Chapter 2

DESIGN OF IP VIRTUAL PRIVATE NETWORKS UNDER END-TO-END QOS CONSTRAINTS

Emilio C.G. Wille
Marco Mellia
Emilio Leonardi
Marco Ajmone Marsan

Abstract Traditional approaches to optimal design and planning of packet networks focus on the network-layer infrastructure. The next generation Internet will be faced with problems concerning end-to-end Quality of Service and Service Level Agreement guarantees. In this chapter, we propose a new packet network design and planning approach, for Virtual Private Networks, that is based on user-layer QoS parameters. Our proposed approach maps the end-user performance constrains 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. Examples of application of the proposed design methodology to different networking configurations show the effectiveness of our approach.

1. Introduction

The pioneering works of Kleinrock (1976) spurred many research activities in the field of optimal design and planning of packet networks, and a vast literature is available on this subject. Almost invariably, however, packet network design focused on the network-layer infrastructure, so that the designer is faced with a trade-off between total cost and average performance (network-wide packet delay, packet loss ratio, link utilization, network reliability, etc.). This approach adopts the viewpoint of network operators, who quite naturally aim at the optimization of some aggregate performance measure, that describe the general behav-

ior of their network, averaging over all traffic relations. 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.

Today, with the enormous success of the Internet, packet networks have reached their maturity and they are used for very critical services. Accordingly, researchers as well as operators are concerned with end-to-end Quality of Service (e2e QoS) issues and Service Level Agreement (SLA) guarantees for IP networks. In this new context, average network-wide performance cannot be taken as the sole metric for network design and planning any longer, specially in the case of corporate virtual private network (VPN).

From the end user's point of view, QoS is driven by end-to-end performance parameters, such as data throughput, web page latency, transaction reliability, etc. Matching the user-layer QoS requirements to the network-layer performance parameters is not a straightforward task. Indeed, the QoS perceived by end users in their access to Internet services is mainly driven by TCP, the reliable transport protocol of the Internet, whose congestion control algorithms dictate the latency of information transfer. Indeed, it is well known that TCP accounts for a great amount of the total traffic volume in the Internet, and among all the TCP flows, a vast majority is represented by short-lived flows (also called mice), while the rest is represented by long-lived flows (also called elephants); see for example: Gribble and Brewer (1977), Claffy et al. (1998), Mellia et al. (2002).

In this chapter, we propose for the first time (to the best of our knowledge) a packet network design and planning approach that is based on user-layer QoS parameters and explicitly accounts for the impact of the TCP protocol [1].

Our proposed approach maps the end-user performance constraints into transport-layer performance constraints first, and then into network-layer performance constraints. The mapping process is then considered together with a realistic representation of traffic patterns at the network layer to design the IP network.

The representation 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, see Paxson and Floyd (1995), but a higher degree of correlation exists, which can be partly due to the TCP control mechanisms. This means that the usual approach of

[1] Fraleigh et al. (2003) account for user-layer QoS constraints focus mainly on voice traffic, and do not consider the impact of TCP at the transport layer.

modeling packet networks as networks of M/M/1 queues as discussed in Gavish and Neuman (1989), Kamimura and Nishino (1991), Cheng and Lin (1995), Gavish (1992), Mai Hoang and Zorn (2001) is not acceptable. In this chapter we adopt a refined IP traffic modeling technique, already presented in Garetto and Towsley (2003), that provides an accurate description of the traffic dynamics in multi-bottleneck IP networks loaded with TCP mice and elephants. The resulting analytical model is capable of producing accurate performance estimates for general topology IP networks loaded by realistic TCP traffic patterns, while still being analytically tractable.

In summary, in this chapter we propose a new approach to the packet network design problem, which considers as constraints the e2e QoS perceived by users. Given (i) the network topology, (ii) the average traffic exchanged by all source/destination pairs (i.e., the traffic matrix), (iii) a routing algorithm (e.g., shortest path), we solve the capacity assignment problem, minimizing the link capacity cost, subject to the e2e QoS constraints expressed by users, i.e., either the average data throughput, or the file transfer latency, obtained by considering TCP as the transport protocol. In addition, our approach is capable of also solving either the droptail buffer dimensioning problem, or the AQM (Active Queue Management) parameter dimensioning problem in the case of AQM buffers (e.g., RED). While the buffer cost is usually considered to be negligible, it is important to have a procedure to dimensioning the correct buffer size, to limit the impact of queueing delay on the performance. Moreover, the availability of buffers in high-capacity router is limited by the cost of high-speed static RAM.

The rest of the chapter is organized as follows. Section 2 describes the general network design methodology. The e2e QoS mapping into transport- and network-layer performance constraints, and some translations examples, are described in Section 2.1. Section 3 provides the formulation of the general optimization problem, and lists the assumptions needed for the modeling phase. Afterwards, the Capacity Assignment (CA) and the Buffer Assignment (BA) problems are presented. Results obtained for both problems are tabulated and compared with results of *ns*-2 simulations in Section 4. Conclusions are given in Section 5.

2. The IP network design methodology

Of course, in any realistic network problem the notion of "optimum design" is an extremely difficult task. The IP network design methodology that we propose in this chapter is based on a "Divide and Conquer" approach, in the sense that it consists of several subproblems. Thus, the

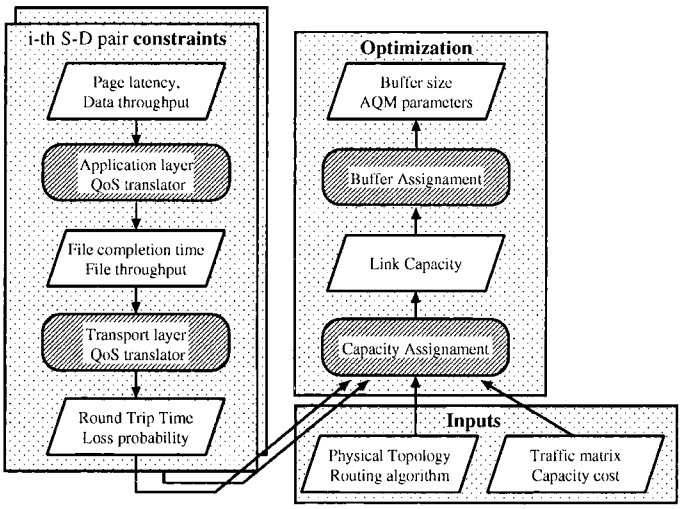


Figure 2.1. Schematic flow diagram of the network design methodology

subproblems are solved separately in a way to obtain a heuristic solution to the general problem.

