Providing Traffic Engineering Capabilities in IP Networks Using Logical Paths

P. Trimintzios, G. Pavlou and I. Andrikopoulos

Networks Group, Centre for Communication Systems Research,
University of Surrey, Guildford, Surrey, GU2 7XH, U.K.
{P.Trimintzios, G.Pavlou, I.Andrikopoulos}@eim.surrey.ac.uk
http://www.ee.surrey.ac.uk/CCSR/Networks/

Abstract. The tremendous growth of IP networks in the past few years has led to the need for traffic engineering in order to enable the effective provision of Quality of Service (QoS). Several mechanisms and service models have been developed to assist traffic engineering solutions. In this paper we propose an approach to provide traffic engineering capabilities in IP Networks by using logical paths and making use of constraint-based optimisation algorithms. Our approach is both proactive and reactive, and considers the existence of the Differentiated Services model. We present an architecture of a management system, which orchestrates the traffic engineering activities. Finally, we identify some important implementation issues for realising, evaluating and validating our approach, and we point out the directions of our future research efforts.

Keywords: IP traffic engineering, capacity and routing management, resource allocation, performance optimisation, constraint-based routing.

1 Introduction

Over the past decade computer networks have quickly evolved into a critical communications infrastructure supporting significant economic, educational, and social activities. Consequently, optimising the performance of large-scale networks at the minimum cost, especially public backbones, has become an important problem. Network performance requirements are multidimensional, complex, and sometimes contradictory, thereby making the performance optimisation very challenging. The network must convey traffic from ingress nodes to egress nodes efficiently, expeditiously, reliably, and economically. Furthermore, in multi-class service environments (e.g. Differentiated Services capable networks), the resource sharing parameters of the network must be appropriately determined and configured according to prevailing policies and service models, to satisfactorily resolve resource contention issues arising from mutual interference between packets traversing the network.

Changing architectural paradigms and simply expanding the physical network capacity, are necessary actions, but they are *not sufficient* to deliver high QoS assurances under all possible circumstances. Some means of controlling and managing the infrastructure are essential for the realisation of the envisioned global QoS network. This means is Traffic Engineering (TE).

IP TE deals with the issues of performance evaluation and performance optimisation of operational IP networks. The IP Traffic Engineering problem is to establish and fine-tune the parameters and operating points of network aspects, in order to address the network growth challenge. Consequently, this is fundamentally a network control and management problem.

In this paper we propose an approach where the management and control system, together with the accompanied constraint-based Traffic Engineering algorithms, are decoupled from the network, i.e. are not embedded in network nodes, but reside outside the network. Note that this does not mean that we are assuming a centralised approach, since the management and control algorithms could be deployed in a distributed fashion.

The paper is organised as follows. Section 2 provides an introduction to the basic concepts and current trends in IP TE. Section 3 presents the definition of the problem of IP TE by using Explicitly Routed Paths. In Section 4 we analyse the problem and we propose algorithms for solving it. Section 5 presents the overall architecture that we propose for realising our IP TE approach. Finally, Section 6 summarises our conclusions and presents the directions for our further study.

2 IP Traffic Engineering

IP TE aims to provide optimisation of network resources in order to satisfy traffic performance objectives at a minimum cost. It encompasses the application of technology and scientific principles to the measurement, characterisation, modeling and control of IP traffic [7]. The purpose of TE is to give the network administrator precise control over the flow of traffic within his/her administrative domain (Autonomous System - AS). Traffic Engineering capabilities are necessary because the standard Interior Gateway Protocols (IGPs) (routing protocols) compute the shortest paths based solely on the metric that has been administratively assigned to each link. This leads to uneven load distribution, where parts of the network become congested, while other parts are under-utilised [24]. These protocols also suffer from lack of dynamics and responsiveness.

There were attempts in the past to provide traffic engineering capabilities to IP networks, but the concept of IP Traffic Engineering was not explicitly defined. Only recently, initiatives for providing a standard framework, as well as solutions, for IP TE within various standardisation bodies started to appear.

Initially, network providers responded to the TE challenge by simply *over-provisioning* more links to provide additional bandwidth. Later by using *adaptive routing*, forwarding decisions were based on the current state of the network. On the other hand, manipulating dynamic link metrics tends to create oscillations since traffic is repeatedly shifted (route flapping) from one location of the network to another. TE was achieved by simply manipulating the routing metric of dynamic routing algorithms. *Metric-based control* was adequate because networks were small but it was not scalable. *Equal Cost Multi-Path* (ECMP) is another technique aiming at addressing the deficiency in Shortest-Path-First (SPF) routing systems. In ECMP, if two or more equal cost paths exist between two nodes, the traffic between the nodes is distributed among multiple equal paths. The main drawbacks of this approach are that packets arrive out-of-order, and that the distribution of traffic does not depend on the congestion status of each path but on the number of ECMPs. Later on, the volume of IP traffic reached a point that forced carriers to redesign their networks to make use of higher speeds supported by switched high-speed ATM and Frame Relay (FR) cores (*overlay model*). Virtual-Circuit (VC) networks provide connectivity among routers that are located at the edges. TE is performed by re-configuring the VCs so that congested physical paths are avoided. Although this approach is quite effective, it has a very important drawback; it requires management of two separate networks, which increases complexity and operational costs.

It is clear that any router-based TE approach *must* provide a level of functionality equivalent to the *overlay model*, since the carriers used to the deterministic performance of the VC TE model and will not settle for anything less. Nowadays, the trend, is to evolve the core IP networks away from the overlay model, towards more *integrated solutions*, which are now possible because of developments like Differentiated Services, Multi-Protocol Label Switching (MPLS) and Constraint-based routing .

2.1 Current Trends in IP Traffic Engineering

Although the tools required to provide TE have been defined, to our knowledge there has not yet been a solution which orchestrates and combines them. In this section we describe some important new technologies and models, which enable to perform traffic engineering and provide QoS capabilities to the otherwise Best Effort Service model of IP networks.

