SACJ 29(1) July 2017
Research Article

Optimising Capacity Assignment in Multiservice MPLS Networks

Abdoul Rassaki[a], Andre Nel[b]

[a] Applied Information Systems, University of Johannesburg
[b] School of Mechanical and Industrial Engineering, University of Johannesburg

ABSTRACT
The general Multiprotocol Label Switch (MPLS) topology optimisation problem is complex and concerns the optimum selection of links, the assignment of capacities to these links and the routing requirements on these links. Ideally, all these are jointly optimised, leading to a minimum cost network which continually meets given objectives on network delay and throughput. In practice, these problems are often dealt with separately and a solution iterated. In this paper, we propose an algorithm that computes the shortest routes, assigns optimal flows to these routes and simultaneously determines optimal link capacities. We take into account the dynamic adaptation of optimal link capacities by considering the same Quality of Service (QoS) measure used in the flow assignment problem in combination with a blocking model for describing call admission controls (CAC) in multiservice broadband telecommunication networks. The main goal is to achieve statistical multiplexing advantages with multiple traffic and QoS classes of connections that share a common trunk present. We offer a mathematical programming model of the problem and proficient solutions which are founded on a Lagrangean relaxation of the problem. Experimental findings on 2-class and 6-class models are reported.

Keywords: MPLS, network capacity assignment, optimal routing, link flow, Lagrangean relaxation, QoS.

Categories: • Networks

Email:
Abdoul Rassaki arassaki@uj.ac.za (CORRESPONDING),
Andre Nel andren@uj.ac.za

Article history:
Received: 9 June 2016
Accepted: 23 February 2017
Available online: 9 July 2017

1 INTRODUCTION

The operation, performance and system cost of today's telecommunication networks are challenged by the routing, flow and capacity assignment strategy. MPLS networks are not immune from these challenges. Of primary importance in the design of the MPLS network is to ascertain the optimal link capacities and routes to be used between each origin-destination (O-D) pair, thereby mitigating system cost. The preferred outcome in network design is to minimize overall system costs which are composed of a) connection costs depending on capacities and end-to-end delay, and b) transfer costs which are incurred due to the limited line and node capacities. A profitable design will make simultaneous decisions on both routing and link capacities as they are closely related. Determining

Rassaki, A., and Nel, A. (2017). Optimising Capacity Assignment in Multiservice MPLS Networks. *South African Computer Journal 29*(1), 69–89. https://doi.org/10.18489/sacj.v29i1.393

SACJ is a publication of the South African Institute of Computer Scientists and Information Technologists. ISSN 1015-7999 (print) ISSN 2313-7835 (online).

the routing between origin-destination pairs and assigning capacities to the links used by those routes is known as the capacity and flow assignment (CFA) problem (Gavish & Altinkemer, 2000).

In the MPLS network design process, tradeoffs have to be made between the response time for clients and the costs of the network. If high capacities are assigned to the network links, then we see elevated connection costs but a low response time. Conversely, if low capacity links are installed, the reverse will be true. This argument shows that the tradeoffs between response time and connection costs are an integral part of the proper design and operation of a communication network. The main focus of this research is on issues concerning routing and capacity assignment particularly in MPLS networks and more generally in all routed telecommunication networks. Currently, network designers use heuristic solution techniques during the design process. However, it is not possible using only such techniques to analyze the quality of the resulting design in terms of cost and response time.

Innovative techniques for MPLS network optimisation, subject to constraints on routing effected by QoS and other parameters, were recently established and illustrated in Rassaki and Nel (2013). The outcome of the investigation was to substantiate how to capitalize on the available bandwidth and diminish the results of network overcrowding with MPLS Traffic Engineering (TE).

An MPLS network is comprised of a pair of nodes known as label switching routers (LSRs). These LSRs have the capacity to switch and route packets using information stored in the label attached to each packet. To begin the process, a label switched path (LSP) is established for the packets being routed and distributed, and QoS considerations are determined for the LSP. The parameters for QoS consist of the queuing and discarding protocols for every LSP on the path as well as the resources that are required to sustain the path. The establishment of the LSP and of the QoS parameters leads to the creation of the forwarding equivalence class (FEC). The FEC denotes a grouping of packets sharing transportation requirements. All packets in an FEC receive the same treatment on the way to the destination. These packets follow the same path and receive the same QoS treatment at each hop. The LSRs simply advance every packet on the strength of its own label value as it is unnecessary to scrutinize or analyze the packet's IP header. Each LSR builds a table, known as label information base (LIB), to specify how a packet must be treated and forwarded. The outcome of this process is that the forwarding mechanism used by an LSR is more straightforward and more timely than that of an IP router.

Packets arrive at an MPLS switching domain via an ingress LSR found at the border of the MPLS network. The ingress LSR reviews the packet to establish the required QoS, then allocates the packet to an FEC and LSP. Next, it attaches the relevant label to the packet which is then directed to the next LSR down the LSP. Within the MPLS network, every LSR on the length of the LSP collects the previously labeled packet and then transmits it to the subsequent LSR along the LSP. As the packet is received by the egress LSR at the periphery of the network closest to its destination, the edge LSR removes the packet's label, reviews the IP packet header and then directs the packet to its ultimate destination.

Multiprotocol label switching (Rosen, Viswanathan, & Callan, 2001) has many valuable attributes for Internet traffic engineering (TE), and as such will be extensively utilised for TE (Awduche, Malcolm, Agogbua, & McManus, 1999). In a straightforward manner, MPLS can establish an explicit-route label switched path (ER-LSP) as necessary, through manual administrative action or traffic requirement by applying CR-LDP (Jamoussi, 2002) or RSVP-TE (Awduche et al., 2001) signaling protocols. Further, traffic trunks that are comprised of traffic flows with comparable traits or traffic needs can simply be mapped on that ER-LSP.

One of the primary motivations behind our research is the emergence of TE support in IP networks, for example in networks found to be using the MPLS protocol. The requirement of choosing the topology and scope of MPLS "explicit routes" to convey QoS multiservice is also discussed here. MPLS facilitates multiple paths from origin to destination since this protocol imparts the promise of advantageous network wide load balancing, which is a benefit that corresponds nicely with our current work. Further, it has been noted that it is equally beneficial to merge differentiated services (DiffServ) and MPLS protocols by applying QoS multiservice traffic on explicit routes (Wu, Cheval, & Vaananen, 1998).

From a "fast forwarding" technology, MPLS has evolved into a set of protocols that offers sophisticated traffic engineering options, Virtual Private Networks (VPNs) and multi-protocol support through logical distinction between IP forwarding and routing. MPLS has been widely deployed on the ISP backbone to replace Asynchronous Transfer Mode (ATM) and traditional IP routing. Further, it has been introduced into metro and access-networks, and even some private enterprise networks (Stallings & Case, 2013).

