \in \mathcal{P}_k$. The delay is usually considered as an additive function over the arcs. Therefore, the end-to-end delay for a path $p \in \mathcal{P}$ is given by the sum of delays on each arc a belonging to p. Let $\phi_s(x_a, c_a)$ and $\psi_s(x_a, c_a)$ be respectively a load and a delay function for CoS $s \in \mathcal{S}$ on arc $a \in A$, where x_a is the bandwidth vector for each CoS. The problem can be formulated as follows:

$$
\begin{array}{lll}
\text{Min} \quad \Phi(x) = \displaystyle\sum_{a \in A} \sum_{s \in \mathcal{S}} \phi_s(x_a, c_a) & & \\
\text{s.t.} & & \\
\quad \displaystyle\sum_{p \in \mathcal{P}_k} x_p^k = 1 & \forall k \in \mathcal{K} & (1) \\
\quad y_p^k - x_p^k \geqslant 0 & \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k & (2) \\
\quad x_a^s = \displaystyle\sum_{k \in \mathcal{K} | s_k = s} q_k \sum_{p \in \mathcal{P}_k | a \in p} x_p^k & \forall a \in A,\ \forall s \in \mathcal{S} & (3) \\
\quad d_p^k = \displaystyle\sum_{a \in A | a \in p} \psi_{s_k}(x_a, c_a) & \forall k \in \mathcal{K},\ \forall p \in \mathcal{P}_k & (4) \\
\quad d_p^k \leqslant \overline{D_{s_k}} + M\left(1 - y_p^k\right) & \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k & (5) \\
\quad x_p^k \in [0, 1] & \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k & \\
\quad y_p^k \in \{0, 1\} & \forall k \in \mathcal{K}, \forall p \in \mathcal{P}_k & \\
\quad x_a^s \geqslant 0 & \forall a \in A,\ \forall s \in \mathcal{S} &
\end{array}
$$

Constraint (1) requires all the bandwidth to be assigned on the paths of commodity k. Constraint (2) is linking bandwidth variables x_p^k with decision variable y_p^k. Constraint (3) computes the bandwidth used on arc a according to the bandwidth that is assigned on each path. Constraint (4) computes the end-to-end delay of each path. Finally, constraint (5) is the QoS constraint for each active path, e.g. for each path that is used to route some of the bandwidth. M is a constant high enough to ensure that a non-active path should not have its end-to-end delay constrained. $\overline{D_s}$ is the upper bound on the end-to-end delay for every commodity associated with CoS s.

4 Solving Strategies

A wide range of efficient methods are well-suited to solve multicommodity flow problems. Yet, most of them cannot be used for this problem since the load and the delay functions, $\phi_s(x_a, c_a)$ and $\psi_s(x_a, c_a)$, are nonlinear. It is possible to remove nonlinear constraints by performing a lagrangean relaxation on the QoS constraints. Therefore, the objective function becomes

$$\Phi(x,\lambda) = \sum_{a\in A}\sum_{s\in\mathcal{S}} \phi_s(x_a,c_a) - \sum_{k\in\mathcal{K},p\in\mathcal{P}_k} \lambda_p^k. \left(\overline{D_{s_k}} + M\left(1-y_p^k\right) - d_p^k\right)$$

where λ_p^k are the lagrangean coefficients associated with the relaxed contraints. Variables d_p^k are computed using constraints (4). The resulting problem has now a nonlinear objective function with linear constraints.

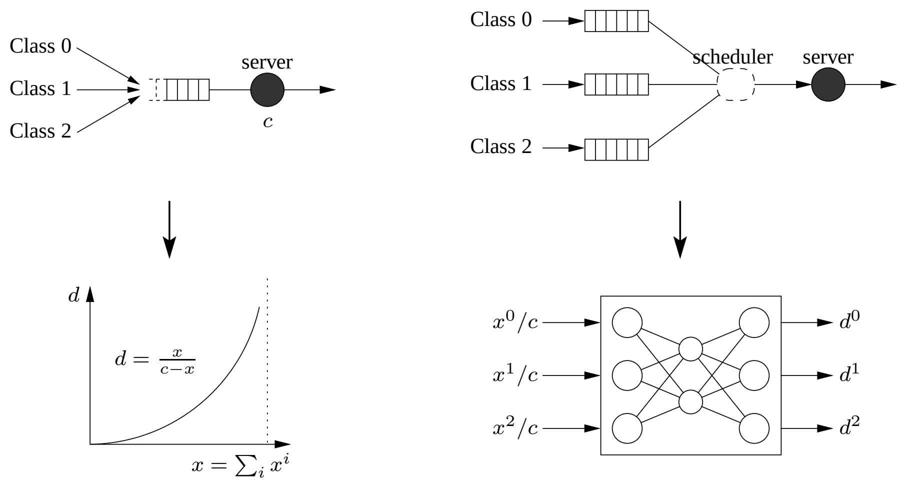
The lagrangean relaxation can be solved by any classical optimization method (subgradient, bundle method). The inner problem, e.g. once lagrangean multipliers λ have been set, will be solved by a specialization of Frank-Wolfe's method to network routing: the flow deviation method ([4, 8]). At each iteration, given the current solution x, a linearization of the objective function is performed and a cheaper routing path is computed for each commodity. Then, a fair amount of bandwidth is transferred from every active path commodity to this new set of paths and the process iterates until no improvement can be made.