Figure 2.1 shows the flow diagram of the design methodology. Shaded, rounded boxes represent function blocks, while white parallelograms represent input/output of functions. There are three 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, the routing algorithm, and the expression of the cost as a function of the 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 and buffer sizes (or AQM parameters).

In our methodology we decouple the CA problem from the BA problem. The optimization starts then with the CA 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.

2.1 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 (see Knoche and de Meer, 1997 and the references therein). 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}).

Matching the user-layer QoS requirements to the network-layer performance parameters is not a straightforward task. In this section we present some examples of QoS constraints translation and propose a new QoS translator tailored for the TCP protocol case.

2.1.1 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).

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 in Markopoulou et al. (2002). 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.

Example – Web page download. In this case, the input of the application-layer QoS translator is a desired download time, expressed 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 1.5s, 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.

2.1.2 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.

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. Jitter and delay variation may also be considered. 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.

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 (L_t), or the minimum file transfer throughput (T_h). 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 over the Internet, presented in Mellia et al. (2002), it is possible to say that 85% of all TCP flows are shorter than 20 segments. For these flows, we im-

pose 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.

To solve the translation problem, we exploit recent research results in the field of TCP modeling (see Garetto and Towsley, 2003 and the references therein). Usually, TCP models take network-layer parameters as inputs, i.e., RTT and packet loss probability, 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 P_{loss}. Among the many models of TCP presented in the literature, when considering file transfer latency, we use the TCP latency model described in Cardwell et al. (2000), 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, from Padhye et al. (2000). Our methodology can however be modified to incorporate more complex/accurate TCP models.

The inversion of TCP models is not simple, since there are at least two 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 CA problem from the BA problem. 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 Figure 2.2 report the maximum admissible RTT which satisfies the most stringent constraint for different values of P_{loss}.

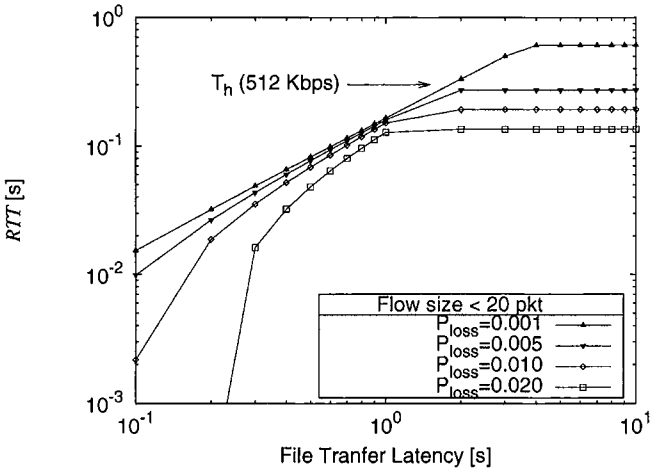


Figure 2.2. *RTT* constraints as given by the transport layer QoS translator

3. Optimization formulation and solutions

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 the capacity is 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 here can be applied to both contexts, as well as others, in this chapter we concentrate on the design of corporate VPNs. The reason we consider VPNs is that we must apply a specific optimization technique for each type of problem.

Several different formulations of the packet network design problem can be found in the literature; generally, they correspond to different choices of performance measures, of design variable, and of constraints. Here, we consider the following general problem:

Given: physical topology, routing algorithm, traffic estimates between node pairs, capacity and buffer costs;

Minimize: total capacity cost, total buffer cost;
With respect to: link capacities, buffer sizes;
Subject to: packet delay constraints, packet loss probability.

In the solution of the CA and BA problems, we need to evaluate the packet delay and loss probability to verify that the constraints are met. We thus first introduce the network model and discuss the relations between performance measures, input parameters, design variables, and constraints that appear in the general design problem. Then we define and solve the CA and BA problems.

3.1 Traffic model

The network model is an open network of queues, where each queue represents an 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 the description of the network model, we assume that all router buffers exhibit a droptail behavior.

Traditionally, $M/M/1/B$ queueing models were considered good representations of packet networks. However, given the well-known correlation of actual IP traffic, we choose to model the increased traffic burstiness induced by TCP using the arrival of packets in groups (batch arrivals), hence using $M_{[X]}/M/1/B$ 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, as discussed in Garetto and Towsley (2003). Our 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 TCP traffic. The decision to model the router output interfaces with $M_{[X]}/M/1/B$ queues is the results of a careful and detailed study of a wide gamut of performance investigations of queue lengths in IP networks, conducted with the *ns*-2 simulator in Garetto and Towsley (2003).

The Markovian assumption for the batch arrival process is mainly due to 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. The average packet loss probability, and the average time spent by packets in the router buffer, are obtained directly from the solution of the $M_{[X]}/M/1/B$ queue. Given the flow length distribution, a stochastic model of TCP

(described in Garetto and Towsley, 2003) is used to obtain the batch size distribution $[X]$.

3.2 Delay analysis

The packet length is assumed to be exponentially distributed with mean $1/\mu$, the transmission time for each packet over a link is $1/\mu C$, and thus the utilization factor is given by $\rho = \lambda/\mu C$, (C is the link capacity, and $f = \lambda/\mu$ is the average data flow on link). The average packet delay in the $M_{[X]}/M/1/\infty$ queue is given in Chao et al. (1999):

$$E[T] = \frac{K}{\mu}\frac{1}{C-f} \tag{2.1}$$

where $K = \frac{m'+m''}{2m'}$, being m' and m'' the first and second moments of the batch size distribution [X].

3.3 Network model, traffic, and routing

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 and E is a set of edges. A node represents a router, and an edge represents a physical link connecting one router to another. For each link l we consider: C_l, the capacity of the link; f_l, the average data flow; d_l, the physical length; and B_l, the buffer size.

Different formulations of the CA problem result by selecting i) the cost functions $f_l(C_l)$, ii) the routing model, and iii) the capacity constraints; different methodologies can be applied to solve them. In this chapter we focus on the VPN case, in which common assumptions are i) linear cost, i.e., $f_l(C_l) = d_l C_l$, ii) non-bifurcated routing, and iii) continuous capacities.