Differentiated Services. The Differentiated Services [8, 27] (DiffServ) approach to providing QoS employs a small, well-defined set of building blocks from which a variety of services may be built. A small bit-pattern in each IP packet, called the *DSCP* (Differentiated Services Code-Point [26]), is used to tell the routers how to process a packet. By marking the *DS field* of packets differently, and processing them accordingly, several differentiated service classes can be created. DiffServ can thus be regarded as a *relative-priority* model [32], in which *traffic aggregation* is the key feature. One of the primary goals of the DiffServ model is to move all the complexity to boundary routers while leaving the core routers as simple as possible, so that boundary routers of a domain have added responsibilities such as: *classification*, *metering*, *marking/re-marking*, *authentication*, *policing* and *shaping/re-shaping* of packets. For traffic management support, carrier backbones are evolving towards the DiffServ architecture because of its major advantage of scalability, since it does not need to keep per-flow state information but only per-class-of-service.

The DiffServ model does not specify the services that will be supported, but instead it defines the building blocks by which an arbitrary number of service classes could be offered. These building blocks are the externally observable behaviours of a node (router) to the packets, and they are called *Per-Hop Behaviours* (PHBs). A Behaviour Aggregate (BA) is the collection of packets with the same DS codepoint traversing a link in a particular direction, and Per-Domain Behaviour (PDB) is the expected treatment that an identifiable or target group of packets will receive from "edge to edge" of a DS domain. A particular PHB (or, if applicable, list of PHBs) and traffic conditioning requirements are associated with each PDB. Different PHBs are applied to different BAs in order to realise certain PDBs, and are implemented using different forwarding treatments [21], i.e. *scheduling disciplines* and *buffer management mechanisms* at each network node. Customers[1] negotiate a *Service Level Specification* (SLS) with a service provider. This specification contains information, such as the characteristics of the traffic that the customer wants to inject in the network, QoS requirements, charging and profiling information, etc.

MPLS. The *Multi-Protocol Label Switching* [9, 29] technology combines the label swapping forwarding paradigm (label switching) with network layer (Layer 3) routing. The basic idea is to assign short, fixed-length labels to packets at the ingress to an MPLS domain, based on the concept of *Forwarding Equivalence Classes* (FECs). An FEC is a group of IP packets, which are forwarded in the same manner (e.g., over the same path, with the same forwarding treatment), i.e. the set of packet flows with common cross-core path forwarding requirements. In the OSI seven-layer model, MPLS would lie between Layer 2 (data link layer) and Layer 3 (network layer).

MPLS is responsible for directing flows of IP packets or an aggregate of flows, across a predetermined path along the network. The routers along that path are called *Label Switched Routers* (LSRs) and the path is called *Label Switched Path* (LSP). An LSP can be manipulated, and managed by the network administrators to direct traffic. MPLS also defines the *traffic trunk* [7, 24], which is the traffic of a single traffic class that is aggregated into a single Label Switched Path (LSP)[2]. It is useful to view traffic trunks as objects that can be routed; that is, the path which a traffic trunk traverses can be changed. MPLS is strategically significant because it can provide router-based networks with some advantages of circuit-switched networks, while avoiding most of their disadvantages. Generally, the route for a given LSP can be established in two ways: a) *Control-Driven* (also called *hop-by-hop LSP*); or b) *Explicitly Routed* (ER-LSP). When setting up a hop-by-hop LSP, each LSR determines the next interface to route the LSP, based on its L3 routing information, i.e. it follows the path normal L3 routed packets would have taken. When setting up an ER-LSP, the route for that LSP is specified in the "setup" message itself, and this route information is carried along the nodes the setup message traverses. ER-LSP can be *specified* and *controlled* by the network management applications to direct network traffic, independent of the L3 technology. For the intra-domain case, for setting up a hop-by-hop LSP, the *Label Distribution Protocol* (LDP) [2] has been proposed as the control protocol. For setting-up intra-domain ER-LSPs, it is proposed to use extensions to the LDP protocol, the Constraint-Based LPD (CR-LDP), or extensions to the Resource ReSerVation Protocol (RSVP), and extensions to BGP-4 for the inter-domain case.

MPLS is strategically significant for TE due to its *path-oriented feature*, so it can provide most of the functionality available from the IP overlay model, while avoiding most of its drawbacks. Although, the concept of TE does not depend on a specific layer 2 technology, MPLS is argued [3, 5, 7, 30] to be the most suitable tool to provide TE. MPLS allows sophisticated routing control capabilities as well as QoS resource management techniques to be introduced into IP networks. The key factors that make MPLS attractive for TE are the following:

[1] By customers it is meant users, organisations, even other service providers.

[2] It is important to emphasize that there is a fundamental distinction between a traffic trunk and path (LSP). An LSP is *only* the *path* through which traffic traverses.

- ER-LSPs can be easily created, maintained and modified, through manual or automated administrative actions.
- Through ER-LSPs, MPLS permits a quasi-circuit switching capability to be superimposed on the current IP routing model.
- Attributes can be associated with traffic trunks and resources, which modulate the trunks' behavioural characteristics and constrain the placement of LSPs on resources.
- MPLS allows both traffic aggregation and disaggregation, whereas classical destination-based IP forwarding permits only aggregation.
- It is relatively easy to integrate a constraint-based routing with MPLS.

Constraint-based Routing. MPLS does not specify how the Explicitly Routed paths (ER-LSPs) are determined or how congestion is detected. Currently, IP traffic is routed and forwarded by using the standard *dynamic routing* protocols, which offer little or no readily available traffic control capabilities and always select the shortest paths to route/forward packets, resulting in some paths to becoming congested while others are idle. These routing protocols are load insensitive. *Constraint-based routing* can be used to compute paths (ER-LSPs) subject to multiple constraints.