We are proposing two ways to compute the load and the delay functions, $\phi_s(x_a,c_a)$ and $\psi_s(x_a,c_a)$ (see Fig. 1):

1. the first approach is based on an analytical formulation. It assumes strong hypotheses on the traffic (Poisson traffic, independant flows). Under such hypotheses,
   (a) the average load can be stated as $\widetilde{\phi_s}(x_a,c_a) = \dfrac{\sum_{s\in\mathcal{S}} x_a^s}{c_a - \sum_{s\in\mathcal{S}} x_a^s}$
   (b) the average delay is then $\widetilde{\psi_s}(x_a,c_a) = \dfrac{1}{c_a - \sum_{s\in\mathcal{S}} x_a^s} = \dfrac{\widetilde{\phi_s}(x_a,c_a)}{\sum_{s\in\mathcal{S}} x_a^s}$ by applying Little's theorem.

   Those functions have interesting properties: they are $\mathcal{C}^\infty$, convex and are acting as barrier functions on the arc capacity. It should worth noting that both $\widetilde{\phi_s}(x_a,c_a)$ and $\widetilde{\psi_s}(x_a,c_a)$ are computed on the aggregated flow, so that values are identical for each class of service.
2. the second approach is based on a more realistic neural network approximation ([2, 3]) of the QoS of $\phi_s(x_a,c_a)$ and $\psi_s(x_a,c_a)$. The neural model is trained by discrete event simulations. Using Little's theorem, $\overline{\phi_s}(x_a,c_a)$ may be computed as $\overline{\phi_s}(x_a,c_a) = x_a^s\,\overline{\psi_s}(x_a,c_a)$. By construction, these two functions are derivable; convexity is only assumed.

We now need to assess the usefulness of the neural-based method by inspecting the optimal routing obtained by the lagrangean relaxation problem.



(a) analytic model of delays

(b) neural model of delays

Figure 1: Two models of delays for an arc with capacity c and three classes of service.

5 Neural estimation of delays

In the section we detail our strategy and first results for the estimation of delays in an output interface of an MPLS router with neural networks.

During the optimization process, input traffics are only characterized by their bandwith. So, from these rates aggregated per class, a neural network will be able to estimate the mean delay for each class. This approach was successfully used with more sophisticated descriptors in [2] and [3].

We consider the input bandwith for the class $s \in \mathcal{S}$ and the arc $a \in A$:

$$x_a^s = \sum_{k \in \mathcal{K} | s_k = s} q_k \sum_{p \in \mathcal{P}_k | a \in p} x_p^k$$

As our neural network will be used for all arcs of the network, their inputs must be independent of capacity of the arc: we use the utilization factor of each class instead of the aggregated bandwith : $\rho_a^s = x_a^s / c_a$.
The neural network was trained to approximate the function: $\{\rho^s\}_{s \in \mathcal{S}} \rightarrow \{d^s\}_{s \in \mathcal{S}}$, where d^s is the average delay for the class s. The neural network was trained on exemples generated by simulation of a simplified model of a router's output interface.

A simplified model of output interface

A simple output interface of an MPLS router may be represented by means of several input buffers, one for each class, with First-In-First-Out (FIFO) queue policy (see Fig. 2). For simplicity, we consider only three traffic classes, each having its own buffer. Only the packet at the head of the FIFO competes for access to the router output server. The first class (class 0) is a priority class (Expedited Forwarding class, for real-time applications such as Voice over IP or Visioconference). The packets in class 0 take precedence over the packets belonging to the other classes: the packets always pass through the router (if the server is available at this time) while the others must wait until queue 0 is empty. Scheduler 1 implements the Priority Queueing algorithm. Note that while a packet is waiting its turn in the input queues 1 or 2, other packets behind the head-of-line are blocked. Obviously, packet waiting time is a function of the service policy that the switch uses when several input queues are waiting to transmit packets at the same time. The fair sharing of the server between the other two classes (Assured Forwarding, e.g. non-interactive data delivery) is achieved by the scheduler which implements the 'round robin' policy [9].

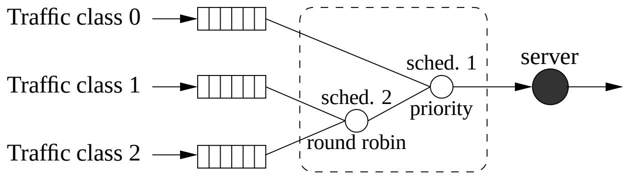


Figure 2: A schematic output card of an MPLS router conveying 3 classes of traffic.

The service time is assumed to be exponential, with the parameter $\mu = 1$ packet per time unit (we can make this assumption without loss of generality because any value of μ can be reduced to 1 by a simple time scale transformation). The buffers have an equal capacity of 100 packets.