This paper develops mathematical programming techniques that directly consider both costs as well as service quality to design and operate telecommunication networks. A major incentive is the vigorous revision of link capacities by considering the same criterion used in the flow assignment problem in combination with a blocking model for describing call admission controls in multiservice broadband networks. Traffic of a number of different types requiring different bandwidth allocations is offered to each source-destination pair. The network manager must implement a call admission control scheme to minimize the packet delay and maximize the throughput earned from the network while maintaining agreed QoS constraints.

Section 2 presents a study of literature of closely related work, and the critical review thereof. In Section 3, we first describe our modeling framework then a nonlinear integer programming formulation of the network design problem is given. The following Sections 4 and 5 present our optimal capacity assignment formulation and algorithm. Section 6 presents the analytical evaluation of the network model. We introduce our multiservice blocking and network models in Sections 7 and 8. The basis of comparing two different call admission control parameters and performance are discussed in Section 9. Lastly, the conclusions are presented in Section 10.

2 REVIEW OF RELATED WORK

The topological design of distributed computer networks in capacity assignment is a subject with comparatively few published results, given that it has been studied since the late 1970s (Gerla, 1975; Maruyama & Tang, 1976; Kleinrock, 1976). As such, this complex field continues to be challenging to study; however, there are some recent findings on relative bandwidth allocation techniques (Martens & Skutella, 2006; Pompili, Scoglio, & Shoniregun, 2007; Truffot, Duhamel, & Mahey, 2010; Rassaki

& Nel, 2015).

Most of the available literature examining this problem handles capacity assignment and flow assignment separately. All these research approaches to capacity assignment, also utilize different performance criteria. In the capacity assignment problem (Gerla, 1975; Maruyama & Tang, 1976; Amiri, 1992) the routing protocol is understood to be given and the optimal capacity for every link is chosen from within a discrete set of line capacities. The flow assignment problem (Bertsekas & Gallager, 1992; Kershenbaum, 1993; Kleinrock, 1976) begins with a given assignment of link capacities, critical paths between O-D pairs are established to minimize either the average message delay or the maximum message delay in the network.

An additional approach is to improve an existing network by redistributing the link capacities while maintaining the total sum of all capacities of the network.

It is important to note the distinction between the uncapacitated MPLS design problem where the capacities (bandwidth) of the physical links are the decision variables, and the capacitated MPLS design problem where the capacities of the physical links appear as constraints. The uncapacitated problem has been studied by, among others, Rohne, Jensen, Svinnset, and Venturin (1998), who first design a physical network under the assumption that all traffic will be routed on a node-by-node basis, and then configure the LSP highways by cross connecting flows in the physical network according to a cost model. Bauschert (1997) also considers the uncapacitated problem and uses a set of iteration loops to simultaneously design a virtual path connection network (VPCN) and virtual connection (VC) routing policy. This scheme relies on an initial pre-selection of paths and includes linear programming models.

A distinction is also made between real valued MPLS configurations and integer valued configurations. In the former, bandwidth is regarded as a resource which can be shared in any fashion while in the latter only certain sharing schemes are possible.

The most conventional manner of resolving the capacitated MPLS design problem is to draw on constrained non-linear programming (NLP) techniques. Unfortunately, with a network of any size, issues such as the quantity of decision variables and computation times grow to such a degree that these techniques become unrealistic to implement. Herzberg and Byes (1993) linearize the problem with a goal of reducing the computational burden while also arriving at integer valued solutions using standard LP solvers, but do not touch upon the problem of a large state space.

Balakrishnan and Graves (1989) study the special case of piecewise linear costs in directed networks, where each link is assigned a capacity and a path is identified for each O-D pair. In this study, the problem of the routing and capacity assignment is devised as a mixed integer program and a composite algorithm is developed to generate both lower bounds and feasible solutions. The model however does not take into account the delay issue that arises when link capacity utilisation reaches certain levels. LeBlanc and Simmons (1989) articulate the routing and capacity assignment problem with continuous link capacity variables. In addition, they propose an innovative convex delay function, demonstrating that, for their assumed length distribution, this novel function forecasts delay with more accuracy than the conventional delay function when flow-capacity ratios are less than 0.80. Computation results have been conveyed for networks with as many as 100 nodes.

Gopal, Kim, and Weinrib (1991) and Arvidsson (1995) devise heuristic optimisation algorithms that maximize a function in a sequence of steps which converge to a local optimum. These algorithms belong to the category of purported greedy algorithms which sees the steps taken at each optimisation point are those immediately giving the utmost reward with no consideration for long term repercussions. Whereas Gopal et al. rely on predefined paths, Arvidsson includes path searching in his algorithm. Pióro and Medhi (2004) surveyed and evaluated a large number of problems related to optimisation techniques. The reader can refer to Pióro and Medhi (2004) for a comprehensive list of references relating to network optimisation from a mathematical viewpoint.

Maruyama, Fratta, and Tang (1977) incorporate heuristic methods for capacity assignment (developed in Maruyama & Tang, 1976) into a more general procedure that iterates between a composite capacity assignment algorithm and flow assignment phase until a local optimum is reached. They also describe a priority assignment scheme which, with high likelihood, yields a less costly capacity assignment satisfying the delay requirements. A similar iterative procedure which alternates between capacity and flow assignment is used by Fratta, Gerla, and Kleinrock (1973) where different heuristic methods based on the flow deviation algorithm (Maruyama et al., 1977) for static route assignment is presented.

Gershet and Weihmayer (1990) examine the issue of assigning capacities to network switches and potential links in order to accommodate traffic demand between nodes and satisfy a performance requirement that specifies an upper bound on the link utilisation. The goal is to minimize switch and link capacity costs which are assumed to be continuous. A solution procedure which alternates between solving an uncapacitated design/routing subproblem and a capacity assignment subproblem is presented. The procedure is applied to a real network with 20 nodes and two levels of link capacities.

Gavish and Altinkemer (2000) expand upon the research of Whitney (1972) by taking into account every potential route for each communicating node pair. They devise the problem, using Lagrangean relaxation inserted in a subgradient optimisation methodology in order to realize lower bounds and a practicable resolution to the problem. They incorporate cut constraints which are redundant in the initial problem to better the lower bounds. These cut constraints are understood to have been classified in advance of the start of the solution procedure. Clearly the value of the solution relies a great deal on the number and selection of the cuts.

As can be seen from the short summary of the literature, the capacity and flow assignment problem is typically separated into two subproblems: capacity assignment, and route determination. These two issues are correlated, as the capacity designated to a link and the delay sustained by a particular flow utilising that link are connected. As such, examining these problems independently of one another may result in unsatisfactory outcomes. Having the routing method remain unchanged and attempting to make the best use of the capacity assignment, or vice versa, does not get to the core of the characteristics integral to the problem and could therefore produce sub-optimal results.

A further shortcoming of the earlier research perspectives is the absence of theoretical or empirical methodology to assess the value of the proposed solutions.

3 MODELING AND ANALYSIS

In this section, we present the appropriate notation and definitions and follow this with the collection of assumptions that define our model of the network. We identify the class of analysis and synthesis problems that confront us in network studies.

