An Augmented Lagrangean Approach for the QoS Constrained Routing Problem

Christophe Duhamel, *LIMOS, Université Blaise Pascal, Clermont-Ferrand, France*
Antoine Mahul *LIMOS, Université Blaise Pascal, Clermont-Ferrand, France*

Keywords: optimal routing, quality of service, nonlinear programming, augmented lagrangean, flow deviation, neural network.

1. Introduction

The demand for a dynamic and efficient usage of network resources is one of the key points in the current evolution of the Internet. However, to support both existing and emerging (VoIP, peer-to-peer, ad-hod, etc.) multimedia applications with distinct Quality of Service (QoS) requirements, additional services and protocols are needed. IETF has proposed several approaches to address QoS requirements in the Internet. DiffServ-based architectures have received the greatest favors of operators and in the industry but the deployment of end-to-end QoS is still limited [6]. Although the basic DiffServ concept (differenciation of packets per traffic classes) is being currently exploited to support voice traffic over IP network (VoIP), handling the traffic in term of performance (delay, loss rate, jitter) is still an open issue [8].

A problem is to design effective routing policies that allow QoS constraints to be respected and ressource occupation to be optimized. MultiProtocol Label Switching (MPLS) is being considered as one (standardized) approach for scalable QoS provisioning in the Internet and may be seen as a possible answer to Traffic Engineering (TE) needs. We adress the problem of routing a set of demands in a MPLS backbone with QoS requirements expressed in term of end-to-end delays. The classical optimization algorithms typically used in TE tasks to maximize network performance and to balance traffic load have a major flaw: they still rely on the assumption that the QoS in routers obeys the standard M/M/1 formula. In this work, we define a multicommodity flow optimization procedure taking into account in which the M/M/1 approximation can be substituted by a more realistic evaluation of the delay using a neural network trained on simulated examples. We restrict ourselves to the QoS constraint defined as the end-to-end delay: for each commodity, it corresponds to the delay incurred by a packet between its arrival time at the destination and its departure time from the origin. The QoS Contrainted Routing Problem (QCRP) 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 all QoS requirements. Note that a somewhat different cost 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 [4]. This is left for future work.

We first present a mathematical formulation for (QCRP). Then we describe an augmented lagrangean approach to solve the problem. The subproblem reduces to a nonlinear minimum cost flow problem, which will be solved by a flow deviation algorithm. Due to the nature of the objective function, the generation of the improving direction will require a specific algorithm. Then, two approximations (the classical Kleinrock's delay function and a neural network-based function) will be presented for the delay and the load functions. Finaly, some numerical results will be presented.

2. Mathematical Formulation

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) $c_a > 0$ 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. The bandwidth requirement q_k is considered to be fixed, e.g. the traffic is not subject to variations. The set $\mathcal{S} = \{s_i,\ i = 1 \dots S\}$ contains the classes of service. To each class of service $s \in \mathcal{S}$, D_s is the upper bound

on the end-to-end delay. Thus, for every commodity $k \in \mathcal{K}$, D_{s_k} is the maximal end-to-end delay for each packet of k.

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, $\mathcal{P} = \cup_{k\in\mathcal{K}}\mathcal{P}_k$ denotes the set of all the paths. Each path $p \in \mathcal{P}$ can be expressed as a set of arcs. For each commodity k, let $x_p^k \geqslant 0$ be the percent of bandwidth assigned to the path $p \in \mathcal{P}_k$. Throughout this work, the delay will be 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 the delays on each arc a belonging to p. Let $\phi_s(x_a, c_a)$ and $\psi_s(x_a, c_a)$ be respectively the load function and the delay function for the class of service $s \in \mathcal{S}$ on the arc $a \in A$, where x_a is the aggregated bandwidth vector for each class of service. The (QCRP) can be formulated as follows:

(QCRP)

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

Constraints (1) require all the bandwidth to be assigned to the paths of commodity k. Equations (2) compute the bandwidth used on each arc according to the bandwidth that is assigned on each path. Equations (3) compute the end-to-end delay of each path. Finally, constraints (4) are the QoS constraint for each active path, e.g. for each path on which a positive amount of bandwidth is routed. These constraints are nonlinear and nonconvex. It is worth noting that only (1) and (4) define constraints, since x_a^s and d_p^k are intermediate variables. Moreover, the capacity constraints are not used in (QCRP) since they are redundant with the delay function ψ_s. (QCRP) is NP-hard as shown by Ben-Ameur and Ouorou [1]. To our knowledge, this is the only work trying to take the end-to-end delay constraint into account.

