A Simple Methodology for IP Network Design with End-to-End QoS Constraints: The VPN Case

Emilio C. G. Wille[1−2], Marco Mellia[1], Emilio Leonardi[1], Marco Ajmone Marsan[1]
[1]Dipartimento di Elettronica - Politecnico di Torino - Italy
[2]Departamento de Eletrônica - Centro Federal de Educação Tecnológica do Paraná - Brazil
e-mail: {wille, mellia, emilio, ajmone}@mail.tlc.polito.it

***Abstract*— In recent years, overprovisioning became the design approach of choice for the Internet, and drove costs to unsustainable levels, thus playing a significant role in the burst of the Internet bubble and in the downturn of the entire telecommunications sector. Today, many packet networks are not profitable because the installed capacity is overprovisioned, and thus largely unused. In order to achieve packet network profitability, we need new, reasonable packet network design methodologies, that allow the choice of the most adequate set of network resources for the delivery of a given mix of services with the desired level of quality. In this paper we describe a simple methodology to tackle the packet network design problem, considering as constraints the end-to-end QoS (Quality of Service) metrics, and we illustrate an example of its application to the optimization of link capacities and routing in a corporate VPN (Virtual Private Network) where traffic is mostly due to TCP connections. An efficient Lagrangean relaxation based heuristic procedure is developed to find bounds and solutions for the considered problem, and numerical results for a variety of problem instances are reported, proving that the design methodology is both efficient and effective.**

***Index Terms*—Network design and planning, Quality of Service, Capacity and Flow Assignment**

I. Introduction

Packet network design is an old problem, that was extensively investigated in the early days of packet networks, starting with the seminal work of Kleinrock in the mid-sixties [1], [2].

In recent years, the widespread diffusion of the Internet, its enormous success, the expectation for a never-ending growth, and a generally accepted incorrect estimation of the traffic increase rate, led to the adoption of overprovisioning as the design approach of choice for the Internet. However, overprovisioning drove costs to unsustainable levels, thus playing a significant role in the burst of the Internet bubble and in the downturn of the entire telecommunication sector. Today, many packet networks are not profitable because the installed capacity is overprovisioned, and thus largely unused.

While traffic slowly grows to make packet networks profitable, it is necessary to devise new, reasonable packet network design methodologies, that allow the choice of the most adequate set of network resources for the delivery of a given mix of services with the desired level of end-to-end quality of service (e2e QoS). This requires optimization approaches capable of considering the dynamics of packet networks, as well as the effect of protocols at the different layers of the Internet architecture on the e2e QoS experienced by end users.

The packet network design methodology that we propose in this paper is quite different from the many proposals that appeared in the literature. Those focus almost invariably on the tradeoff between total cost and average performance (expressed in terms of average network-wide packet delay, average packet loss probability, average link utilization, network reliability, etc.). This may lead to situations where the average performance is good, but, while some traffic relations obtain very good QoS, some others suffer unacceptable performance levels. On the contrary, our packet network design methodology for the first time (to the best of our knowledge) is based on user-layer QoS parameters, and explicitly accounts for each source/destination QoS constraint.

The key element of the proposed packet network design methodology consists in the mapping of the e2e QoS constraints into transport-layer performance constraints first, and then into network-layer performance constraints. The latter are then considered together with a realistic representation of traffic patterns at the network layer to design the IP network.

The description of traffic patterns inside the Internet is a particularly delicate issue, since it is well known that IP packets do not arrive at router buffers following a Poisson process [3], but a higher degree of correlation exists. This means that the usual approach of modeling packet networks as networks of M/M/1 queues [4], [5], [6], [7], [8] is not acceptable. In this paper we adopt a somewhat more refined IP traffic modeling technique, already presented in [9], that provides a more accurate description of the traffic dynamics in multi-bottleneck IP networks loaded with TCP mice and elephants. The resulting analytical model is both simple and capable of producing accurate performance estimates for general-topology packet networks loaded by realistic traffic patterns.

Designing a packet network today may have quite different meanings, depending on the type of network that is being designed. If we consider the design of the physical topology of the network of a large Internet Service Provider (ISP), the design must very carefully account for the existing infrastructure, for the costs associated with the deployment of a new connection or for the upgrade of an existing link, and for the very coarse granularity in the data rates of high-speed links. Instead, if we consider the design of a corporate VPN (Virtual Private Network), where connections are leased from a long distance carrier, the set of leased lines is not a critical legacy, costs are directly derived from the leasing fees, and the data rate granularity is much finer. While the general methodology for packet network design and planning that we describe in this paper can be applied to both contexts, as well as others, in this paper we concentrate on the design of corporate VPNs.

Traditionally, packet network design focused on optimizing either network cost or performance by tuning link capacities and routing strategies. Since the routing and link capacities optimization problems are closely interrelated, it is appropriate to jointly solve them in what is called the CFA (Capacity and Flow Assignment) problem. In this paper, we present a nonlinear mixed-integer programming formulation for the generic CFA problem and solve it in the case of corporate VPNs. An efficient Lagrangean relaxation based heuristic procedure is developed to find bounds and solutions. When explicitly considering TCP traffic it is also necessary to tackle the Buffer Assignment (BA) problem, for which we propose an efficient solution for the droptail case as well as for more advanced Active Queue Management (AQM) schemes, like RED [10].

Numerical results for a variety of problem instances are reported, proving that the proposed design methodology is both efficient and effective.

The rest of the paper is organized as follows. Section II briefly mentions some previous works in the field of packet network de-

sign. Section III describes the general design methodology and provides the formulation of the optimization problem. Section IV specializes the optimization problem to the case of corporate VPNs, and illustrates a Lagrangean relaxation of the problem, as well as a heuristic solution procedure. Numerical results are discussed and compared against results of ns-2 simulations in Section V. Conclusions are given in Section VI.

II. Related Work

The literature focusing on the routing problem, where link capacities are assumed to be known, is abundant; see, for example, [2], [5], [7], [11]. Papers where the routing and capacity assignment problems are treated simultaneously include [2], [4], [6], [8], [12], [13].

Gerla and Kleirock [2] presented a series of heuristics to solve continuous (concave) and discrete versions of the CFA problem, based on the flow deviation algorithm. A difficulty experienced with heuristic methods is that no information can be obtained about the distance between the best solution found, and the actual optimal solution.

Gavish and Neuman [4] formulated the CFA problem as a non-linear integer programming problem, and proposed a Lagrangean relaxation based approach. Gersht and Weihmayer [8] presented a mixed integer/linear programming (MILP) formulation of the optimal network design and facility engineering problem, which corresponds to finding network topologies that minimize the total network cost while selecting facility types, allocating capacity, and routing traffic to accommodate traffic demands and performance requirements. The MILP formulation is decomposes into two subproblems, which can be solved sequentially. The solution of the first subproblem yields the topological design, facility selection, and flow assignment. The second subproblem consists in the capacity assignment.

All these works use M/M/1 queueing systems to model the network behavior, and aggregate packet delay in the problem formulation.

Ng and Hoang [12] present a global optimal solution technique for the CFA problem. A continuous lower bound of the average packet delay is used in the formulation of the cost objective function. They consider an m-$M/M/1$ queueing system to model the network behavior (where a link is implemented by m transmission lines, each of capacity C); therefore, the objective function is shown to be convex with respect to the network multicommodity flow. The convexity property ensures the global optimum solution of the CFA problem, that is obtained using the flow deviation method.

Cheng and Lin [6] consider the problem of minimizing the maximum end-to-end delay in the network. They propose a two-phase algorithm to solve the CFA problem, where in a first phase a minimum-hop heuristic routing is used, and in a second phase the capacity assignment problem is solved. They too adopt M/M/1 queueing systems to model the network.

Medhi and Tipper [13] proposed four approaches based on the Lagrangean relaxation with sub-gradient optimization method and genetic algorithms to obtain solutions to a multi-hour combined capacity design and routing problem, neglecting however packet delay constraints.