For each source-destination pair, traffic is transmitted over exactly one directed path in the network. Each path p_{sd} from source s to destination d (that is an input to the problem) is determined by a minimum-cost algorithm. Considering that TCP is a closed-loop control protocol, we define as *transport path* (route) $r_{sd} = p_{sd} \cup p_{ds}$. For each path r_{sd} and link $l \in E$, let $\delta_l(r_{sd}) \in \{0, 1\}$ denote the indicator function which is one if link l is in path r_{sd} and zero otherwise. This allows the direct evaluation of the average data flow f_l on a link l as a function of traffic requirements.

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. $P_{loss}(r_{sd})$ is the desired e2e loss probability for path r_{sd}.

3.4 The Capacity Assignment problem

As previously said, we solve the Capacity Assignment (CA) problem by considering infinite buffers. The only constraint that has to be met is therefore the e2e packet delay, which is evaluated thanks to the adoption of the $M_{[X]}/M/1/\infty$ model for links. Given the network topology, the traffic requirements, and the link flows in the general problem, it is possible to formulate the CA problem as follows.
Minimize:

$$Z_{CA} = \sum_{l \in E} f_l(C_l) \tag{2.2}$$

Subject to:

$$\frac{K}{\mu} \sum_{l \in E} \frac{\delta_l(r_{sd})}{C_l - f_l} \leq RTT_{sd} - \tau_{sd} - \tau_{ds}, \quad \forall\, s, d \in V \tag{2.3}$$

$$f_l = \sum_{s,d \in V} \delta_l(r_{sd}) \gamma_{sd}, \quad \forall\, s, d \in V \tag{2.4}$$

$$C_l \geq f_l, \quad \forall\, l \in E \tag{2.5}$$

The objective function (2.2) represents the total link cost, which is the sum of the cost functions of link l, $f_l(C_l)$. Equation (2.3) is the packet delay constraint for each source/destination node pair; where RTT_{sd} is the desired Round Trip Time for (aggregated) TCP traffic from node s to node d, and τ_{sd} is the propagation delay for path p_{sd}. Equation (2.4) defines the average data flow on the link. Constraints (2.5) are non–negativity constraints. The only design variables are the link capacities C_l.

We notice that the objective function and the constraint functions are (weakly) convex, therefore the CA problem is a convex optimization problem.

3.5 The Buffer Assignment problem

Given the network topology, the traffic requirements and the link flows, by fixing the link capacities in the general problem, it is possible

to formulate the Buffer Assignment (BA) problem as follows. Minimize:

$$Z_{BA} = \sum_{l \in E} g_l(B_l) \tag{2.6}$$

Subject to:

$$\sum_{l \in E} \delta_l(r_{sd}).p(B_l, C_l, f_l, [X]) \leq P_{loss}(r_{sd}), \quad \forall\, s, d \in V \tag{2.7}$$

$$B_l \geq 0, \quad \forall\, l \in E \tag{2.8}$$

The objective function (2.6) represents the total buffer cost, which is the sum of the cost functions of buffer l, $g_l(B_l) = B_l$. Equation (2.7) is the loss probability constraint for each source/destination node pair. Where $p(B_l, C_l, f_l, [X])$ is the average loss probability for the $M_{[X]}/M/1/B$ queue, which is evaluated by solving its Continuous Time Markov Chain (CTMC). Constraints (2.8) are non–negativity constraints.

In the previous formulation we have considered the following upper bound on the value of P_{loss} (constraint (2.7)).

$$\begin{aligned} P_{loss}(r_{sd}) = \; & 1 - \prod_{l \in E} (1 - \delta_l(r_{sd}).p(B_l, C_l, f_l, [X])) \leq \\ & \leq \sum_{l \in E} \delta_l(r_{sd}).p(B_l, C_l, f_l, [X]) \end{aligned} \tag{2.9}$$

Notice also that the first part of equation (2.9) is based on the assumption that link losses are independent.

Therefore, the solution of the BA problem is a conservative solution to the full problem. Notice also that, to evaluate the packet dropping probability, we explicitly consider the bidirectional transport path r_{sd}, taking into account the fact that the performance of TCP is affected by data segments lost on the forward path p_{sd}, and by ACKs lost on the reverse path p_{ds}. While the second event has less impact on TCP performance, it is not negligible for short file transfers.

The proof that the BA problem is a convex optimization problem is not a straightforward task. 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, we conjecture that the BA problem is a convex optimization problem by considering that: (i) for an $M/M/1/B$ queue, $p(B, C, f)$ is a convex function (see Nagarajan and Towsley, 1992); and (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 loss probability is a convex function of B.

We can thus classify both the CA and BA problems as multivariable constrained convex minimization problems; therefore, the global minimum (for each subproblem) can be found using convex programming techniques. We solve the minimization problems applying first a constraints reduction procedure which reduces the set of constraints by eliminating redundancies. Then the solution of the CA and BA problems is obtained using the *logarithm barrier method*, see Wright (1992).

3.6 Setting the AQM parameters

The output of the BA problem is the buffer size B_l 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 chapter, we consider Random Early Detection (RED), see Floyd and Jacobson (1993), as an example, and discuss how to set its parameters.

The basic 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 avg 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 $B_l = \alpha.max\_th$, $\alpha > 1$, e.g., $\alpha = 2$ as suggested in the "gentle" Rosolen et al. (1999) variation of RED[2]

Therefore, the RED parameter dimensioning problem can be solved by imposing that:

$$p(B_l, C_l, f_l, [X]) = \frac{E_l[N] - min\_th_l}{max\_th_l - min\_th_l} max\_p_l \tag{2.10}$$

Note that (2.10) fixes max_p_l by imposing that the average RED dropping probability evaluated at the average queue length $E_l[N]$ (obtained considering the $M_{[X]}/M/1/B$ queue) satisfies the $P_{loss}(r_{sd})$ constraint

[2]Gentle-RED is a modification of RED that allows a smoother transition of the dropping probabilities when the average queue length exceeds the maximum threshold, making it more robust to the setting of the parameters.

in (2.7). Finally, we set $min\_th_l = \beta.max\_th_l$, $\beta < 1$. In the numerical examples that follow, we selected $\alpha = 2$, $\beta = 1/16$.

4. Numerical examples and simulations

In this section we present some selected numerical results, showing the accuracy of the IP network designs produced by our methodology. We first applied our optimization method to design some networks topologies and next used a simulation procedure to evaluate if the QoS constraints were actually respected. The tool used for simulation is *ns* version 2. For all simulations, the "batch means" technique, with 30 batches, was considered.

