Considering End-to-End QoS in IP Network Design [1]

Emilio C. G. Wille[1-2], Michele Garetto[1], 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

Abstract. Traditional approaches to optimal design and planning of packet networks focus on the network-layer infrastructure, In this paper, we propose a new packet network design and planning approach 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 [1] spurred many research activities in the field of 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 tradeoff between total cost and *network average* performance (expressed in terms of average network-wide packet delay, average packet loss probability, average link utilization, network reliability, etc.). This approach adopts the viewpoint of network operators, who quite naturally aim at the optimization of some aggregate performance measures, that describe the general behavior of their network, averaging over all traffic relations. Of course, this approach is subject to the risk that a few traffic relations suffer unacceptable performance levels.

Today, with the enormous success of the Internet, packet networks have reached their maturity, are used for very critical services, and researchers as well as operators are concerned with end-to-end QoS (Quality of Service) issues, and SLA (Service Level Agreement) 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 big part 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) [2, 3].

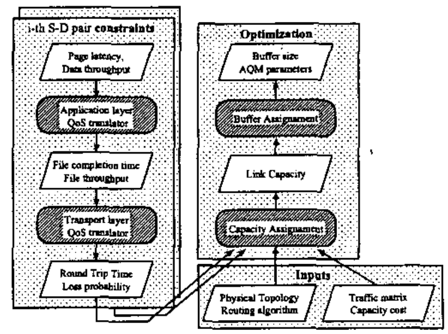


Figure 1. Schematic flow diagram of the network design methodology

In this paper, we propose 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[2]. Our proposed approach maps the end-user performance 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 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 [4], but a higher degree of correlation exists, which can be partly due to the TCP control mechanisms. This means that the usual approach of modeling packet networks as networks of M/M/1 queues [5, 6, 7] is not acceptable. In this paper we adopt a refined IP traffic modeling technique, already presented in [8], 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.

[1] This work was supported by the Italian Ministry for Education, University and Research under project TANGO. The first author was supported by a CAPES Foundation scholarship from the Ministry of Education of Brazil.

[2] The only previous work [9] which accounts for user-layer QoS constraints focuses mainly on voice traffic, and does not consider the impact of the transport layer.

2 IP Network Design Methodology

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, such as web page download latency. Thanks to the definition of *QoS translators*, all the user-layer constraints are then mapped into lower-layer constraints, down to the IP layer.

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, and reliability. An overview of the QoS translation problem is given in [10]. According to the Internet protocol architecture, at least two QoS mapping procedures should be considered in our case; the first one 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}).

2.1.1 Application-Layer QoS translator

This module takes as input the application-layer QoS constraints, such as web page transfer latency, data throughput, audio quality, etc. Given the multitude of Internet applications, it is not possible to devise a generic procedure to solve this problem, and in this paper we do not focus on generic translators, since ad-hoc solutions should be used depending on the application.

2.1.2 Transport-Layer QoS translators

The translation from transport-layer QoS constraints to network-layer QoS parameters, such as Round Trip Time and Packet Loss Probability is more difficult. This is mainly due to the complexity of the TCP protocol, because of the error, flow and congestion control algorithms it implements. 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., mice) meet the maximum file transfer latency constraint, while longer flows (i.e., elephants) are subjected to the throughput constraint. Obviously, the more stringent constraints among latency and throughput will be considered. For example, from the knowledge of the file length distribution [3], is possible to say that 85% of all TCP flows are shorter than 20 packets. For these flows, the latency constraint must hold.

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 [8] and the references therein). Our approach is based on the inversion of such 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, we use the TCP latency model described in [11]. We will refer to this model as CSA model (from authors' name).

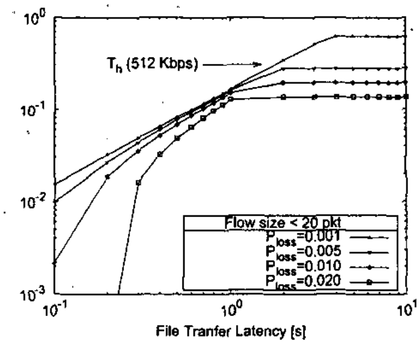


Figure 2. RTT constraints as given by the transport layer QoS translator

When considering throughput, we instead exploit the formula in [12], referred as PFTK model.