Take into account a physical network which is comprised of a group of N nodes represented by $\mathscr{N}$ and a collection of L physical links indicated by $\mathscr{L}$. The nodes convey the routers in an MPLS network. The traffic requirements are stipulated by an $N \times N$ matrix $\mathscr{R}_e = r_{ij}$, called the requirement matrix, whose entries are non-negative and with $i \neq j$. Let C_{ij} denote the capacity in bandwidth units of the physical link from an origin node (ingress) i to a destination node (egress) j. The set of routes connecting O-D pair (o, d) is denoted by $R_{o,d}$. Every route is comprised of a non-cycling sequence of physical links.

In the network design problem, the messages are offered to O-D pair (i, j) according to a Poisson process with mean rate λ_{ij}. The average message length from node i to node j is exponentially distributed with mean $1/\mu_{ij}$. Let $\rho_{ij} = \lambda_{ij}/\mu_{ij}$ denote the intensity of the offered traffic stream. Let f_r denote the flow on route r. The total flow F_{ij} on link (i, j) is denoted by

$$F_{ij} = \sum_{r \in \mathscr{A}_{ij}} f_r \tag{1}$$

where $\mathscr{A}_{ij}$ is the set of routes that use link (i, j).

Our interest is in the numerical resolution of the ensuing network flow problem, subject to a constraint that the total link capacity not exceed C_{ij}:

Minimize: The average end to end network delay.

$$T = \sum_{(i,j)} \frac{F_{ij}}{\gamma} T_{ij} \tag{2}$$

where $\gamma = \sum_{(i,j)} \lambda_{ij}$ is the total message arrival rate from external sources (bits/sec) and T_{ij} is the average delay experienced by a message on link (i, j) (sec) *subject to:*

$$0 \leq F_{ij} \leq C_{ij} \qquad\qquad \forall i, j \in \mathscr{N} \tag{3}$$

We further believe that the cost of building the channel with capacity C_{ij} is given by $d_{ij}C_{ij}$, an arbitrary function of the capacity and of the channel. If we say that D signifies the cost of the whole network, which we understand to be comprised of only the cost[1] of channel construction, we then have

[1] The cost of the nodes may be incorporated in the channel cost directly.

$$D = \sum_{(i,j)} d_{ij} C_{ij} \tag{4}$$

where d_{ij} is the positive cost per unit capacity on link (i, j).

We have earlier defined message delay as the total time it takes for a message to travel across a network. What we are more interested in, though, is the *average* message delay (2) and we consider this to be our key performance indicator.

In any practical network design procedure, a large number of design variables suggest themselves. Among these we include: the selection of channel capacities; the form of routing procedure; the form of flow control procedure; the topological design of the network; the storage capacity at each node; the choice of hardware and software programs to be used for the switching computer; the partitioning of messages into various-size packets; and so on. Since we are interested mainly in the queueing problems in this paper, we discuss neither the hardware nor many aspects of the software design of the switching computer itself any further.

4 PROBLEM FORMULATION

This section begins by considering the problem of assigning optimal capacities to the links in the network given the link topology (location of the nodes and links) and link flows. We consider a network consisting of N nodes and L links. The links have original capacities $C_1, C_2, \ldots, C_L$ measured in bits/sec. Let $i - j$ denote the link connecting O-D pair (i, j). Link $i - j$ has capacity C_{ij} measured in bits/sec. There are S classes of messages.The traffic requirements between the node pairs are measured in bits per second. We assume a flow distribution – a flow on each link which satisfies the requirements.

The objective is to compute the optimal link capacities for a network where the topology and traffic flows are known and fixed which minimizes the average network delay subject to the linear overall cost of the system:

$$\sum_{(i,j)} d_{ij} C_{ij} \tag{5}$$

In essence, we want to establish both the scope of resource capacity required for the given demand volume, and the manner in which to reasonably and efficiently disseminate it through the network using a series of routing/flow constraints. This determination, which is typically found in medium to long-term network planning, is broadly known as uncapacitated design.

When the network capacity is established and the demand volume is understood, the question becomes how to assign flows on distinct paths in such a way that specified network objectives (e.g., minimum cost routing or maximum total revenue) are optimised. The system costs are composed of connection costs which depend on link capacities and delay costs incurred by users due to the limited capacities of the links and the resulting queueing at intermediate nodes.

We wish to focus on three very basic design parameters that we must consider: first is the selection of the channel capacities C_{ij}; second is the selection of the channel flows F_{ij}; and third there is the topology itself. All of these may be varied to improve network performance. The notion of “optimum design” is extremely difficult to achieve in any realistic network design; however we define, the performance criterion, which is the average message delay T, and attempt to minimize this quantity (thereby optimising performance). This approach will allow us to make some important qualitative statements about network design and performance. Of course, any optimisation problem must be subject to some form of cost constraint, and here we choose the fixed cost constraint given in Eq. (4). Therefore we have a performance measure T, a cost constraint D, and three variables design “parameters,” C_{ij}, F_{ij}, and the topology.

This study overcomes serious shortcoming of previous methods suggested in past research. In the routing selection process, these methods assume that a set of pre-specified candidate routes chosen from among all possible routes is given for every communicating O-D pair. Evidently, the value of the solutions offered by these methods relies greatly on the selection of the candidate route sets determined before the procedure is employed. The use of only a subset of all possible routes by these methods results in a practical limitation which is the potential of producing lower bounds higher than the values of the optimal solutions to the routing and capacity assignment problem. Our solution method removes this shortcoming by considering every promising route for each communicating node pair.

What is particularly attractive about the method known as Lagrangean relaxation is the way it offers both upper and lower bounds on the value of the objective function (Berezner & Krzesinski, 2001). In other words, it is understood that the optimal objective function value rests between the value of the best feasible solution found and a value that it can be no better than. In this paper, a Lagrangean problem is established by multiplying some of the constraints by Lagrange multipliers and adding them to the objective function. As a result, the Lagrangean problem is divisible to a routing subproblem and a link subproblem. Every category of subproblem is further divisible into subproblems for every link and for every communicating pair. The link subproblem consists of assigning a capacity to a link and the route subproblem deals with choosing a route for a communicating O-D pair.

To minimize the objective function, we proceed by using a Lagrange multiplier β and by forming the Lagrangean relaxation function as follows:

$$L = T + \beta\Big(\sum_{(i,j)} d_{ij}C_{ij} - D\Big) \tag{6}$$

where D is the total cost of the network and T is given by the $M/M/1$ delay function:

$$T = \frac{1}{\gamma}\sum_{(i,j)}\Big(\frac{F_{ij}}{C_{ij} - F_{ij}}\Big) \tag{7}$$

In Eq. (6), if we find the minimum value of L with respect to the capacity assignment, then we will have found the solution to the capacity assignment problem since the bracketed term is identically equal to zero. The parameter β is the undetermined multiplier to be evaluated.