We assume that New Reno is the TCP version of interest. In addition, we assume that TCP connections are established choosing at random a server–client pair, and are opened at instants described by a Poisson process. Connection opening rates are determined so as to set the link flows f_l to their desired values. The packet size is assumed constant, equal to the maximum segment size (MSS), the maximum window size is assumed to be 32 segments. The amount of data to be transferred by each connection (i.e., the file size) is expressed in number of segments. We consider a mixed traffic scenario where the file size follows the distribution shown in Figure 2.3, which is derived from one-week long measurements, conducted in Mellia et al. (2002), in three different time periods. In particular, we report the discretized CDF, obtained by splitting the flow 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 inside one reports the same distribution taking today's most common MSS of 1460 bytes as unit. We use the most recent measurements in the following simulations.

4.1 Single-bottleneck topology

We start by considering a very simple single bottleneck topology. We assume one-way TCP New Reno connections with uncongested backward path. The topology comprises the bottleneck link, and a number of peripheral links, whose capacities are equal to 25 Mbps and whose propagation delays are uniformly distributed between 0.01 and 0.03ms.

Table 2.1 reports the capacity and buffer size of the bottleneck link obtained with our method. In order to obtain some comparisons, we also implemented a design procedure using the classical formula, see Kleinrock (1976), which considers an $M/M/1$ queue model in the CA problem. We also extended the classical approach to the BA problem,

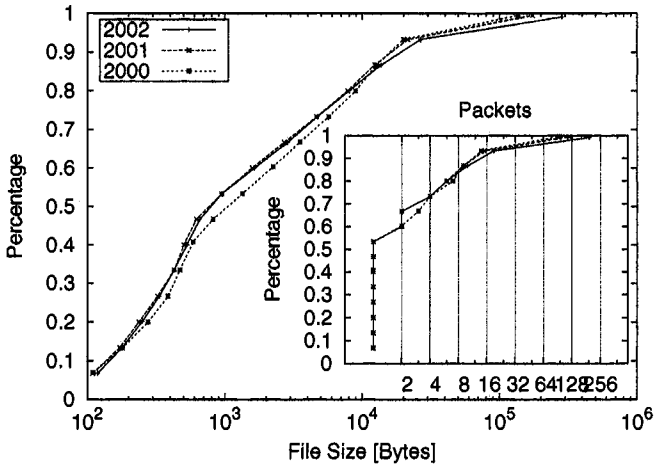


Figure 2.3. TCP connection length cumulative distributions

which is solved considering $M/M/1/B$ queues. We choose as target parameters the following: latency $L_t \leq 0.3$s for flows shorter than 20 segments, throughput $T_h \geq 512$ Kbps for flow longer than 20 segments and $P_{loss} = 0.01$. Using the transport layer QoS translator, we obtain the equivalent constraint $RTT \leq 0.03$s (for the sake of simplicity, in the examples we will consider $RTT_{sd} = RTT$, $\forall\ s, d \in V$), which corresponds to the most stringent latency constraint (Figure 2.2). We imposed these same constraints also in the classical approach. Looking at the CA solution we observe that using our methodology a much higher data rate than the classical approach is required, as shown by the average link utilization $\rho_l = 0.64$. Also when considering the buffer size design, we observe that the adoption of the $M_{[X]}/M/1/B$ model leads to larger buffer requirements than the one with a simpler $M/M/1/B$ model.

Table 2.2 shows the average packet delay $E[T]$, the average queue size $E[N]$, and the packet loss probability P_{loss}, from the $M_{[X]}/M/1/B$ queue model and from *ns*-2 simulations (considering droptail and RED buffers). We can observe good agreement between model and simulations results. Notice also that the assumption of exponential length packets does not affect the performance evaluation. Indeed, recalling that in the simulation all data packets have a fixed length of 1460 bytes, no significant differences are noticed. This point out as an indication that packet length distribution is not a critical factor.

Table 2.3 reports file transfer latencies for different flow sizes (in number of segments), as estimated by the CSA model (second column) and

Table 2.1. Design results for bottleneck network

	$M_{[X]}/M/1/B$	$M/M/1/B$
$f_l[Mbps]$	16	16
$C_l[Mbps]$	25	17
ρ_l	0.64	0.93
$B_l[pkts]$	79	28

Table 2.2. Model and simulation results for bottleneck network (network layer)

	$M_{[X]}/M/1/B$	*droptail*	*RED*
$E[T]$	0.0095	0.010	0.0091
$E[N]$	13.2	13.4	12.4
P_{loss}	0.0098	0.0044	0.0039

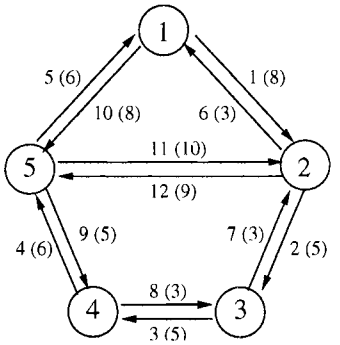
as observed by simulations. Results are shown considering both our approach and the classical methodology, and by considering either droptail or RED buffers. We can observe that the accuracy of the network design obtained with our methodology is extremely good, with flow latencies always meeting the QoS constraints. Note also that longer flows obtain a much higher throughput than the target, because the flow transfer latency constraint is more stringent (as also shown in Figure 2.2). On the contrary, the network design obtained with the classical formula fails to meet the QoS constraints. This is mainly due to the adoption of an $M/M/1$ queue model, which fails to capture the high burstiness of IP traffic. No major differences are visible when RED buffers are present in the network, if our methodology is adopted, while a degradation of performance is observed if the classical approach is used. This is due to the very small buffer sizes resulting from the classical design, which do not allow RED to work properly, and therefore cause a large packet dropping probability.