Constraint-based Routing evolves from *QoS routing* [11, 25]. QoS routing returns the route that is most likely to be able to meet QoS requirements, given a QoS request of a flow or an aggregation of flows. Constraint-based Routing extends QoS routing by considering additional constraints, such as *policies*. The goals of Constraint-based Routing are three-fold: i) select routes that can meet certain QoS requirements, ii) avoid congestion and improve the user's satisfaction (*user's utility maximisation* [10]), iii) maximise the network utilisation (optimise the resource utilisation). Constraint-based Routing allows a demand-driven, resource-reservation-aware routing paradigm, to co-exist with current topology-driven hop-by-hop IGPs. A constraint-based routing framework uses as input the following: attributes associated with traffic trunks, attributes associated with resources, and topology and network state information.

Based on this information, a constraint-based traffic engineering system automatically computes explicit routes for each traffic trunk originating from the node. In this case, an explicit route for each traffic trunk is the specification of a label-switched path that satisfies the demand requirements expressed in the trunk's attributes, subject to constraints imposed by resource availability, administrative policy and other network state information. Note that similar constraint-based TE algorithms could be used by a management system [6] (either centralised or distributed). We propose such a management system in this paper.

3 Problem Statement

3.1 Problem Definition

The problem we are addressing in this work can be described as an optimisation problem, as follows:
Given a fixed physical topology and a source-destination matrix of offered traffic, which routing and capacity management decision offers the best overall performance, at the minimum cost?
The practical function of the above problem is the mapping of traffic onto the network infrastructure, and the dimensioning of this infrastructure to achieve specific performance objectives.

We propose the use of logical paths, and we introduce the concept of Explicitly Routed Paths (ERPs) (for example the MPLS ER-LSPs) for the realisation of the capacity and routing decisions stated in the problem definition above. By using constraint-based traffic engineering algorithms, which compute the routing and the dimensioning of ERPs, we seek to optimise the network performance objectives. Management decisions based on the output of these algorithms are employed, by configuring the various network elements, i.e. routers. By monitoring the performance of the network, we are able to evaluate the efficiency of the configuration. If the network performance is not satisfactory, then either dynamic control algorithms, resulting in small modifications in the current configuration, or global reconfiguration algorithms, need to be employed. In this sense, performance optimisation becomes an interactive (both *proactive* and *reactive*) procedure.

Similar problems concerning logical topology design and management have been studied in the literature for their application to ATM Networks, and more specifically by using Virtual Paths as the means to establish logical topologies (see [15], [18] and the references therein). Although these works for ATM and generally for circuit-switched networks, include some concepts which might be fairly generic, our approach differs because we are considering completely different technologies (DiffServ, MPLS, CBR), which do not have the same limitations and thus impose different sets of constraints. For example the constraint of the number of logical paths in ATM networks due to the limitation of the Virtual Path Identifier (VPI) size does not exist any more. Also limitations like the maximum call setup time, call blocking probability etc. are not applicable to IP DiffServ capable networks. Of course very generic concepts for example the triggering mechanism or generic optimisation heuristics, (see [18]), provide useful enlightening to our endeavours to solve the problem.

Performance Objectives. The performance objectives that we want to optimise can be categorised as *traffic-oriented* and *resource-oriented.*[3] Performance optimisation is accomplished by addressing the traffic requirements (constraints), while utilising network resources efficiently and economically. The *traffic-oriented* performance metrics may include: packet loss, delay, delay variation, and goodput. A measure of how effective a traffic-oriented policy is could be the relative proportion of the offered traffic satisfying its performance requirements to the overall offered traffic. The *resource-oriented* objectives include the optimisation of network resources, and throughput. Efficient resource management is the basic approach to secure resource-oriented performance objectives.

Another crucial objective is to *minimise the congestion problems* that are prolonged rather than the ones that are results of instantaneous bursts (i.e. transient congestion). Congestion typically occurs under two scenarios: a) when network resources are insufficient to accommodate the offered load, or b) when traffic streams are inefficiently mapped onto available resources, causing subsets of network to become over-utilised while others remain under-utilised. In the first scenario, the arising problems can be addressed by: augmenting network capacity, or modulating and conditioning, or throttling the demand, or applying classical congestion control techniques (flow control, rate shaping, tariffs etc.). In the second scenario, the problems can only be addressed through effective control and management techniques, i.e. by increasing the efficiency of resource allocation (routing and capacity management). In this work we are considering the cases where congestion occurs according to the second scenario.

3.2 Mathematical Model

Conceptually, the establishment and configuration of ERPs results in having a logical network on top of the physical network. We call this logical network *ERP graph* (also known as induced MPLS graph in [7]). The mathematical discipline we use to describe and model this logical network is Graph Theory. The reader should refer to [12] for the basic terminology, theorems and algorithms of Graph Theory. Describing the set of Explicitly Routed Paths as a graph is very important because the basic problem we defined in the previous section is essentially an issue of how to efficiently map such an ERP graph onto the physical network graph. Generally, an ERP graph abstraction problem can be formalised as follows:

Let $\mathcal{G} = (V, E, C)$ be a capacitated graph depicting the physical topology of the network, V is the set of nodes in the network and E is the set of links; that is,

$$\forall\, v, u \in V,\ (v, u) \in E \qquad \Longleftrightarrow \qquad (v \rightleftharpoons u) \triangleright \mathcal{G}\ . \tag{1}$$

i.e. (v, u) are in E if and only if v, u are directly connected under $\mathcal{G}$. The parameter c ($\in C$) is a set of capacity and other constraints associated each $v, u \in V$ and $(v, u) \in E$. $\mathcal{G}$ represents the physical network topology and its restrictions.

[3] Note that these objectives are of a trade-off relation, and the optimisation of both needs to be balanced. The balance parameter, i.e. the decision for which objective has higher priority, depends on the network operator's policies.