If β is large enough, it is a penalty for violating the constraint on total capacity. If the sum of the C_{ij} exceeds D, the term in the brackets is positive and if multiplied by β increases the value of the objective to be minimised. Values of C_{ij} are sought which do not violate the constraint. If β is too large, however, it is possible to make this new objective function smaller by letting the sum of the C_{ij} become strictly less than D, thus minimising the new objective function but not the original one. As β increases from zero, the sum of the C_{ij} decreases as the first term is traded in the objective against the second. There is a unique value of β which makes the sum exactly equal to D. This is the value sought along with the corresponding values of the C_{ij}.

As is usual in Lagrangean optimisation problems, we set the partial derivatives $\partial L/\partial C_{ij}$ to zero:

$$\frac{\partial L}{\partial C_{ij}} = \beta d_{ij} - \frac{F_{ij}}{\gamma(C_{ij} - F_{ij})^2} = 0 \tag{8}$$

Solving for C_{ij} gives:

$$C_{ij} = F_{ij} + \frac{1}{\sqrt{\beta\gamma}}\sqrt{\frac{F_{ij}}{d_{ij}}} \tag{9}$$

The objective now is to find the value of β. Once we have evaluated the constant β, this will be our solution.

$$\sum_{(i,j)} d_{ij}C_{ij} = \sum_{(i,j)}\left(F_{ij}d_{ij} + \frac{1}{\sqrt{\beta\gamma}}\sqrt{F_{ij}d_{ij}}\right) \tag{10}$$

From this equation, solving for β gives,

$$\frac{1}{\sqrt{\beta\gamma}} = \frac{D - \sum\limits_{(i,j)} F_{ij}d_{ij}}{\sum\limits_{(i,j)} \sqrt{F_{ij}d_{ij}}} \tag{11}$$

Using this last form in the Eq. (9), the optimal solution to the capacity assignment problem is

$$C_{ij} = F_{ij} + \frac{D - \sum_{(i,j)} F_{ij}d_{ij}}{\sum_{(i,j)} \sqrt{F_{ij}d_{ij}}}\sqrt{\frac{F_{ij}}{d_{ij}}} \tag{12}$$

The solution to the Lagrangean problem for any specific values of the Langrange multipliers often breaches one or more of the relaxed constraints. Several Lagrangean-based algorithms include additional heuristics which allow these infeasible solutions to become feasible. As such, the algorithms can provide welcome solutions to the initial framework. The top feasible solutions of those offered by the procedure at any point denote the upper bound on the value of the true optimal solution.

The disparity that is seen between the upper and the lower bounds is known as the gap. Should the value of the gap become zero (or some minimum value determined using the integer properties of the model), we will have arrived at the optimal solution. If not, but the gap becomes satisfactorily small (e.g. less than 1%), the analyst might halt the procedure, content that the current best solution is within 1% of optimality.

In implementing Lagrangean relaxation, we are challenged to select which constraints will be relaxed. The objective is to have a relaxed problem which can be easily resolved and bring about ideal lower bounds. As the relaxed model may require solving hundreds or thousands of times while seeking the best multiplier values, the simplicity of the solution is vital to achieving the desired outcomes. Preferably, the solution to the relaxed problem will found by inspection or simply by sorting the objective function coefficients.

The algorithm assumes:

1. The nodes of the network and the input traffic flow for each pair of nodes are known.

2. A routing model determines the optimal flows F_{ij} of all links (i, j) given the link original capacities C_{ij}. We assume that the link flows minimize a cost function $\sum_{ij} D_{ij}(F_{ij})$ and F_{ij} can be determined by minimising the average packet delay,

$$T = \frac{1}{\gamma}\sum_{(i,j)}\left(\frac{F_{ij}}{C_{ij} - F_{ij}} + F_{ij}p'_i\right)$$

   based on the $M/M/1$ channel model, where γ is the total input traffic into the network, and C_{ij} and p'_i are the capacity and the processing and propagation delay, respectively, of link (i, j). The algorithms described in Rassaki and Nel (2013) can be used for this purpose.

4.1 The algorithm

This section describes the different steps of the capacity assignment algorithm and how it works.

The steps of the capacity assignment algorithm

Step 1: Select a network topology with initial capacities and requirements.

Step 2: Compute optimal link flows that minimize the average delay for the network using the FOA algorithm.

Step 3: Allocate the link capacities to minimize the delay with the link flows computed in step 2, given the constraints on the total cost of a system.

Step 4: Use these link capacities instead of the original capacities with the original requirements and go to step 2. The delay calculated in this step will be less than in step 2.

Step 5: Reallocate the optimal link capacities with the optimal link flows from step 4. The new delay will be smaller than the delay in step 3.

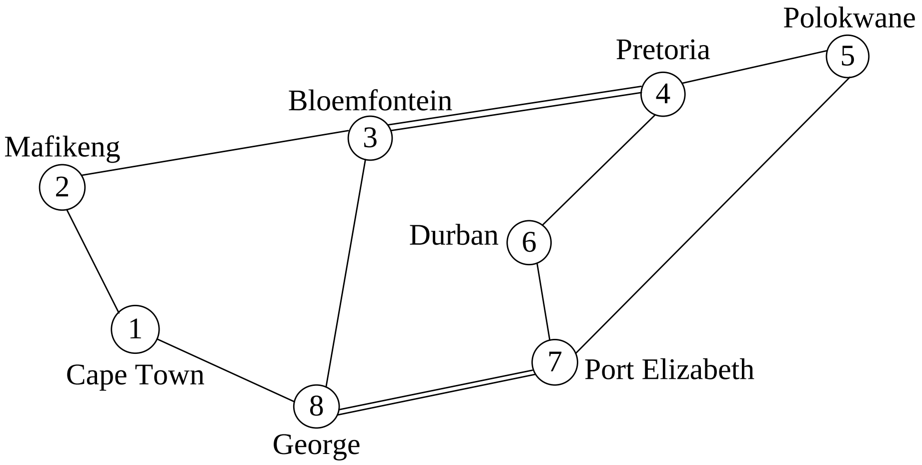

Figure 1: An SA infrastructure network

There is a guarantee of the convergence of the algorithm due to the finite (albeit large) number of shortest route flows. Because of the decrease in delay, repetitions of the same flow do not exist. And since the delay decreases with every iteration and it is positive, the algorithm converges.

In general, this approach only leads to a local minimum, not a global optimum. However, for the special case (i.e., $d_{ij}C_{ij} = d_{ij}C_{ij}^{\alpha} + d_{ij0}$ where d_{ij0} is a positive start-up cost, and $0 \leq \alpha \leq 1$), it is still possible to find a global minimum (Berezner & Krzesinski, 2001).

5 NETWORK PERFORMANCE ANALYSIS

An application of our algorithm to the network topology shown in Figure 1 is next introduced. The network model consists of 8-nodes and 10-links which is a fictitious network connecting eight South African cities in different provinces to compute the optimal link capacities. Each link carries traffic in both directions. The double lines between nodes 3 and 4 and 7 and 8 indicate that there are two links in each direction connecting these nodes. The network carries 2 traffic classes: the bandwidth requirement of the first class is 1 unit and the bandwidth requirement of the second class is 40 units.