In [14], the authors for the first time abandon the Markovian assumption in favor of a Long Range Dependent (LRD) traffic model, i.e., a Fractional Brownian Motion model. They solve the discrete capacity assignment problem under network e2e delay constraints only, using simulated annealing metaheuristic. However, it is difficult to extend this approach to consider more general CFA problems, because the relation among traffic, capacity and queueing delay is not expressed by a closed formula.

To the best of our knowledge, no previous work solves the CFA problem for packet networks accounting for user layer e2e QoS constraints.

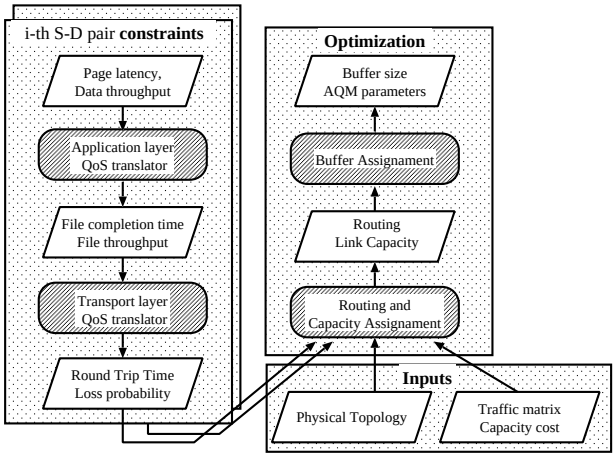

Fig. 1. Schematic Flow Diagram of the Network Design Methodology

III. The IP Network Design Methodology

The simple IP network design methodology that we propose in this paper is based on a "Divide and Conquer" approach, in the sense that it consists of several sub-tasks, which are solved separately. Such an approach is a necessity, because, even if the resulting methodology provides sub-optimal solutions, the complexity of the problem makes a global solution impossible.

Fig. 1 shows the flow diagram of the design methodology. Shaded, rounded boxes represent function blocks, while white parallelograms represent input/output of functions. Three are the main blocks, which correspond to the classic blocks in constrained optimization problems: *constraints* (on the left), *inputs* (on the bottom right) and *optimization procedure* (on the top right). As constraints we consider, for every source/destination pair, the specification of user-layer QoS parameters, e.g., download latency for web pages or perceived quality for real-time applications. Thanks to the definition of *QoS translators*, all the user-layer QoS constraints are then mapped into lower-layer performance constraints, down to the network layer, where performance metrics are typically expressed in terms of average delay and loss probability.

The optimization procedure needs as inputs the description of the physical topology, the traffic matrix, and the expression of the cost as function of link capacities. The objective of the optimization is to find the minimum cost solution that satisfies the user-layer QoS constraints. The solution identifies link capacities, flow assignment (i.e. routing) and buffer sizes (or AQM parameters).

In our methodology we decouple the CFA problem from the BA problem. The optimization starts then with the CFA subproblem, solved considering infinite buffers. A second optimization is then performed to solve the BA sub-problem. Motivations for this choice are given in the following sections, where we briefly comment on the main steps of the design methodology, and we provide a formal description for the optimization problem.

A. QoS translators

The process of translating QoS specifications between different layers of the protocol stack is called QoS translation or QoS mapping. Several parameters can be translated from layer to layer, for example: delay, jitter, throughput, or reliability. An overview of the QoS translation problem is given in [15]. According to the Internet protocol architecture, at least two QoS mapping procedures should be considered in our case; the first translates the application-layer QoS constraints into transport-layer QoS constraints, and the second translates transport-layer QoS constraints into network-layer QoS constraints, such as *Round Trip Time* (RTT) and *Packet Loss Probability* (P_{loss}).

1) Application-layer QoS translator: **This module takes as inputs the application-layer QoS constraints, such as web page transfer latency, data throughput, audio quality, etc. Assuming then that for each application we know which transport protocol is used, i.e., either TCP or UDP, this module maps the application-layer QoS constraints into transport-layer QoS constraints. Given the multitude of Internet applications, it is not possible to devise a generic procedure to solve this problem, and we do not focus on generic translators, since ad-hoc solutions should be used, depending on the application.**

For real-time applications over UDP, the output of the application-layer translator is given in terms of packet loss probability, and maximum network e2e delay.

For elastic applications exploiting TCP, the output of the application-layer translator is still a set of high-level constraints, expressed as *file transfer latency* **(L_t), or** *throughput* **(T_h).**

a) Example: Voice over UDP: **In this case, the application-layer QoS translator is in charge of translating the high-level QoS constraint, such as the Mean Opinion Score (MOS), into transport-layer performance constraints, expressed in terms of packet loss probability, maximum network e2e delay. Several studies were conducted on this subject [16]. For example, good vocal perceived quality is associated with an average packet loss probability of the order of 1%, and a maximum e2e delay smaller than 200 ms.**

b) Example: Web page download: **In this case, the input of the application-layer QoS translator is a desired download time, expresses as a function of the page size, the protocol type, the number of objects in the page, etc. As output, the TCP latency constraint is evaluated. For example, given a desired web page download time smaller than 1.5 s, a web page which contains 20 objects, downloaded using 4 parallel TCP connections at most, each object must be transferred with a TCP connection of average duration smaller than 0.3s.**

2) Transport-layer QoS translator: **The Transport-layer QoS translator maps transport-layer performance constraints into network-layer performance constraints; the translator in this case must be tailored to the transport protocol used: either UDP or TCP.**

a) Real Time Applications - UDP: **The translation from transport-layer performance constraints into network-layer performance constraints in the case of real-time UDP applications is rather straightforward, since the transport-layer performance constraints are usually expressed in terms of packet loss probability and maximum e2e network delay, which can be directly used also as network-level performance parameters. The only effect of UDP that must be taken into account is related to the protocol overhead, which increases the offered load to the network. This effect may be significant, specially for applications like voice, that use small packets.**

b) Elastic Traffic - TCP: **The translation from transport-layer QoS constraints to network-layer QoS parameters, such as Round Trip Time (RTT) and packet loss probability (P_{loss}) in this case is more difficult. This is mainly due to the complexity of the TCP protocol, and in particular to the error, flow and congestion control algorithms.**

The TCP QoS translator accepts as inputs either the maximum file transfer latency, or the minimum file transfer throughput. We impose that all flows shorter than a given threshold (i.e., TCP mice) meet the maximum file transfer latency constraint, while longer flows (i.e., TCP elephants) are subjected to the throughput constraint. For example, from measurements of the file length distribution [17] over the Internet, it is possible to say that 85% of all TCP flows are shorter than 20 segments. For these flows, we impose that the latency constraint must hold. Instead, for flows longer than 20 segments we impose that the throughput constraint must be met. Obviously, the most stringent constraint must be considered in the translation. The maximum RTT and P_{loss} that satisfy both constraints constitute the output of this translator.

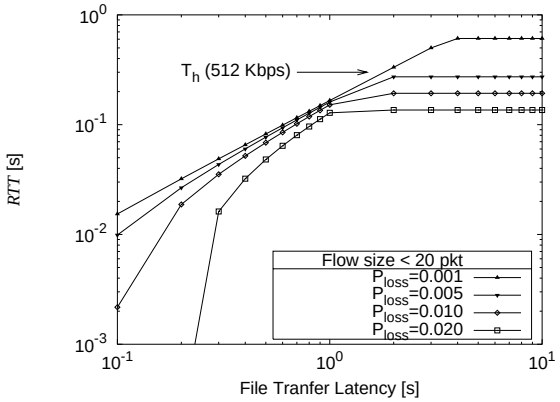

Fig. 2. RTT Constraints as given by the Transport Layer QoS Translator

To solve the translation problem, we exploit recent research results in the field of TCP modeling (see [9] and the references therein). Usually, TCP models take network-layer parameters as inputs, i.e., RTT and packet loss, and give as output either the average throughput or the file transfer latency. Our approach is based on the inversion of known TCP models, taking as input either the connection throughput or the file transfer latency, and obtaining as outputs RTT and packet loss. Among the many models of TCP presented in the literature, when considering file transfer latency, we use the TCP latency model described in [18], which offers a good tradeoff between computational complexity and accuracy of performance predictions. We will refer to this model as CSA (from the last name of authors). When considering throughput, we instead exploit the well-known PFTK formula [19]. Our methodology can however be modified to incorporate more complex/accurate TCP models.