Let $\mathcal{ERP} = (U, F, D)$ be a capacitated graph depicting the logical network, i.e. represents the Explicitly Routed Path graph. U is a subset of V (i.e. $U \subseteq V$), representing the set of ERP nodes (e.g. Label Switched Routers) in the network, or more precisely the set of these ERP nodes that are endpoints of at least one ERP. F is the set of ERPs, so that for every $x, y \in U$, the object $(x, y) \in F$, if there is a logical path with x, y as endpoints. The parameter d ($\in D$) is the set of demands and restrictions associated with each $(x, y) \in F$ (i.e. with each logical path). $\mathcal{ERP}$ is a directed graph, and it can be seen that it depends on the transitivity characteristics of $\mathcal{G}$.

Let S be the set of boundary nodes (ingress or egress). It is obvious that $S \subseteq V$. Let M be a multidimensional matrix denoting the forecasted traffic. Each entry $m \in M$ is a set of constraints based on traffic characteristics and QoS requirements, associated with every node $s \in S$. Note that in this work we are considering only traffic aggregates (as defined by the DiffServ model), so m is a vector, containing constraints associated to class i of traffic aggregate. Let also R denote the set of network-wide resource-oriented constraints that are not captured by the previous constraints, for example an $r \in R$ might be the vector of network-wide throughput requirements per class i of the traffic aggregate.

So the problem we have described earlier can be modeled as follows:

$$\begin{array}{rl} \text{Find} & \mathcal{ERP}^* \mapsto \mathcal{G} \\ \text{subject to} & \\ \text{all} & c \in C,\ d \in D,\ m \in M,\ r \in R \quad \text{are met.} \end{array} \tag{2}$$

4 Problem Analysis

4.1 Problem Decomposition

In the previous section we have described the problem in its general form. We now decompose it into three sub-problems so that we can analyse it more easily:

1. **ERP topology design:** which pairs of nodes should be connected by ERPs, i.e. what is the topology of the logical path network? Note that at this level each ERP appears as a single logical link between two terminator nodes.
2. **ERP layout design:** how should the ERPs be mapped onto the physical network topology, i.e. what is the physical route for each logical path (which nodes should a logical path contain)?
3. **ERP dimensioning:** what is the dimension of each ERP, i.e. how much capacity should be assigned to each logical path? By capacity here we do not only bandwidth. Since we consider a DiffServ-capable network, in addition to bandwidth, the resources on each node include the scheduling weight, as well as the buffer dropping thresholds, applied to different BAs.

We assume that the MPLS, Constraint-Based Routing and the DiffServ model will be used in order to enable QoS and TE capabilities on IP networks. MPLS and Constraint-based traffic engineering algorithms are the means to build the logical path topology. In this work we consider a multi-class service environment (DiffServ), where traffic streams with different service requirements are in contention for network resources and this plays important role in dimensioning the ERP graph, since traffic engineering tasks must provide preferential treatment to some service classes in accordance with a utility model. There are limits to the number of BAs that an ERP can support [16].

When MPLS is used to realise the concept of ERPs, by determining ER-LSPs; there are two additional [7] sub-problems: a) how to do the mapping of incoming packets to Forwarding Equivalent Classes (FECs); and b) how to do the mapping of the Forwarding Equivalent Classes onto traffic trunks. In this work we do not focus on these sub-problems. Even though they are quite important, they are more or less subject to local policies and standardisation procedures and their definition does not affect the way the network resources are controlled.

4.2 Complexity Evaluation

Each of the three sub-problems in Sect. 4.1 is difficult to solve, since from the algorithmic point of view they all contain NP-complete (intractable) optimisation sub-problems. This is why any approach to solve such problems faces the following dilemma: either to give up *optimality* by using some heuristic solution, or, if optimality is required, either exactly or at least with a bounded error, then *scalability* is lost in the sense that for realistic (i.e. large) networks, algorithms become computationally infeasible.

Regarding the complexity of the first and second sub-problems, it can be easily proven that the known NP-complete problem DISJOINT CONNECTING PATHS (see [19]) can be obtained as a special case of both the ERP topology and layout design sub-problems. So both the ERP topology design and ERP layout design sub-problems are NP-hard problems. [4] Any formulation of the optimisation problem involved in even restricted versions of the third sub-problem (ERP dimensioning) contains the capacity non-linear integer-value multi-commodity flow problem and as such it is also NP-complete [19].

The problem becomes even more complex with respect to the global optimum ($\mathcal{ERP}^*$), if optimisation is done for the joint three sub-problems. Even if we have found ways to "solve" optimally the three sub-problems, then most probably the global optimum has not been reached. This means that by decomposing the general problem and trying to reach to a solution by solving each of the three sub-problems separately we have compromised the global optimality. In addition, using heuristics to solve the sub-problems causes loss of optimality. Note that this loss is unavoidable due to the nature of the problem. The relative merit of proposed solutions in terms of this loss is necessary to be evaluated and is for further work.

4.3 Algorithmic Approach

As discussed in Sect. 4.2 the problem is computationally infeasible. Therefore heuristic algorithms need to be employed for solving it. We propose a number of heuristics in order to cope with each of the three sub-problems. We propose to exercise TE in two timescales.

In the longer timescale (days - weeks) we propose to have off-line Contstraint-based TE algorithms, which specify the appropriate ERPs based on the forecasted traffic ($m \in M$). This process takes into account global network conditions and constraints ($r \in R$), and traffic loads, and it involves the global trade-offs of traffic and resource oriented objectives. Its output is the ERP graph for the constraints and demands imposed at that point in time t ($\widetilde{\mathcal{ERP}}^*_{(t)}$). Note that since the method is based on heuristics therefore the resulted ERP graph is different from the optimal the optimal at that particular time t ($\mathcal{ERP}^*_{(t)}$). The distance:

$$\mathcal{D}_{(t)} = ||\mathcal{ERP}^*_{(t)} - \widetilde{\mathcal{ERP}}^*_{(t)}|| \tag{3}$$

is the metric which gives us the ability to test the quality of the proposed heuristics. The problem is that $\mathcal{ERP}^*_{(t)}$ is difficult to obtain for any other than trivial network configurations. Therefore in order to test the validity of our heuristics we rely only on measuring the resource utilisation and if the traffic-oriented objectives are met, information provided by network resource and performance monitoring. In this timescale the Constraint-based TE Algorithms are triggered either periodically, or when the network performance or the demand changes significantly. The thresholds for the latter case are very important factors for the performance of the system.