3. Augmented lagrangean method for the (QCRP)

A wide range of efficient methods are well-suited to solve convex multicommodity minimum cost flow problems (see [9] for a comprehensive survey and comparison). 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. The nonlinear QoS constraints (4) will be first dualized by performing a lagrangean relaxation. Let $g_p^k(x) = x_p^k \left(D_{s_k} - d_p^k\right)$. Then the objective function becomes

$$L(x, \lambda) = \sum_{a\in A}\sum_{s\in\mathcal{S}} \phi_s(x_a, c_a) - \sum_{k\in\mathcal{K}, p\in\mathcal{P}_k} \lambda_p^k g_p^k(x) \tag{1}$$

where $\lambda_p^k \geqslant 0$ are the lagrangean coefficients associated to the relaxed constraints. The resulting problem has a nonlinear objective function with linear constraints.

The lagrangean relaxation cannot be solved directly as the objective function $L(x, \lambda)$ is nonconvex (due to

$g_p^k(x)$). An augmented lagrangean relaxation scheme can be built by adding a quadratic penalty term to the objective function (see [3]). Let $h_p^k(x,\lambda,c) = \max\left\{0, \lambda_p^k + c g_p^k(x)\right\}$. Then the augmented lagrangean function is

$$L_c(x,\lambda) = \sum_{a\in A}\sum_{s\in\mathcal{S}} \phi_s(x_a, c_a) + \frac{1}{2c}\sum_{k\in\mathcal{K}, p\in\mathcal{P}_k}\left(\left(h_p^k(x,\lambda,c)\right)^2 - \left(\lambda_p^k\right)^2\right) \tag{2}$$

$L_c(x,\lambda)$ is locally convex at the neighborhood of x. The resulting problem can be solved by any classical optimization method (e.g. subgradient, bundle method). The inner problem, e.g. once the lagrangean multipliers λ have been set, will be solved by an adaptation of the Frank-Wolfe method to network routing: the Flow Deviation method ([5, 2]). At each iteration, given the current solution x, the objective function is linearized and an improving routing path is computed for each commodity. Then, a fair amount of bandwidth is transferred from every active path to this new set of paths and the process iterates until no improvement can be done.

As $L(x,\lambda)$ is nonconvex, the augmented lagrangean method converges towards a local optimum. Thus, the computed solution x^* is only a local optimum. However, this solution is still interesting as it is feasible, especially with respect to the QoS constraints. Moreover, a lower bound can be computed within the method. This will give us an estimation of the quality of our primal solution.

At each iteration of the augmented lagrangean method, the lagrangean coefficients are fixed and the subproblem (L_t^+) is as follows:

$$(L_t^+)\quad \boxed{\begin{array}{lll} \text{Min} & L_c(x,\lambda_t) & \\ \text{s.t.} & & \\ & \displaystyle\sum_{p\in\mathcal{P}_k} x_p^k = 1 & \forall k\in\mathcal{K} \quad (1)\\ & & \\ & x_p^k \geqslant 0 & \forall k\in\mathcal{K}, \forall p\in\mathcal{P}_k \end{array}}$$

At iteration t, let x_t^+ be the solution of (L_t^+). The multipliers are updated using the formula

$$\lambda_{t+1} = \max\{0,\ \lambda_t + c_t\, g(x_t)\} \tag{3}$$

The choice for the serie $\{c_t\}$ is derived from [3]. It consists in multiplying c_t by a constant factor β (here $\beta = 10$) each time the overall violation of the QoS constraints has not decreased by a factor γ (here $\gamma = 1.0005$)

