Riikka Susitaival, Jorma Virtamo, and Samuli Aalto. Load balancing by MPLS in differentiated services networks. In Proceedings of the International Workshop on Architectures for Quality of Service in the Internet (Art-QoS 2003), pages 252-264, March 2003.

Load Balancing by MPLS in Differentiated Services Networks

Riikka Susitaival, Jorma Virtamo, and Samuli Aalto

Networking Laboratory, Helsinki University of Technology
P.O.Box 3000, FIN-02015 HUT, Finland
{riikka.susitaival, jorma.virtamo, samuli.aalto}@hut.fi

Abstract. Multi Protocol Label Switching (MPLS) assigns a short label to each packet and packets are forwarded according to these labels. The capability of MPLS of explicit routing as well as of splitting of the traffic on several paths allows load balancing. The paper first concentrates on two previously known approximations of the minimum-delay routing. Using these load balancing algorithms from the literature as a starting point, the main goal of this paper is to develop optimization algorithms that differentiate classes in terms of mean delay using of both routing and WFQ-scheduling. Both optimal and approximative algorithms are developed for the joint optimization of the WFQ-weights and routing. As a result it is found that the use of the approximations simplifies the optimization problem but still provides results that are near to optimal.

Keywords: MPLS, load balancing, differentiated services, scheduling

1 Introduction

In the conventional IP routing, forwarding decisions are made independently in each router, based on the packet's header and precalculated routing tables. MPLS (Multi Protocol Label Switching) is a flexible technology that enables new services in IP networks and makes routing more effective [1,2]. It combines two different approaches, datagram and virtual circuit, as a compact technology. MPLS is based on short fixed length labels, which are assigned to each packet at the ingress node of the MPLS cloud. These labels are used to make forwarding decisions at each node. This simplifies and improves forwarding. The architecture of MPLS is defined in [3].

One of the most significant applications of MPLS is Traffic Engineering. Traffic Engineering (TE) concerns performance optimization of operational networks [4]. Traffic Engineering using MPLS provides mechanisms to route traffic that have equal starting point and destination along several paths. The most important benefit of traffic splitting is the ability to balance load.

MPLS and its Traffic Engineering capabilities could provide technical support to the implementation of Quality of Service (QoS). The differentiation of services can be obtained by an alternative flow allocation that has the same

W. Burakowski et al. (Eds.): Art-QoS 2003, LNCS 2698, pp. 252–264, 2003.

principles as the load balancing methods. In order to make differentiation more effective, scheduling mechanisms, like WFQ-scheduling, can be utilized in the same context.

Our goal is to adjust routing and scheduling parameters that optimize differentiation of experienced service of different classes in terms of their mean delays. We use load balancing methods as a starting point in the further development. The routing and scheduling methods to be introduced are divided into two types. The first type tries to optimize only flow allocation so that differentiation is achieved. The second type of methods makes use of WFQ-weights. In each node, each service class has a WFQ-weight and the bandwidth is shared according to these weights using approximation of parallel independent queues. More details of these methods can be found in [5].

The rest of this paper is organized as follows: In the second section we concentrate on the three previously known load balancing algorithms. In section 3 we introduce flow allocation methods that differentiate traffic classes by routing only. We present the flow allocation model that makes use of WFQ-scheduling in section 4. We develop both optimal and approximative algorithms. In section 5 we present numerical results of all algorithms. Finally, section 6 makes some conclusions.

2 Load Balancing Algorithms

Load balancing methods make an attempt to balance load in the network and therefore achieve better performance in terms of delay. The basic optimization problem minimizes the mean delay in the network. The use of the link delays of M/M/1-queues leads to a non-linear optimization problem (NLP). Many exact algorithms have been introduced to this optimization, the most famous one being Gallager's algorithm from year 1977 [6].

2.1 Minimum-Delay Routing

First we formulate the load balancing as an optimization problem, which minimizes the mean delay of the network. Consider a network consisting of N nodes. A pair of nodes (m, n) can be connected by a directed link (m, n) with bandwidth equal to $b_{(m,n)}$. The number of links is denoted by L and the topology is denoted by T, which is a set of node-pairs. Let $A \in \mathbb{R}^{N \times L}$ be the matrix for which $A(i, j) = -1$ if link j directs to node i, $A(i, j) = 1$ if link j leaves from node i and $A(i, j) = 0$ otherwise.