The capacity of each link is 5624 units. The traffic intensity matrix is given in Table 1. Note that the table has zeros on the main diagonal and infinity (denoted by '–' here) wherever no links exists. Note also, that the mapping of the distance between cities is being done randomly and does not necessarily reflect the physical distance between cities. The class dependent intensities are given by $\rho_{ij}^{s} = \rho_{ij}\gamma_s b_s$ where ρ_{ij} represents the traffic intensity between link (i, j), γ_s is the class load factor and b_s the bandwidth. These values are given in Table 2.

Our algorithm and its variants adapted to the problems have been coded in C++ and compiled through gcc - on the GNU/Linux system. The numerical tests were performed on an Intel Core

Table 1: The SA network: traffic intensity matrix

nodes	1	2	3	4	5	6	7	8
1	0	13	15	2	20	10	4	6
2	–	0	49	6	64	29	11	17
3	–	–	0	7	76	34	13	21
4	–	–	–	0	9	4	2	2
5	–	–	–	–	0	45	17	27
6	–	–	–	–	–	0	8	12
7	–	–	–	–	–	–	0	4
8	–	–	–	–	–	–	–	0

Table 2: Load factor and slots per service

	class 1	class 2
γ_s	20	0.1
b_s	1	40

i3-2310 CPU@2.10GHz computer with 2.5Gb RAM.

We now compare the network optimal link capacities with their initial values. The experiment results are expressed by presenting the values of the optimal routes, the optimal flow solution, the optimal link capacities and the average end to end delay in the network corresponding to the best feasible solution. An outcome of these experiments was that we were able to verify ways to optimize available bandwidth usage and also minimize the impact of traffic congestion on the network using MPLS TE.

Table 3 shows the optimal link flows and capacities computed after the convergence of the capacity assignment algorithm. The optimal and initial link capacities' values calculated are displayed in Figure 2.

Figure 3(a) plots the capacity assigment for link (1-2) as the algorithm executes while the network delay is plotted in Figure 3(b) as the algorithm executes. The delays converge after 44 iterations. That is, when the newly calculated network delay is not significantly better than the previous one. One can see the convergence of the algorithm and how the network delay gets better and better after each iteration in Figure 3(b). These results are obtained through explicit routing, in that an explicit LSP is already established through the network, profiting from all available network resources. Further, we were able to offer a degree of resource assurance with MPLS QoS elements. In particular, we were able to verify how to best plot traffic into a specific LSP with the aim of IP network performance enhancement.

Comparison with Figure 3(a) shows a strong correlation between the delay and the assigned

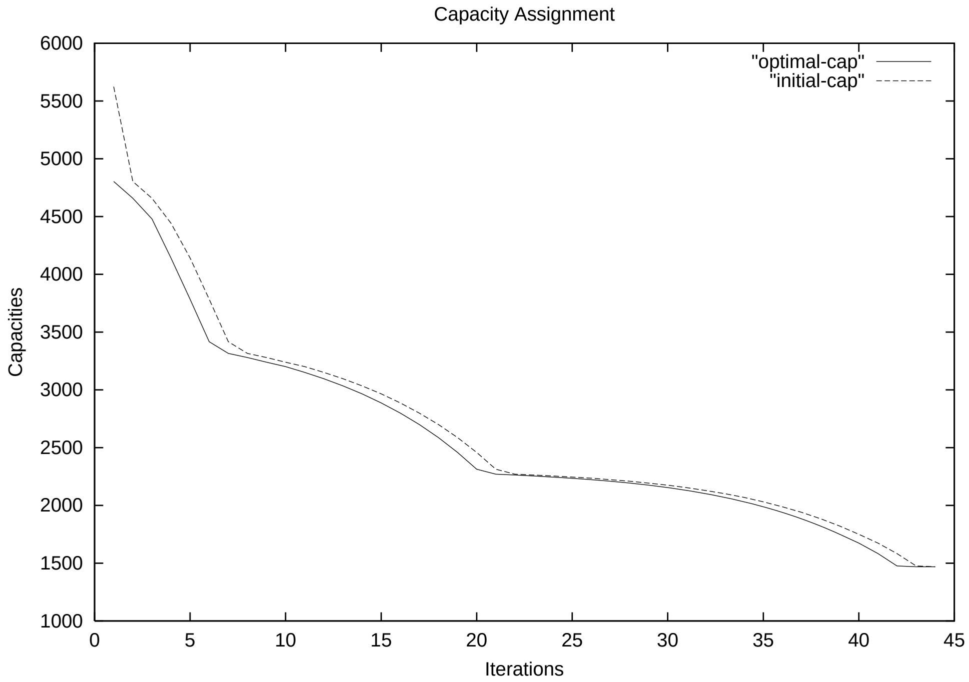

Figure 2: The optimal versus the initial link capacities

capacity. This can be expected since the delay is a function of capacity. The delays decrease since the flows are shifted onto optimal paths in order to reduce the delays on congested links. The smaller flow on the congested links in turn leads to a decrease in the capacity required to achieve a given delay.

There are several approaches to capacity assignment, utilising different performance criteria. An additional approach is to improve an existing network by redistributing the link capacities while maintaining the total sum of all capacities of the network.

Kleinrock (1976) notes that the selection of an appropriate algorithm to allocate capacities will depend on the cost-capacity structure, on the presence of additional topological constraints, on the degree of human interaction allowed and, finally, on the tradeoff between cost and precision required by the particular application. Kershenbaum (1993) describes a capacity assignment approach that guarantees an optimal solution but can take an inordinate amount of time. The algorithm can yield some approximate results by alternatively strengthening the dominance criterion. Another approach originally proposed by Whitney guarantees a solution in a reasonable amount of time and also gives a bound on the quality of the solution it obtains (which is not, in general, optimal).

