A Tool for MPLS Traffic Engineering

Luís Cardoso *, Jorge Patrão **, Carlos Lopes **, Amaro de Sousa **, Rui Valadas **

* Portugal Telecom Inovação, Rua Eng. José F. Pinto Basto, 3810 Aveiro, Portugal

** Instituto de Telecomunicações/ Universidade de Aveiro, 3810-193 Aveiro, Portugal

Abstract

This paper presents a tool for MPLS Traffic Engineering that allows for multi-hour dimensioning of networks supporting simultaneously peer-to-peer and client-server services. It is assumed that all LSPs to be routed and their attributes are known. The dimensioning problem is a combined capacity design and routing problem where the LSP sets are calculated in order to minimise the network operational costs. This problem is formulated as an integer programming problem, which is solved through an heuristic based on Lagrangean relaxation with sub-gradient optimisation. The network design tool, named PTPlan MPLS, includes a graphical interface for an easy introduction and edition of the network parameters. Results show that the tool can design large networks with good computational speed.

I. INTRODUCTION

Multi-Protocol Label Switching (MPLS) is an advanced forwarding scheme for IP networks supporting Quality of Service (QoS). MPLS adds labels to the header to switch packets, in a similar way to ATM. A router that supports MPLS is known as a Label Switching Router (LSR). An MPLS path is called a Label Switched Path (LSP) and an explicit LSP has its route determined at its originating node. In MPLS, packets are encapsulated at ingress points of an MPLS domain with labels that are then used to forward the packets along LSPs. All packets can be divided into subsets called Forward Equivalence Classes (FECs) [1]. The idea is that packets belonging to the same subset are forwarded in the same manner. Classification into FECs is done using packet filters that examine header fields such as source address, destination address, and type-of-service bits. The granularity of the packet forwarding mechanism may therefore vary depending on this classification. The mapping of packets to FECs is performed only once when the packets enter an MPLS domain. The purpose of classifying packets into FECs is to enable the service provider to differentiate packet flows with different QoS requirements and route each FEC in the most appropriate manner. This is done by mapping arriving packets belonging to an FEC to one of the LSPs associated with the FEC. When the packets are mapped onto an LSP, there are two possible cases. The LSP is already established and its capacity is augmented through a resource reservation signalling protocol such as Resource Reservation Protocol (RSVP). The LSP is not yet established and it is set up, along an explicit route if specified, using a resource reservation signalling protocol such as Resource Reservation Protocol (RSVP) or Constrained Routing Label Distribution protocol (CR-LDP). Using such protocols enables the set up of LSPs with appropriate resources to meet the required QoS for the supported FEC.

Traffic Engineering (TE) is defined as the part of Internet network engineering that deals with the performance evaluation and optimisation of operational IP networks [1]. Traffic Engineering is needed in the Internet mainly because current routing protocols forward traffic based only on destination addresses and along shortest paths computed using mostly static and traffic-insensitive link metrics. While this shortest path routing is enough to achieve connectivity, it does not always make good use of available network resources. A prime problem is that some links in the shortest path between certain origin-destination router pairs may get congested while links on possible alternate paths remain free. This shortcoming of network operation, coupled with the phenomenal growth of Internet usage, makes it very difficult to manage IP-based network performance [2]. The explicit routing feature of MPLS was introduced to address the shortcomings associated with current IP routing schemes, providing means for ingress routers to control traffic trajectory precisely. Controlling the way in which traffic flows are routed into the network is of fundamental importance for resource optimisation and it is one of the main objectives of TE.

In MPLS, explicit LSPs can be used to configure different logical networks on top of the physical network. These LSPs can be thought of as virtual trunks that carry flow aggregates generated by classifying the packets arriving at the ingress routers of an MPLS network into FECs. Therefore, a logical network composed by a set of explicit LSPs can be configured in the network to support the traffic flows of each FEC. A similar approach has been used in the past for ATM network dimensioning [3], [4], where Virtual Path Connections are used instead of LSPs.