The traffic demands are given by the matrix $[d_{(i,j)}]$, where i is the ingress and j is the egress node. $R_{(i,j)} \in \mathbb{R}^{N \times 1}$ is a vector for each ingress-egress pair (i, j) such that $R_{(i,j),k} = d_{(i,j)}$, if k is the ingress node, $R_{(i,j),k} = -d_{(i,j)}$, if k is the egress node, and $R_{(i,j),k} = 0$ otherwise. Demands and capacities are assumed to be constant (or average traffic rates). Let $x_{(i,j),(m,n)}$ be the allocated traffic of ingress-egress pair (i, j) on link (m, n). Then the total traffic on the link (m, n) is

$$X_{(m,n)} = \sum_{(i,j)} x_{(i,j),(m,n)}. \tag{1}$$

The formulation of the optimization problem that minimizes the mean delay in the whole network is as follows:

$$\begin{aligned} &\text{Minimize } E[D] = \tfrac{1}{\Lambda} \textstyle\sum_{(m,n)} \tfrac{X_{(m,n)}}{b_{(m,n)} - X_{(m,n)}}, \\ &\text{subject to the constraints} \\ &X_{(m,n)} < b_{(m,n)}, && \text{for each } (m,n), \\ &Ax_{(i,j)} = R_{(i,j)}, && \text{for each } (i,j), \end{aligned} \tag{2}$$

where Λ is the total offered traffic of the network. The last equation in (2) states that, at every node n, incoming traffic of each ingress-egress pair must be equal to outgoing traffic.

2.2 Flow Allocation Using Two-Step Algorithm

The algorithm that calculates an optimal flow allocation in terms of mean delay may be approximated by using the approach presented in [7]. The approximative algorithm solves first the paths to be used by LP-optimization and, after that, allocates the traffic to these paths using NLP-optimization.

The pair-based flow formulation that minimize the maximum link load is as follows:

$$\begin{aligned} &\text{Minimize } [-\epsilon Z + \sum_{(m,n)} w_{(m,n)} \sum_{(i,j)} x_{(i,j),(m,n)}] \\ &\text{subject to the constraints} \\ &x_{(i,j),(m,n)} \geq 0; \quad Z \geq 0, \\ &\textstyle\sum_{(i,j)} x_{(i,j),(m,n)} + \sqrt{b_{(m,n)}} Z \leq b_{(m,n)}, && \text{for each } (m,n) \text{ with } b_{(m,n)} > 0, \\ &Ax_{(i,j)} = R_{(i,j)}, && \text{for each } (i,j), \end{aligned} \tag{3}$$

where $w_{(m,n)}$ is a cost weight of link (m,n) and Z is a free parameter that describes the minimum value of the proportional unused capacity.

When the LP-problem (3) is solved and variables $x_{(i,j),(m,n)}$ found, we have to find paths for each ingress-egress pair. The algorithm to define paths to ingress-egress pair (i,j) is as follows: We have the original topology T, which consists of the set of directed links. Because one ingress-egress pair uses only part of the whole topology, we define a new topology $T'_{(i,j)}$, which consists of the links for which $x_{(i,j),(m,n)}$ differs from zero. Let $L'_{(i,j)}$ be the number of links in $T'_{(i,j)}$. Topology $T'_{(i,j)}$ is loop-free, because if there were loops, the original allocation would not be optimal. We search all possible paths to ingress-egress pair (i,j) by a breadth-first-search algorithm. The set of these paths is denoted by $P_{(i,j)}$. Let $K_{(i,j)}$ be the number of paths and $Q_{(i,j)} \in \mathbb{R}^{L'_{(i,j)} \times K_{(i,j)}}$ be the matrix, where

$$Q_{(i,j),(l,k)} = \begin{cases} 1, & \text{if path } k \text{ uses link } l \text{ of topology } T'_{(i,j)}, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

for each ingress-egress pair (i,j). Let $y_{(i,j),k}$ be the traffic allocation in path k. Unknown $y_{(i,j),k}$'s can be solved from matrix equation

$$Q_{(i,j)} y_{(i,j)} = {x'}_{(i,j)}, \quad \text{for each } (i,j), \tag{5}$$

where $x'_{(i,j)} \in \mathbb{R}^{L'_{(i,j)} \times 1}$ is the flow vector for each ingress-egress pair (i,j). Finally, if element k of the solution vector $y_{(i,j)}$ differs from zero, ingress-egress pair (i,j) uses path k, else not. So reducing the unused paths from path set $P_{(i,j)}$ we get actual path set $P'_{(i,j)}$.