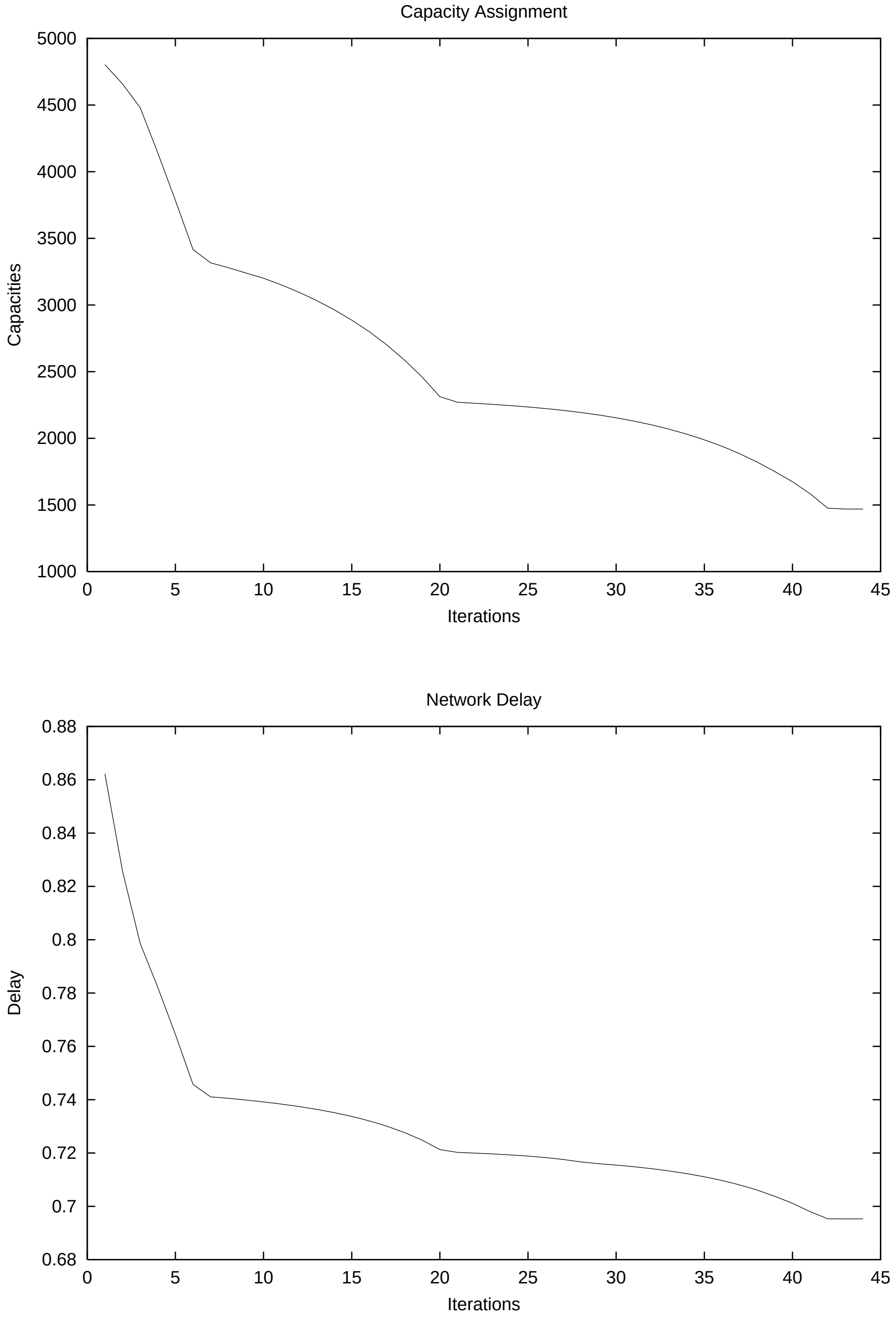

Figure 3: (a) Capacity assignment algorithm. (b) Convergence of the algorithm

https://doi.org/10.18489/sacj.v29i1.393

Table 3: Optimal link flows and link capacities

Links	**Optimal Flows**	**Optimal Cap**	**Initial Cap**
(1,2)	520	1469.4	5624
(1,8)	2280	4268.1	5624
(2,3)	7040	10533.4	5624
(3,4)	8800	12705.8	11248
(3,8)	3240	5609.9	5624
(4,6)	4480	7266.8	5624
(5,7)	2640	4779.3	5624
(6,7)	1200	2642.3	5624
(7,5)	2640	4779.3	5624
(7,8)	4040	6686.4	11248

6 MULTISERVICE BLOCKING MODEL

In this section, we again model a network as a grouping of available means that allows telephone calls to arrive randomly. These telephone calls all have a related holding time and class. However, this time a call can be blocked.

The FOA (Rassaki & Nel, 2013) uses an objective function based on the $M/M/1$ queue. In this system, if a message or customer arrives when the channel is not busy, (i.e., no message in transmission), the message is transmitted immediately. If the channel is busy when the message arrives, the message is placed in a queue where it waits until the channel becomes free and then begins serving the next message.

Not all the systems deal with congestion by allowing messages to wait. Most traditional telephone systems block calls from entering the system if no capacity is available. Ross (2012) defines a loss system as a set of accessible means to which calls, each with a linked holding time and class, arrive at random instances.

This kind of system is fundamentally different from a queueing system because a call's system time is equal to its holding time. Here, a call arrives and requires a fixed amount of capacity, enough to handle a conversation. If the capacity is available, it is dedicated to the call for its duration. If not the call is blocked and lost. We consider such a system in this section.

6.1 Network model

In our blocking model, a network consists of N nodes with L physical links which carries S classes of calls. Each link $i-j$ has capacity of C_{ij} bandwidth units. We define a route as a sequence of physical links. Let $\mathscr{R}_{ij}$ denote the set of routes that connect i and j. Class-s calls are offered to O-D pair

Table 4: The link blocking probabilities per service class

	Service Integration		Service Separation	
Links	class1 $\times 10^{-4}$	class2 $\times 10^{-3}$	class1 $\times 10^{-3}$	class2 $\times 10^{-3}$
(1,2)	2.04698	9.99201	9.78443	9.69459
(1,8)	2.13866	9.95198	9.87183	9.22791
(2,3)	2.15838	9.99911	9.96748	9.00554
(3,4)	2.22972	9.96107	9.96809	8.91993
(3,8)	1.97635	9.95442	9.67632	6.76622
(4,5)	2.16964	9.98509	9.89086	9.74935
(4,6)	2.04468	9.97948	9.86695	9.85335
(5,7)	2.15711	9.93149	9.88864	9.41679
(6,7)	2.04277	9.92063	9.79374	7.07872
(7,8)	2.16638	9.96009	9.85324	8.57969

(i, j) according to a Poisson process with rate $\lambda^s_{i,j}$. The average holding time of a call of class-s is exponentially distributed with mean $1/\mu^s_{i,j}$. We refer to $\rho^s_{i,j} = \lambda^s_{i,j}/\mu^s_{i,j}$ as the class-s intensity of the offered traffic stream. The bandwidth requirement of a class-s call is b_s. A class-s connection between O-D pair (i, j) is admitted if there is sufficient bandwidth available on at least one route in $\mathscr{R}^s_{ij}$ to accommodate its effective bandwidth and is lost otherwise.

Our goal is to obtain a capacity assignment such that the link blocking probabilities satisfy a certain grade of service (GoS).

6.2 Implementation

We compute the link blocking probabilities B^s_{ij} as

$$B^s_{ij} = E^s_{ij}(\rho_{ij}, \mathbf{b}, C_{ij}) \tag{13}$$

where $\rho_{ij} = (\rho^s_{ij})_{s\in S}, \mathbf{b} = (b_s)_{s\in S}$ and $E^s_{ij}(.)$ is a function returning the blocking probabilities of class-s calls on link (i, j).

6.3 Blocking probabilities

There are several options for the blocking function $E^s_{ij}(.)$ (Ross, 2012; Ross, Tang, & Wang, 1994) and references therein for examples). The authors introduce the stochastic knapsack to which objects arrive and depart at random times. We have opted to use the stochastic knapsack algorithm which, along with the computation of multiservice blocking probability, is described in Berezner and Krzesinski (2001).

