14-10-2002

Traffic engineering for MPLS-based virtual private networks

Chuntung Chou
University of Wollongong, ctchou@uow.edu.au

Traffic engineering for MPLS-based virtual private networks

Abstract

This paper considers the traffic engineering of MPLS-based virtual private networks (VPNs) with multiple classes of service. We focus on two main issues. Firstly, we point out that the one LSP per ingress-egress pair constraint can be relaxed for the case of MPLS-based VPNs due to the ease in classifying flows on a per-VPN basis. This allows us to use LSP with finer granularity and thus better load balancing. Secondly, we point out that the single objective traffic engineering formulations proposed in literature address only one particular aspect of the traffic engineering problem. We propose a multiobjective traffic engineering problem which takes both resource usage and link utilisation into account. This optimisation problem is NP-complete and involves a large number of variables. We propose an heuristic to solve this problem.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Chou, CT, Traffic engineering for MPLS-based virtual private networks, Proceedings Eleventh International Conference on Computer Communications and Networks, 14-16 October 2002, 110-115. Copyright IEEE 2002.

Traffic Engineering for MPLS-based Virtual Private Networks

Chun Tung Chou
School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia
Email: ctchou@titr.uow.edu.au, Fax: +61-2-4227 3277

Abstract— **This paper considers the traffic engineering of MPLS-based Virtual Private Networks (VPNs) with multiple classes of service. We focus on two main issues. Firstly, we point out that the one LSP per ingress-egress pair constraint can be relaxed for the case of MPLS-based VPNs due to the ease in classifying flows on a per-VPN basis. This allows us to use LSP with finer granularity and thus better load balancing. Secondly, we point out that the single objective traffic engineering formulations proposed in literature address only one particular aspect of the traffic engineering problem. In this paper, we propose a multiobjective traffic engineering problem which takes both resource usage and link utilisation into account. This optimisation problem is NP-complete and involves a large number of variables. We propose an heuristic to solve this problem.**

Keywords: Traffic engineering, MPLS, Virtual private networks, Quality of Service (QoS)

I. INTRODUCTION

This paper considers the traffic engineering of MPLS-based Virtual Private Networks (VPNs) with multiple classes of service. Traffic engineering for MPLS-based networks has been considered in, for example, [1], [2], [3], [4], [5]. A common assumption made in these papers is that the traffic from an ingress-egress pair is to be put on one LSP. The reason behind this assumption is that it will take considerable amount of work to classify the packets at the edge router so that a flow is not routed over multiple paths.

The one LSP per ingress-egress pair assumption is valid for the case where all the traffic flows are belonging to one single network service provider (NSP). However, this assumption can be relaxed if we are considering the case of VPNs. We argue in this paper that the BGP/MPLS VPN [6] provides a convenient way to classify packets on a per-VPN basis with only minor modification in the edge routers. This allows us to use granularity finer than one LSP per ingress-egress pair in VPN traffic engineering. With this finer granularity, one can potentially achieve better load balancing in the network.

Another aspect of traffic engineering that we will consider in this paper is the choice of optimisation criterion for traffic engineering. Two common optimisation criteria have been proposed in the literature. The first one is to minimise a linear function of the link bandwidth usage [3]. From a NSP's point of view, this optimisation criterion minimises the network operation cost. However, a drawback of this criterion is that it may result in an uneven distribution of traffic in the network where some links are over-utilised and some links are under-utilised. This is demonstrated in the example in section IV of this paper.

Another optimisation criterion that has been proposed in literature is to minimise the maximum link utilisation [4]. This optimisation criterion will produce an even traffic distribution and will also maximise the room for traffic growth. However, a drawback of this criterion is that the network resource usage is not minimised. In fact, the example in section IV shows that this criterion may use 70% more network resources than the case where network resources are minimised.

In this paper, we propose a multiobjective formulation of the traffic engineering problem which takes into account both resource usage (which can be viewed as network operation cost) and link utilisation. We demonstrate with an example that this multiobjective formulation can produce near Pareto optimal result. Although we have formulated this multiobjective problem in terms of the MPLS-based VPN traffic engineering problem, this multiobjective framework is equally applicable to traffic engineering problems in other settings.