The objective is to find an optimal flow allocation to these paths. Let $P'_{(i,j),k}$ be the k:th path of node-pair (i,j), $K'_{(i,j)}$ be the number of paths in $P'_{(i,j)}$, and $\phi_{(i,j),k}$ be the fraction of $d_{(i,j)}$ allocated to path $P'_{(i,j),k}$. The structure of path k is defined by $\delta_{(m,n),(i,j),k}$ as follows:

$$\delta_{(m,n),(i,j),k} = \begin{cases} 1, & \text{if path } P'_{(i,j),k} \text{ uses link } (m,n), \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Note that the traffic allocation $y_{(i,j),k}$ above is a special case of traffic allocation $d_{(i,j)}\phi_{(i,j),k}$. Our objective is to minimize the mean delay of the total network. So the optimization problem is as follows:

$$\begin{aligned} &\text{Minimize } \frac{1}{\Lambda} \sum_{(m,n)} \frac{\sum_{(i,j),k} \delta_{(m,n),(i,j),k} d_{(i,j)} \phi_{(i,j),k}}{b_{(m,n)} - \sum_{(i,j),k} \delta_{(m,n),(i,j),k} d_{(i,j)} \phi_{(i,j),k}}, \\ &\text{subject to the conditions} \\ &\sum_{(i,j),k} \delta_{(m,n),(i,j),k} d_{(i,j)} \phi_{(i,j),k} < b_{(m,n)}, && \text{for each } (m,n), \\ &\phi_{(i,j),k} \geq 0, && \text{for each } (i,j), k, \\ &\sum_{k=1}^{K'_{(i,j)}} \phi_{(i,j),k} = 1, && \text{for each } (i,j). \end{aligned} \tag{7}$$

2.3 Heuristic Approach

The heuristics presented in [8] to allocate traffic into the network using a particular level of granularity is very simple. The traffic granularity refers to the level of traffic aggregation [8]. The finer the level of the granularity the finer the traffic partitioning to different paths. In this algorithm, depending on the level of granularity, traffic demands from ingress to egress node are divided into streams (e.g. using some hashing function). The streams are sorted in descending order in terms of their traffic demand. After that each stream is routed sequentially one at a time to the shortest path defined by Dijkstra's algorithm using the mean delay of an $M/M/1$-queue as a link cost.

3 Methods to Achieve Differentiation by Routing

In this section we present such flow allocation methods that try to differentiate traffic classes by routing only. The traffic classes with a higher priority are routed along the paths that are not congested, for example. We differentiate classes by minimizing the sum of weighted mean delays, by fixing the ratio of mean delays, and using a heuristic approach.

3.1 Optimization Using Cost Weights

We consider the situation where the total traffic is divided into traffic classes, into the gold, silver and bronze classes, for example. The gold class has the highest priority and the bronze class the lowest priority. Each traffic class l has its own traffic matrix $[d_{l,(i,j)}]$, where i is the ingress node and j is the egress node. $R_{l,(i,j)} \in \mathbb{R}^{N\times 1}$ is an array for each class l and ingress-egress pair (i,j), where $R_{l,(i,j),k} = d_{l,(i,j)}$, if k is the ingress node, $R_{l,(i,j),k} = -d_{l,(i,j)}$, if k is the egress node and $R_{l,(i,j),k} = 0$ otherwise. The total traffic offered by class l is denoted by Λ_l. Let $x_{l,(i,j),(m,n)}$ be the allocated traffic of ingress-egress pair (i,j) of class l on link (m,n). So the total traffic of class l on link (m,n) is

$$X_{l,(m,n)} = \sum_{(i,j)} x_{l,(i,j),(m,n)}, \text{ for each } l, (m,n). \tag{8}$$

Let w_l be the cost weight associated to traffic class l. Additional notation used is the same as in section 2.2.