The inversion of TCP models is not simple, since at least two are the parameters that impact TCP throughput and latency, i.e., RTT and P_{loss}. An infinite number of possible solutions for these two parameters satisfies a given constraint at the TCP level. We decided therefore to fix the P_{loss} parameter, and leave RTT as the free variable. This choice is due to the considerations that the loss probability has a larger impact on the latency of very short flows, and that it impacts the network load due to retransmissions. Furthermore, P_{loss} is also constrained by real-time applications. Finally, fixing the value of the loss probability allows us to decouple the CFA problem from the BA problem, as shown in Section III-B.1. Therefore, after choosing a value for P_{loss}, a set of curves can be derived, showing the behavior of RTT versus file latency and throughput. From these curves it is then possible to derive the maximum allowable RTT. The inversion of the CSA and PFTK formulas is obtained using numerical algorithms.

For example, given a maximum file transfer latency and a minimum throughput $T_h = 512$ Kbps constraint, the curves of Fig. 2 report the maximum admissible RTT which satisfies the most stringent constraint for different values of P_{loss}.

B. Optimization formulation and solution

For the sake of simplicity, in the rest of this paper we restrict our discussion to TCP traffic only, since TCP is known to carry over 90% of the Internet traffic. However, the problem formulation can be easily extended to more general cases.

In the solution of the CFA and BA problems, we need to evaluate the packet delay and loss probability for given values of the network parameters, in order to verify that the QoS constraints are met. Before presenting the problem formulation, we thus first introduce the network model and discuss the relations between performance measures, input parameters, design variables, and constraints that appear in the design problem.

1) Network model: **The network model is an open network of queues, where each queue representsan output interface of an IP router, with its buffer. The routing of customers on this queuing network reflects the actual routing of packets within the IP network.**

In order to obtain a useful formulation of the CFA problem, it is necessary on one side to be accurate in the prediction of the performance metrics of interest (average delay, packet loss probability), while on the other side limiting the complexity of the model, (i.e., we are forced to adopt models allowing a simple closed-form solution).

Traditionally, either $M/M/1$ or $M/M/1/B$ queueing models were considered as good representations of packet queueing elements in the network. However, the traffic flowing in IP networks is known to exhibit LRD behaviors, which cause queue dynamics to severely deviate from the above model predictions. For these reasons, the classical queuing models appear now inadequate for the design of packet networks.

Unfortunately, explicitly considering LRD traffic models is not practical. Indeed, queues driven by LRD processes are very difficult to study, and only few asymptotic results exist. To the best of our knowledge, no closed formula exists for queues fed by LRD processes, which relates the queue performance to input parameters [14].

In [9], a simple and quite effective expedient was proposed to accurately predict the performance of network elements subject to TCP traffic, using Markovian queueing models. The main idea behind the approach in [9] consists in reproducing the effects of traffic correlations on network queueing elements by means of Markovian queueing models with batch arrivals. The choice of using batch arrivals following a Poisson process has the advantage of combining the nice characteristics of Poisson processes (analytical tractability in the first place) with the possibility of capturing the burstiness of the IP traffic. Hence, we model network queueing elements using $M_{[X]}/M/1$ queues. The batch size varies between 1 and $W$with distribution $[X]$, where W is the maximum TCP window size expressed in segments. The distribution $[X]$ is obtained considering the number of segments that TCP sources send in one RTT [9] for a given flow length distribution. The Markovian assumption for the batch arrival process is mainly justified by the Poisson assumption for the TCP connection generation process (when dealing with TCP mice), as well as the fairly large number of TCP connections simultaneously present in the network. Given the flow length distribution, a stochastic model of TCP (described in [9]) is used to obtain the batch size distribution $[X]$. The evaluation of $[X]$ is done only once before starting the CFA optimization.

2) Problem formulation: **In the mathematical model, the network infrastructure to be designed is represented by a directed graph $G = (V, E)$ in which V is a set of nodes (with cardinality n) and E is a set of edges (with cardinality m). A node represents a node, and an edge represents a physical link connecting one router to another.**

The average (busy-hour) traffic requirements between nodes can be represented by a requirement matrix $\hat{\Gamma} = \{\hat{\gamma}_{sd}\}$, where $\hat{\gamma}_{sd}$ is the average packet transfer rate from source s to destination d. The $\hat{\Gamma}$ matrix can be derived from a higher-level description of the (maximum) traffic requests, expressed in terms of "pages per second", or "flows per second" for a given source/destination pair.

We consider as traffic offered to the network $\gamma_{sd} = \frac{\hat{\gamma}_{sd}}{1-P_{loss}}$, thus accounting for the retransmissions due to the losses that flows experience along their path to the destination. Recall that P_{loss} is the desired e2e loss probability.

The decision of fixing "a-priori" the loss probability allows us to decouple the CFA solution from the BA solution. We first solve the CFA problem (properly selecting the capacity of links and routing of flows) considering the e2e delay constraints only. Then, we enforce the loss probability to meet the P_{loss} constraints by properly choosing buffer sizes. In the first optimization, a queueing model with infinite buffers will be used, i.e., a $M_{[X]}/M/1/\infty$ queueing model. This provides a pessimistic estimate of the queueing delay that packets suffer with finite buffers, which will results from the second optimization step, during which an $M_{[X]}/M/1/B$ queueing model is used.

The following notation is necessary for developing a mathematical model for the CFA and BA problems:

C_{ij}	**the capacity of link (i, j).**
f_{ij}	**the average data flow on link (i, j).**
d_{ij}	**the physical length of link (i, j).**
RTT_{sd}	**the Round Trip Time of path (s, d).**
B_{ij}	**the buffer size of link (i, j).**
δ_{ij}^{sd}	**auxiliary variables (which represent the fraction of traffic from s to d flowing on link (i, j)**

a) CFA formulation: **Our goal is to minimize the total link costs while determining the best route for the traffic that flows on each source/ destination path, and meeting the maximum e2e packet delay constraint. The following optimization problem is thus formulated:**

$$Z_{CFA} = \min \sum_{i,j} g_{ij}(C_{ij}) \tag{1}$$

subject to:

$$\sum_j \delta_{ij}^{sd} - \sum_j \delta_{ji}^{sd} = \begin{cases} 1 & \text{if } s = i \\ -1 & \text{if } d = i \\ 0 & \text{otherwise} \end{cases} \quad \forall\,(i, s, d) \tag{2}$$

$$K_1 \sum_{i,j} \frac{\delta_{ij}^{sd}}{C_{ij} - f_{ij}} \le RTT_{sd} - K_2 \sum_{i,j} \delta_{ij}^{sd} d_{ij} \qquad \forall\,(s, d) \tag{3}$$

$$f_{ij} = \sum_{s,t} \delta_{ij}^{sd} \gamma_{sd} \qquad \forall\,(i, j) \tag{4}$$

$$0 \le \delta_{ij}^{sd} \le 1 \quad \forall\,(i, j), (s, j); \qquad C_{ij} \ge f_{ij} \ge 0 \quad \forall\,(i, j) \tag{5}$$

The objective function (1) represents the total link cost, which is the sum of the cost functions of link (i, j), $g_{ij}(C_{ij})$. Constraint set (2) contains the flow conservation equations, which define routes for the traffic of each source/destination pair. Equation (3) is the e2e packet delay constraint for each source/destination pair. It says that the total amount of delay experienced by all the flows routed on a path should not exceed the maximum RTT minus the propagation delay of the route. The average queueing delay is expressed by considering an $M_{[X]}/M/1/\infty$ queue [20]:

$$E[T] = \frac{K}{\mu} \frac{1}{C - f} \tag{6}$$

$$K = \frac{m'_{[X]} + m''_{[X]}}{2m'_{[X]}} \tag{7}$$

where $m'_{[X]}$ and $m''_{[X]}$ are the first and second moments of the batch size distribution [X] and $1/\mu$ is the average packet length.