The implementation starts by computing link flows that are optimal in terms of the network delay. Then we compute the link blocking probabilities using these flows for each link. To calculate the probabilities we need the traffic intensities for different classes which are given by $\rho^s_{ij} = \rho_{ij}\gamma_s b_s$ where ρ_{ij} represents the traffic intensity on link (i, j), γ_s is the class load factor and b_s the bandwidth. After calculating these intensities, we run the multiservice blocking probability algorithm. The algorithm uses the link capacity as a loop index. Thus we first calculate blocking probabilities for the link with capacity 1, then use this to calculate for link with capacity 2, and so on. Each step reduces probabilities and it is quite natural: we expect calls to be lost less frequently on links with higher capacities. When the capacities are large enough, the blocking probability tends to zero. The iteration terminates as soon as the average blocking probability becomes less than the GoS required.

6.4 Service Separation

Under service integration, all the bandwidth C_{ij} of link (i, j) is available to all service classes. With service separation, each class has access only to a different bandwidth $C^1_{ij}, C^2_{ij}, \cdots, Cij^S$ where $C^1_{ij} + C^2_{ij} + \cdots + C^S_{ij} = C_{ij}$.

In this case, we calculate the optimal link capacities required for every class with certain blocking probability, then we sum all capacities to obtain the capacity on a link. Consider again the network model in Section 6. The network consists of 8 nodes and carries 2 traffic classes. The traffic intensity matrix is given in Table 1 and the class dependent intensities are given by multiplying these values by the load factors.

Table 4 summarizes the results of the computational tests. It presents the link blocking probabilities per service class for service integration as well as service separation and we compare the link capacities produced by the two services in Table 5.

We can see from both Table 5 and Figure 4 that links with larger flows have larger capacities and therefore smaller service times which is a factor contributing to the reduction in the total delay of the network. The network model described in the previous section was capacitated with a GoS of 1%.

Ross (2012) and others have written papers on the issue of determining grade of service allocation to multi services connections (Chlebus, Coyle, Henderson, Pearce, & Taylor, 1994; Bean, Gibbens, & Zachary, 1995).

Note that our experiment combines two different models: the blocking model and the queueing model. However, these models are applied at different stages. First we have the queueing model associated with a network delay. Secondly, we have the blocking model associated with calls blocked and packets dropped. In first case, the capacities must be enough to accomodate the optimal flows, otherwise the network delay becomes indefinite. In the second situation, some capacities might be insufficient and calls will be dropped.

In order to evaluate the effect of applying service separation, we also compute the optimal link capacities for service integration and then compare the two. The total link capacity computed in service separation is the sum of the capacities required in both traffic classes.

Table 5: Optimal link flows and capacities per service class

Links	$\mathbf{F}^{s}_{i,j}$		**Cap**$_{int}$	**Cap**$_{sep}$
	class1	*class2*		
(1,2)	1408.4	845.1	2671	2678
(1,8)	2273.1	1363.8	4130	4140
(2,3)	2386.9	1422.9	4320	4333
(3,4)	3834.1	2300.4	6723	6732
(3,8)	1085.1	651.1	2118	2155
(4,5)	2581.5	1548.9	4645	4647
(4,6)	1410.7	846.4	2675	2680
(5,7)	2574.4	1544.7	4634	4640
(6,7)	1478.5	887.1	2791	2828
(7,8)	2615.4	1569.3	4702	4721

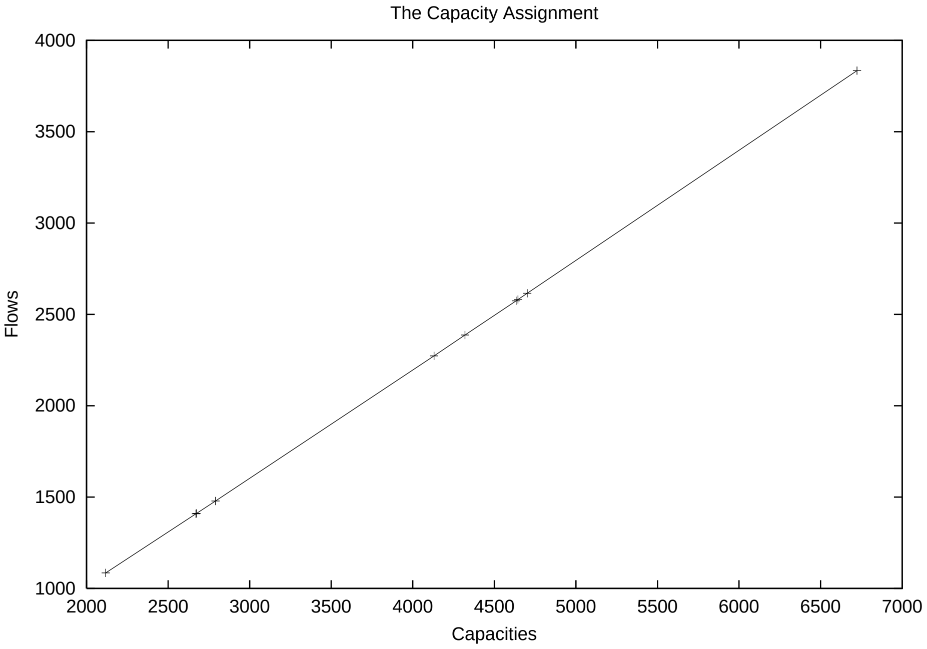

Figure 4: Optimal link capacity versus link flow

https://doi.org/10.18489/sacj.v29i1.393

Our conclusion is that the new capacity is sometimes larger than in the case of complete sharing; this is the penalty for service separation; the penalty is however not too large. For example, for link (4,5) the link capacity on complete sharing is 4645 while the separation case is 4647.

7 CONCLUSIONS AND FUTURE WORK

The thrust of this work is twofold: a) to select the optimal routes and link capacities for MPLS networks and b) to conduct the analysis at the call level in the presence of multiservice traffic sources. Overall system costs are minimised by trading off link capacity costs versus expected network delay costs.

The model is developed for MPLS networks managers and users who deal with these tradeoffs. It is an extension of the method described in Rassaki and Nel (2015) that is applied to thousands of IP networks. The new approach combines two different models: the queueing model and the blocking probability model. First, we have the queueing model associated with the expected network delay. Secondly, we have the blocking model associated with calls blocked and packets dropped. In first case, the capacities must be enough to accommodate the optimal flows, otherwise the network delay becomes indefinite. In the second situation, some capacities might be insufficient and calls will be dropped.

Considering the most basic model, we see one individual transmission link accepting input traffic that results from the superposition of N traffic classes. Two CAC strategies are defined for the multiservice structure in which a distinct level of service is necessary for every class. The comparison between the two strategies brings the following conclusions: a) service separation is not the most appealing option for two reasons: weak performance, and a higher degree of resource management complexity and service deployment; b) the test results for service integration reveal concerns about fairness in some circumstances, even though it is less difficult to put into practice; and c) test outcomes are somewhat influenced by the holding time distribution (specifically, blocking probabilities are dependent on the offered traffic, but also on the arrival rate and holding times).