A path-based flow deviation method is used to solve the subproblem (L_t^+). At each iteration, a linearization of the objective function is performed at the current point x_t to get an extremal point $\overline{x_t}$. Then, the linear combination $x_{t+1} = x_t + \theta(\overline{x_t} - x_t)$ minimizing the objective function $L_c(x,\lambda_t)$ is computed. The method stops as soon as no sufficient improvement can be obtained. In the context of classical routing problems, computing the extremal point reduces to the computation of a shortest path for each commodity and computing the optimal linear combination can be seen as a fair re-routing of some amount of flow from the old flow x_t to the new flow $\overline{x_t}$. In the context of (QCRP), the computation is a bit more complex due to the formulation of the gradient:

$$\frac{\partial L_c(x,\lambda)}{\partial x_p^k} = \sum_{a\in A}\sum_{s\in\mathcal{S}} \frac{\partial \phi_s}{\partial x_p^k}(x_a, c_a) + \sum_{k'\in\mathcal{K}}\sum_{p'\in\mathcal{P}_{k'}} h_{p'}^{k'}(x,\lambda,c)\frac{\partial g_{p'}^{k'}(x)}{\partial x_p^k} \tag{4}$$

This formula can be transformed into

$$\frac{\partial L_c(x,\lambda)}{\partial x_p^k} = \sum_{a\in p}\left[\sum_{s\in\mathcal{S}}\frac{\partial \phi_s}{\partial x_a^{s_k}}(x_a,c_a) - \sum_{k'\in\mathcal{K}}\sum_{\substack{p'\in\mathcal{P}_{k'}\\ a\in p'}} x_{p'}^{k'} h_{p'}^{k'}(x,\lambda,c)\frac{\partial \psi_{s_{k'}}}{\partial x_a^{s_k}}(x_a,c_a)\right] + h_p^k(x,\lambda,c)\left(D_{s_k} - d_p^k\right) \tag{5}$$

The computation of the solution minimizing the gradient (5) cannot be reduced to the computation of a shortest path anymore as the last term of this function depends on the path. However, it can be separated over the commodities. Thus, one has to compute an extreme point (that is, a path) for each commodity. This path is no more a shortest path in the classical sense but an optimal path, taking path-based cost into account, as said before. One can note these path-based costs can only be positive for paths that have already been used in the previous solutions. Let $\mathcal{H}^k(x) = \left\{p \in \mathcal{P}^k \mid h_p^k \neq 0\right\}$. This set contains the active paths violating the QoS constraints as well as some paths that once were active. Then, for each commodity, the algorithm works in two steps: it first computes a shortest path $\overline{p}$ using the first term of the formula (5) as the arc cost, then it computes the path $p^* \in \mathcal{H}^k(x) \cup \{\overline{p}\}$ minimizing (5).

4. Approximating the functions ϕ_s and ψ_s

We propose two ways to compute the load and the delay functions, $\phi_s(x_a, c_a)$ and $\psi_s(x_a, c_a)$. The first approach (the classical one) is based on an analytical formulation. It assumes strong hypotheses on the traffic (Poisson traffic, independent flows). Let $f_a = \sum_{s\in\mathcal{S}} x_a^s$ be the aggregated flow on the arc a. Under such hypotheses, the average delay is then $\widetilde{\phi_s}(x_a, c_a) = \frac{f_a}{c_a - f_a}$. This approximation is mostly referred as the Kleinrock's delay function. By using Little's theorem, the load can be obtained from the delay: $\widetilde{\phi_s}(x_a, c_a) = x_a^s\widetilde{\psi_s}(x_a, c_a)$. Those functions have interesting properties: they are $\mathcal{C}^\infty$, convex and are defining a barrier on the arc capacity. As both $\widetilde{\phi_s}(x_a, c_a)$ and $\widetilde{\psi_s}(x_a, c_a)$ are computed on the aggregated flow, they are identical for each class of service which means that no differenciation of service can be done.

The second approach tries to compute a more realistic approximation with respect to the differenciation of service (DiffServ). The QoS is usually expressed in terms of average response time and / or loss rate. It is difficult to express analytically in terms of the incoming traffic characteristics. Thus, it is particularly appealing to train a feed-forward neural network as a "black-box" on simulation data for the evaluation of the QoS values. As the delay is monotonically increasing with respect to the incoming traffic rates, inclusion of this prior knowledge into the training procedure can lead to better generalization. Here, neural networks (multi-level perceptron) are trained to approximate $\phi_s(x_a, c_a)$ and $\psi_s(x_a, c_a)$. By construction, these two functions are derivable. The monotonicity is imposed with a penalty algorithm during the learning procedure (see [7]). However, convexity can only be assumed.

