Design and Analysis of IP Networks with End-to-End QoS Guarantees

Emilio C. G. Wille[1−3], Marco Mellia[2], Emilio Leonardi[2], Marco Ajmone Marsan[2]

***Abstract*— Traditional approaches to optimal design and planning of packet networks focus on the network-layer infrastructure. In this paper we describe a simple methodology to tackle the packet network design problem, considering as constraints the end-to-end Quality of Service (e2e QoS) metrics, and we illustrate its application to the optimization of link capacities in a corporate Virtual Private Network (VPN). We use 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.**

***Index Terms*— Networks design and planning, TCP/IP, queueing theory, optimization**

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].

The traditional approaches to optimal design and planning of packet networks focus on the network-layer infrastructure, thus neglecting end-to-end Quality of Service (e2e QoS) issues, and Service Level Agreement (SLA) guarantees. This is quite inappropriate, since the Internet today carries a wide range of critical telecommunication services, and design approaches based on end-to-end QoS are a must.

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. 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) [2], [3].

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 [4], but a higher degree of correlation exists. 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 Long Range Dependent (LRD) behaviors, which cause queue dynamics to severely deviate from the above model predictions. For these reasons, the usual approach of modeling packet networks as networks of M/M/1 queues [5], [6], [7] appear now inadequate for the design of such networks.

In [8], the authors for the first time abandon the Markovian assumption in favor of a LRD traffic model, i.e., a Fractional Brownian Motion model. They solve the discrete Capacity Assignment (CA) problem under network e2e delay constraints only, using Simulated Annealing metaheuristic. 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.

The challenge in the area of network design is how to devise 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 e2e QoS and, at the same time, consider the traffic dynamics of today packet networks.

In this paper, we propose a packet network design and planning approach that considers 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. 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 refined IP traffic modeling technique, already presented in [9], that is both simple and capable of producing accurate performance estimates for general-topology packet networks loaded by realistic traffic patterns.

We present a nonlinear programming formulation for the continuous Capacity Assignment (CA) problem and solve it in the case of corporate Virtual Private Networks (where the capacity is leased from a long distance carrier, and costs are directly derived from the leasing fees). 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].

This paper is organized as follows. Section II 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 II-A. Section II-B lists the assumptions needed for the modeling phase, and discusses CA and BA problems. Results obtained for both problems are tabulated and compared with results of ns-2 simulations in Section III. Conclusions are given in Section IV.

II. IP Network Design Methodology

The IP network design methodology that we propose in this paper is based on a "Divide and Conquer" approach. Fig. 1

[1]CEFET/PR - CPGEI, Curitiba-PR, Brazil. E-mail: ecgwille@cefetpr.br

[2]Politecnico di Torino - DELEN, Torino, Italy. E-mail: mellia, emilio, ajmone@mail.tlc.polito.it.

[3]The first author was supported by a CAPES Foundation scholarship from the Ministry of Education of Brazil.

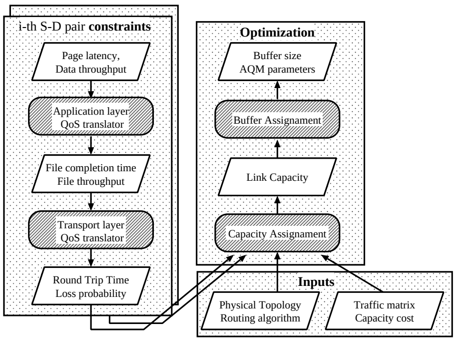


Fig. 1. Schematic Flow Diagram of the Network Design Methodology

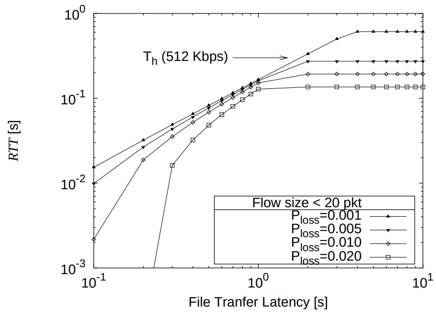


Fig. 2. RTT Constraints as Given by the Transport Layer QoS Translator

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, 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.

A. QoS translators

The process of translating QoS specifications between different layers of the protocol stack is called QoS translation or QoS mapping. 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}).