4.2 Multi-bottleneck topologies

As a second example, we present results obtained considering the multi-bottleneck mesh network shown in Figure 2.4 with 5 nodes and 12 links. In this case, link propagation delays are all equal to 0.5ms, that correspond to a link length of 150 Km. Figure 2.4 shows link identifiers, link weights (in parentheses), and the traffic requirements matrix Γ. Link weights are chosen in order to have one single path (by using a minimum cost routing algorithm) for every source/destination pair. A

Table 2.3. Model and simulation results for bottleneck network (L_t)

		$M_{[X]}/M/1/B$		$M/M/1/B$	
seg.	*CSA*	*droptail*	*RED*	*droptail*	*RED*
1	0.05s	0.08s	0.05s	1.84s	1.56s
2	0.08s	0.09s	0.08s	2.12s	2.31s
4	0.12s	0.12s	0.11s	2.44s	3.71s
6	0.16s	0.13s	0.13s	2.56s	5.26s
10	0.20s	0.15s	0.16s	2.84s	6.59s
19	0.26s	0.18s	0.19s	3.16s	10.41s
195	2.07Mbps	5.2Mbps	5.1Mbps	180kbps	15kbps



O/D	1	2	3	4	5
1	0	7	9	8	3
2	9	0	3	9	2
3	4	1	0	8	7
4	8	1	6	0	2
5	3	8	4	9	0

Traffic Matrix [Mbps]

Figure 2.4. 5-node network: topology and traffic requirements

number of peripheral links (not shown in the picture) are attached to each node. These links are not congested, their capacities being equal to 30 Mbps, and their propagation delays are uniformly distributed between 0.01 and 0.03ms.

We considered the same QoS target constraints for all source/destination pairs, which are: (i) file latency $L_t \leq 0.5$s for TCP flows shorter than 20 segments, and (ii) throughput $T_h \geq 512$ Kbps for TCP flows longer than 20 segments. Selecting $P_{loss} = 0.01$, we obtain as design constraint $RTT \leq 0.07$s as can be seen in Figure 2.2.

The CA and BA problems associated with this network have 12 unknown variables and 11 constraint functions (we have discarded 9 redundant constraint functions). Table 2.4 reports the link capacities, link utilizations, and buffer sizes obtained with the proposed method. We also report in the same table the average packet delay $E[T]$, average queue size $E[N]$, and the average packet loss probability P_{loss}, computed using the $M_{[X]}/M/1/B$ queue model.

Table 2.4. Design results for 5-node network

$M_{[X]}/M/1/B$						
Link	C [Mbps]	ρ	B	$E[T]$	$E[N]$	P_{loss}
1	18.9	0.85	196	0.033	47.8	0.006
2	23.9	0.88	261	0.035	65.2	0.004
3	26.9	0.89	265	0.034	73.2	0.006
4	11.9	0.75	137	0.031	25.3	0.004
5	4.4	0.67	88	0.061	16.1	0.010
6	25.4	0.82	184	0.021	40.2	0.004
7	18.4	0.76	160	0.021	26.8	0.002
8	23.4	0.81	188	0.022	37.0	0.003
9	23.9	0.88	243	0.034	63.3	0.005
10	13.9	0.79	154	0.032	31.1	0.005
11	9.4	0.85	163	0.062	43.9	0.01
12	3.4	0.58	72	0.061	10.8	0.009
	ΣC_i	$\overline{\rho}$	$\overline{B}$	$\overline{E[T]}$	$\overline{E[N]}$	
	204.16	0.794	175	0.037	40.0	

It can be noticed that the link utilization factors are in the range $[0.67, 0.89]$, with average equal to about $\overline{\rho} = 0.8$. Buffer sizes are in the range $[70 : 270]$, with average $\overline{B} = 175$, which is about 4 times the average number of packets in the queue ($\overline{E[N]} = 40$). This is due to the bursty arrival process of IP traffic, which is well captured by the $M_{[X]}/M/1/B$ model.

To complete the evaluation of the new methodology, we compare the link utilization factors and buffer sizes obtained when considering the classical algorithm, i.e., by using an $M/M/1/B$ queueing model. Figure 2.5 shows the link utilizations (first plot) and buffer sizes (second plot) obtained with our method and with the classical approach. 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 smaller than those produced by the classical approach, and buffers are much longer.

To evaluate the quality of the design results, we ran *ns*-2 simulations for droptail and RED buffers. Considering first network layer QoS parameters ($E[T]$ and P_{loss}), Table 2.5 reports *ns*-2 simulation results for the average packet delay and the average packet loss probability on every link (considering droptail and RED buffers). It can be noticed that the resulting delay and loss probability are very close to the de-

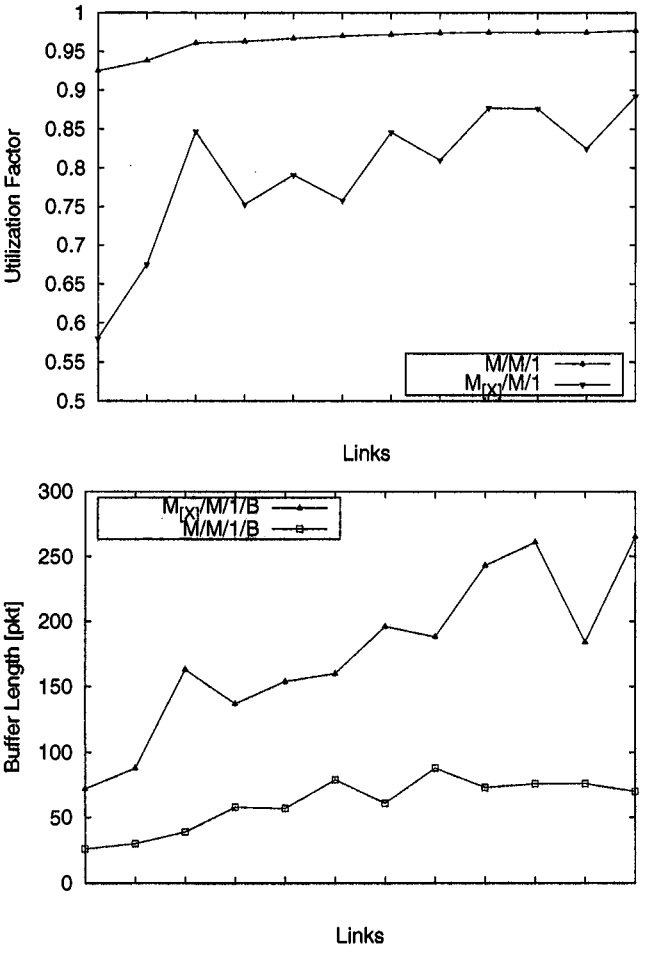


Figure 2.5. Link utilization factor and buffer size for a 5-node network

sired one; in fact, simulated $E[T]$ has few results larger than the targets, while simulated P_{loss} results smaller than target one (considering that the simulation margin of error for $E[T]$ and P_{loss}, for each link, are about $\pm 13.5\%$ and $\pm 20\%$, respectively).