The purpose is to divide traffic into paths so that classes with higher priority achieve smaller mean delay than other classes. The optimization problem, where we minimize the sum of the weighted mean delays of the classes, is as follows:

$$\begin{aligned}
&\text{Minimize } \sum_l w_l E[D_l] = \sum_{(m,n)} \frac{\sum_l \frac{w_l X_{l,(m,n)}}{\Lambda_l}}{b_{(m,n)} - \sum_l X_{l,(m,n)}},\\
&\text{subject to the constraints}\\
&\sum_l X_{l,(m,n)} < b_{(m,n)}, && \text{for each } (m,n),\\
&A x_{l,(i,j)} = R_{l,(i,j)}, && \text{for each } l,(i,j),
\end{aligned} \tag{9}$$

where traffic allocations $x_{l,(i,j),(m,n)}$ are decision variables.

When the cost weights of different classes differ sufficiently, the optimization function tries to minimize the mean delays of gold class at the expense of lower priority classes. As a result, the routing obtained by the optimization function above differs from the load balanced routing, because in the load balancing the delays of the links are balanced to be almost equal and the differentiation could occur only if the paths are of different length.

3.2 Optimization with a Fixed Mean Delay Ratio

Now we fix the ratio of the mean delays of various classes to some value in order to differentiate classes. For example, in the case of two classes (gold and silver), the ratio of the mean delays between the silver and the gold class could be fixed to parameter q:

$$\frac{E[D_{l_2}]}{E[D_{l_1}]} = \frac{\frac{1}{\Lambda_{l_2}} \sum_{(m,n)} \frac{X_{l_2,(m,n)}}{b_{(m,n)} - \sum_l X_{l,(m,n)}}}{\frac{1}{\Lambda_{l_1}} \sum_{(m,n)} \frac{X_{l_1,(m,n)}}{b_{(m,n)} - \sum_l X_{l,(m,n)}}} = q. \tag{10}$$

After that the optimization can be done by minimizing the mean delay of either class.

Compared to the optimization in the previous section, this approach does not include cost weights and the actual ratio of the mean delays is known before the optimization. In order to make the optimization procedure easier it is useful to constraint the ratio of the mean delays to some small interval rather than to the exact value.

3.3 Heuristics

There exists a demand to provide also simple routing methods that can be implemented without heavy optimization. The approach that routes traffic to the network near optimally but still obtains the differentiation in terms of mean delay tries to adapt the heuristic approach presented in section 2.3.

The heuristic approach in the case of two classes (gold and silver) is as follows: The gold class is routed using heuristics introduced in 2.3. Then the allocated traffic of the gold class is multiplied by $1 + \Delta$. The silver class is then routed using heuristics in 2.3.

The idea of the heuristics is that the links used by the gold class look more congested than they actually are. So the routing scheme tries to balance load by routing the silver class by some other way. That is, the artificial congestion forces the silver class to avoid links used by the gold class and therefore the gold class should achieve more bandwidth. The choice of the parameter Δ depends on how much there is traffic offered compared to the capacity of the network.

4 Methods to Achieve Differentiation in Mean Delay Using Routing and WFQ-Scheduling

The possibilities to provide differentiated services using routing only are limited. To achieve certain ratio of mean delays may lead up to disadvantageous routing because the low priority class is routed along long and congested paths in order to artificially obtain the desired ratio of the mean delay.

Weighted Fair Queueing (WFQ) as a packet scheduling mechanism divides bandwidth among the parallel queues. Each queue achieves a guaranteed bandwidth, which depends on the WFQ-weight of that queue and the link capacity. We try to find optimal routing that differentiates the quality of service by including the WFQ-weighs to the optimization function as free parameters.

Because WFQ-scheduling is a work-conserving discipline, the bandwidth that is guaranteed for a class in our model can be viewed as the lower bound. Actually, the bandwidth available to a class can be much greater as the other classes may not always use the bandwidth reserved for them.