Typically, existing services are either peer-to-peer or client-server based. In peer-to-peer services, there are traffic flows between any pair of nodes with attached users. Thus, an LSP must be configured for each user node pair. In client-server services (e.g., video-on-demand, database access), there are traffic flows between a user node and one of the server nodes of the service. Thus, an LSP must be configured between each user node and a server node.

It is common sense that existing services do not have the same traffic behaviour during time. For example, business services have higher traffic flows in periods that are complementary to residential services. If the network has dynamic reconfiguration capabilities, i.e., time is partitioned in periods and the explicit LSPs can be reconfigured between time periods, a multi-hour design procedure can lead to significant savings in the overall network cost.

This paper presents a software tool for MPLS Traffic Engineering, called PTPlan MPLS. This tool is an evolution of another network dimensioning tool for ATM networks [5]. PTPlan MPLS performs multi-hour network dimensioning and considers mixed peer-to-peer and client-server services. It is assumed that all LSPs to be routed and their attributes are known. In the case of client-server services the attributes are origin node, a set of pre-defined candidate server nodes, maximum bandwidth and, optionally, preferred routes. In the case of peer-to-peer services the attributes are origin node, destination node, maximum bandwidth and, optionally, usable colours, maximum hop count, preferred routes, and survivability. Link attributes must also be defined: maximum utilisation and colour. The attributes defined for each LSP act like constraints that must be satisfied when routing each LSP. The preferred routes attribute is a set of routes from which the LSP route must be selected. The usable colours can be used to forbid a link to be used by a certain LSP, e.g., red links cannot be used for a green LSP. Maximum hop count attribute can be used to limit the maximum number of hop count for a particular LSP. For the survivability, two types are considered: link disjoint and node disjoint. In both cases, when this attribute is set for a particular LSP, the tool divides the LSP in two, giving each one a bandwidth between 50% and 100% of the original LSP. When node disjoint is considered, the routes of the LSPs must be node disjoint along the entire path between origin and destination. When link disjoint is considered, the routes of the LSPs must be link disjoint, but in this case they can have common nodes.

In section II we present the model that defines the network dimensioning problem and describe how it can be solved. In section III we show the main features of the PTPlan MPLS tool. In section IV we discuss a case study and in section V present some conclusions.

II. MPLS NETWORK DIMENSIONING

Let the network be represented by an undirected graph *(N, A)* whose nodes and arcs represent LSRs locations and available transmission facilities between LSR locations. Each element of A is defined by an undirected arc (i,j), with $i, j \in N$. The set of possible interface types to install in any arc is denoted by T and α_t is the bandwidth of interface type $t \in T$. We also define f_{ij} as the colour of the arc (i,j) and u_{ij} as the maximum utilisation. Let Y_{ij}^{t} denote the maximum number of interfaces of type $t \in T$ that can be installed on (i,j). The operational and maintenance cost associated with the use of one interface of type $t \in T$ in the arc (i,j) is denoted by C_{ij}^{t}. Let H be the set of time periods h. Each LSP k_h is defined by the origin node $o(k_h)$, the set of possible destination nodes $d(k_h) \in D(k_h)$, the bandwidth in the direction from origin to destination $b(k_h)$, the bandwidth in the direction from destination to origin $\underline{b}(k_h)$, the set of preferred routes $R(k_h)$, the set of usable colours $U(k_h)$, the maximum hop count $m(k_h)$ and the type of survivability $s(k_h)$ (no_survivability, link_disjoint and node_disjoint). Set $D(k_h)$ is (i) a single destination for peer-to-peer services and (ii) the set of server nodes for client-server services. The optimisation model uses the following set of variables. Integer variables y_{ij}^{t} that define the number of interfaces of type t that are installed on arc $(i,j) \in A$. Route binary variables x_{ij}^{kh}, when equal to one, define that LSP k_h passes through arc $(i,j) \in A$ in the direction from node i to node j. Route binary variables $\underline{x}_{ij}^{kh}$, when equal to one, define that LSP k_h passes through arc $(i,j) \in A$ in the direction from node j to node i. The following integer programming model determines the lowest cost physical network given by the optimal constrained routes for each LSP and the optimal server locations:

$$\text{Minimise} \quad \sum_{(i,j)\in A} \sum_{t \in T} C_{ij}^{t} y_{ij}^{t} \tag{1}$$