Equation (4) defines the average data flow on the link. Constraints (5) are non–negativity constraints. Finally, $K_1 = K/\mu$, and K_2 is a constant to convert distance in time.

The above formulation can be specialized by enforcing non-bifurcated routing, i.e. by forcing the traffic of a given source/destination pair to follow one path. This is obtained by introducing the following integrity constraints:

$$\delta_{ij}^{sd} \in \{0, 1\} \qquad \forall\,(i, j, s, d) \tag{8}$$

Explicitly considering non-continuous capacity values is possible by adding the following constraints, in which $\mathcal{C}$ is the set of possible integer capacity values:

$$C_{ij} \in \mathcal{C} \qquad \forall\,(i, j) \tag{9}$$

Different formulations of the CFA problem result by selecting i) the cost functions $g_{ij}(C_{ij})$, ii) the routing model, and iii) the capacity constraints; different methodologies can be applied to solve them. In this paper we focus on the VPN case, in which common assumptions are i) linear cost, i.e., $g_{ij}(C_{ij}) = d_{ij}C_{ij}$, ii) non-bifurcated routing, and iii) continuous capacities. Solution techniques for this sub case are presented in Section IV.

3) BA formulation: As final step in our methodology, we need to dimension buffer sizes, i.e., to solve the following problem:

$$Z_{BA} = \min \sum_{i,j} h_{ij}(B_{ij}) \tag{10}$$

Subject to:

$$\sum_{ij} \delta_{ij}^{sd} p(B_{ij}, C_{ij}, f_{ij}, [X]) \leq P_{loss}, \quad \forall\,(s,d) \tag{11}$$

$$B_{ij} \geq 0, \quad \forall\,(i,j) \tag{12}$$

where $p(B_{ij}, C_{ij}, f_{ij}, [X])$ is the average loss probability for the $M_{[X]}/M/1/B$ queue, which is evaluated by solving the CTMC model.

Notice that constraint (11) has been linearized thanks to the following inequality:

$$\begin{aligned} \hat{P}_{loss} = \; & 1 - \prod_{i,j} \left(1 - \delta_{ij}^{sd} p(B_{ij}, C_{ij}, f_{ij}, [X])\right) \leq \\ & \leq \sum_{ij} \delta_{ij}^{sd} p(B_{ij}, C_{ij}, f_{ij}, [X]) \end{aligned} \tag{13}$$

The solution of the linearized problem is a conservative solution for the original BA problem.

We conjecture that the BA problem is a convex optimization problem; however, we were not able to generate a formal proof. The difficulty in this proof derives from the need of showing that $p(B, C, f, [X])$ is convex. Since, to the best of our knowledge, no closed form expression for the $M_{[X]}/M/1/B$ stationary distribution is known, no closed form expression for $p(B, C, f, [X])$ can be derived. However, (i) considering an $M/M/1/B$ queue, $p(B, C, f)$ is a convex function [21]; (ii) approximating $p(B, C, f, [X]) = \sum_{i=B}^{\infty} \pi_i$, where π_i is the stationary distribution of an $M_{[X]}/M/1/\infty$ queue, the dropping probability is a convex function of B.

We can thus classify the BA problem as a multi-variable constrained convex minimization problem; therefore, the global minimum can be found using convex programming techniques. We solve the minimization problem applying first a constraints reduction procedure which reduces the set of constraints by eliminating redundancies. Then, the solution of the BA problem is obtained via the *logarithm barrier method* [22]

The output of the BA problem is the buffer size B_{ij} for each router interface, assuming a droptail behavior. If more advanced AQM schemes are deployed by network providers to enhance the TCP performance, it is possible to derive guideline for the configuration of the AQM parameters as well. In this paper, we consider Random Early Detection (RED) [10] as an example, and discuss how to set its parameters.

The original RED algorithm has three static parameters min_th, max_th, max_p, and one state variable avg. When the average queue length avg exceeds min_th, an incoming packet is dropped with a probability that is a linear function of the average queue length. In particular, the packet dropping probability increases linearly form 0 to max_p, as avg increases from min_th to max_th. When the average queue size exceeds max_th, all incoming packets are dropped.

Ideally, the buffer size should be sufficiently large to avoid that packets are dropped at the queue due to buffer overflow. Therefore, we choose $B_ij = \alpha max\_th, \; \alpha > 1$, e.g., $\alpha = 2$ as suggested in the "gentle" variation of RED.

Therefore, the BA problem can be solved by imposing that

$$p(B_{ij}, C_{ij}, f_{ij}, [X]) = \frac{E_{ij}[N] - min\_th_{ij}}{max\_th_{ij} - min\_th_{ij}} max\_p_{ij} \tag{14}$$

Note that (14) fixes max_p_l by imposing that the average RED dropping probability evaluated at the average queue length $E_{ij}[N] = E_{ij}[N](C_{ij}, f_{ij}, [X])$ satisfies the P_{loss} constraint in (11). Finally, we set $min\_th_{ij} = \beta max\_th_{ij}, \; \beta < 1$. Replacing (14) in (11) and solving the resulting problem, we obtain max_p_l. In the numerical examples that follow, we selected $\alpha = 2, \; \beta = 1/16$.

IV. CFA Problem: the VPN Case

In this section we focus on the case in which the CFA formulation includes constraints (8), which force the routing to be non-bifurcated. The resulting problem is a nonlinear mixed-integer programming problem, which is difficult in general. Except for the nonlinear constraint (3), this is basically a multicommodity flow problem [23], since each source/destination pair transmits a different quantity of traffic over the network. Multicommodity flow problems belong to the class of NP-hard problems. In [6] it is proved that also the continuous relaxation of constraints (8) leads to a non-convex programming problem by verifying the Hessian of (3). As a consequence of this property, in general several local minima exist.

In the following, we propose a composite upper and lower bounding procedure based on a Lagrangean relaxation of the problem.

A. Lagrangean relaxation

The CFA problem is complicated by the nonlinear constraints (3). We first apply the change of variable $w_{ij} = \frac{1}{C_{ij} - f_{ij}}$:

$$Z_{CFA} = min\Big[\sum_{i,j} \frac{d_{ij}}{w_{ij}} + \sum_{i,j}\sum_{s,d} d_{ij}\delta_{ij}^{sd}\gamma_{sd}\Big] \tag{15}$$

subject to:

$$K_1 \sum_{i,j} \delta_{ij}^{sd} w_{ij} + K_2 \sum_{i,j} \delta_{ij}^{sd} d_{ij} \leq RTT_{sd} \qquad \forall\,(s,d) \tag{16}$$

$$w_{ij} \geq 0 \qquad \forall\,(i,j) \tag{17}$$

and (2), (8).

Our next step toward obtaining a lower bound on the cost of the full problem is to linearize constraints (16) (by using a logical constraint). We use the new variables w_{ij}^{sd} (whose dimension is seconds per bit) for each link (i,j) on path (s,d). Thus we have the equivalent problem:

$$Z_{CFA} = min\Big[\sum_{i,j} \frac{d_{ij}}{w_{ij}} + \sum_{i,j}\sum_{s,d} d_{ij}\delta_{ij}^{sd}\gamma_{sd}\Big]$$

subject to:

$$w_{ij}^{sd} \leq M_{sd}\delta_{ij}^{sd} \qquad \forall\,(i,j,s,d) \tag{18}$$

$$K_1 \sum_{i,j} w_{ij}^{sd} + K_2 \sum_{i,j} \delta_{ij}^{sd} d_{ij} \leq RTT_{sd} \qquad \forall\,(s,d) \tag{19}$$

$$w_{ij}^{sd} \geq 0 \qquad \forall\,(i,j,s,d) \tag{20}$$

and (2), (8), (17).

Note that constraints (18) force the packet delay of link (i,j) on path (s,d) to be 0 if the link is not used. The constant M_{sd} corresponds to the minimum value of w_{ij}^{sd} that is able to satisfy

the packet delay constraints for path (s,d). We have $M_{sd} = RTT_{sd}/K_1$. We refer to the this problem as problem P in the rest of this paper.