In order to verify the e2e QoS constraints at the transport layer, we report detailed results selecting traffic from node 4 to node 1, which is routed over one of the most congested path (three hops, over links: 8,7,6). Figure 2.6 plots the file transfer latency for all flow sizes for the selected source destination pair (95% confidence intervals are shown). The QoS constraint of 0.5s for the maximum latency is also reported.

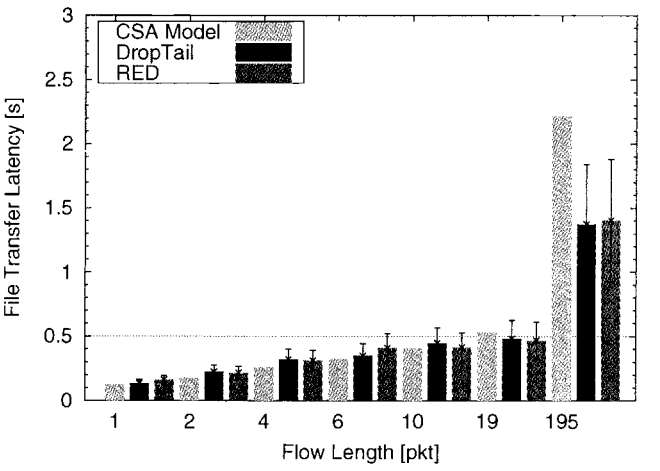


Figure 2.6. Model and simulation results for latency; 3-link path from the 5-node network

In this case we can see that model results and simulation estimates are in perfect agreement with specifications, being the constraints perfectly satisfied for all flows shorter than 20 segments. Note also that longer flows obtain a much higher throughput than the target, because the flow transfer latency constraint is more stringent (as also shown in Figure 2.2).

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, i.e., the network designed with the classical approach is not capable of supporting the offered load and therefore cannot satisfy the QoS constraints.

As a second example of multi-bottleneck topology we chose a network of 10 nodes and 24 links. For all (90) source/destination pairs, traffic is routed over a single path. Link propagation delays are uniformly distributed between 0.05 and 0.5ms, i.e., link lengths vary between 15 Km and 150 Km. The traffic requirement matrix is set to obtain an average link flow of about 15 Mbps.

The CA and BA problems associated with this network have 24 unknown variables and 66 constraint functions (we have discarded 24 redundant constraint functions). We considered the same design target parameters as for the previous example. In order to observe the impact

Table 2.5. Simulation results for 5-node network (network layer)

	droptail			*RED*		
Link	$E[T]$	$E[N]$	P_{loss}	$E[T]$	$E[N]$	P_{loss}
1	0.029	40.3	0.002	0.027	38.3	0.004
2	0.015	26.8	0.001	0.019	35.9	0.002
3	0.019	40.3	0.001	0.029	61.6	0.003
4	0.033	25.5	0.004	0.033	25.6	0.004
5	0.079	20.4	0.005	0.086	22.3	0.008
6	0.018	33.4	0.003	0.021	37.8	0.005
7	0.020	24.4	0.004	0.026	31.2	0.002
8	0.023	38.8	0.005	0.033	54.8	0.004
9	0.024	44.5	0.002	0.020	36.6	0.002
10	0.031	29.2	0.003	0.031	29.3	0.003
11	0.055	38.2	0.002	0.052	36.3	0.005
12	0.096	16.5	0.010	0.090	15.4	0.010

of traffic load and performance constraints on our design methodology, we consider different numerical experiments.

Figure 2.7 shows the range of network link utilizations versus traffic load (first plot). Looking at how traffic requirements impact the CA problem, we observe that the larger is the traffic load, the higher the utilization factor. This is quite intuitively explained by a higher statistical multiplexing gain, and by the fact that the RTT is less affected by the transmission delay of packets at higher speed. The behavior of buffer sizes versus traffic requirements is shown in the second plot. As expected, the larger is the traffic load the higher the space needed in queue (buffer sizes).

The impact of more stringent QoS requirements is considered in Figure 2.8 ($P_{loss} = 0.01$, link traffic load = 15 Mbps). Notice that, in order to satisfy a very tight constraint (file latency $L_t \leq 0.2$s), it is necessary an utilization factor close to 20% on some particularly congested links (first plot). Tight constraints mean packet delays with small values and thus larger capacity values concerning the link flows. On the contrary, relaxing the QoS constraints, we note a general increase in the link utilization, up to 90%. The behavior of buffer sizes versus file transfer latency requirements is shown in the second plot.

Finally, Figure 2.9 shows link utilizations and buffer sizes considering different packet loss probability constraints, while keeping fixed the file transfer latency $L_t \leq 2$s and throughput $T_h \geq 512$ Kbps (link traffic load

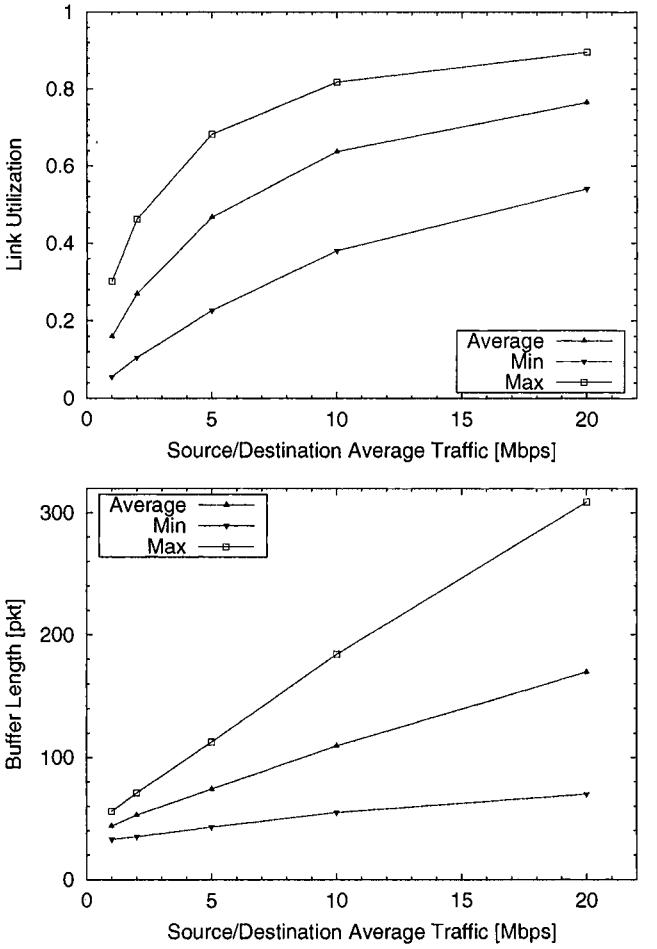


Figure 2.7. Link utilization factor and buffer length for a 10-node network (considering different source/destination traffics)

= 15 Mbps). Obviously, an increase of P_{loss} values forces the transport layer QoS translator to reduce the RTT to meet the QoS constraints. As a consequence, the utilization factor decreases (first plot).