Recently, Faragó et al. [14] have studied a similar problem to the first sub-problem of section 4.1. They proposed a solution, which does not suffer from the *optimality* vs. *scalability* problem. The methodology they used to accomplish this is derived from the theory of *Random Graphs.* Their approach maximises the *connectivity*[5] given a processing capacity bound; and minimises the *diameter* D of the resulting ERP graph, that is, any two nodes can reach each other by following at most D logical links. The algorithm converges in polynomial time and the resulting solution is asymptotically optimal. At that stage we have the ERP topology.

[4] A formal detailed proof of a similar problem can be found in Appendix A of [20].

[5] Connectivity is defined as the minimum number of nodes whose deletion can disconnect the graph.

The other two sub-problems are treated sequentially and not simultaneously because of the complexity of the joint problem. For the ERP layout design sub-problem, i.e. the physical routing of the ERPs, we reduce the space of feasible solutions in the following manner: Given the topology of the $\mathcal{ERP}$ graph, for each ERP we prune all the resources from $\mathcal{G}$ that do not satisfy the requirements of the predicted traffic load (captured in matrix M) and possibly other network-wide restrictions (captured in matrix R). We then run a shortest-path algorithm on the residual graph. This procedure gives a rough solution and possible optimisations on this are for further study.

Having the logical ERP topology and its mapping onto the physical topology we need to dimension the ERPs. In order to optimise the capacity assignment we propose to use some kind of a hill-climbing procedure, with two steps, the first step tries to satisfy the current traffic demands/objectives and the next step, if there is any bandwidth left, will try to assign to ERPs in order to capture future demands. Note that the following are computed for each class i of traffic aggregate. Initially we assign no capacity resources to every ERP and we will try incrementally to add units of capacity to these ERPs in order to meet the current traffic demands. We define the gain g_n for an the n ERP, as a function of the constraints associated with that ERP (d), the traffic constraints and demands that use this ERP and the network wide constraints r associated with class i of the traffic traversing that ERP:

$$g_n = f(d, m, r) \tag{4}$$

Then we calculate the gain if we assign a unit of capacity to the ERP. We also calculate the loss l, which is defined as the accumulation the gain that might have been achieved if this unit of capacity resources was assigned to other ERPs that share parts of the physical path with the one we are interested in. Let $\mathcal{E} \subset \mathcal{ERP}$ be the set of ERPs that share parts of the physical path with ERP n. Then the loss is defined as:

$$l_n = \sum_{x \in \mathcal{E}} g_x \tag{5}$$

We assign capacity resources to the ERP x, which gives the best gain-to-loss ratio, i.e.

$$\max_{x \in \mathcal{ERP}} \frac{g_x}{l_x} \tag{6}$$

This procedure stops when all $m \in M$ have met, or we have reached the physical capacity constraints $c \in C$. At that point we have satisfied the traffic constraints. Going a step further we might want to improve our solution by adding capacity to ERPs, in order to satisfy the network wide constraints $r \in R$. For example if r_i denotes the throughput requirements per traffic class i, we might continue adding capacity to these ERPs in order to maximise the total throughput for each class i. Note that fine-tuning the details of this algorithm is ongoing work.

In the shorter timescale (minutes - hours) we consider dynamic route and resource CBR solutions. These solutions are only based on the observed state of the operational network. Dynamic Constraint-Based TE solutions need to be employed in order to adapt to current network state within the bounds determined by the longer term solutions described above. Real-time measurements, performance monitoring and accounting (see Sect. 5.2) are very important for such solutions.

Given that we have computed the $\widetilde{\mathcal{ERP}}^*_{(t)}$ from the previous process, this solution needs fine tuning since as time passes then it is not the optimal (in terms of the heuristics). The idea is to keep track of the network and the matrix M, and try to keep close to $\widetilde{\mathcal{ERP}}^*$ whenever changes occur[6]. For example, merging of ERPs might be desirable. Also resizing ERPs might be desirable in order to capture the changes on the ERP utilisation. In some cases splitting of ERPs might be useful, for example if, after splitting, we can merge[7] part of the path with other paths. Also in that category of algorithms we can consider solutions which balance the traffic load among multiple

[6] Note that the dynamic control procedures will not replace the more "heavyweight" global reconfiguration, which will be activated either periodically or when the distance from $\widetilde{\mathcal{ERP}}^*$ is very big.

[7] Merging ERPs must follow the constraints imposed in [16].