The rest of this paper is organised as follows. Section II discusses the MPLS-based VPN traffic engineering problem. This section discusses the issues of LSP granularity and optimisation criterion, and ends with a mixed integer multiobjective programming formulation of the VPN traffic engineering problem. The proposed optimisation problem is NP-complete and involves a large number of binary decision variables. In section III, we propose a heuristic solution to tackle this problem. Finally, an example is given in section IV and the conclusions are presented in section V.

II. TRAFFIC ENGINEERING OF MPLS-BASED VPN

In this section, we will formulate the traffic engineering problem for MPLS-based VPN with multiple classes of service. Section II-A gives an overview of the VPN traffic engineering problem. Section II-B addresses two issues: the granularity of the LSP and the optimisation criterion. Based on the discussion in section II-B, we present a multiobjective formulation of the VPN traffic engineering problem in section II-C.

A. Overview of the traffic engineering problem for MPLS-based VPN

According to [7], Internet traffic engineering is concerned with performance optimisation of operational IP networks. A common problem that is faced in today's Internet, which is caused by the use of destination based shortest path routing, is that part of the network is over-utilised while another part of the network is under-utilised. A goal of traffic engineering is to correct this imbalance in resource usage. This can be achieved by using the route pinning property of MPLS which allows the NSP to control the routes used by the different LSPs. The route pinning property of MPLS is also important in providing QoS in the Internet. A fundamental requirement of being able to provide QoS guarantee is to ensure that there are sufficient resources for the QoS traffic. By using MPLS and resource reservation, a NSP can ensure that QoS traffic is given sufficient network resources.

Since the main degree-of-freedom in MPLS traffic engineering is the choice of routes for the LSPs, a traffic engineering problem is often formulated as an integer or mixed-integer optimisation problem whose aim is to find a suitable route for each of the LSPs [4], [3]. In the VPN traffic engineering problem to be considered in this paper, we assume that we perform offline computation to obtain these routes. Furthermore, we assume that the NSP owns a physical network for providing the VPN service. In order to simplify the discussion here, we assume for the time being that only one service class is offered by this NSP. We also assume that each VPN customer provides the NSP with a traffic demand matrix whose elements are the bandwidth requirement between an ingress-egress pair of the VPN. In this context, the goal of the VPN traffic engineering problem is to find a route for each of these demands. However, this description has overlooked two important issues:

1) The granularity of the LSPs to be used to implement the VPNs in the physical network.
2) The optimisation criterion to be used.

We will discuss these two issues further in the next section.

B. Traffic engineering issues

1) Granularity of LSPs: In the traffic engineering overview in the previous section, we mention that the goal of the VPN traffic engineering is to find a suitable path for each demand of each VPN. (We continue to assume a single service class in this section). An issue is how these demands should be mapped to the LSPs. On one extreme, we can aggregate all the demands using the same ingress-egress pair from all VPNs into an LSP. For a physical network with N nodes, this will result in $\mathcal{O}(N^2)$ LSPs in the network.

The idea of using only one LSP per ingress-egress pair is implicit in the implementation of the BGP/MPLS VPN scheme presented in the IETF RFC 2547 [6]. The implementation makes use of MPLS label stack where the bottom label is VPN specific while the top label is VPN independent. The core routers in the network only require the top label for routing and are therefore completely oblivious of the existence of the various VPNs. This results in a scalable implementation where the number of routes in the network can be made independent of the number of VPNs.

However, sometimes it may not be possible to put the aggregate of all the demands of an ingress-egress pair in one LSP. This happens if the aggregate demand is larger than the capacity of any single link in the network. Also, an aggregate with a large demand may be hard to load balance.

The mapping of the aggregate demands onto a single LSP represents the coarsest granularity that we can use. On the other extreme, each of the demand of each VPN can be mapped onto an individual LSP. This will result in $\mathcal{O}(N^2\times$ #VPNs) LSPs or routes in the network. This is clearly a non-scalable solution and is precisely what the authors of RFC 2547 [6] are trying to avoid. However, we see in the last paragraph that there are occasions where it is appropriate to use more than one LSP for the aggregate demand between an ingress-egress pair. We therefore believe that the granularity of a LSP should not be fixed *a priori* but should be determined by the optimisation process. However, a limit on the number of LSPs should be imposed in order to avoid an unscalable number of routes.