Subject to:

$\{ x_{ij}^{kh}, \underline{x}_{ij}^{kh} : x_{ij}^{kh} = 1 \wedge \underline{x}_{ij}^{kh} = 1 \}$ is a path subject to constraints set $\{R(k_h), U(k_h), m(k_h), s(k_h)\}$ from $o(k_h)$ to $d(k_h) \in D(k_h)$, $k_h \in K_h$, $h \in H$ (2)

$$\sum_{k_h \in K_h} (b(k) \cdot x_{ij}^{kh} + \underline{b}(k) \cdot \underline{x}_{ij}^{kh}) < u_{ij} \sum_{t \in T} \alpha_t y_{ij}^{t}, \ (i,j) \in \text{A}, \ h \in \text{H} \tag{3}$$

$$\sum_{k_h \in K_h} (b(k) \cdot \underline{x}_{ij}^{kh} + \underline{b}(k) \cdot x_{ij}^{kh}) < u_{ij} \sum_{t \in T} \alpha_t y_{ij}^{t}, \ (i,j) \in \text{A}, \ h \in \text{H} \tag{4}$$

$$y_{ij} < Y_{ij}, \ (i,j) \in \text{A}, \ t \in \text{T} \tag{5}$$

$$x_{ij}^{kh} \in \{1,0\}; \ \underline{x}_{ij} \in \{1,0\}; \ y_{ij} > 0 \text{ and integer} \tag{6}$$

The objective function (1) represents the total cost of the network solution as a function of the number of interfaces of each type installed in each network arc. Constraint (2) forces the solution to be a constrained path from origin to destination for all LSPs to be supported by the network. As it will be understood later, there is no need to explicitly define constraint (2). Constraints (3) and (4) impose that the total bandwidth installed in each arc is enough to support the bandwidth occupied by the LSPs that cross the arc in both directions and in all time periods. Finally, constraint (5) guarantees that the number of interfaces of each type in each arc is not greater then their maximum values.

The solution to the optimisation problem is obtained using Lagrangean relaxation with sub-gradient optimisation. Departing from the original problem, a new optimisation problem is obtained by applying Lagrangean relaxation to constraints (3) and (4), which we refer to as the Lagrangean Lower Bound Problem (LLBP). To derive the LLBP, a set of Lagrangean multipliers is introduced, one for each of the relaxed constraints. For any arbitrary set of non-negative Lagrangean multipliers, the solution of LLBP is a lower bound of the original problem [6]. Using the x variables solution of LLBP, it is possible to calculate a feasible solution in the original problem using constraints (3) and (4) to find the minimum values for variables y. To compute different sets of Lagrangean multipliers, we use sub-gradient optimisation [7]. This technique is an iterative process that, for a given set of Lagrangean multipliers, calculates another set of multipliers that try to maximise the objective function value of the LLBP. At the end of the procedure, a final solution is obtained which is the best of all calculated feasible solutions. The solution of the LLBP for the route variables is a shortest path calculation in the case of client-server services. For the peer-to-peer services, we developed an algorithm that deals with multiple constraints. In this algorithm, we use the SPLDP (shortest pair of link disjoint paths algorithm) and the SPNDP (shortest pair of node disjoint paths algorithm) presented in [8], which gives the shortest pair of link disjoint and node disjoint paths, respectively. We also use the MDSPHL (modified Dijkstra shortest path hop-limit algorithm) presented in [9], that gives the shortest path which satisfies hop-limit constraint.