At least two are the parameters that affect TCP throughput (or latency), i.e., RTT and P_{loss}. We decided to fix the P_{loss} parameter, and leave RTT as 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 may impact the network load due to retransmissions. 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 the file transfer latency and a fixed throughput of 512 kbps constraints, the curves of Fig. 2 report the maximum admissible RTT which satisfies the most stringent constraint for different values of P_{loss}.

2.2 Optimization formulation and solutions

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 very general formulation:

Given: physical topology, routing algorithm, peak-hour traffic estimates between node pairs, capacity cost;
Minimize: total capacity cost, total buffer cost;
Over the design variables: link capacities, buffer sizes;
Subject to: packet delay constraints, packet loss probability constraints on every source/destination pair.

In the solution of the Capacity Assignment (CA), and Buffer Assignment (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.

2.2.1 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.

Instead of considering traditional $M/M/1/B$ queueing models, we choose to model the increased traffic burstiness induced by TCP by means of batch arrivals, hence using $M_{[X]}/M/1/B$ queues, according to recent findings in [8]. Given the flow length distribution, a stochastic model of TCP (described in [8]) is used to obtain the batch size distribution $[X]$, considering the number of segments that TCP sources send in one RTT. 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.

2.2.2 *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 [13]:

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

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

2.2.3 *Costs*

We consider two types of cost: the total link capacity and to the total buffer size.

The total "link" cost D_C, which is a function of both the capacity C_l and the length d_l of link l, is defined as $D_C = \sum_l h(C_l, d_l)$, where $h(C_l, d_l)$ is supposed to be a weakly convex function. For the sake of simplicity, we will assume that $h(C_l) = K_1 \cdot d_l \cdot C_l$ in the remaining of this paper, being K_1 a constant.

The "buffer" cost D_B is defined similarly as $D_B = \sum_l g(B_l)$, where B_l is the buffer size on link l, and $g(B_l)$ is a weakly convex function. Also in this case, we will consider a linear cost, i.e., $g(B_l) = K_2 \cdot B_l$, being K_2 a constant.

While the optimization of link capacities and buffer sizes should be performed by considering only a discrete set of values, for computational efficiency it is convenient to consider them as continuous variables. Fortunately, the optimal values of capacities and buffer sizes obtained by a continuous optimization can then be rounded, still obtaining solutions which are close to optimality. This is particularly true for VPN design, in which link capacities have a much smaller granularity than for backbone design.

2.2.4 *Network model, traffic, and routing*

In the mathematical model, the IP 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 network router and an edge represents a physical link connecting one router to another.

For each source/destination pair, traffic is transmitted over exactly one directed path in the network. Each path is determined by the fixed routing algorithm choosing from a set of paths $P = \{p_{sd}\}$. 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.

For simplicity, we assume fixed routing, as is currently used in IP networks. This allows the direct evaluation of the average data flow f_l on a link l as a function of routing tables and 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 transmission rate from source s to destination d. It is possible to derive the $\hat{\Gamma}$ matrix also from a higher-level description of the (maximum) traffic requests, given in terms of "pages per second", or "flows per second" for a given source/destination pair. From the knowledge of the page/flow size it is then straightforward to obtain $\hat{\gamma}_{sd}$.

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

2.2.5 *The Capacity Assignment problem*

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

$$Z_{CA} = \sum_{l \in E} h(C_l, d_l) \tag{2}$$

Subject to:

$$\frac{K}{\mu} \sum_{l \in E} \frac{\delta_l(r_{sd})}{C_l - f_l} \leq delay(r_{sd}), \quad \forall\, s, d \in V \tag{3}$$

$$delay(r_{sd}) = RTT_{sd} - \tau_{sd} - \tau_{ds}, \quad \forall\, s, d \in V \tag{4}$$

$$f_l = \sum_{s,d \in V} \delta_l(r_{sd})\gamma_{sd}, \quad \forall\, s, d \in V \tag{5}$$

$$C_l \geq f_l, \quad \forall\, l \in E \tag{6}$$

where $delay(r_{sd})$ is the end-to-end queueing and transmission delay, RTT_{sd} is the desired round trip time for TCP connections from node s to node d, and τ_{sd} is the propagation delay for path p_{sd}.