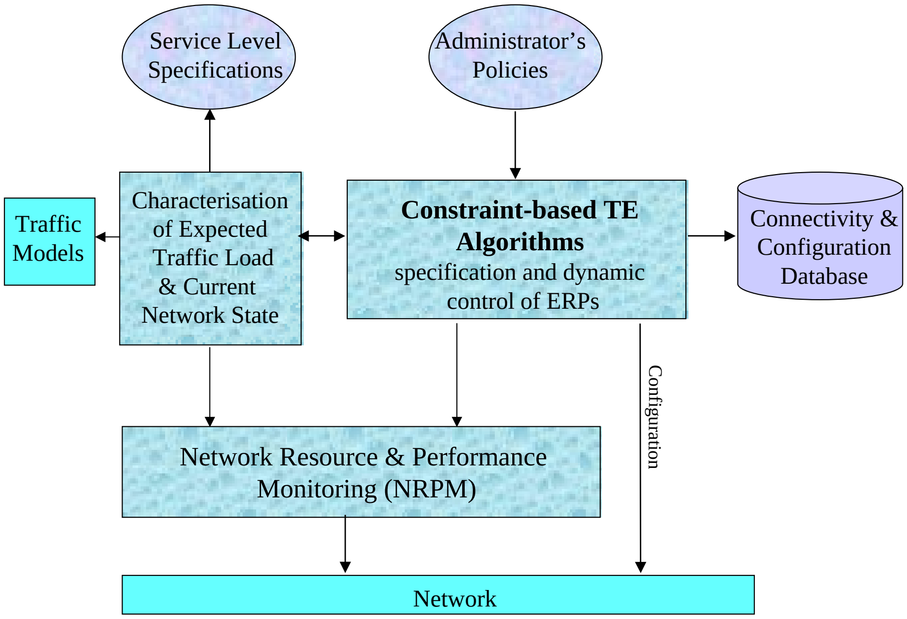
Service Level Specifications
Administrator's Policies
Traffic Models
Characterisation of Expected Traffic Load & Current Network State
Constraint-based TE Algorithms
specification and dynamic control of ERPs
Connectivity & Configuration Database
Configuration
Network Resource & Performance Monitoring (NRPM)
Network

Fig. 1. The functional architecture of the proposed IP Traffic Engineering Management and Control System.

EPRs proportionally to their load (see [31] for an example). Other examples of such a real-time optimisation include IGP/BGP metric-tuning (see [17]).

We argue that the network performance optimisation is a continuous process. Therefore optimisation iterations consist of both the non-real-time capacity and routing management algorithms as well as the real-time dynamic changes. The two processes are mutually complementary activities. A well-planned and dimensioned network makes real-time optimisation easier, while a systematic approach to real-time network performance optimisation allows decisions to focus on long-term issues rather than immediate considerations. This is why we propose algorithms for both *proactive* and *reactive* scenarios.

5 Architecture

In this section we present the architecture that we propose for enabling Traffic Engineering capabilities on IP-based networks by using the ERP concept.

5.1 Functional Architecture

In Fig. 1, we present graphically the functional architecture of the IP Traffic Engineering Management and Control System that we are proposing. The directionality of the arrows represents which part is responsible for initiating information requests, and not the directionality of the information flow.

The main logic of the architecture lies in the constraint-based TE algorithms functional block. It contains the families of algorithms which are responsible for both determining which are the necessary ERPs (global ERP graph (re)-configuration), and dynamically controlling the ones that are already active. The algorithms presented in Sect. 4.3 reside in this functional block[8]. These algorithms need input about the expected traffic, the current network load, the connectivity and the current configuration of the physical network as well as the established ERPs. They also need to define which are the resource and performance parameters that need to be monitored. Finally,

[8] Algorithms which are responsible for fast restoration of ERPs might also reside in this functional block.

according to their output, configuration actions need to be taken. The two timescale algorithms described in Sect. 4.3, reside in this functional module. The longer timescale algorithms operate with an AS-wide (Autonomous System) view as a management plane function. The dynamic management algorithms operate on a more localised view of the network and reside in the control plane.

The functionality of the algorithms is driven by the administrator's policies and the characterisation of the expected traffic and the current network state. There are two general types of policies, the ones that have to do with the *admission control*, and the ones that have to do with the *network dimensioning* and *dynamic management of resources.* Here we consider only the latter, for example what is the balancing parameter between optimisation of traffic-oriented and resource-oriented objectives or the triggering mechanism (synchronous or asynchronous) and its parameters. The policy issues are an active research area, but they are outside the scope of this paper. Changes in the expected traffic or of the current network state might also trigger the constraint-based TE algorithms.

The Service Level Specifications (SLSs) module, is a component which is responsible for the subscription and admission control of user requests. This module is complex and needs to be further decomposed to other components, for example long-term SLS subscription, Dynamic Admission Control. For the exact functionality of this module we need to define an SLS template and further to provide specific SLSs types according to this template. At this stage of our work we only use this module for the SLS repository it contains in order to obtain the traffic demands, the other specifics of this module are out of the scope of this paper.

For the characterisation of the expected traffic, in addition to the network state information provided by the monitoring system (current network state as well as historical trends data), we need to obtain information from the customer subscriptions (SLSs), and traffic models. This is a very critical part of the architecture, and generally of every IP TE system, so we further elaborate on it in Sect. 5.2. Monitoring also plays a very important role in our system, since it provides the appropriate input to several other components of our architecture. We consider monitoring as "passive" module in architecture in the sense that other components (e.g. constraint based algorithms, policies) request the monitoring of several network parameters, either network-wide or of a particular node. We describe our monitoring system in more detail in Sect. 5.3.

5.2 Traffic Characterisation

Traffic Engineering encompasses the application of technology and scientific principles to the *measurement*, *modeling* and *characterisation*, in addition to *control* of traffic [7], and the application of such knowledge and techniques to achieve performance objectives. It is, therefore, important to have accurate estimate of the offered load between the various points in a network domain. Expected traffic estimates can be derived from:

- *customer subscriptions* (e.g. SLSs),
- *traffic projections* (historical data),
- *traffic models*,
- *actual measurements* at different levels of abstraction (from *packet* level to *network-wide* level, characteristics), and
- *economical models*, which describe the users' behaviour.

Modeling involves constructing an abstract or physical representation, which depicts relevant *traffic* and *network* attributes and characteristics. Accurate source models for traffic aggregates are very useful for our analysis. It is inaccurate to apply the classic telecommunications traffic modeling theory to IP networks, because it has been proved [23, 28] that Poisson modeling is inadequate for IP traffic, since IP traffic exhibits *heavily tailed* distributions. This type of traffic is much better modeled using *self-similar* processes.