Note that it requires only minor modification to the edge routers in the BGP/MPLS scheme in order to have multiple LSPs between an ingress-egress pair. For example, if we are to set up two LSPs between an ingress-egress pair, we can divide the VPNs using this ingress-egress pair into two groups where traffic from each group will be assigned to one particular LSP. The edge router will again insert two labels into the packets. The bottom label is VPN specific. The top label will specify one of the two LSPs. Note that: (1) Even if multiple LSPs are used, only the edge routers have to know about the different VPNs but the core routers remain unaware of the existence of various VPNs. (2) We do not advocate the use of granularity that is finer than per-VPN level because significant workload, in the form of IP packet classification, will be required to ensure that an IP flow is not split across multiple LSPs.

2) Optimisation criterion: A goal of traffic engineering is to optimise network performance. Various optimisation criteria have been proposed for this traffic engineering optimisation problem. For example, [3] proposes a criterion which minimises a weighted linear sum of per-link bandwidth usage. However, such an optimisation criterion has the same drawback as minimising the resource usage, i.e. some links being over-utilised. This will be illustrated in an example in section IV.

An alternative optimisation criterion suggested in the literature is to minimise the maximum link utilisation in the network [4]. The motivation for introducing such criterion is that, in the case of fixed routing and linear traffic growth, the minimisation of the maximum link utilisation will maximise the linear growth factor before re-routing will be required. However, such criterion has two drawbacks. Firstly, it ignores the resource usage as a factor. Secondly, it puts its emphasis on the bottleneck link only. In fact, the example in section IV shows that this criterion may use 70% more network resources than the case where network resources are minimised.

Note that both of these criteria, if used on their own, address only one aspect of the traffic engineering problem. In section II-C, we propose a multiobjective programming optimisation problem which uses both of these criteria. This results in a solution which takes both network resource usage and network traffic growth into account. We will demonstrate in section IV that this multiobjective programming formulation gives a near Pareto optimal solution in both resource minimisation and maximum link utilisation.

C. Mathematical formulation of the VPN traffic engineering problem

1) Notation: This section defines the notation that will be used. We assume that the physical network is given by a capacitated directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ where $\mathcal{V}$ and $\mathcal{E}$ are respectively the set of network nodes and links. The elements in $\mathcal{E}$ are denoted by e_{uv} where $u, v \in \mathcal{V}$ are the end points of the link. Associated with each $e_{uv} \in \mathcal{E}$ is a bandwidth (capacity) which is denoted by b_{uv}. Finally, let N denote the number of nodes in the network.

We assume the NSP offers a number of different service classes indexed by $s \in \mathcal{S} = \{1, 2, \ldots\}$. The total number of service classes is denoted by $|\mathcal{S}|$.

We assume there are altogether M different VPNs and they will be indexed by m. Each of these VPNs will supply the NSP with $|\mathcal{S}|$ traffic demand matrices, one for each traffic class. Let $t_{ij}^{m,s}$ be the traffic demand of the m-th VPN for service class s between ingress-egress pair (i, j) where $i, j \in \mathcal{V}$. Note that each VPN may have different virtual topologies and may not have demands for all the different service classes. In this case, a zero value in the demand matrix will be used.

Let i, j be two distinct nodes in $\mathcal{V}$. For each ingress-egress pair (i, j) and service class s, each individual demand between i and j will be routed over one of the potential paths in the set $P_{ij}^{s} = \{p_{ij}^{s,1}, p_{ij}^{s,2}, \ldots, p_{ij}^{s,k}, \ldots\}$. Note that the set of potential paths is dependent on the ingress-egress pair and the service class, and is independent of individual VPNs. Let P denote the order of magnitude of the number of potential routes per ingress-egress pair per service class. This quantity will be used later on to quantify the number of variables in the optimisation problem.