In the augmented lagrangean method, traffic flows are only characterized by their bandwidth. After aggregating these traffic rates per class of service, the neural network will be able to estimate the mean delay for each class since it was trained to approximate the function $\{x_a^s\}_{s\in\mathcal{S}} \rightarrow \{d_a^s\}_{s\in\mathcal{S}}$, where d_a^s is the average delay for the class s on the arc a.

4..1 Preliminary numerical results

We consider several randomly generated planar network instances, whose size ranges from 10 nodes, 20 edges and 40 commodities to 30 nodes, 78 edges and 180 commodities. The program has been coded in C and compiled with gcc version 3.3.2 on a Pentium 4 computer with 4Gb memory running linux 2.6.6.

Table 1 displays the results for all the instances and for the analytical delay and load approximation (using Kleinrock's function). For each instance, two strategies are compared. The first one, "FD", consists in running the flow deviation algorithm on (QCRP) without the QoS constraints. The second one, "AL", consists in solving (QRCP) using the augmented lagrangean method described in section 3. For both strategies, the value of the optimal solution $\Phi^* = \Phi(x^*)$, the maximal violation $||g||_\infty$ of the solution over all the paths, the number of active paths violating QoS constraints over the active paths (column "viol."), and the CPU time in seconds are reported. The maximal violation is computed as $||g||_\infty = \max_{p \in \mathcal{P}_k, k \in \mathcal{K}} \max\left\{0, g_p^k(x^*)\right\}$, where $g_p^k(x^*)$ denotes the QoS constraint on the path p for demand k (cf. constraint (4) of (QCRP)). In the objective function, the analytical approximation function $\widetilde{\phi_s}$ is kept in all cases for two reasons: first, this evaluation function is not critical because it only gives a direction for the optimization, and also because it is a barrier function for the capacity on the arcs, which simplifies sub-problems.

instance	strategy	Φ^*	$\|\|g\|\|_\infty$	viol.	cpu (sec.)
T_10_20	FD	22.455944	4.522463×10^{-2}	10 / 45	12
	AL	28.215005	9.886708×10^{-7}	4 / 48	477
T_10_60	FD	60.211124	3.406511×10^{-1}	2 / 101	13
	AL	61.733936	1.488083×10^{-8}	1 / 240	50851
T_20_40	FD	46.993028	2.237437×10^{-1}	25 / 150	45
	AL	48.299828	8.354941×10^{-7}	4 / 273	2069
T_20_120	FD	92.888238	1.454175×10^{-1}	15 / 358	159
	AL	93.326225	3.660244×10^{-7}	1 / 409	7692
T_30_180	FD	164.437816	4.764601×10^{-3}	29 / 1034	562
	AL	164.437470	9.992447×10^{-7}	3 / 982	2539

Table 1: results for the analytical approximation.

As can be seen, taking the QoS into account (strategy AL) consumes a lot more time as it calls iteratively the FD algorithm. The difference can be really high, as for instance T_10_60. However, the computed solutions produce a very limited violation on the QoS, and for only few paths. In general, the number of paths carrying the traffic is higher as AL tries to use more paths to reduce the delay on each path. This has an impact on the value of the objective function. For the instance T_30_180, the FD strategy behaves rather poorly and stops too early. Thus, the AL strategy produces a better solution.

Table 2 displays a comparision between the analytical (KL) approximation and the neural network (NN) approximation (column "approx."). As the neural network consumes a lot of time, the experiments have only been done on a small instance (10 nodes, 22 edges and 40 commodities). In order to measure the impact of the global congestion, several throughputs (0.9, 1.0 and 1.1) have been tested. Each throughput t is a common multiplying factor for all the demands. Thus, the bandwidth demands are both equally and globally increased. For the neural network, two traffic scenarios (Poissonian sources and OnOff sources, which better approximate the nature of the traffic) are considered.