More interesting is the effect of selecting different values of P_{loss} on buffer sizes (second plot). Indeed, to obtain $P_{loss} \leq 0.005$, buffer sizes longer than 350 packets are required, while $P_{loss} \leq 0.02$ can be guaranteed with buffers shorter than 70 packets. This result stems from the correlation of TCP traffic and is not captured by a Poisson model.

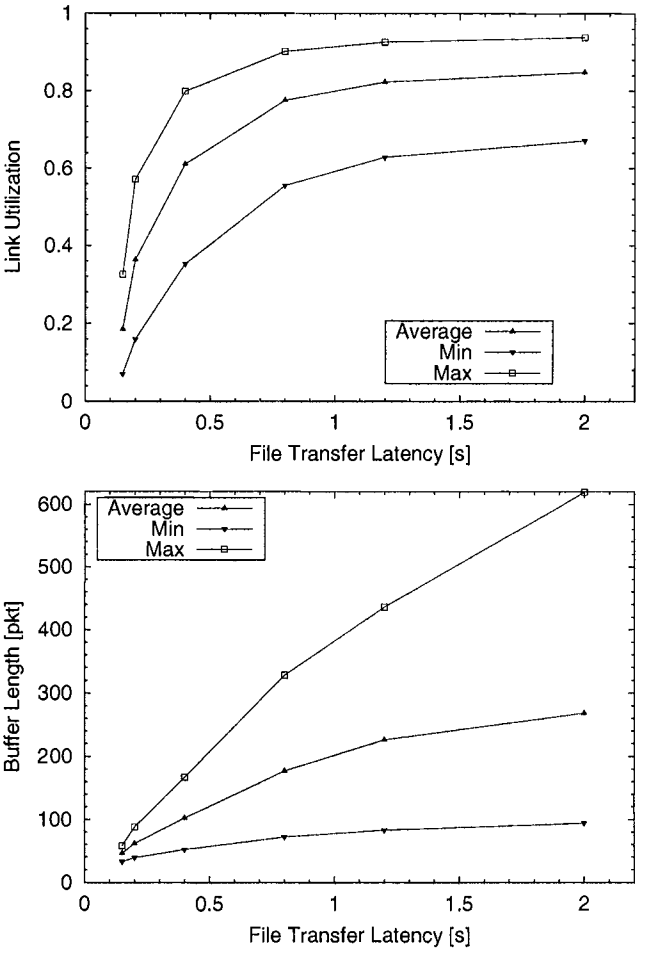


Figure 2.8. Link utilization factor and buffer length for a 10-node network (considering different target file transfer latencies)

Running simulations in *ns*-2 with more than about 1000 TCP connections becomes very expensive with standard computers, so in this case we performed *path simulations* rather than simulating the complete network, i.e., we selected a single path, and simulated only that one, using *ns*-2. TCP connection opening rates are determined so as to set the link flows to their determined values. The results obtained by path simulations are in general a worst case with respect to what would be obtained by running simulations of the entire network, because the "interfering"

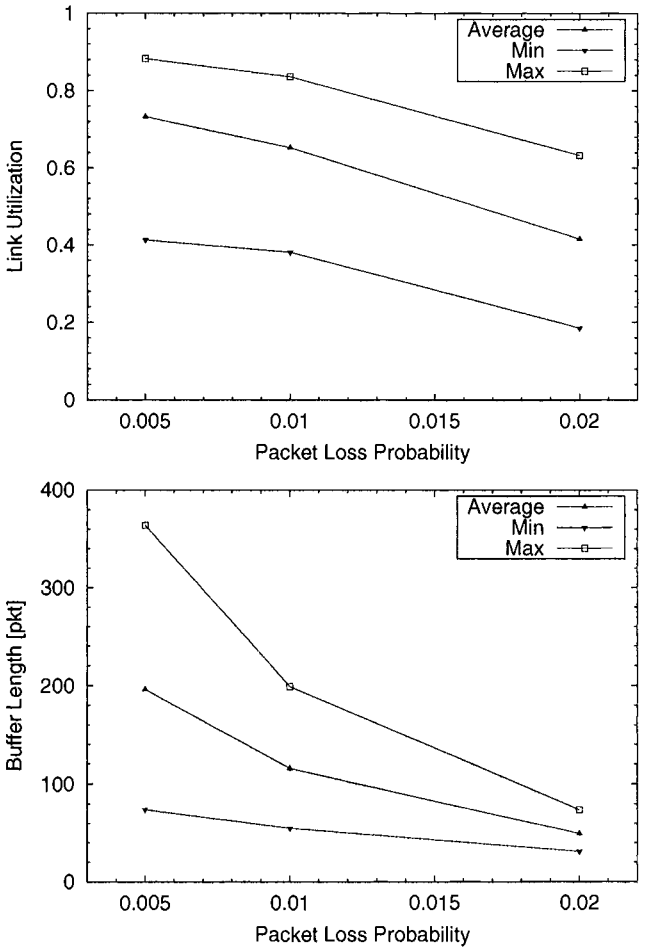


Figure 2.9. Link utilization factor and buffer length for a 10-node network (considering different target packet loss probabilities)

traffic is more aggressive, since it does not cross all links along its path, hence no loss and no traffic shaping occurs.