The algorithm that determines the route for each LSP is briefly described below.

```
if R(k_h)<>∅
    Calculate minimum cost route from R(k_h);
else
{
    bool exist_colour_shortest_path = false;
    if U(k_h)<>∅
    {
        Calculate a new set of arcs A', where all unusable arcs are
        pruned;
        Calculate colour shortest path using Dijkstra(N,A'), if path
        exists set exist_colour_shortest_path = true;
    }
    if s(k_h)<>no_survivability and exist_colour_shortest_path
    {
        bool survivability_exist = false;
        if s(k_h) == link_disjoint
            survivability_exist = SPLDP(o(k_h), d(k_h), N, A');
        else if s(k_h) == node_disjoint
            survivability_exist = SPNDP(o(k_h), d(k_h), N, A');
        if survivability_exist = false
            Use colour shortest path for both LSPs;
    }
    else if s(k_h)<>no_survivability
    {
        bool survivability_exist = false;
        if s(k_h) == link_disjoint
            survivability_exist = SPLDP(o(k_h), d(k_h), N, A);
        else if s(k_h) == node_disjoint
            survivability_exist = SPNDP(o(k_h), d(k_h), N, A);
        if survivability_exist = false
            Dijkstra(o(k_h), d(k_h), N, A);
    }
    else if m(k_h)<>∞ and exist_colour_shortest_path
    {
        bool maximum_hops_path_exist = false;
        maximum_hops_path_exist = MDSPHL(o(k_h), d(k_h),
        m(k_h), N, A');
        if maximum_hops_path_exist = false
            Use colour shortest path;
    }
    else if m(k_h)<>∞
    {
        bool maximum_hops_path_exist = false;
        maximum_hops_path_exist = MDSPHL(o(k_h), d(k_h),
        m(k_h), N, A);
        if maximum_hops_path_exist = false
            Calculate shortest path using (N,A);
    }
    else if exist_colour_shortest_path
        Use colour shortest path;
    else
        Calculate shortest path using (N,A);
}
```

This algorithm is proposed in such a way that when it is not possible to comply with some of the constraints, a solution is selected taking into account only the remaining constraints.

III. THE PTPlan MPLS TOOL

The PTPlan MPLS tool includes two blocks: one for network dimensioning and another for routing management considering QoS constraints. The dimensioning problem is a combined capacity design and routing problem where the LSP sets are calculated in order to minimise the network operational costs (already described in previous section). The routing management problem assigns routes to each LSP in a network that is deployed on the field.

In PTPlan MPLS, the user can define the network topology, the service characteristics, the traffic scenario and (optionally) impose constraints to the LSPs. Based on this information, the tool determines a physical network configuration and the LSP routes and bandwidths that can support the traffic scenario. If the user defines also the physical configuration, the tool performs routing management. PTPlan MPLS runs on a Windows platform and includes a graphical interface (Figure 1, Figure 2) through which the user can enter and edit the network topology (nodes and links between nodes). The user can create a library of interfaces that can be used to define the network nodes. A library of nodes can also be created. After node definition, each link is automatically assigned the set of interfaces that are common to its adjacent nodes. In addition the user can associate a distance, a maximum utilisation and a colour to each link and define the maximum number of interfaces of each type that the link can support. For the routing management problem, the user must also define the existing number of interfaces of each type. Both a switching and a transmission cost can be assigned to each interface in each link. Transmission costs can be based on link length. The cost of each interface is calculated as 2 × switching cost + transmission cost × link length. There is also the possibility of creating subnets. In the “service characterisation” window (Figure 1), services can be selected as being peer-to-peer or client-server. In the first case, the user must define the bandwidth per flow, that is equal in both directions, and in the second case the bandwidth per flow in each direction.