In the optimisation problems to be formulated, we will need to ensure that the total capacity allocated to any physical link does not exceed its physical capacity. We therefore require a way to keep track of whether a particular potential path uses a certain physical link. We define the following indicator functions:

$$\mathcal{I}_{ij}^{uv,s,k} = \begin{cases} 1 & \text{if } e_{uv} \in \mathcal{E} \text{ is on the path } p_{ij}^{s,k} \in P_{ij}^{s} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Let μ_{uv} denote the link utilisation of the physical link $e_{uv} \in \mathcal{E}$.

Also let $\bar{R}$ denote a pre-specified upper limit on the total number of LSPs or routes in the NSP's physical network.

Note that in the above definitions, and in the rest of the paper, we have adopted the convention of using i and j to index the ingress and egress of a VPN demand. The end points of a physical link will be indexed by u and v.

2) A multiobjective VPN traffic engineering problem: The aim of this section is to formulate the multiobjective VPN traffic engineering problem. In this paper, we will make the assumption that the physical network $\mathcal{G}$ has sufficient capacity to meet the demands from all VPNs. With this assumption, the traffic engineering problem becomes one of choosing a suitable physical route for each VPN demand such that a certain criterion is optimised. Define the decision variables

$$\delta_{ij}^{m,s,k} = \begin{cases} 1 & \text{if demand } t_{ij}^{m,s} \text{ uses path } p_{ij}^{s,k} \in P_{ij}^{s} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In terms of these decision variables and the indicator function defined in equation (1), the capacity allocated on link $e_{uv} \in \mathcal{E}$ for the VPN requests is

$$y_{uv} = \sum_{s} \sum_{m} \sum_{i,j} t_{ij}^{m,s} \left(\sum_{k} \mathcal{I}_{ij}^{uv,s,k} \delta_{ij}^{\ell,s,k} \right) \quad (3)$$

In order to control the number of LSPs to be used, we introduce an additional set of decision variables

$$\eta_{ij}^{s,k} = \begin{cases} 1 & \text{if the LSP between ingress-egress pair } (i,j) \\ & \text{ uses path } p_{ij}^{s,k} \in P_{ij}^{s} \\ 0 & \text{otherwise} \end{cases}$$

The total number of LSPs or routes R that will be used to implement these VPNs will be

$$R = \sum_{ij} \sum_{s} \sum_{k} \eta_{ij}^{s,k}. \quad (4)$$

The multiobjective programming VPN traffic engineering problem consists of two steps. In the first step, we minimise the maximum link utilisation and is stated as follows:

Optimisation problem **OPT1a**

$$\min \mu \quad (5)$$

subject to the constraints

$$y_{uv} \leq \mu b_{uv} \quad \forall e_{uv} \in \mathcal{E} \quad (6)$$

$$\sum_{k} \delta_{ij}^{m,s,k} = 1 \, \forall i, j \in \mathcal{V}, m = 1, ..., M, s = 1, ..., |\mathcal{S}| \quad (7)$$

$$\delta_{ij}^{m,s,k} \leq \eta_{ij}^{s,k} \, \forall i, j \in \mathcal{V}, m = 1, ..., M, s = 1, ..., |\mathcal{S}| \quad (8)$$

$$R \leq \bar{R} \quad (9)$$

$$\delta_{ij}^{\ell,s,k}, \eta_{ij}^{s,k} \in \{0, 1\} \quad (10)$$

The constraint (7) ensures that only one path is chosen for the demand $t_{ij}^{m,s}$. The constraint (8) enforces the fact that if the LSP between ingress-egress pair (i, j) does not use path $p_{ij}^{s,k}$, no demands in $t_{ij}^{m,s}$ can use this path. Finally, the inequality (9) is a constraint on the number of routes.