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) 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* (L_t), or the minimum *file transfer throughput* (T_h). 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 *flow 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 [9] 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). When considering throughput, we instead exploit the formula in [12], referred as PFTK model.

There are at least two 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}.

B. Optimization formulation and solution

1) Network model: In order to obtain a useful formulation of the CA 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).

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 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 TCP/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 for a given flow length distribution [9]. 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 CA optimization.

2) Problem formulation: The decision of fixing "a-priori" the loss probability allows us to decouple the CA solution from the BA solution. We first solve the CA problem (properly selecting the capacity of links) 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 CA and BA problems:

C_l	the capacity of link l.
f_l	the average data flow on link l.
d_l	the physical length of link l.
RTT_{sd}	the Round Trip Time of path r_{sd}.
B_l	the buffer size of link l.
$\delta_l(r_{sd})$	indicator function (which is one if link l is in path r_{sd} and zero otherwise).
$P_{loss}(r_{sd})$	the desired e2e loss probability for path r_{sd}.
γ_{sd}	the traffic offered on path r_{sd}.

a) The Capacity Assignment problem: Different formulations of the CA problem result by selecting i) the cost functions $g_l(C_l)$, ii) the routing model, and iii) the capacity constraints. In this paper we focus on the VPN case, in which common assumptions are i) linear cost, i.e., $g_l(C_l) = d_l C_l$, ii) non-bifurcated routing, and iii) continuous capacities.

For each source/destination pair (s, d), the 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) $r_{sd} = p_{sd} \cup p_{ds}$.

As previously said, we solve the CA problem by considering infinite size 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 routing, it is possible to formulate the CA problem as follows.

$$Z_{CA} = min \sum_l g_l(C_l) \tag{1}$$

subject to:

$$\frac{K_1}{\mu} \sum_l \frac{\delta_l(r_{sd})}{C_l - f_l} \leq delay(r_{sd}) \quad \forall\,(s,d) \tag{2}$$

$$delay(r_{sd}) = RTT_{sd} - \tau_{sd} - \tau_{ds} \quad \forall\,(s,d) \tag{3}$$

$$f_l = \sum_{(s,d)} \delta_l(r_{sd}) \gamma_{sd} \quad \forall\, l \tag{4}$$

$$C_l \geq f_l \geq 0 \quad \forall\, l \tag{5}$$

The objective function (1) represents the total link cost, which is the sum of the cost functions of link l, $g_l(C_l)$. Equation (2) 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 (see section II-A) minus the propagation delay of the route.

The average queueing delay is expressed by considering an $M_{[X]}/M/1/\infty$ queue [13]:

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

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

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. The average 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. We consider as traffic offered to the network $\gamma_{sd} = \hat{\gamma}_{sd}/(1 - P_{loss})$, thus accounting for the retransmissions due to the losses that flows experience along their path to the destination.

Constraints (5) are non–negativity constraints.

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

b) The Buffer Assignment problem: As final step in our methodology, we need to dimension buffer sizes, i.e., to solve the following problem:

$$Z_{BA} = min \sum_l h_l(B_l) \tag{8}$$

subject to:

$$\sum_l \delta_l(r_{sd}).p(B_l, C_l, f_l, [X]) \leq P_{loss}(r_{sd}) \quad \forall\,(s,d) \tag{9}$$

$$B_l \geq 0 \quad \forall\, l \tag{10}$$

The objective function (8) represents the total buffer cost, which is the sum of the cost functions of buffer l, $h_l(B_l) = B_l$. Equation (9) is the loss probability constraint for each source/destination pair. Constraints (10) are non–negativity constraints. $p(B_l, C_l, f_l, [X])$ is the average loss probability for the $M_{[X]}/M/1/B$ queue, which is evaluated by solving the CTMC.

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 $M/M/1/B$

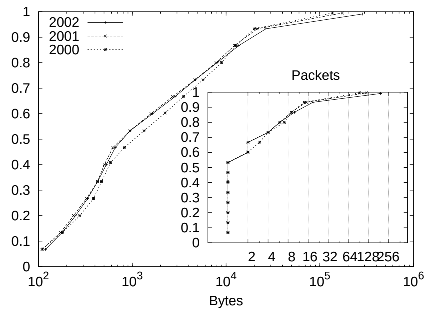
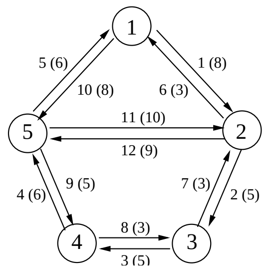


Fig. 3. TCP Connection Length Cumulative Distributions.