An ideal approach for characterising the expected traffic would have to consider the following: *(a)* heavily tailed models, *(b)* the notion of *equivalent capacity/bandwidth* [1, 22] (the Gaussian approximation) predicts capacity requirements of traffic aggregates, *(c)* customer subscription information and *(d)* adapting the *forecasts* of the expected load by using real-time measurements, in order to devise dynamic traffic engineering. The application of all four is not a trivial task, and it may prove impossible to realise under reasonable constraints. But even if this is impossible, recent works [17] prove that the application of measurement-based only techniques for traffic prediction might be satisfactory.

5.3 Monitoring

Our monitoring system includes a two level approach, low level node monitoring/metering and higher-level network-wide monitoring.

At the network-wide level network monitoring builds network-wide view of current traffic and quality conditions. Accepts monitoring and measurement requests the Constraint-based TE Algorithms module (long term), policy manager and the traffic characterisation modules. These modules may also provide specific thresholds in order to receive notification when they are crossed. The network monitoring system determines what needs to be measured where and directs the monitoring/metering modules accordingly, identifies the type of measurements needed and which nodes need to participate in the measurements. This level also includes database of historical load and quality measurements, in order to provide input to the characterisation of expected traffic whenever necessary.

At the node monitoring level we measure load, losses, etc. at a local level. Probes are downloaded from the network-wide monitor. It is necessary at this level to include an *active* monitoring engine, which performs delay and jitter measurements, because otherwise we cannot measure them. This engine puts necessary test streams into the forwarding path in order to realise *active probing.* Caching at this level is necessary in order to avoid loading the network with too much test traffic. The classic *passive* monitoring is included in this module. Passive monitoring relies on counters (*MIB probing*) to perform monitoring. The counters are available in various parts of the network: meters, classifiers and forwarding engines. The passive node monitoring measures used/available bandwidth and aggregate traffic statistics (packet lost/dropped etc.).

5.4 Implementation Considerations

In this section we describe our approach to implementing the proposed IP Traffic Engineering Management and Control System, as described in Sect. 5.1, for testing and validation purposes.

We argue that it is very important for proof of concept issues to use a simulator to run evaluation experiments. The advantages of using a simulated network include: visualisation of network characteristics under different conditions, ability to test the system under extreme scenarios, get hints for possible solutions (algorithm fine-tuning), reveal pathologies such as single points of failure and potential bottlenecks, which may require additional capacity. Simulators can also be used to conduct scenario-based and perturbation-based analysis, as well as sensitivity studies.

We have deliberately not provided information about the network shown in Fig. 1 because the proposed architecture is network independent. When coming to realisation issues this independence is not a necessity, since if a simulator is used, the whole system can be implemented in a tightly-coupled manner inside the simulator. We argue that it is equally important for an IP TE system to be validated and tested over real networks, since Traffic Engineering is an applied discipline in its nature.

The key idea behind the realisation is to implement our traffic engineering system in such a way that it could be used with both a simulated and a real network. By decoupling the simulator from the TE system we increase the degrees of freedom in terms of experimentation and validation scenarios at the price of additional implementation effort. The decoupling of the network and the TE system is illustrated in Fig. 2. We are using the U.C. Berkeley NS [13] (Network Simulator) and add management capabilities to it. Note that to the system we proposed we only have to add an interface to the simulator[9]. By removing this simulator-specific interface we can use it on a real network. The real network in our case is a testbed that we are maintaining at our research centre, and which consists of PC-based routers.

[9] We also use a proprietary management protocol between the simulator and the interface.

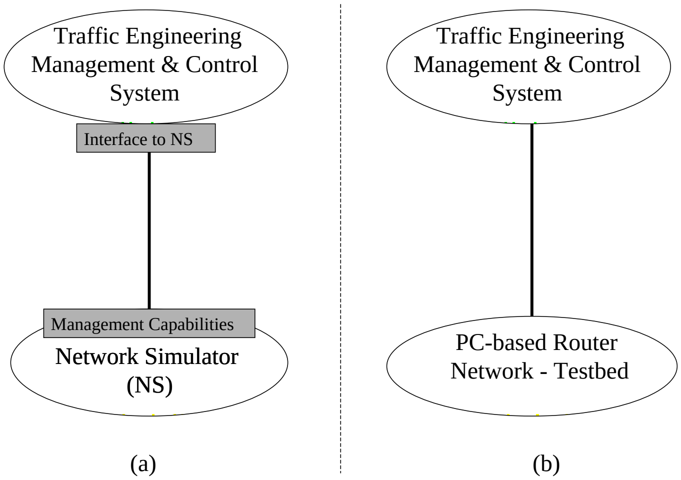
Traffic Engineering Management & Control System
Interface to NS
Management Capabilities
Network Simulator (NS)
(a)
Traffic Engineering Management & Control System
PC-based Router Network - Testbed
(b)

Fig. 2. Experimentation and validation scenarios: (a) using a simulated network, (b) using a real network.

Implementation issues include two parts: to provide the simulator with the management interface, and to build of the various components of the TE management system. We are building the management system components by using a Distributed Processing Environment (DPE).[10] In addition, we are working on how to make the constraint-based TE algorithms distributed. Note that the TE system needs very careful design and implementation, particularly the parts that need to interact with external entities, in order to ensure interoperability when used in different scenarios.

6 Conclusions

In this paper we have presented an approach for IP Traffic Engineering using the concept of Explicitly Routed Paths. In this approach we have assumed that the Differentiated Services model is provided over connection-oriented layer 2 technologies such as MPLS.

We first provided a precise definition of the relevant resource optimisation problem, including its mathematical formulation. In order to cope with the problem complexity, we have decomposed the problem into three sub- problems: topology design, layout design and dimensioning. For long term solutions we have proposed heuristics for each of these sub-problems, including an initial design of the relevant algorithms. We argued that an effective IP TE management system must be both reactive and proactive, therefore we have also catered for dynamic control actions to cope with small timescale changes.