The bandwidth of each link (m, n) is shared according to WFQ-weights. We approximate the behavior of WFQ-scheduling as follows: Let $\gamma_{l,(m,n)}$ be the WFQ-weight that determines the proportion of total bandwidth that is given to class l. The bandwidth $b_{l,(m,n)}$ of class l on link (m, n) is thus

$$b_{l,(m,n)} = \gamma_{l,(m,n)} b_{(m,n)}, \text{ for each } l, (m, n). \tag{11}$$

The sum of the WFQ-weights of the classes on each link must equal to one. As a result, we have changed over from the WFQ-scheduling system to the system of parallel independent queues with link capacities described above.

4.1 Optimization Using Cost Weights

In this section we obtain the differentiation between classes by using the cost weights as in (9). The gold class gets the greatest cost weight and so on. The joint optimization of flow allocation and WFQ-weights where the sum of weighted mean delays is minimized is as follows:

$$\begin{aligned}
&\text{Minimize } \sum_l w_l E[D_l] = \sum_l \frac{w_l}{\Lambda_l} \sum_{(m,n)} \frac{X_{l,(m,n)}}{\gamma_{l,(m,n)} b_{(m,n)} - X_{l,(m,n)}},\\
&\text{subject to the constraints}\\
&X_{l,(m,n)} < \gamma_{l,(m,n)} b_{(m,n)}, && \text{for each } l, (m, n),\\
&A x_{l,(i,j)} = R_{l,(i,j)}, && \text{for each } l, (i, j),\\
&0 < \gamma_{l,(m,n)} < 1, && \text{for each } l, (m, n),\\
&\textstyle\sum_l \gamma_{l,(m,n)} = 1, && \text{for each } (m, n),
\end{aligned} \tag{12}$$

where traffic allocations $x_{l,(i,j),(m,n)}$ and WFQ-weights $\gamma_{l,(m,n)}$ are decision variables. This straightforward optimization is referred to "Str".

The optimization problem presented in (12) is quite heavy and time-consuming. We introduce near optimal algorithms that make the size of the problem smaller and the calculation easier. The first two algorithms first allocate the traffic into the network and after that optimize WFQ-weights. The structure of both algorithms is as follows:

1. Allocate the traffic into the network without WFQ-weights so that the weighted sum of mean delays is minimized. The formulation of the optimization algorithm is presented in section 3.1.
2. Fix the traffic allocation obtained in the first step.
3. Determine WFQ-weights using optimization problem (12) applied to the fixed link flows. Now the number of free variables equals the number of WFQ-weights, the number of links in the network multiplied by the number of classes minus one.

The cost weights of the optimization function are selected twice, in steps 1 and 3. In the first two-step algorithm (referred to as "2StepV1") we select the cost weights in the first step to be

$$w_l = \frac{\Lambda_l}{\sum_k \Lambda_k}. \tag{13}$$

Now the flow allocation is optimal and differences in mean delays do not appear in the first step. This algorithm makes only use of WFQ-scheduling when trying to differentiate classes. In the second two-step algorithm (referred to as "StepV2") the cost weights in the first and third steps are equal. So this algorithm utilizes both routing and WFQ-scheduling when differentiating classes and can therefore be closer to the optimal.

The third approximative algorithm makes use of the two-step algorithm presented in section 2.2 and is referred to as "QoS-LP-NLP". The paths are first calculated using the linear optimization that minimizes the maximum link load. Then the traffic is allocated to these paths and WFQ-weights are determined using the non-linear optimization (12) applied to the fixed paths.

4.2 Fixing the Link Delay Ratio

In the routing with WFQ-weights, if the ratio of total mean delays is fixed like in the algorithm presented in section 3.2, the optimization problem is demanding. One possibility is to fix the mean delay ratios at the link level. If the lengths of paths of different classes do not differ significantly, this approach should result approximately in the same mean delay ratio in the whole network.

We consider the case with two classes, gold and silver. We fix the ratio of link delays to parameter q and solve WFQ-weights $\gamma_{l_1,(m,n)}$'s as a function of the traffic of both classes, that is

$$\gamma_{l_1,(m,n)}(X_{l_1,(m,n)}, X_{l_2,(m,n)}) = \frac{q b_{(m,n)} + X_{l_1,(m,n)} - q X_{l_2,(m,n)}}{b_{(m,n)}(q+1)}, \text{ for each } (m,n). \tag{14}$$

The optimization problem of the flow allocation with the fixed ratio of mean delays is almost similar to optimization problem (12). The difference is that the WFQ-weights are not free parameters in the approach with fixed link delays. If we want to differentiate the classes by fixing the link delays only, the cost weights of the optimization function are equal to one (referred to as "FLD"). We can also optimize the sum of the weighted mean delays (referred to as "FLDW"). The problem is now how to determine the cost weights in relation to the ratio of link delays.