Let μ^{*} be the optimal value of μ obtained in the first optimisation step. The second optimisation step is to minimise the cost subject to the constraint that all link utilisation remains under μ^{*}. The problem can be stated as follows:

Optimisation problem **OPT1b**

$$\min \sum_{uv} c_{uv} y_{uv} \tag{11}$$

subject to the constraints

$$y_{uv} \leq \mu^* b_{uv} \quad \forall e_{uv} \in \mathcal{E} \tag{12}$$

$$\sum_k \delta_{ij}^{m,s,k} = 1 \quad \forall i,j \in \mathcal{V}, m = 1,\ldots,M, s = 1,\ldots,|\mathcal{S}| \tag{13}$$

$$\delta_{ij}^{m,s,k} \leq \eta_{ij}^{s,k} \quad \forall i,j \in \mathcal{V}, m = 1,\ldots,M, s = 1,\ldots,|\mathcal{S}| \tag{14}$$

$$R \leq \bar{R} \tag{15}$$

$$\delta_{ij}^{\ell,s,k}, \eta_{ij}^{s,k} \in \{0,1\} \tag{16}$$

The constraints in the second optimisation step are the same as those in the first step except for constraint (12), where we enforce the condition that the maximum utilisation of the network remains at the same level as that given by the first optimisation.

In order to understand why the second optimisation step is necessary, we need to realise that the solution to **OPT1a** is generally not unique. Without loss of generality, we will assume in the following discussion that $c_{uv} = 1$. This means the objective of **OPT1b** is to minimise the total resource usage. We now argue that there are many solutions to **OPT1a** which give the same value of μ^* but they consume different level of network resources. Consider the situation depicted in figure 1. The number next to a link indicates the utilisation of that link. There are two bottleneck links, 2-3 and 1-8, with link utilisation 0.8. Let us consider the demands to be routed between the ingress-egress pair 3-8. These demands can be routed using the direct path 3-8 or it can be routed using a longer path, e.g. 3-4-6-7-8. Provided that the demands are not too large, the demands for ingress-egress pair 3-8 can take either of two these paths without affecting the maximum network utilisation. However, the choice of paths will make a difference in the total resource usage in the network.

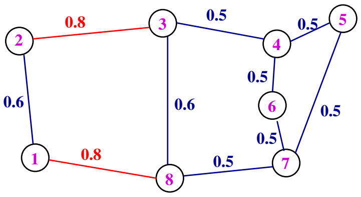

Fig. 1. This figure is used in section II-C.2 to explain why the solution to minimising the maximum utilisation is generally not unique.

Before we finish this section, we would like to make a remark on our problem formulation based on the set of potential paths P_{ij}^s. The basic idea behind this is that we will use this selection of paths to enforce the QoS specifications for each service class. For example, we may use P_{ij}^s to set a limit on the number of hops used by each traffic class.

3) Complexity of the problem: The optimisation problems **OPT1a** and **OPT1b** formulated earlier belong to the class of mixed linear integer programming (MILP). A special case of the optimisation problem **OPT1a** is that considered in [4] where there is only 1 VPN and 1 service class, and without constraints on the number of LSPs. That problem is proved in [4] to be NP-hard. Thus the problem **OPT1a** is also NP-hard. The second optimisation problem **OPT1b** is NP-complete [3].

In addition, these two optimisation problems also involve a large number of binary decision variables, which is of the order $\mathcal{O}(N^2 \times M \times |\mathcal{S}| \times P)$. We will provide an heuristic solution to this optimisation problem in the following section.

III. An heuristic solution

In this section, we present an heuristic solution to the optimisation problem **OPT1a** and **OPT1b** that we have formulated earlier. We will make two simplifications from the outset.

1) We will perform the optimisation with one service class after another. This reduces the number of binary decision variables per optimisation problem to the order $\mathcal{O}(N^2 \times M \times P)$.
2) We ignore the constraints on the number of routes for the time being. In other words, we drop the constraints (8) and (9) for **OPT1a**, and the constraints (14) and (15) for **OPT1b**.
   Note that the removal of the constraint on the number of routes means that we have lost control over this requirement. However, we will demonstrate in section IV that our heuristic gives a solution which mostly uses one route per ingress-egress pair.

Even with the first simplification in place, the number of binary decision variables that we have to deal with is still large. In fact, the complexity of the problem grows with the number of VPNs. We will approach this problem in two steps. We will show in section III-A how we can obtain an approximate solution using linear programming (LP). In section III-B, we show how we can obtain an integer solution using the approximation obtained in section III-A.