Furthermore, we proposed a high-granularity functional architecture of a management system which orchestrates the TE activities, in cooperation with control functions such as constraint-based routing. This management system, though decoupled from the network and logically centralised, we argue it should be physically distributed in order to cope with scalability, react timely to network events and avoid traffic concentration around a single node.

In order to test our algorithms and validate the overall approach in providing effective and efficient IP TE solutions, we propose an approach in which we will use a management system both over a simulated network, in order to capture extreme conditions, and over a real testbed, in order to validate the approach under real world scenarios.

Acknowledgements

This work was undertaken in the context of the PRONET project (PROduction of Broadcast Content in an Object-Oriented IP-Based NETwork) which is funded by the UK LINK Broadcasting programme.

[10] Note that while of writing this paper a few other publications [4, 17] also used DPEs to implement TE capabilities.

References

[1] H. Ahmandi and R. Guérin. Equivalent Capacity and its Application to Bandwidth Allocation in High-Speed Networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, September 1991.

[2] L. Andersson et al. *LDP Specification.* Internet draft, <draft-ietf-mpls-ldp-06.txt>, work in progress, October 1999.

[3] G. Armitage. MPLS: The Magic Behind the Myths. *IEEE Communications Magazine*, pages 124–132, January 2000.

[4] P. Aukia, M. Kodialam, P.V.N. Koppol, T.V. Lakshman, H. Sarin, and B. Suter. RATES: A Server for MPLS Traffic Engineering. *IEEE Network Magazine*, 14(2):34–41, March/April 2000.

[5] D. Awduche. MPLS and Traffic Engineering in IP Networks. *IEEE Communications Magazine*, pages 42–47, December 1999.

[6] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. *A Framework for Internet Traffic Engineering.* Internet draft, <draft-ietf-tewg-framework-01.txt>, work in progress, May 2000.

[7] D Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. MacManus. RFC2702 – *Requirements for Traffic Engineering Over MPLS*, September 1999.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC2475 – *An Architecture for Differentiated Services*, December 1998.

[9] R. Callon et al. *A Framework for Multiprotocol Label Switching.* Internet draft, <draft-ietf-mpls-framework-05.txt>, work in progress, September 1999.

[10] R. Comerford. State of the Internet: Roundtable 4.0. *IEEE Spectrum*, October 1998.

[11] E. Crawley, R. Nair, B. Jajagopalan, and H. Sandick. RFC2386 – *A Framework for QoS-based Routing in the Internet*, August 1998.

[12] S. Even. *Graph Algorithms.* Computer Science Press, 1979.

[13] K. Fall and K. Varadhan (eds.). *ns* Notes and Documentation, February 2000. available at: http://www-mash.cs.berkeley.edu/ns.

[14] A. Faragó, I. Chlamtac, and S. Basagni. Virtual Path Network Topology Optimisation Using Random Graphs. In *Proc.INFOCOM '99.* IEEE, 1999.

[15] A. Faragó et al. A New Degree of Freedom in ATM Network Dimensioning: Optimising the System of Virtual Paths. *IEEE JSAC*, 13(7):1199–1206, Sept. 1995.

[16] F. Le Faucheur et al. *MPLS Support for Differentiated Services.* Internet draft, <draft-ietf-mpls-diff-ext-04.txt>, work in progress, March 2000.

[17] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford. NetScope: Traffic Engineering for IP Networks. *IEEE Network*, 14(2), March/April 2000.

[18] V.J. Friesen, J.J. Harms, and J.W. Wong. Resource Management with Virtual Paths in ATM Networks. *IEEE Network Magazine*, 10(5):10–19, Sept./Oct. 1996.

[19] M.R. Garrey and D.S. Johnson. *Computers and Intractability - A guide to the Theory of NP-Completeness.* W.H. Freeman and Co., San Francisco,1979.

[20] O. Gerstel, I. Cidon, and S. Zaks. Optimal Virtual Path Layout in ATM Networks with Shared Routing Table Swithces. *Chicago Journal of Theoretical Computer Science*, October 1996. available at: www-mitpress.mit.edu/jrnls-catalog/chicago.html.

[21] R. Guérin and V. Peris. Quality-of-Service in Packet Networks Basic Mechanisms and Directions. *Computer Networks*, 31(3):169–189, Elsevier Science B.V., February 1999.

[22] F.P. Kelly. Notes on Effective Bandwidths. In F.P. Kelly, S. Zachary, and I.B. Ziedins, editors, *Stochastic Networks: Theory and Applications*, pages 141–168. Oxford University Press, 1996. Royal Statistical Society Lecture Notes Series.

[23] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[24] T. Li and Y. Rekhter. RFC2430 – *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*, October 1998.

[25] Q. Ma. *QoS Routing in the Integrated Services Networks.* PhD thesis, CMU, January 1998. CMU-CS-98-138.

[26] K. Nichols, S. Blake, F. Baker, and D. Black. RFC2474 – *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, December 1998.

[27] K. Nichols, V. Jacobson, and L. Zhang. *A Two-bit Differentiated Services Architecture for the Internet.* Internet draft, <draft-nichols-diff-svc-arch-02.txt>, work in progress, April 1999. ftp://ftp.ee.lbl.gov/papers/dsarch.pdf.

[28] V. Paxson and S. Floyd. Wide-Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[29] E.C. Rosen, A. Viswanathan, and R. Callon. *Multi-Protocol Label Switching Architecture.* Internet draft, <draft-ietf-mpls-arch-06.txt>, work in progress, August 1999.

[30] G. Swallow. MPLS Advantages for Traffic Engineering. *IEEE Communications Magazine*, pages 54–57, December 1999.

[31] I. Widjaja and A. Elwalid. *MATE: MPLS Adaptive Traffic Engineering.* Internet draft, <draft-widjaja-mpls-mate-01.txt>, work in progress, October 1999.

[32] X. Xiao and L.M. Ni. Internet QoS: The Big Picture. *IEEE Network*, 13(2):8–18, March/April 1999.