Feasible solutions as well as lower bounds for the optimal solution of problem P, can be obtained by using Lagrangean relaxation. First, constraints in (18) and (19) are relaxed, and the corresponding Lagrangean problem is constructed; next, a sub-gradient optimization procedure [24] is used in order to improve the quality of the Lagrangean lower bound.

The Lagrangean relaxation is motivated by two objectives: first, to derive a Lagrangean subproblem that is easy to solve (ideally, solvable in polynomial time); second, to minimize the number of dualized constraints. There are two obvious reasons to aiming at this last objective: we want a Lagrangean subproblem that is as close as possible to the original formulation, and we also wish to minimize the number of Lagrangean multipliers.

The complexity of the original problem, together with the fact that the Lagrangean subproblems have a simple structure, and that the sub-gradient procedure is very effective in narrowing the gap between the lower and the upper bound, justify the use of the Lagrangean technique in this case.

Consider the Lagrangean relaxation of problem P obtained by dualizing constraints (18) and (19) using the nonnegative multipliers α_{ij}^{sd} and β_{sd}, respectively.

$$\begin{aligned} L(\alpha,\beta) \quad = min \Bigg\{ & \sum_{i,j} \frac{d_{ij}}{w_{ij}} + \sum_{i,j} \sum_{s,d} d_{ij}\delta_{ij}^{sd}\gamma_{sd} \\ & + \sum_{s,d} \sum_{i,j} \alpha_{ij}^{sd} \left[(w_{ij}^{sd} - M_{sd}\delta_{ij}^{sd}) \right] \\ & + \sum_{s,d} \beta_{sd} \left[\sum_{i,j} (K_1 w_{ij}^{sd} + K_2 \delta_{ij}^{sd} d_{ij}) - RTT_{sd} \right] \Bigg\} \end{aligned} \tag{21}$$

subject to (2), (8), (17) and (20).

Problem $L(\alpha,\beta)$ can now be decomposed into two independent subproblems as follows:
Subproblem $L_1(\alpha,\beta)$:

$$L_1(\alpha,\beta) = min \sum_{s,d} \sum_{i,j} \delta_{ij}^{sd} (d_{ij}\gamma_{sd} - M_{sd}\alpha_{ij}^{sd} + K_2 d_{ij}\beta_{sd}) \tag{22}$$

subject to (2) and (8).

Subproblem $L_2(\alpha,\beta)$:

$$\begin{aligned} L_2(\alpha,\beta) = \quad & min \sum_{i,j} \left(\frac{d_{ij}}{w_{ij}} + \sum_{s,d} w_{ij}^{sd} (\alpha_{ij}^{sd} + K_1\beta_{sd}) \right) \\ & - \sum_{s,d} \beta_{sd} RTT_{sd} \end{aligned} \tag{23}$$

subject to (17) and (20).

Subproblem $L_1(\alpha,\beta)$ can be further decomposed into $n*(n-1)$ shortest path problems (one for each source/destination pair) and solved using the classic Bellman-Ford's algorithm.

To be able to solve problem $L_2(\alpha,\beta)$, we decompose it into m independent subproblems (one for each link):

$$L_2^{(i,j)}(\alpha,\beta) = min \left(\frac{d_{ij}}{w_{ij}} + w_{ij} \sum_{s,d} y_{ij}^{sd} (\alpha_{ij}^{sd} + K_1\beta_{sd}) \right) \tag{24}$$

where the variables y_{ij}^{sd} can be seen as estimates of the network routing. Subproblem $L_2^{(i,j)}(\alpha,\beta)$ is minimized by minimizing $\sum_{s,d} y_{ij}^{sd}(\alpha_{ij}^{sd} + K_1\beta_{sd})$. It is straightforward to see that at least one variable y_{ij}^{sd} must be 1, for all (s,d); otherwise w_{ij} tends to infinity. The solution consists in set $y_{ij}^{sd} = 1$ for (s,d) of minimum $(\alpha_{ij}^{sd} + K_1\beta_{sd})$, and $y_{ij}^{sd} = 0$ otherwise. Then, the optimum values of w_{ij} are given by:

$$w_{ij}^* = \sqrt{\frac{d_{ij}}{\sum_{s,d} y_{ij}^{sd} (\alpha_{ij}^{sd} + K_1\beta_{sd})}} \tag{25}$$

B. Solving the Lagrangean dual problem

The Lagrangean dual problem typically produces solutions that after recovering primal feasibility tend to be close to optimal. Like for all relaxation procedures, the success of the approach depends heavily on the ability to generate good Lagrangean multipliers. In order to solve the Lagrangean dual problem, we employ a sub-gradient algorithm to search for "good" multipliers, while to recover primal feasibility we employ a heuristic.

The value of the Lagrangean for any set of multipliers $\upsilon = (\alpha,\beta)$ will be equal to the sum of the optimal solutions to the subproblems, $L(\upsilon) = L_1(\upsilon) + L_2(\upsilon)$. It is well known from optimization theory, by using the weak Lagrangean duality theorem [25], that for any vector of multipliers, $L(\upsilon)$ is a lower bound for the objective function value of the original problem, i.e., $L(\upsilon) \leq Z_{CFA}$; $\forall\ \upsilon \geq 0$. We are interested in obtaining the tightest possible lower bound, i.e., in the multipliers vector υ^*, that corresponds to $L(\upsilon^*) = max_\upsilon\{L(\upsilon)\}$ (the Lagrangean Dual Problem).

C. Sub gradient optimization procedures

In order to solve the dual problem, the sub gradient method is used to update the multiplier υ. Thus, given υ, once the solutions to the subproblems $L_1(\upsilon)$ and $L_2(\upsilon)$ are obtained, a dual sub gradient, $\xi = \{\xi_{ij}^{sd}\}$, is computed using:

$$\xi_{ij}^{sd} = \left(\sum_{i,j} (K_1 w_{ij}^{sd} + K_2 \delta_{ij}^{sd} d_{ij}) - RTT_{sd}, w_{ij}^{sd} - M_{sd}\delta_{ij}^{sd} \right) \tag{26}$$

The subsequent value of the Lagrangean multiplier is updated as follows:

$$\upsilon^{p+1} = max\{0, \upsilon^p + t^p \xi^p\} \tag{27}$$

where t^p and ξ^p are the step size and the sub-gradient, respectively, at the pth iteration.

The step size is defined by [24]:

$$t^p = s^p \frac{\overline{Z_{FCA}} - L(\upsilon^p)}{\| \xi^p \|^2} \tag{28}$$

where $\overline{Z_{FCA}}$ is the value of the best feasible solution found so far, and the relaxation parameter s^p is a scalar between 0 and 2.

Roughly speaking, υ^p acts as a penalty for the violation of the relaxed constraints. If constraints are not violated by the current solution w^p,δ^p, then υ^p is decreased according to the current step t^p and the value of ξ^p, otherwise the penalty υ^p is increased by the amount $t^p\xi^p$. At termination, the sub-gradient algorithm reports $L(\upsilon^*)$ as the highest lower bound.

Crucial to the effectiveness of the algorithm are its parameters and the schedule for decreasing t^p. In particular, if the value of $L(\upsilon^p)$ remains approximately the same over several iterations, the parameter values are decreased. The number of iterations before decrease and the percentage decrease are the values to be selected or "tuned". The sub-gradient procedure is started with $s^p = 2$. If $L(\upsilon^p)$ has not given any improvement of the lower bound in $MaxImp$ iterations, we let $s^p = s^p/2$. When a better lower bound is obtained, we reset $s^p = 2$.

Since no monotone increase of $L(\upsilon^p)$ is likely to occur, we save the best lower bound, $L(\upsilon^*)$, as the maximal objective function value obtained for the dual problem.