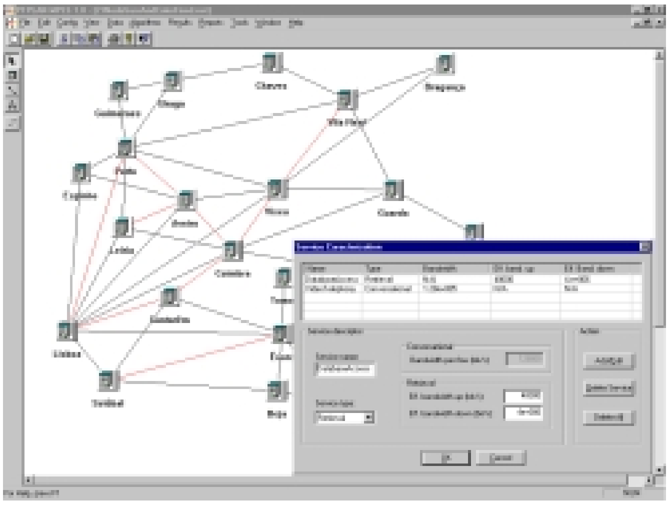

Figure 1 GUI of PTPlan MPLS – service characterisation

In the "case study" window (Figure 2), the user defines the number of time periods (for multi-hour designs), which nodes have attached users and, in the case of client-server services, which nodes can have servers. The tool automatically generates the origin-destination pairs, in the case of peer-to-peer services, and the origin nodes in the case of client-server services. After that the user is asked to define the maximum number of flows expected for each origin-destination pair. Optionally, it is possible to impose constraints for each LSP. Those constraints include usable colours, maximum hop count, type (node or link disjoint) of survivability and preferred routes. After defining all network parameters, the user may perform network dimensioning or routing management, and to select between using uni- or multi-hour design. The results given by the tool are the LSP routes and, in the case of network dimensioning, the number of interfaces of each type to install in each link and the overall network cost.

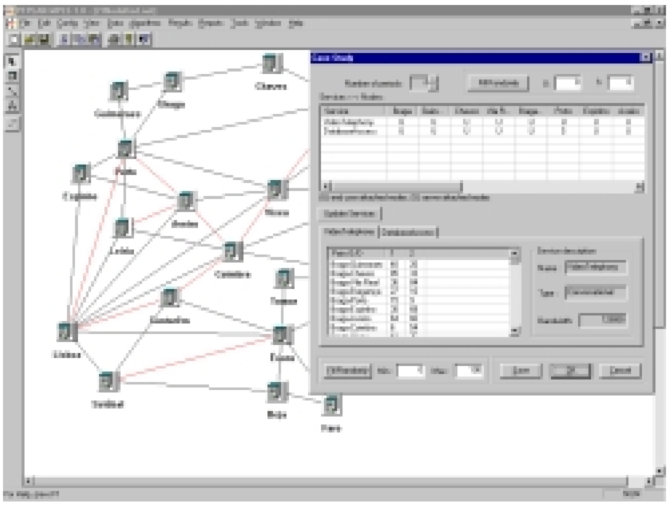

Figure 2 GUI of PTPlan MPLS – case study

IV. CASE STUDY

The network topology is represented in Figure 1, and has 21 nodes and 46 links. There are 3 interface types: Channelized E1, Unchannelized E3 and OC3c/STM-1, with the following costs:

	Bandwidth (Mbps)	Switching Cost	Transmission Cost
Channelized E1	2	100	1
Unchannelized E3	34	1000	10
OC3c/STM-1	155	3000	40

We have considered two services, *Video-telephony (VT)* and *Database Access (DA)*, and two time periods. Service *VT* is a conversational service, with flow bit rate of 128kbps. The *DA* service is a retrieval service, with flow bit rates of 40kbps in the client-server direction and 4000kbps in the server-client direction.

There are end users for both services at all nodes and two servers for the *DA* service located at Porto and Lisboa nodes. The number of flows in each connection was randomly assigned using a uniform distribution between 0 and 100.

We tested this network in two cases: (i) without constraints and (ii) imposing only link survivability (85% on all LSPs), for uni- and multi-hour design. For the client-server service, we considered two cases: (i) fixed server (each user is attached to the closest server) and (ii) selected server (the tool is allowed to select the best server for each node).