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

2.2.6 *The Buffer Assignment problem*

Given the network topology and the traffic requirements, by fixing the link flows and the link capacities in the general problem, it is possible to formulate the BA problem as follows. Minimize:

$$Z_{BA} = \sum_{l \in E} g(B_l) \tag{7}$$

Subject to:

$$\sum_{l \in E} \delta_l(r_{sd}).p(B_l, C_l, f_l, [X]) \leq P_{loss}(r_{sd}), \quad \forall\, s, d \in V \tag{8}$$

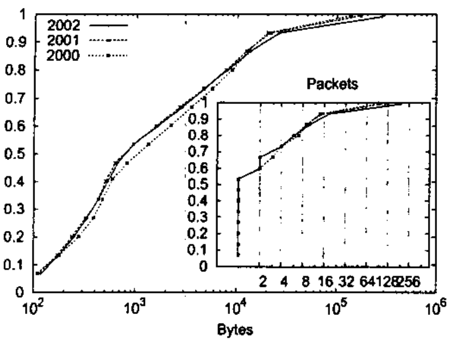


Figure 3. TCP connection length cumulative distributions.

$$B_l \geq 0, \ \ \forall \, l \in E \tag{9}$$

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 the corresponding continuous time Markov chain.

Notice that in the previous formulation, constraint (8) has been linearized thanks to the following inequality:

$$\begin{aligned} P_{loss}(r_{sd}) = \ & 1 - \prod_{l \in E} (1 - \delta_l(r_{sd}).p(B_l, C_l, f_l, [X])) \leq \\ & \leq \sum_{l \in E} \delta_l(r_{sd}).p(B_l, C_l, f_l, [X]) \end{aligned} \tag{10}$$

The solution of the linearized problem is a conservative solution to the original BA 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. In addition, also this hypothesis will produce conservative solutions the original BA problem. We conjecture that the BA problem is a convex optimization problem [14]. We can thus classify both the CA and BA problems as multivariable constrained convex minimization problems; therefore, the global minimum 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 via the *logarithm barrier method* [15].

3 Numerical Examples

In this section we present some selected numerical results, showing the accuracy of the IP network designs produced by our methodology. In order to validate our approach, we compare the target performance parameters against the performance measured from very detailed simulation experiments. The tool used for simulation is ns version 2. For all simulations, the "batch means" technique was used, with 30 batches.

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 utilization factors, ρ_l to their desired values. The segment size is assumed constant, equal to 1460 bytes. The amount of data to be transferred by each connection (i.e., the file size) is expressed in number of packets. We consider a mixed traffic scenario where the file size follows the distribution shown in Fig. 3, which is derived from measurements [3]. 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 inset reports the same distribution taking 1460 bytes as unit. We use the most recent measurements in the following simulations.

3.1 Single-bottleneck topology

We start by considering a very simple single bottleneck topology. We assume oneway TCP New Reno connections with not congested backward path. Table 1 reports the capacity and

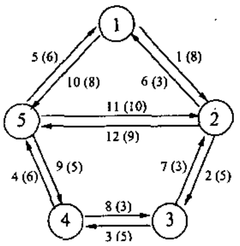
Table 1. Design results for Bottleneck Network