5 Numerical Results

The formulations of the optimization problems were written using a General Algebraic Modelling System (GAMS), which is a high-level modelling language

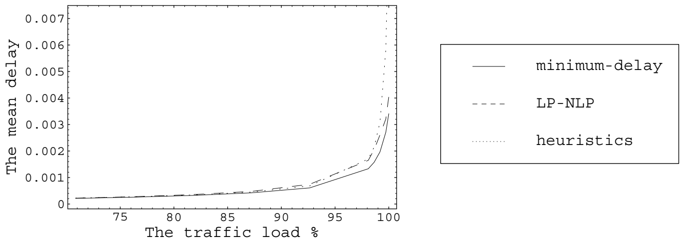

Fig. 1. The mean delay as a function of the traffic load

for mathematical programming problems [9]. We have used solver module Minos 5 in our optimizations.

The algorithms are tested in a known test-network, which consists of 10 nodes, 58 links and 72 ingress-egress pairs. The link capacities and the traffic demands are available at web-page http://brookfield.ans.net/omp/random-test-cases.html. In order to make optimizations simpler, we study only the case of two classes, gold and silver. The traffic matrices of both classes are equal, the half of the original demands, so that the comparison between the traffic classes is easier.

5.1 Load Balancing Routing

The mean delay and the relative deviation of the mean delay from the optimal as a function of the traffic load of the three routing methods of section 2 are presented in Figure 1 and 2. With the heuristic approach, we use the granularity level 32, except in the cases of heavy load (the traffic load is over 95%) when the used granularity level is 128. We can see that the mean delays of different methods do not differ significantly. Only when the traffic load is near to the total capacity of the network is the performance of the minimum-delay routing notable.

We find that the maximum number of paths used per each ingress-egress pair is only 3 in both minimum-delay routing and LP-NLP routing in the case of heavy traffic load (98%). Only 20% of the pairs in LP-NLP routing and 30% of the pairs in minimum-delay routing use more than one path. The computation time for minimum-delay routing is about 6.6 seconds, whereas it is for LP-NLP routing ten times smaller, about 0.6 seconds.

5.2 Methods to Achieve Differentiation Using Routing Only

The three approaches in section 3 make an attempt to differentiate the classes by routing only. The first optimization problem minimizes the sum of weighted mean delays. As a result, the ratio of the mean delay of the silver class to the mean delay of the gold class and the growth of the total mean delay as a function of the cost weight of the gold class is presented in Figure 3.

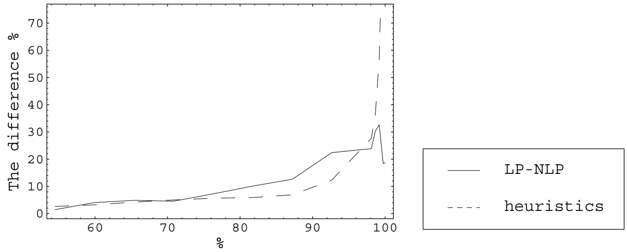

Fig. 2. The relative deviation of the mean delay from the optimal as a function of the traffic load

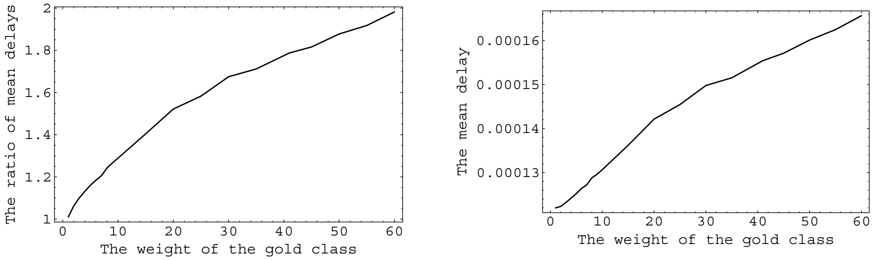

Fig. 3. The ratio of mean delays and the total mean delay as a function of the weight of the gold class

We take the total mean delay of the network as a function of the ratio of mean delay as a performance indicator. The increase in mean delay describes the cost of achieving a certain level of differentiation. All three methods are compared in Figure 4. The figure shows that the first and second optimizations generate the same result. However, the benefit of optimization function (10) is that the ratio of mean delays is known a priori, while the cost weights of optimization function (9) must be determined.