A. A continuous approximation

The aim of this section is to formulate two LP problems which give us an approximation of the simplified version of **OPT1a** and **OPT1b**. This approximation is meant to be effective when the number of VPNs is large.

In the problem formulation of **OPT1a** and **OPT1b**, we have assumed that each demand $t_{ij}^{m,s}$ is to be routed independently. Instead of doing this, we will route the aggregate demand per ingress-egress pair. We further introduce two assumptions:

1) The aggregate demand is infinitely divisible.
2) The aggregate demand can be routed over multiple routes.

Let T_{ij}^s be the aggregate demand from all VPNs for ingress-egress pair (i,j) for service class s, i.e.

$$T_{ij}^s = \sum_{m=1}^{M} t_{ij}^{m,s} \tag{17}$$

Since we will be performing the optimisation on a per-class basis, the index s should be treated as a constant here. We have chosen to retain the index s instead of dropping it so that we do not have to redefine the notation.

We now define a set of continuous decision variables in the range $[0, 1]$. Define

$$x_{ij}^{s,k} = \text{The fraction of aggregate demand } T_{ij}^{s} \text{ to be routed over the path } p_{i,j}^{s,k}$$

Based on these decision variables and the indicator function (1), the capacity being used on physical link e_{uv} can be written as

$$z_{uv}^{s} = \sum_{ij} T_{ij}^{s} \sum_{k} x_{ij}^{s,k} \mathcal{I}_{ij}^{uv,s,k} \tag{18}$$

Based on the simplifications that we have introduced earlier, we define the following two LP problems.

OPT2a

$$\min \mu \tag{19}$$

subject to the constraints

$$z_{uv} \leq \mu b_{uv} \ \forall e_{uv} \in \mathcal{E} \tag{20}$$

$$\sum_{k} x_{ij}^{s,k} = 1 \ \forall i, j \in \mathcal{V} \tag{21}$$

$$x_{ij}^{s,k} \in [0, 1] \ \forall i, j \in \mathcal{V}, \forall k \tag{22}$$

Let μ^{*} be the minimum value of μ given by **OPT2a**. The second LP is:

OPT2b

$$\min \sum_{uv} c_{uv} z_{uv} \tag{23}$$

subject to the constraints

$$z_{uv} \leq \mu^{*} b_{uv} \ \forall e_{uv} \in \mathcal{E} \tag{24}$$

$$\sum_{k} x_{ij}^{s,k} = 1 \ \forall i, j \in \mathcal{V} \tag{25}$$

$$x_{ij}^{s,k} \in [0, 1] \ \forall i, j \in \mathcal{V}, \forall k \tag{26}$$

The problems **OPT2a** and **OPT2b** are, respectively, the continuous approximations of the problems **OPT1a** and **OPT1b**, after the simplifications that we have introduced. Note that both of these LPs have $\mathcal{O}(N^2 \times P)$ variables, which is independent of the number of VPNs.

B. Recovering the integer solution

In section III-A, we assume that the aggregate demand is infinitely divisible in order to use LP to compute a continuous approximation. We will show in this section, how we can retrieve the integer solution. The solution to **OPT2b** tells us how the aggregate demand T_{ij}^{s} is split among the potential paths $\{p_{ij}^{s,1}, p_{ij}^{s,2}, \ldots, p_{ij}^{s,k}, \ldots\}$. The problem is then to distribute the VPN demands $t_{ij}^{m,s}$ for $m = 1, \ldots, M$ among the potential routes with non-zero traffic such that after the distribution process, the actual fraction of aggregate demand in each potential route with non-zero traffic matches as closely as possible to that given by the continuous solution. It is instructive to point out here that this matching is only feasible if an aggregate demand is not split into too many routes. We will demonstrate in section IV that this is indeed the case. Based on the problem description earlier, we will define the problem in a general setting.