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

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 [1], which considers an $M/M/1$ queue model in the CA problem. The BA problem is solved considering $M/M/1/B$ queues. We choose as target parameters the following: latency $\leq$ 0.3s for flows shorter than 20 segments, throughput larger than 512kbps for flow longer than 20 segments and $Ploss = 0.01$. Using the transport layer QoS translator, we obtain the equivalent constraints $RTT \leq 0.03s$, which corresponds to meet the most stringent latency constraint. We imposed these same constraints also in the classical approach. Looking at the CA solution, we observe that our methodology requires a much higher data rate than the classical approach, as shown by the average link utilization $\rho_l = 0.64$. Also when considering the B_l design, we observe that the adoption of the $M_{[X]}/M/1/B$ model leads to larger buffer requirements than the with a simpler $M/M/1/B$ model. Table 2 reports flow latencies for different flow sizes (in number of segments), as estimated by the CSA model (second column) and as observed by simulations. Results are shown considering both our approach and the classical methodology, and by considering either droptail or RED buffers (whose parameters have been set according to [14]). 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. 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

Table 2. Simulation Latency for Bottleneck Network

		$M_{[X]}/M/1/B$		$M/M/1/B$	
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 4. 5-Node Network : topology and traffic requirements

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 impose a very large packet dropping probability.

3.2 Multi-bottleneck topologies

As a second example, we present results obtained considering the multi-bottleneck mesh network shown in Fig. 4. The network topology comprises 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. Fig. 4 reports link identifiers, link routing weights (in parenthesis), and the traffic requirements matrix Γ. Routing weights are chosen in order to have one single path for every source/destination pair.

We considered the same QoS target constraints for all source/destination pairs, which are: (i) file latency smaller than 0.5s for TCP flows shorter than 20 segments, and (ii) throughput larger than 512kbps for TCP flows longer than 20 segments. Selecting $Ploss = 0.01$, we obtain as design constraint $RTT \leq 0.07$s (Fig. 2).

Table 3 reports the link capacities, link utilizations, and buffer sizes obtained with our method. We also report in the same table the average packet delay $E[T]$, average packet size $E[N]$, and the average packet loss probability $Ploss$, computed using the $M_{[X]}/M/1/B$ queue model.

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 asses the quality of the design results, we ran ns-2 simulations for droptail and RED buffers.

In order to verify the end-to-end 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). Fig. 5 plots the file transfer latency for all flow size classes for the selected source/desti nation pair. The QoS constraint of 0.5s for the maximum latency is also reported. 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.

To complete the evaluation of our methodology, we compare the link utilization factor and buffer size obtained when considering the classical algorithm, i.e., by using an $M/M/1/B$ queueing model. Fig. 6 shows the link utilizations (left plot) and buffer sizes (right plot) obtained with our method and with the classical approach. It can be immediately noticed that considering the burstiness of TCP/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. 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. As a second example of multi-bottleneck topology we chose a network comprising 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 0.5 ms, i.e., link lengths vary between 15km and 150km. The traffic requirement matrix is set to obtain an average link flow of about 15 Mbps; however, in order to observe the impact of traffic load on our design methodology, we consider also two scaled version of the traffic matrix. We considered the same design target parameters as for the previous example. Fig. 7 shows the link utilizations obtained with our method, considering the three different traffic loads. In the plot, links are sorted in increasing utilization factor. 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.

Table 3. Design results for 5-Node Network

5-Node Network - $M_{[X]}/M/1/B$						
$Link$	C [Mbps]	ρ	B	$E[T]$	$E[N]$	$Ploss$
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	

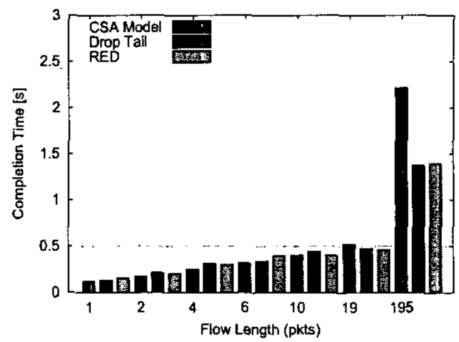


Figure 5. Model and Simulation Results for latency; 3-Link Path from the 5-Node Network