Note the importance of the upper bound $\overline{Z_{CFA}}$, since a too large value of $\overline{Z_{CFA}}$ will make the steps too long, and hence slow down the convergence. Therefore, we update the value of $\overline{Z_{CFA}}$ using feasible solutions from a Primal Heuristic algorithm (see the next subsection).

Ideally, the procedure terminates when $\| \xi^p \|^2 = 0$, which indicates that the dual optimum is found, and that the solution is primal feasible. However, this is unlikely to occur in most cases, since the Lagrangean dual problem is solved approximately, and a duality gap may exist. In practice, there are several stopping criteria that may be used to terminate the algorithm. If $\| \xi^p \|^2 \leq \epsilon$ or $t^p \leq \epsilon$, for some very small $\epsilon > 0$, the algorithm should stop, since v^p is not varying enough. We also use a maximum number of iterations, $MaxItr$, as stopping criterion.

For the forthcoming analysis, the parameters were set as follows: v^0 is a random number between 0.1 and 10, $MaxImp = 20$, $MaxItr = 500$, and $\epsilon = 10^{-3}$.

D. Obtaining feasible solutions

Because of the used decision variables and stopping criterion, the solution to the dual problem is generally associated with an infeasible project, i.e., some of the end-to-end packet delay constraints and/or routing constraints may be violated.

To construct a feasible solution to the CFA problem, we use as a basis the trial solutions to the Lagrangean problem obtained at each of the iterations of the sub-gradient procedure (Primal Heuristic). At each iteration, we test if the routing obtained from the solution of subproblem $L_1(v)$ can generate a feasible solution to the primal problem P. The test corresponds to verifying whether RTT is strictly greater than $K_2 \sum_{i,j} \delta_{ij}^{sd} d_{ij}$ for all source/destination pairs; in this case the values for w_{ij}^{sd} can be obtained. The algorithm stops if no feasible solution can be found, so that the requirements of the problem must be relaxed.

Therefore, given the routing, a capacity assignment solver provides the values for the C_{ij}. We apply two techniques to solve the problem: i) a fast approximation solution which is described in the next subsection (it gives a fast upper bound for the CFA problem); and ii) a second approach using the logarithmic barrier method [22] (it gives a solution whose accuracy is a priori known). Consequently, the value of the primary objective function can be obtained. As the iteration progresses, we check for decreases in the primal objective value, and store the best primal solution, and the best primal cost so far.

E. Approximate solution to the CA problem

If we assume that the routing is known, problem P reduces to the following capacity assignment (CA) problem:

$$Z_{CA} = min\Big[\sum_{i,j} \frac{d_{ij}}{w_{ij}} + \sum_{i,j} \sum_{s,d} d_{ij} \delta_{ij}^{sd} \gamma_{sd}\Big]$$

subject to:

$$\sum_{i,j} w_{ij}^{sd} \leq b_{sd} \qquad \forall\,(s,d) \tag{29}$$

$$w_{ij}^{sd} \geq 0 \qquad \forall\,(i,j,s,d) \tag{30}$$

where:

$$b_{sd} = \frac{1}{K_1}\big(RTT_{sd} - K_2 \sum_{i,j} \delta_{ij}^{sd} d_{ij}\big) \qquad \forall\,(s,d) \tag{31}$$

We can obtain a simple approximate solution to the CA problem in the following way. First, for each path (s,d) we construct the Lagrangean:

$$L(\psi) = min\Big[\sum_{i,j} \frac{d_{ij}^{sd}}{w_{ij}^{sd}} + \psi\big(\sum_{s,d} w_{ij}^{sd} - b_{sd}\big)\Big] \tag{32}$$

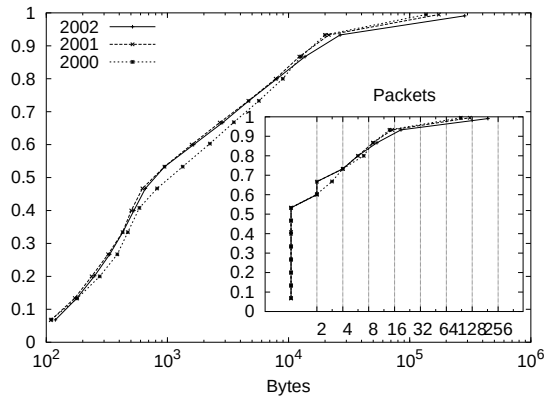

Fig. 3. TCP connection length cumulative distribution

subject to (30). The solutions to this problem are given by:

$$w_{ij}^{sd} = \frac{b_{sd}\sqrt{d_{ij}^{sd}}}{\sum_{k,l} \sqrt{d_{kl}^{sd}}} \qquad \forall\,(s,d) \tag{33}$$

Second, since a link can be used by several paths, the following expression is used to obtain admissible values for the variables w_{ij}.

$$w_{ij} = \min_{s,d}\{w_{ij}^{sd}\} \tag{34}$$

Finally, the capacities are computed using $C_{ij} = \frac{1}{w_{ij}} + f_{ij}$.

V. Numerical Results

In order to prove the effectiveness of the design methodology, we run a large number of numerical experiments and computer simulations. Some of the results are briefly presented and discussed in this section.

We consider a mixed traffic scenario where the file size follows the distribution shown in Fig. 3, which is derived from one-week long measurements [17] in three different time periods. In particular, we report the discretized Cumulative Distribution Function (CDF), obtained by splitting the flow length distribution in 15 groups with the same number of flows per group, from the shortest to the longest flow, and then computing the average flow length in each group. The large plot reports the discretized CDF using bytes as unit, while the inset reports the same distribution, taking today's most common MSS of 1460 bytes as unit.

We present results obtained considering several topologies, which have been generated using the BRITE topology generator [26] with the router level option. Random traffic matrices were generated by picking the traffic intensity of each source/destination pair from a uniform distribution. For each topology, we solved both the CFA and BA problems using the approach described in previous sections. Simulation experiments at the packet level were run using the ns-2 simulator.

A. 10-Node Networks

In this section, we present results obtained considering a 10-nodes, 20-links network topology. In the design we considered the following target QoS constraints for all source/destination pairs: i) file latency $L_t \leq 0.4$ s for TCP flows shorter than 20 segments, ii) throughput $T_h \geq 512$ Kbps for TCP flows longer than 20 segments. Selecting $P_{loss} = 0.01$, we obtain a network-level design constraint equal to $RTT \leq 0.052$s (see Fig. 2) for all source-destination pairs. Each traffic relation offers an average aggregate traffic equal to $\hat{\gamma}_{sd} = 1$Mbps. Link propagation delays ranges

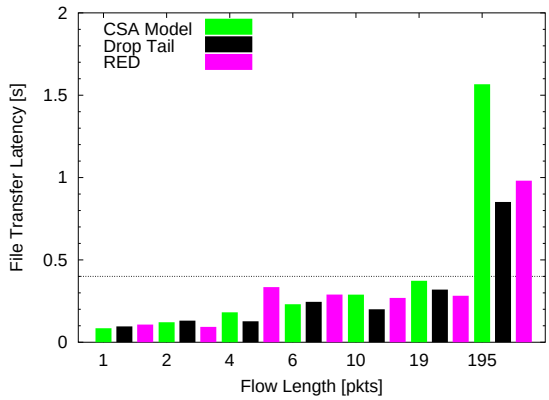

Fig. 4. Model and simulation results for latency; 5-link path from the 10-node network

10-Node Network				
	$M_{[X]}/M/1/B$		ns-2 (RED)	
$Link$	$E[T]$	P_{loss}	$E[T]$	P_{loss}
1	0.006	0.0031	0.007	0.0023
2	0.008	0.0015	0.010	0.0016
3	0.008	0.0018	0.010	0.0018
4	0.006	0.0018	0.008	0.0016
5	0.009	0.0016	0.012	0.0038
tot	0.037	0.0098	0.047	0.0111