Selecting an acceptance threshold is significant as far as implementation in a realistic network environment. It is essential to be aware of, in advance, every potential traffic class and to establish a GoS administrative protocol for these classes. Although there will be supplementary complexity costs, the bonus of expanded capacity makes them defensible.

In future work, we plan to improve and assess unique effective procedures to deliver QoS differentiation to the IP flows passing through the MPLS domain.

References

Amiri, A. (1992). *Routing and capacity assignment in backbone communication networks* (Doctoral dissertation, Ohio State University).

Arvidsson, Å. (1995). High level B-ISDN/ATM traffic management in real time. In *Performance modelling and evaluation of ATM networks* (pp. 177–207). Springer. https://doi.org/10.1007/978-0-387-34881-0_10

Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., & Swallow, G. (2001, December). *IETF RFC 3209: RSVP-TE extension to RSVP for LSP tunnels*. Internet Engineering Task Force.

Awduche, D., Malcolm, J., Agogbua, J., & McManus, J. (1999, September). *RFC 2702: Requirements for traffic engineering over MPLS*. Internet Engineering Task Force.

Balakrishnan, A. & Graves, S. C. (1989). A composite algorithm for a concave-cost network flow problem. *Networks*, *19*(2), 175–202. https://doi.org/10.1002/net.3230190202

Bauschert, T. (1997). Multihour design of multi-hop virtual path based wide-area ATM networks. *Proceedings of 15th International Teletraffic Conference*, 1019–1030.

Bean, N. G., Gibbens, R. J., & Zachary, S. (1995, March). Asymptotic analysis of single resource loss systems in heavy traffic with applications to integrated networks. *Journal of Advances in Applied Probability*, *27*(1). https://doi.org/10.1017/s0001867800046346

Berezner, S. & Krzesinski, A. E. (2001). An efficient stable recursion to compute multiservice blocking probabilities. *Performance Evaluation*, *43*(2), 151–164. https://doi.org/10.1016/S0166-5316(00)00042-0

Bertsekas, D. & Gallager, R. (1992). *Data networks* (2nd). Prentice-Hall, Englewood Cliffs.

Chlebus, E., Coyle, A., Henderson, W., Pearce, C. E. M., & Taylor, P. G. (1994). Mean-value analysis for examining call admission thresholds in multiservice networks. In *The Fourteenth International Teletraffic Conference, Antibes, France*.

Fratta, L., Gerla, M., & Kleinrock, L. (1973). The flow deviation method: An approach to store-and-forward communication network design. *Networks*, *3*(2), 97–133. https://doi.org/10.1002/net.3230030202

Gavish, B. & Altinkemer, K. (2000, December). Backbone network design tools with economic tradeoffs. *ORSA Journal of Computing*, *2*, 226–252.

Gerla, M. (1975). Approximations and bounds for the topological design of distributed computer network. In *ACM 4th Int Conf on Data Communications Symposium, Quebec, Canada*.

Gershet, H. & Weihmayer, R. (1990). Joint optimization of data network design and facility location. *IEEE Journal on Selected Areas in Communications*, *8*, 149–152.

Gopal, G., Kim, C.-K., & Weinrib, A. (1991). Algorithms for reconfigurable networks. In *13th International Teletraffic Congress Proceedings* (pp. 288–289). Citeseer.

Herzberg, M. & Byes, S. (1993). Bandwidth management for reconfigurable multi-service networks. *Australian Teletraffic Research*, *25*(2).

Jamoussi, B. e. a. (2002, January). *IETF RFC 3212: Constraint-based LSP setup using LDP*. Internet Engineering Task Force.

Kershenbaum, A. (1993). *Telecommunications network design algorithms*. McGraw-Hill.

Kleinrock, L. (1976). *Queueing systems vol 2: Computer applications*. John Wiley and Sons.

LeBlanc, L. J. & Simmons, R. V. (1989). Continuous models for capacity design of large packet-switched telecommunication networks. *ORSA Journal on Computing*, *1*(4), 271–286. https://doi.org/10.1287/ijoc.1.4.271

Martens, M. & Skutella, M. (2006). Flow on few paths: Algorithms and lower bounds. *Networks Journal*, *48*(2), 68–76. https://doi.org/10.1002/net.20121

Maruyama, K., Fratta, L., & Tang, D. T. (1977). Heuristic design algorithm for computer communication networks with different classes of packets. *IBM Journal of Research and Development*, *21*(4), 360–369. https://doi.org/10.1147/rd.214.0360

Maruyama, K. & Tang, D. T. (1976). Discrete link capacity assignment in communication networks. In *ICCC 3rd Int Conf, Toronto, Canada* (pp. 92–97).

Pióro, M. & Medhi, D. (2004). *Routing, flow, and capacity design in communication and computer networks*. Elsevier.

Pompili, D., Scoglio, C., & Shoniregun, C. A. (2007). Virtual-flow multipath algorithms for MPLS. *Journal of Internet Technology and Secured Transactions*, *1*(1/2), 1–19. https://doi.org/10.1504/IJITST.2007.014832

Rassaki, A. & Nel, A. (2013, August). Quality of service in MPLS networks. In *IASTED 15th Int Conference on Control and Applications, Honolulu, USA, pp 67-74, Aug. 2013.*

Rassaki, A. & Nel, A. (2015). Optimal capacity assignment in IP networks. In *Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference on* (pp. 256–261). IEEE. https://doi.org/10.1109/icdipc.2015.7323038

Rohne, M., Jensen, T., Svinnset, L., & Venturin, R. (1998). Designing VP networks. In *Fourteenth Nordic Teletraffic Seminar, Copenhagen, Danemark* (pp. 17–30).

Rosen, E., Viswanathan, A., & Callan, R. (2001, January). *IETF RFC 3031: Multiprotocol label switching architecture*. Internet Engineering Task Force.

Ross, K. W. (2012). *Multiservice loss models for broadband telecommunication networks*. Springer Science & Business Media.

Ross, K. W., Tang, D., & Wang, J. (1994). Monte Carlo summation and integration applied to multichain queueing networks. *Journal for the Association of Computing Machinery*, *41*(6). https://doi.org/10.1145/195613.195630

Stallings, W. & Case, T. (2013). *Business data communications infrastructure, networking and security*. Upper Saddle River, NJ. Pearson, Education Inc.

Truffot, J., Duhamel, C., & Mahey, P. (2010). K-splittable delay constrained routing problem: A branch and price approach. *Network Optimization*, *55*(1), 33–45. https://doi.org/10.1002/net.20311

Whitney, V. (1972). Lagrangian optimization of stochastic communication systems models. In *Proceedings MRI Symposium on Computer-Communication Networks, Brooklyn, New York* (pp. 332–342).

Wu, L., Cheval, P., & Vaananen, P. (1998). *MPLS extensions for differential services*. Internet Engineering Task Force. Retrieved from https://tools.ietf.org/html/draft-wu-mpls-diff-ext-00