First, we can observe that the AL approach converges towards a feasible solution or a *quasi*-feasible solution. As the value $||g||_\infty$ remains small, output from the algorithm can be used after a cleaning step. This holds for all values of t. The FD approach does not take the QoS constraints into account. Therefore, the QoS model cannot have any impact on the behavior of this method. This explains why, for any given value of t, the optimal value of FD remains the same whatever the approximation. For the same reason, the number of violated QoS constraints n_{vio} if higher than for AL. This number increases with higher congestion (higher throughput), along with the number of active paths (which is a classical result). About the AL approach, it can be noted that the impact of the delay model on the solution increases as the overall traffic load (i.e. the throughput t) increases. For moderate values of t, AL gives solutions with fewer active paths. However this situation becomes different when the network comes close to saturation (to the QoS contraint point of view). At this point, it is difficult to keep a small number of active paths and the solution must use a lot more paths (e.g. $t = 1.0$ with OnOff neural approximation of delay). The last line of the table should indicate that the problem has become infeasible with respect to the OnOff QoS constraint. It still remains feasible with less accurate QoS approximations (namely Kleinrock and neural approximation with Poissonian traffic).

t	approx.	strategy	Φ^*	$\|\|g\|\|_\infty$	viol.	cpu (sec.)
0.9	KL	FD	18.190555	3.76×10^{-3}	2 / 42	12
		AL	18.190524	0.	0 / 39	640
	Poisson	FD	18.190555	1.67×10^{-3}	1 / 42	13
		AL	18.190524	0.	0 / 39	5358
	OnOff	FD	18.190555	1.90×10^{-3}	2 / 42	13
		AL	18.190524	0.	0 / 39	6461
1.0	KL	FD	22.455972	2.02×10^{-2}	7 / 45	16
		AL	22.483721	0.	0 / 43	3320
	Poisson	FD	22.455972	1.04×10^{-2}	4 / 45	17
		AL	22.456047	0.	0 / 40	13312
	OnOff	FD	22.455972	2.25×10^{-1}	5 / 45	17
		AL	22.730098	8.08×10^{-8}	1 / 61	6575
1.1	KL	FD	27.780724	8.28×10^{-2}	8 / 46	14
		AL	28.384249	1.71×10^{-5}	3 / 69	15051
	Poisson	FD	27.780724	9.22×10^{-2}	5 / 46	16
		AL	28.384249	9.00×10^{-9}	2 / 69	10526
	OnOff	FD	27.780724	1.38	8 / 46	16
		AL		Failed		

Table 2: results for both Kleinrock and NN approximations.

Overall, these experiments are useful in proving that the way QoS is approximated has a strong impact on the routing solution computed. According to the QoS approximation used, the routes that are obtained differ significantly. It is worth mentioning that the higher the overall demand, the better balanced the load throughout the network. The numerical results presented in this work show that the neural network approach provides reliable estimates of the delay and it has been successfully applied on a simple context. It shows promises for QoS-based traffic control, routing and congestion avoidance in multiservice telecommunication networks.

References

[1] Ben-Ameur, W., and Ouorou, A., "Mathematical models of the delay constrained routing problem", *Algorithmic Operations Research*, 1(2),2006.

[2] Bertsekas, D., and Gallager, R., *Data Networks, second edition*, Prentice-Hall, 1992.

[3] Bertsekas, D., *Nonlinear Programming, second edition*, Athena Scientific, 1999.

[4] Bienstock, D., and Raskina, O., "Asymptotic analysis of the flow deviation method for the maximum concurrent flow problem", *Mathematical Programming*, 91(3), pp. 479–492, 2002.

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

[6] Giordano, S., Potts, M., Smirnov, M., and Ventre, G., "Advances in QoS", *IEEE Communications Magazine*, 41(1), pp. 28–29, 2003.

[7] Mahul, A., and Aussem, A., "Learning with monotonicity requirements for optimal routing with end-to-end quality of service constraints", in *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN'2006)*, pp. 455-460, Bruges, Belgium, 2006.

[8] Mykoniati, E., and al., "Admission control for providing QoS in DiffServ IP networks: the TEQUILA approach", *IEEE Communications Magazine*, 41(1), pp. 38–44, 2003.

[9] Ouorou, A., Mahey, P., and Vial, J.-P., "A survey of algorithms for convex multicommodity flow problems", *Management Science*, 47(1), pp. 126–147, 2000.