TABLE I

MODEL AND SIMULATION RESULTS: 5-LINK PATH

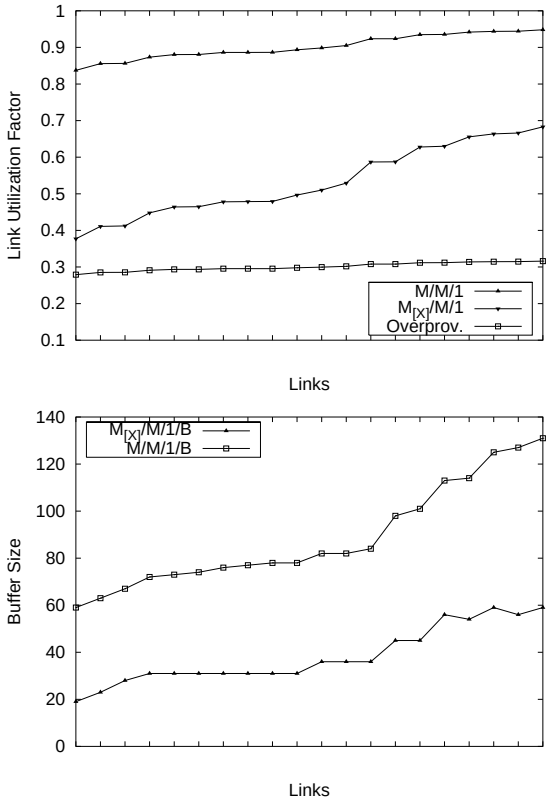

Fig. 5. Link utilization factor and Buffer size for the 10-node network.

from 0.25 ms to 1.5 ms, i.e., link lengths vary between 50 km and 300 km. After solving the CFA problem, we solved the BA problem in both the DropTail and RED cases.

To verify the accuracy of the IP network design produced by the methodology, we performed packet-level simulations to check whether the QoS constraints are actually met. In the experiments we assumed that TCP New Reno is adopted, and that TCP connections are established at instants described by a Poisson process, choosing at random a server-client pair. Connection opening rates are determined so as to meet the offered traffic, γ_{sd}. The amount of data to be transferred by each TCP connection (i.e., the file size) is expressed in number of packets according to the measured values. We performed *path simulations* rather than simulating the entire network, i.e., we selected a path referring to a single source/destination pair, and simulated only links in that path, considering also interfering cross traffic. This approach is necessary due to scalability problems in ns-2, which did not allow us to simulate the entire topology. Moreover, results obtained in path simulations are worst-case results with respect to entire network simulations, because cross traffic is more aggressive, since it is directly injected into the simulated path, without traversing all links along its path, hence not suffering losses or shaping.

Among all possible source/destination pairs, we selected the longest path in the network, which comprises 5 links. Results are plotted in Fig. 4, which reports the file transfer latency for all flow size classes. The QoS constraint of 0.4s for the maximum latency is also shown. We can clearly see that model predictions and simulation results are in perfect agreement with specifications, since the latency constraint is satisfied for all flows shorter than 20 segments. The flow transfer latency constraint for mice is more stringent than the throughput constraint for elephants, represented by 195 packet long flows, therefore the throughput of the latter is 2.2 Mbps in the RED case, instead of the minimum desired 512 kbps. Notice that the predicted throughput obtained by applying the CSA model is a pessimistic estimate. This is due to the limit in the CSA model itself, and not to a mismatch in the network-layer parameters between model and simulation. Indeed, Table I reports the average packet delay $E[T]$, and the average packet loss probability P_{loss} predicted by the $M_{[X]}/M/1/B$ queueing model and measured in simulations (with RED buffers), for the selected 5-link path. As it can be observed, there is a very good match between model predictions and simulation results.

To complete the evaluation of our methodology, we compare the link utilization factor and buffer size obtained when considering the classical $M/M/1$ queueing model instead of the $M_[X]/M/1$ model. Fig. 5 shows the link utilizations (top plot) and buffer sizes (bottom plot) obtained with our method and with the classical model. It can be immediately noticed that considering the burstiness of IP traffic radically changes the network design. Indeed, the link utilizations obtained with our methodology are much lower than those produced by the classical approach, and buffers are much longer.

It is important to observe that the test of the QoS perceived by end users in a network dimensioned using the classical approach cannot be performed, since simulations fail even to run, because the dropping probability experienced by TCP flows is so high that retransmissions cause the offered load to become larger than 1 for some links. These means that the network designed with the classical approach is not capable of supporting the offered load and therefore cannot satisfy the QoS constraints.

In addition in Fig.5, we also compare our results to those of an overprovisioned network, in which the capacities obtained by using the traditional $M/M/1$ model are multiplied a posteriori by the minimum factor which allows the QoS constraints to be met. The overprovisioning factor was estimated by a trial and error procedure based on path simulations at the packet level. Since it is difficult to define an overprovisioning factor for the BA problem, we fixed a priori the buffer size to be equal to 150. The final overprovisioned network is capable of satisfying the QoS constraints, but a larger cost is incurred, which is directly proportional to the increase in link capacities. Note also that the heuristic used to find the minimum overprovisioning factor can not be applied for

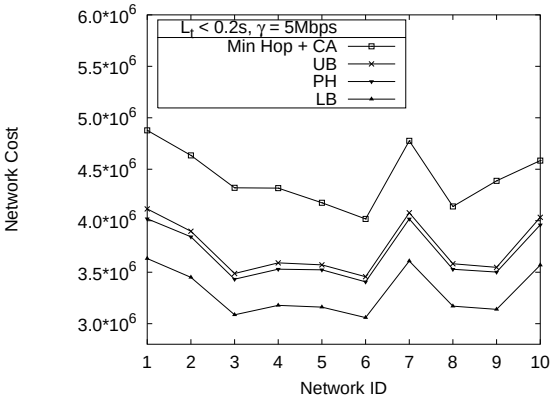

Fig. 6. Network cost for 40-node networks with random topologies

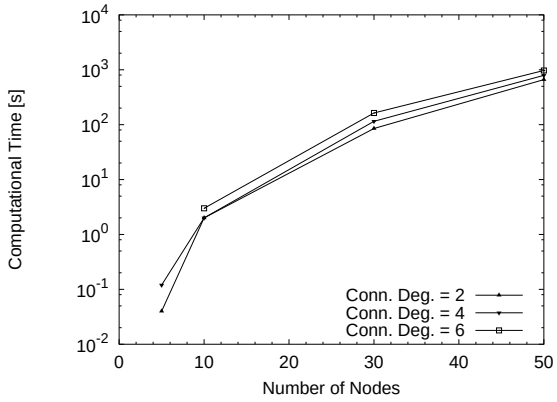

Fig. 8. Computation times for the CFA problem

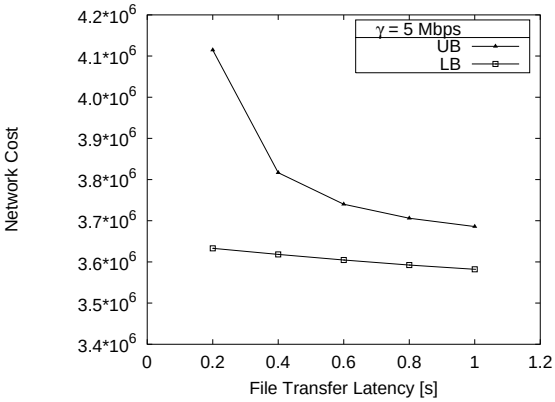

Fig. 7. Network cost versus latency (40-node, 160-link network)

large/high-speed networks, due to scalability problem of packet level simulators.

B. 40-Node Networks

In this section we present results for 40-node, 160-link network topologies where link propagation delays are uniformly distributed between 0.5 and 1.5 ms, i.e., where link lengths vary between 100 and 300 km.

Two sets of experiments were performed. In the first set of experiments, we compare the results obtained with four different techniques: i) Lagrangean relaxation (LB), ii) primal heuristic with logarithmic barrier CA solution (PH), iii) primal heuristic with approximate CA solution (UB), iv) CA with minimum-hop routing (MinHop). Results for 10 random topologies are presented in Fig. 6. The average source/destination traffic requirement is set to $\gamma = 5$ Mbps. For all source/destination pairs, the target QoS constraints are: i) latency $L_t \leq 0.2$ s for TCP flows shorter than 20 segments, ii) throughput $T_h \geq 512$ kbps for TCP flows longer than 20 segments, iii) $P_{loss} = 0.001$. Using the transport-layer QoS translator, we obtain the equivalent network-layer performance constraint $RTT \leq 0.032$ s, which derives from the most stringent latency constraint ($L_t \leq 0.2$ s for 20-segment flows).