Training of the neural network

The data sets used for training and validation purposes were obtained by discrete-event simulation and selected so that they cover as many traffic patterns as possible in a critical range where congestion is observed. The more exact the desired approximation is, the more data points are needed at the cost of extensive computation time. Since the simulation process is burdensome, the performance of the NNs were tested using a ten-fold cross-validation technique; 80% of the data was randomly selected for training and the remaining data was set aside for performance assessment. The target values, namely the average delay throughout the output interface of a router were estimated from the simulation. They are transformed with a simple logarithmic processing to improve generalization capabilities of the neural network.
The MLPs used in these experiments are trained by the standard back-propagation algorithm. Momentum, weight decay and an adaptive learning step were used to prevent local minima and avoid overfitting. A standard measure of fitting for the NN is given by the *normalized mean squared error*, that is the mean squared error divided by the empirical variance of the simulation values calculated over all the training patterns. A value of the NMSE = 1 thus corresponds to predicting the unconditional mean.
For the sake of simplicity, we assume first the traffic matches a Poisson distribution in each class. We have 3 Poisson sources, so only three parameters: $(\rho_i)_{i=0..2}$, utilization factors of each class of traffic, chosen randomly in the domain $[0, 1[$, and such as $\rho_0 + \rho_1 + \rho_2 \leqslant 1$. The input vector for neural network is simply (ρ_0, ρ_1, ρ_2). We estimate only the logarithm of the delay for each class. The neural network has 3 inputs, 15 hidden neurons and 3 outputs, that is 108 adjustable parameters. The results summarized below clearly show that the method works satisfactorily on this simple setting with well behaving traffic.

	class 0	class 1	class 2
delay	$5.0 \cdot 10^{-4}$	$1.2 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$

Table 1: Average NMSE for 3 Poisson classes.

6 Conclusion

The preliminary results presented here are useful in proving that the neural network approach provides reliable estimates of the delay in a MPLS router, using simple source models. Satisfactorily tested on simple settings, the approach shows promises for a variety of applications to QoS-based traffic control, routing and congestion avoidance in multiservice telecommunication networks. However, our model of delay can easily be improved by considering more realistic traffic models and by replacing the simple round robin algorithm by the WFQ scheduling in simulations of output interface.

We are currently comparing the results obtained by the optimization algorithm with the classical M/M/1 model of delay and the neural network one. We will also measure the gap from the solution computed using the discrete event simulation and will propose several ways to improve the quality of the approximations.

References

[1] A. Aussem, "Call admission control in ATM networks with the random neural network," *IEEE ICNN'94*, Vol. 4, pp. 2482-2487, 1994.

[2] A. Aussem, Sébastien Rouxel and Raymond Marie, "Neural-based queueing system modeling for service quality estimation in B-ISDN networks", *ICANN'99*, Edinburgh, Scotland, pp. 970-975, 1999.

[3] A. Aussem, A. Mahul and Raymond Marie, "Queueing network modelling with distributed neural networks for service quality estimation in B-ISDN networks", *IEEE IJCNN'00*, Vol. 5, Como, Italy 24-27 July 2000, pp. 392-397.

[4] D. Bertsekas and R. Gallager, "Data Networks", Prentice-Hall, 1987, Englewood Cliffs.

[5] D. Bienstock and O. Raskina, "Analysis of the Flow Deviation Method for the Max Concurrent Flow Problem", *Mathematical Programming*, Vol. , No. , 2002.

[6] R. Bolla, F. Davoli, P. Maryni, T. Parsini, "An adaptive neural network admission controller for dynamic bandwidth allocation," *IEEE Trans. Syst., Man, and Cyb.*, Vol. 28, No. 4, 1998.

[7] E. Casilari, A. Alfaro, A. reyes, A. Diaz-Estrella, F. Sandoval, "Neural modeling of Ethernet traffic over ATM networks", *EANN'98*, June, Gibraltar, pp. 304-307, 1998.

[8] L. Fratta and M. Gerla and L. Kleinrock, "The Flow Deviation Method: an Approach to Store-and-forward Computer-communication Network Design", Networks, Vol. 3, 1973, pp. 97-133.

[9] E. Gelenbe and G. Pujolle, *Introduction to Queueing Networks*, 2nd Ed., Wiley, 1998.

[10] I.W. Habib, "Applications of neurocomputing in traffic management of ATM networks", *Proceedings of the IEEE*, No. 10, Vol. 84, pp. 1430-1441, 1996.

[11] A. Mahul and A. Aussem, "Distributed Neural Networks for QoS Estimation in Communication Network", International Journal of Computational Intelligence and Applications, to appear, 2003.

[12] Moh, W. M., Chen, M.-J., Chu, N.-M., and Liao, C.-D. (1995). "Traffic prediction and dynamic bandwith allocation over ATM: a neural network approach", *Computer Communications*, 18(8).

[13] Soh, W.-S. and Tham, C.-K. (2001). "Modular neural networks for multi-service connection admission control", *Computer Networks*, 36, pp. 181-202.

[14] E. Rosen, A. Viswanathan, and R.Callon, Multiprotocol Label Switching Architecture, RFC 3031, Jan. 2001.