5.3 Methods to Achieve Differentiation by Routing and WFQ-Scheduling

First we have implemented the optimization method that minimizes the weighted sum of mean delays straightforwardly and the optimization methods that divide the problem into two steps (introduced in section 4.1).

A near optimal routing can also be achieved using an iterative approach (referred to as "2StepIt"). The flow allocation and the WFQ-weights are optimized alternatingly. The number of iterations is ten, which means that ten flow allocations and ten WFQ-weight determinations are done in the optimization.

The ratio of the mean delays and the total mean delay of all five algorithms as a function of the weight of gold class are presented in Figure 5 and 6. Note

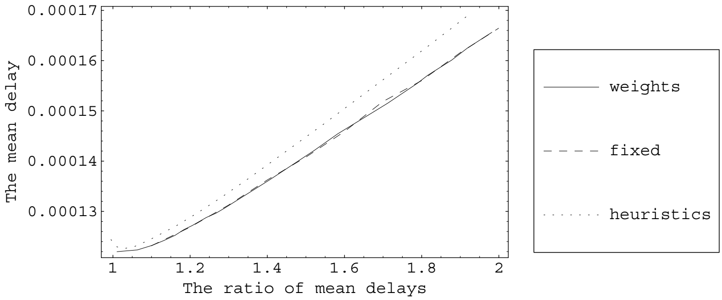

Fig. 4. The total mean delay as a function of the ratio of mean delays

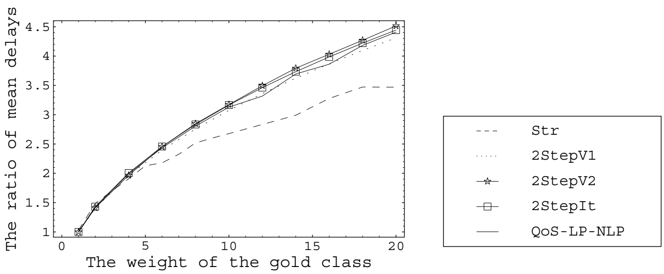

Fig. 5. The ratio of mean delays as a function of the weight of the gold class

that also the total mean delay is calculated using the approximation of parallel queues. Thus the figures in this subsection are not comparable with the figures in subsection 5.2. The irregularities in the curve of the straightforward routing are perhaps a consequence of numerical errors in the optimization procedure.

We have also implemented optimizations that fix the ratio of link delays to some parameter q (problems "FLD" and "FLDW"). In the latter problem we have implemented only one case, where the cost weight is three times greater than link delay ratio q. In Figure 7 we present the relation between the ratio of link delays and the ratio of mean delays of different classes. In the problem, where we fix only q ("FLD"), the ratio of mean delays seems to be smaller than the ratio of link delays. The explanation is that the routing algorithm tries to balance traffic load by routing classes that achieve more bandwidth through long routes.

Finally, we compare all the methods. The performance metric is the same as in section 5.2, the total mean delay of the network as a function of the ratio of mean delays. The results are presented in Figure 8. The straightforward optimization seems to have the smallest mean delay. The difference to other algorithms is significant when the ratio of mean delays is small. When the ratio is greater, the performance of the two-step algorithm that utilizes both routing