Let $\{t_1, \ldots, t_D\}$ be a set of non-zero demands to be distributed to B different bins where a bin is an LSP in our context. Let also $\rho_1, \ldots, \rho_B$ be B strictly positive numbers such that $\sum_{h=1}^{B} \rho_h = 1$. Finally, define $T = \sum_{i=1}^{D} t_i$. In the context of our work, $\{t_1, \ldots, t_D\}$ correspond to the non-zero traffic demands for an ingress-egress pair (i, j). The optimisation problem that we have formulated in section III gives a solution which distribute the aggregate demand T into B different LSPs with a fraction ρ_h in the h-th LSP.

In order to formulate this assignment problem, we define binary decision variables

$$q_{gh} = \begin{cases} 1 & \text{if demand } t_g \text{ is to be put into bin } h \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

The assignment problem can be stated as the following optimisation problem:

OPT3

$$\min \sum_{h=1}^{B} \left| T\rho_h - \sum_{g=1}^{D} t_g q_{gh} \right| \tag{28}$$

subject to the constraint

$$\sum_{h=1}^{B} q_{gh} = 1 \ \forall g = 1, \ldots, D \tag{29}$$

$$q_{gh} \in \{0, 1\} \tag{30}$$

The optimisation problem **OPT3** is NP-complete. We will prove that for the case $B = 2$. In this case, the decision variables are q_{g1} and q_{g2}. By substituting $q_{g2} = 1 - q_{g1}$ in the optimisation problem, **OPT3** is equivalent to

$$\min \left| T\rho_1 - \sum_{g=1}^{D} t_g q_{g1} \right| \text{ with } q_{g1} \in \{0, 1\}$$

The solution to this optimisation is given by the minimum of the following two optimisation problems:

$$\min T\rho_1 - \sum_{g=1}^{D} t_g q_{g1} \text{ s.t. } T\rho_1 \geq \sum_{g=1}^{D} t_g q_{g1}, q_{g1} \in \{0, 1\}$$

$$\min -T\rho_1 + \sum_{g=1}^{D} t_g q_{g1} \text{ s.t. } T\rho_1 \leq \sum_{g=1}^{D} t_g q_{g1}, q_{g1} \in \{0, 1\}$$

Both of these problems are subset sum problems [8], which are known to be NP-complete [9, p.247].

A possible way to obtain an approximation to **OPT3** is by first solving the multiple subset sum (MSS) problem:

$$\max \sum_{h=1}^{B} \sum_{g=1}^{D} t_g q_{gh} \text{ s.t. } T\rho_h \geq \sum_{g=1}^{D} t_g q_{gh}, q_{gh} \in \{0, 1\}.$$

MSS can be solved in pseudo-polynomial time [10]. Let T_{MSS} denote the optimal solution to the above MSS problem. Then, it can be shown that the optimal solution to **OPT3** is bounded from above by $T - T_{\text{MSS}}$. However, for moderate value of D and small value of B, **OPT3** can often be solved directly.

IV. Example

In this section we demonstrate the effectiveness of our algorithms. We use a network with 17 nodes and 58 links. We assume there are 100 VPNs and the demand for these VPNs are randomly generated. There are altogether 3 service classes. The set of potential paths for Service Class 1 has 6 hops or less. Those for Service Class 2 have 9 hops or less, with no restriction on the number of hops for Service Class 3. There are more than 50,000 potential paths in Service Class 3. If we are to solve the integer programming problem for Service Class 3, it will have over 5 million binary decision variables.

We will compare the effect of the choice of optimisation criteria on the traffic distribution. Three criteria are used. The first criterion is based on minimising the total network resource usage. The second criterion is based on minimising the maximum link utilisation. The last criterion is the multiobjective programming formulation proposed in this paper. For each choice of optimisation criteria, we solve the optimisation problem first for Service Class 1, and then for Service Class 2 using the residual network, and finally for Service Class 3. The results are summarised in table I. We see that if we minimise the resource usage alone, it gives the smallest total resource usage among the three criteria but some links (2 in this case) are fully utilised. In contrary, minimising the maximum utilisation gives the smallest maximum link utilisation but results in a large resource usage. However, the multiobjective formulation gives a near Pareto optimal result.