The results are as follows:

			Uni-hour		Multi-hour	
			Cost	Time* (sec)	Cost	Time* (sec)
No Constraints		Fixed Server	941,344	11.08	842,589	16.35
		Selected Server	934,849	11.05	836,679	16.55
Survivability	Node Disjoint	Fixed Server	1,128,400	24.47	975,690	42.55
		Selected Server	1,117,280	24.25	973,934	41.96
	Link Disjoint	Fixed Server	1,114,240	23.39	949,513	40.99
		Selected Server	1,107,680	23.00	947,125	41.11

*Using an AMD Athlon 800Mhz with 128 MB RAM

As expected, gains are obtained when adopting a multi-hour approach and a selected server strategy: the cost difference between fixed server/uni-hour and selected server/multi-hour is between 11.1% (no constraints) and 15.0% (imposing a link disjoint survivability). We note that, although the results can vary significantly with the network and traffic scenario, in general considerable gains are obtained by resorting to a multi-hour approach and selected server strategy.

Imposing survivability significantly increased the cost of the achieved solution, and the safest option (node disjoint) is the most expensive.

Finally, we created a more complex scenario combining different constraints and setting all links *Black*, except Porto-Lisboa, Porto-Aveiro, Aveiro-Leiria, Aveiro-Coimbra, Lisboa-Santarém, Santarém-Coimbra, Viseu-Coimbra, Viseu-Vila Real and Setúbal-Évora, set to *Red*.

We imposed on some of the LSPs a maximum of 6 hops, a node disjoint survivability of 85% and used the colour constraint to forbid them from using red links. The results were as follows:

	Uni-hour		Multi-hour	
	Cost	Time* (sec)	Cost	Time* (sec)
Fixed Server	1,195,890	25.31	1,029,070	43.98
Selected Server	1,095,510	24.66	973,172	44.25

*Using an AMD Athlon 800Mhz with 128 MB RAM

The cost of the solution and the computational time required significantly increased, when compared to the case of a non-constrained network.

The tool was able to comply with all the restrictions imposed and design a network in less than 45 seconds.

V CONCLUSIONS

In this paper we described a traffic engineering tool that performs MPLS network dimensioning. The dimensioning procedure is based on the determination of appropriate explicit LSPs for the support of each FEC. The solution method also allows the user to impose routing constraints to the LSPs and different kinds of survivability. Computational results show that the tool can find solutions in low computing times.

V. REFERENCES

[1] Awduche et al, "A Framework for Internet Traffic Engineering", *Internet Draft - Work in Progress*, January 2000.

[2] W. Lai, "Capacity Engineering of IP-based Networks with MPLS", *Internet Draft - Work in Progress*, March 2000.

[3] A. de Sousa, R. Valadas, L. Cardoso, A. Duarte, "ATM Network Dimensioning for Mixed Symmetrical and Asymmetrical Services with Dynamic Reconfiguration in a Multi-Network Provider Environment", *Third IFIP Workshop on Traffic Management and Design of ATM Networks*, England, 1999, pp. 5/1-5/15.

[4] D. Medhi, "Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-Based Dynamically Reconfigurable Wide-Area ATM Networks", *IEEE/ACM Transactions on Networking*, Vol. 3, No. 6, 1995, pp. 809-818

[5] L. Cardoso, C. Lopes, A. Sousa, R. Valadas, "A Tool for Multi-Hour ATM Network Design considering Mixed Peer-to-Peer and Client-Server Services", *Networks 2000*, Toronto, 2000

[6] J. Beasley, "Lagrangean Relaxation", *Modern Heuristic Techniques for Combinatorial Problem*, Ed. by Colin Reeves, Blacwell Scientific publications, 1993

[7] M. Held, P. Wolfe, H. Crowder, "Validation of subgradient optimisation", *Mathematical Programming*, vol. VI, pp. 62-88, 1974.

[8] R. Bhandary, "Optimal Diverse Routing in Telecommunication Fiber Networks", IEEE, 1994

[9] S. Choudhury, L. Berry, "Two methods for Minimum Cost Synthesis of Link-disjoint, Hop-limited Virtual Paths in ATM Networks", ITC 16, pp. 613-622, 1999