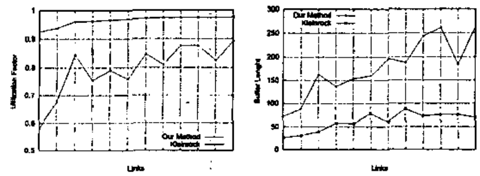


Figure 6. Link Utilization Factor and Buffer Size for a 5-Node Network

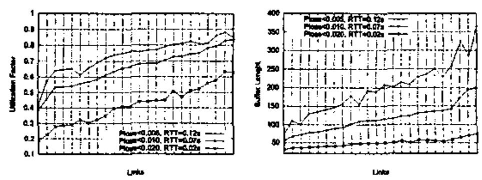


Figure 8. Link Utilization Factor and Buffer Size for a 10-Node Network (considering different target packet loss probabilities)

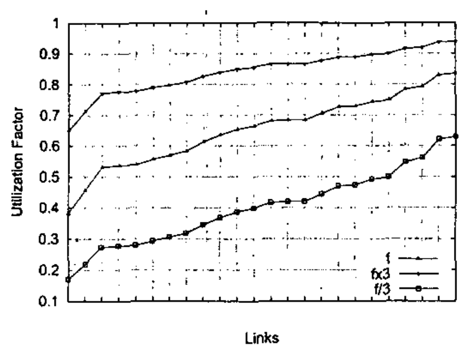


Figure 7. Link Utilization Factor for a 10-Node Network for different traffic requirements

Performing packet level simulations in this scenario, we observe that the target QoS constraints are met in all cases.

Finally, Fig. 8 shows link utilizations and buffer sizes considering different packet loss probability values, while keeping fixed the file transfer latency $\leq$ 2s and throughput $\geq$ 512kbps (average link flow =15Mbps). Obviously, this forces the transport layer QoS translator to reduce the RTT to meet the QoS constraints. Looking at the left plot, we observe that the utilization factor decreases for decreasing RTT constraints. More interesting is the effect of selecting different values of P_{loss} on buffer sizes (right 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.

3.3 Computational complexity

Finally, we briefly discuss the computation times needed to solve the CA problem. The solver algorithm was implemented in 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).

4 Conclusion

In this paper, 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. A second important improvement with respect to traditional packet network design approaches lies in the fact that we considered more realistic packet traffic models, accounting for both long-lived and short-lived TCP connections, and considering more complex systems of queues. Examples of application of the proposed design methodology to different networking configurations have shown the effectiveness of our approach.

REFERENCES

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

[2] K. Claffy, Greg Miller, and Kevin Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone". In *Proceedings of INET '98*, July 1998.

[3] M. Mellia, A. Carpani, R. Lo Cigno, "Measuring IP and TCP behavior on Edge Nodes", *IEEE Globecom 2002*, Taipei, TW, Nov. 2002.

[4] 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.

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

[6] K. Kamimura and H. Nishino. "An efficient method for determining economical configurations of elementary packet-switched networks". *IEEE Transactions on Communications*, Vol.39, N.2, pp. 278-288, Feb. 1991.

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

[8] M. Garetto, D. Towsley, "Modeling, Simulation and Measurements of Queuing Delay under Long-tail Internet Traffic", *ACM SIGMETRICS 2003*, San Diego, California, June 10-14, 2003

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

[10] H. Knoche and H. de Meer, "Quantitative QoS Mapping: A Unifying Approach", In *Proc. of the 5th Int. Workshop on Quality of Service (IWQoS97)*, pp. 347-358, New York, May 1997.

[11] N. Cardwell, S. Savage, T. Anderson, "Modeling TCP Latency", *Infocom 2000*, Tel Aviv, Israel, March 2000.

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

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

[14] E.Wille, *Design and Planning of IP Networks Under End-to-End QoS Constraints*, PhD dissertation, Politecnico di Torino, Available at http://www.tlc-networks.polito.it/mellia/papers/wille_phd.pdf

[15] G.L. Nemhauser, A. H. G. Rinnooy Kan, M. J. Todd, *Optimization*, Amsterdam : North-Holland, 1989