We discuss in section II-C.2 (in the paragraph above figure 1) that there are numerous solutions which minimise the maximum utilisation but with different resource usage. Figure 2 shows the link utilisations at the end of the optimisation. It can be seen if we minimise the maximum link utilisation only, the traffic is evenly distributed but many links have maximum utilisation. If we optimise the resource alone, figure 2 shows that the traffic is non-evenly distributed.

In section III, we have set aside the constraints on the total number of routes in the network. For the multiobjective formulation, we need to use altogether 851 routes for all the three service classes together. If we have used a fully meshed network for each service class, this would have required 816 routes. This means that most of the aggregate demands are routed using one path. Of all those aggregate demands that are split into multiple routes, all but one uses 2 routes and only one uses 3 routes.

V. Conclusions

In this paper we have proposed a multiobjective formulation for the MPLS-based VPN traffic engineering problem. This multiobjective formulation takes both resource usage and maximum link utilisation into account. We demonstrate that this multiobjective formulation overcomes the problems of single objective formulations (e.g. minimising resource usage and minimising maximum link utilisation) that have appeared in the literature. The optimisation problem that we have formulated is NP-complete and involves a large number of binary decision variables. We have proposed an heuristic solution, which allows tractable solution.

Optimisation criterion	Maximum link utilisation	Network resource usage (Gbps)
Minimum resource	1.000	74.7
Minimax link util.	0.727	128.2
Multiobjective	0.728	74.8

TABLE I

Results for the example in section IV.

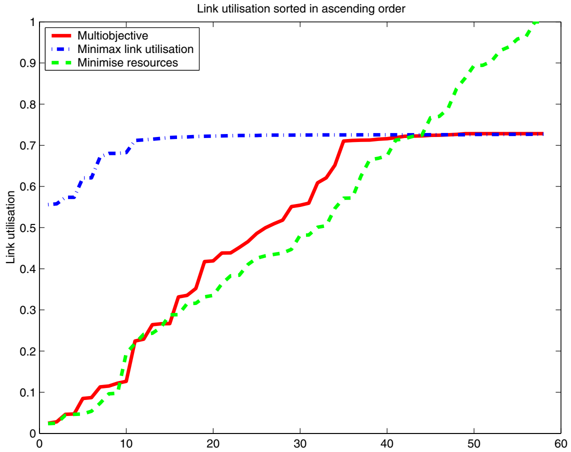

Fig. 2. This graph shows the link utilisations resulted from using the three optimisation criteria. The link utilisations have been sorted in ascending order.

References

[1] X. Xiao, A. Hannan, B. Bailey, and L.M. Ni, "Traffic engineering with MPLS in the Internet," *IEEE Network*, pp. 28–33, March-April 2000.

[2] A. Juttner, B. Sziatovszki, A. Szentesi, D. Orincsay, and J. Harmatos, "On-demand optimization of label switched paths on mpls networks," in *Proc. 9th Int. Conf. on Computers Communications and Networks 2000*, 2000, pp. 107–113.

[3] M.K. Girish, B. Zhou, and J.Q. Hu, "Formulation of the traffic engineering problems in MPLS based IP networks," in *Proc. of the 5th Int. Sym. on Computers and Communications(ISCC 2000)*, 2000, pp. 214–219.

[4] Y. Wang and Z. Wang, "Explicit routing algorithms for Internet traffic engineering," in *Proc. 8th Int. Conf. Computer Communications and Networks*, 1999, pp. 582–588.

[5] K. Kar, M. Kodialam, and T.V. Lakshman, "Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering," *IEEE Journal of Selected Areas in Communications*, vol. 18, no. 12, pp. 2566–2579, 2000.

[6] E. Rosen and Y. Rekhter, "BGP/MPLS VPNs," Request for Comments 2547, Internet Engineering Task Force (IETF), March 1999.

[7] D. Awduche et al., "Requirements for traffic engineering over mpls," Request for Comments 2702, Internet Engineering Task Force (IETF), September 1999.

[8] S. Martello and P. Toth, *Knapsack problems*, John Wiley, 1990.

[9] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman and Co., 1979.

[10] A. Caprara, H. Kellerer, and U. Pferschy, "A PTAS for the multiple subset sum problem with different knapsack capacities," *Information Processing Letters*, vol. 73, pp. 111–118, 2000.