As an example, by choosing a 4-link path, we obtained average packet delay $E[T]$ and average packet loss probability P_{loss} results that are reported in Table 2.6, considering three scaled versions of the traffic matrix (and network designs). It can be observed that, for all the traffic scenarios, simulated $E[T]$ are in accordance to the targets, and simulated P_{loss} have results smaller than the targets (considering that the simu-

Table 2.6. Simulation results for a 4-link path from the 10-node network (RED)

	γ	3γ	$\gamma/3$
$E[T]$	0.070	0.071	0.069
P_{loss}	0.0052	0.0060	0.0079

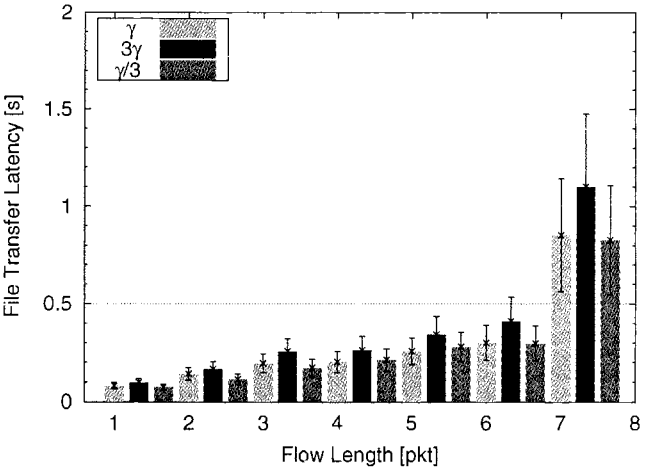


Figure 2.10. Latency simulation for a 4-link path from the 10-node network (RED)

lation margin of error for $E[T]$ and *Ploss* are about $\pm 15\%$ and $\pm 20\%$, respectively).

File transfer latency results are also obtained (for the chosen 4-link path) and are reported versus the file size, and scaled versions of the traffic matrix, in Figure 2.10 (in the case of RED buffers). It can be noted that the target QoS constraints are met in all cases (95% confidence intervals are shown).

4.3 Computational complexity

Finally, we briefly discuss the computation times needed to solve the CA problem. The solver algorithm was implemented in the C language, and the computation was carried out on a 1GHz processor under Linux O.S.

We considered networks with different numbers of nodes (from 10 to 100) and different number of ingoing/outgoing links per node (from 3 to 9). For each number-of-nodes/number-of-links pair, we obtain problems

with different number of variables and different number of constraints. CPU times range from less than 1 second (to solve a 10-nodes/30-links network design problem) to about than 40 minutes (to solve a 100-node/900-links network design problem).

5. Conclusion

In this chapter, we have proposed a new packet network design and planning approach that is based on user-layer QoS parameters.

The main novelty of our approach is that it considers the end-to-end performance constrains at the application layer, mapping them into transport layer QoS constraints first, and finally into network layer performance constraints. Traditional packet network design approaches model a communication network as a Jackson queueing network, thus assuming packet flows to be Poisson. A second important improvement with respect to traditional approaches lies in the fact that we have tried to consider more realistic packet traffic models, accounting for both long-lived and short-lived TCP connections, and considering more complex systems of queues which have been recently proved to effectively represent the performance of modern IP networks (Garetto and Towsley, 2003). Examples of application of the proposed design methodology to different networking configurations have shown the effectiveness of our approach.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful comments and suggestions.

References

Cardwell, N., Savage, S., and Anderson, T. (2000). Modeling TCP latency. In: *Proceedings of Infocom 2000*, pp. 1742–1751, Tel Aviv, Israel.

Chao, X., Miyazawa, M., and Pinedo, M. (1999). *Queueing Networks, Customers, Signals and Product Form Solutions*. John Wiley.

Cheng, K.T. and Lin, F.Y.S. (1995). Minmax end-to-end delay routing and capacity assignment for virtual circuit networks. In: *Proceedings of IEEE Globecom 1995*, pp. 2134–2138.

Claffy, K., Miller, G., and Thompson, K. (1998). The nature of the beast: Recent traffic measurements from an Internet backbone. In: *Proceedings of INET'98*, Geneva, CH.

Floyd, S. and Jacobson, V. (1993). Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413.

Fraleigh, C., Tobagi, F., and Diot, C. (2003). Provisioning IP backbone networks to support latency sensitive traffic. In: *Proceedings of IEEE Infocom 2003*, pp. 375–385, San Francisco, CA.

Garetto, M. and Towsley, D. (2003). Modeling, simulation and measurements of queuing delay under long-tail Internet traffic. In: *Proceedings of ACM SIGMETRICS*

2003, pp. 47–57, San Diego, CA.

Gavish, B. and Neuman, I. (1989). A system for routing and capacity assignment in computer communication networks. *IEEE Transactions on Communications*, 37(4):360–366.

Gavish, B. (1992). Topological design of computer communication networks - the overall design problem. *European Journal of Operational Research*, 58:149–172.

Gribble, S.D. and Brewer, E.A. (1997). System design issues for Internet middleware services: Deductions from a large client trace. In: *USITS'97*.

Kamimura, K. and Nishino, H. (1991). An efficient method for determining economical configurations of elementary packet-switched networks. *IEEE Transactions on Communications*, 39(2):278–288.

Kleinrock, L. (1976). *Queueing Systems, Volume II: Computer Applications*. Wiley Interscience, New York.

Knoche, H. and de Meer, H. (1997). Quantitative QoS mapping: A unifying approach. In: *Proceedings of the 5th Int. Workshop on Quality of Service (IWQoS97)*, pp. 347–358, New York, NY.

Mai Hoang, T.T. and Zorn, W. (2001). Genetic algorithms for capacity planning of IP-based networks. In: *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, pp. 1309–1315.

Markopoulou, A., Tobagi, F., and Karam, M. (2002). Assessment of VoIP quality over Internet backbones. In: *Proceedings of IEEE Infocom 2002*, pp. 747–760, New York, NY.

Mellia, M., Carpani, A., and Lo Cigno, R. (2002). Measuring IP and TCP behavior on edge nodes. In: *Proceedings of IEEE Globecom 2002*, pp. 2533–2537, Taipei, TW.

Nagarajan, R. and Towsley, D. (1992). A Note on the convexity of the probability of a full buffer in the M/M/1/K queue. *CMPSCI Technical Report TR 92-85*.

Padhye, J., Firoiu, V., Towsley, D., and Kurose, J. (2000). Modeling TCP Reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking*, 8(2):133–145.

Paxson, V. and Floyd, S. (1995). Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244.

Rosolen, V., Bonaventure, O., and Leduc, G. (1999). A RED discard strategy for ATM networks and its performance evaluation with TCP/IP traffic. *ACM Computer Communication Review*, 29(3):23–43.

Wright, M. (1992). Interior methods for constrained optimization. *Acta Numerica*, 1:341–407.

http://www.springer.com/978-0-387-25550-7

Performance Evaluation and Planning Methods for the Next Generation Internet

Girard, A.; Sansò, B.; Vazquez-Abad, F. (Eds.)

2005, XVI, 365 p., Hardcover
ISBN: 978-0-387-25550-7