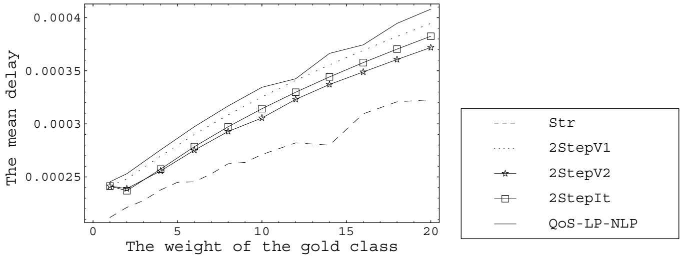

Fig. 6. The total mean delay as a function of the weight of the gold class

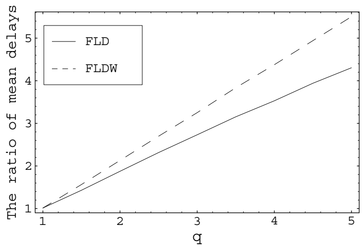

Fig. 7. The ratio of mean delays as a function of q

and WFQ-weights ("2StepV2") is near to optimal. As a conclusion, the differentiation methods that make use of both routing and WFQ-scheduling ("Str", "2StepV2", "2StepIt", "FLDW") perform better than the differentiation methods that make use of only WFQ-scheduling ("2StepV1", "QoS-LP-NLP" and "FLD").

6 Conclusions

We have used load balancing algorithms as a starting point when developing methods that try to differentiate traffic classes in terms of the mean delay. In the first model differentiation is achieved by using routing only. The mean delay using the algorithm that uses cost weights and the algorithm that fixes the ratio of mean delays are equal. The advantage of the latter algorithm is that the cost weights need not be known in advance. The mean delay using the heuristic approach is a little greater than using the other two algorithms.

We have also presented a model where the bandwidth of each link is shared among the traffic classes according to the WFQ-weights. The optimization problem is to minimize the weighted mean delay. In addition, near-optimal heuristic algorithms have been introduced. We notice that the use of the algorithm that makes use of both routing and WFQ-scheduling gives the best result.

The bandwidth guaranteed by WFQ-scheduling to each class is the theoretical minimum. As a result, the actual ratio of mean delays may differ from the

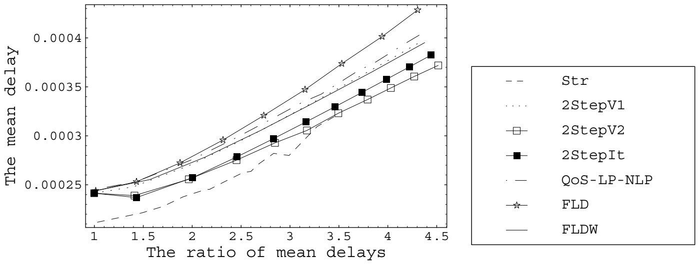

Fig. 8. The total mean delay as a function of the ratio of mean delays

result obtained by the optimizations. It would be interesting to know whether the actual ratio of mean delays is greater or smaller. A simulation study of WFQ-scheduling may provide some answers. That would also help the comparison between differentiation with routing only and differentiation with scheduling and routing. Adaptive algorithms used in the situation where the traffic matrices of the classes are unknown would be useful.

References

1. A. Viswanathan, N. Feldman, Z. Wang and R. Callon, Evolution of Multi-Protocol Label Swithing, IEEE Communication Magazine, Vol 36, Issue 5, pp. 165–173, May 1998.
2. G. Armitage, MPLS: The Magic Behind the Myths, IEEE Communications Magazine, Vol 38, Issue 1, pp. 124–131, January 2000.
3. E. Rosen, A. Viswanathan and R. Callon, Multiprotocol Label Switching Architecture, IETF RFC 3031, January 2001.
4. D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, Requirements for Traffic Engineering over MPLS, IETF RFC 2702, September 1999.
5. R. Susitaival, Load Balancing by MPLS in Differentiated Services Networks, Master's Thesis, Helsinki University of Technology, 2002.
6. R.G. Gallager, A Minimum Delay Routing Algorithm Using Distributed Computation, IEEE Transactions on Communication, Vol COM-25, Nr 1, pp. 73–85, January 1977.
7. T. Ott, T. Bogovic, T. Carpenter, K. R. Krishnan and D. Shallcross, Algorithms for Flow Allocation for Multi Protocol Label Switching, Telcordia Technical Memorandum TM-26027, 2001.
8. A. Sridharan, S. Bhattacharyya, R. Guérin, J. Jetcheva and N. Taft, On The Impact of Aggregation on The Performance of Traffic Aware Routing, Technical report, University of Pennsylvania, July 2000.
9. http://www.gams.com.