First of all, the results allow us to conclude that ignoring the routing optimization when solving the CA problem (MinHop) leads to poor results.

In addition, we can observe that the feasible solutions (PH) and sub optimum solutions (UB) for all considered topologies always fall rather close to the lower bound (LB). The gap between UB and LB is about 16%. Using the PH solution, the gap is reduced to 13%.

The second set of experiments aimed at investigating the impact of the latency constraints on the optimized network cost. Fig. 7 shows the LB and UB values for latency constraint values ranging from 0.2 to 1.0 s. The plots clearly show the trade off between cost and latency; as expected, costs grow when the latency constraints become tighter. It is interesting to observe that when the latency constraints become very tight (latencies become close to zero), the sensitivity of the network cost increases.

C. Complexity

Finally, we briefly discuss the computation times needed to solve the CFA problem. The optimization algorithms (the sub-gradient algorithm and the heuristic) were implemented in C language and run on a workstation with a 1GHz processor running Linux. The computation times (in CPU seconds) for several CFA problems are presented in Fig. 8. We tested our approach on networks with different numbers of nodes and different connection degrees (number of ingoing/outgoing links in a node). As can be seen, CPU times range from less than 1 second to more than 15 minutes.

It is straightforward to obtain the time complexity of the sub-gradient algorithm delineated in section IV-C. At each iteration it is necessary to solve subproblem $L_1(v)$; this requires $O(n^3 m)$ operations, since $n(n-1)$ shortest paths are found using the Bellman-Ford algorithm of complexity $O(nm)$. If the number of iterations, $MaxItr$, is the only stopping criterion used, the resulting time complexity is $O(n^3 m\ MaxItr)$.

VI. Conclusions

In this paper, we have considered the the QoS design of packet networks, and in particular the joint Capacity and Flow Assignment problem where both the routing assignments and capacities are considered to be decisions variables. Our new formulation to the CFA problem differs in two important points from previous formulations. First, the novelty of our approach is that it considers end-to-end QoS constraints for all source/destinations pairs on the network. A second important improvement with respect to earlier approaches is the use of a refined IP traffic modeling technique that provides an accurate description of the traffic dynamics in multi-bottleneck networks loaded with TCP mice and elephants. By explicitly considering TCP traffic, we also need to consider the impact of finite buffers, therefore facing the Buffer Assignment problem.

We have formulated the problem as a nonlinear mixed-integer programming problem. A Lagrangean Relaxation approach was used to obtain both lower bounds and feasible solutions in the VPN case. A sub-gradient method was used to find the optimal Lagrangean multipliers. Numerical results suggest that the proposed methodology provides a quite efficient approach to obtain near-optimal solutions with small computational effort.

Examples of application of the proposed design methodology to different networking configurations have been discussed. The

network target performances are validated against detailed simulation experiments, proving that the proposed design approach is both accurate and flexible.

REFERENCES

[1] **Kleinrock, L., "Queueing Systems, Volume II: Computer Applications", *Wiley Interscience*, New York, 1976.**

[2] **Gerla, M., L. Kleinrock, "On the Topological Design of Distributed Computer Networks", *IEEE Transactions on Communications*, Vol.25, pp.48-60, Jan. 1977.**

[3] **V.Paxson, S.Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling,"** *IEEE/ACM Transactions on Networking*, **Vol.3, N.3, pp.226–244, Jun. 1995.**

[4] **B.Gavish, I.Neuman, "A system for routing and capacity assignment in computer communication networks", *IEEE Transactions on Communications*, Vol.37, N.4, pp.360–366, April 1989.**

[5] **B.Gavish, S.Hantler, "An Algorithm for Optimal Route Selection in SNA Networks", *IEEE Transactions on Communications*, Vol.31, N.10, pp1154–1161, Oct. 1983.**

[6] **K.T.Cheng, F.Y.S.Lin, "Minimax End-to-End Delay Routing and Capacity Assignment for Virtual Circuit Networks", *IEEE Globecom 95*, Singapore, pp. 2134-2138, Nov. 1995.**

[7] **E. Rolland, A. Amiri, R. Barkhi, "Queueing Delay Guarantees in Bandwidth Packing", In *Computers and Operations Research*, Vol.26, pp. 921-935, 1999.**

[8] **A.Gersht, R.Weihmayer, "Joint optimization of data network design and facility selection", *IEEE Journal on Selected Areas in Communications*, Vol.8, N.9, pp.1667-1681, Dec. 1990.**

[9] **M.Garetto, D.Towsley, "Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic", *ACM SIGMETRICS 2003*, San Diego, CA, June 2003.**

[10] **S.Floyd, V.Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Transactions on Networking*, Vol.1, N.4, pp.397–413, 1993.**

[11] **H.Pirkul, A.Amiri, "Routing in Packet-switched Communication Networks", *Computer Communications*, Vol.17, N.5, pp.307-316, May 1994.**

[12] **T.Ng, D.Hoang, "Joint Optimization of Capacity and Flow Assignment in a Packet Switched Communication Network", *IEEE Transaction on Communications*, Vol.35, N.2, pp.202-209, Feb. 1987.**

[13] **D.Medhi, D.Tipper, "Some approaches to solving a multi-hour broadband network capacity design problem with single-path routing", *Telecommunication Systems*, Vol.13, N.2, pp.269-291, 2000.**

[14] **C.Fraleigh, F.Tobagi, C.Diot, "Provisioning IP Backbone Networks to Support Latency Sensitive Traffic", *IEEE Infocom 03*, San Francisco, CA, Mar 2003.**

[15] **H.Knoche, H.de Meer, "Quantitative QoS Mapping: A Unifying Approach", *5th Int. Workshop on Quality of Service (IWQoS97)*, New York, NY, pp.347–358, May 1997.**

[16] **A.Markopoulou, F.Tobagi, M.Karam, "Assessment of VoIP Quality over Internet Backbones" *IEEE Infocom 02*, New York, NY, June 2002.**

[17] **Reference removed to guarantee authors anonymity.**

[18] **N.Cardwell, S.Savage, T.Anderson, "Modeling TCP Latency", *IEEE Infocom 00*, Tel Aviv, IS, March 2000.**

[19] **J.Padhye, V.Firoiu, D.Towsley, J.Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation", *N*etworking, IEEE/ACM Transactions on, Vol.8, N.2, pp.133–145, Apr. 2000.**

[20] **X.Chao, M.Miyazawa, M.Pinedo, *Queueing Networks, Customers, Signals and Product Form Solutions*, John Wiley, 1999.**

[21] **R.Nagarajan, D.Towsley, "Note on the Convexity of the Probability of a Full Buffer in the M/M/1/K queue", *CMPSCI Technical Report TR 92-85*, Aug. 92.**

[22] **M.Wright, "Interior methods for constrained optimi- zation", *Acta Numerica*, Vol.1, pp.341–407, 1992.**

[23] **B.Gendron, T.G.Crainic, A.Frangioni, "Multicommodity Capacitated Network Design". B. Sans ò and P. Soriano (eds), *Telecommunications Network Planning*, pp.1–19, Kluwer, MA, 1998.**

[24] **M.Held, P.Wolde, H.Crowder, "Validation of subgradient optimization", *Mathematical Programming*, Vol.6, pp. 62-88, 1974.**

[25] **A.M. Geoffrion, "Lagrangean relaxation and its uses in integer programming", *Mathematical Programming Study*, Vol.2, pp.82–114, 1974.**

[26] **A.Medina, A.Lakhina, I.Matta, J.Byers, "BRITE: Boston university representative internet topology generator", Boston University, http://cswww.bu.edu/brite, April 2001.**