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]

Fig. 4. 5-Node Network : Topology and Traffic Requirements.

queue, $p(B, C, f)$ is a convex function (see [14]); 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 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 [15].

c) 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 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 $B_l = \alpha.max\_th,\ \alpha > 1$, e.g., $\alpha = 2$ as suggested in the "gentle" variation of RED.

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

$$p(B_l, C_l, f_l, [X]) = \frac{E_l[N] - min\_th_l}{max\_th_l - min\_th_l} max\_p_l \qquad (11)$$

Note that (11) 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 in (9). Finally, we set $min\_th_l = \beta.max\_th_l,\ \beta < 1$. In the numerical examples that follow, we selected $\alpha = 2,\ \beta = 1/16$.

III. 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. In order to validate our approach, we ran simulation experiments using the software ns-2.

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 Fig. 3, which is derived from one-week long measurements, conduced in [3], 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 inset 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.

A. Multi-bottleneck topologies

As a first 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 parentheses), and the traffic requirements matrix $\hat{\Gamma}$. Routing weights are chosen in order to have one single path for every source/destination pair. A number of peripheral links (not shown in the picture) are attached to each node. These links are not congested, being their capacities equal to 30 Mbps, and their propagation delays are uniformly distributed between 0.01 and 0.03ms.

We choose as target parameters the following: latency $L_t \leq$ 0.3s 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 $RTT_{sd} = RTT,\ \forall\ s, d$), which corresponds to meet the most stringent latency constraint (Fig. 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). As results, 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.

In order to obtain some comparisons, we also implemented a design procedure using the classical formula, see [1], which

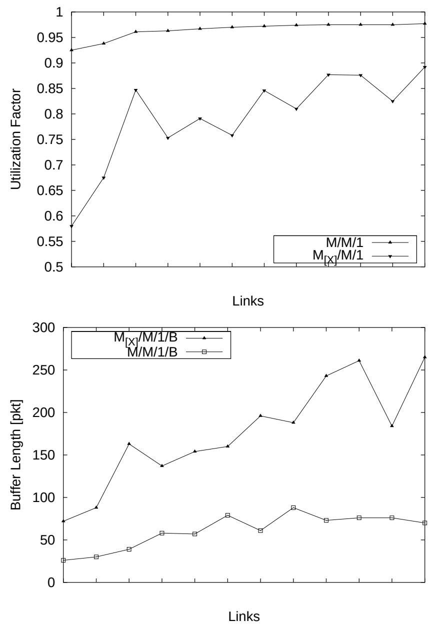


Fig. 5. Link Utilization Factor and Buffer Size for a 5-Node Network.

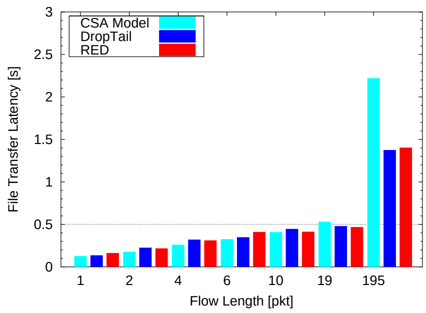


Fig. 6. Model and Simulation Results for Latency; 3-Link Path from the 5-Node Network.

considers an $M/M/1$ queue model in the CA problem. We also extended the classical approach to the BA problem, which is solved considering $M/M/1/B$ queues. We imposed these same constraints also in the classical approach. In Fig. 5, 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 asses the quality of the design results, we ran ns-2 simulations for droptail and RED buffers. 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. 6 plots the file transfer latency for all flow size classes for the selected source destination pair. The QoS constraint of 0.5s for the maximum latency is also reported. 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 Fig. 2). It is important to observe that a network dimensioned using the classical approach cannot satisfy all the QoS constraints.

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 0.5ms, 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 of traffic load and performance constraints on our design methodology, we consider different numerical experiments.

Fig. 7 shows the range of network link utilizations versus traffic load (top 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 on the bottom 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 Fig. 8 ($P_{loss} = 0.01$, link traffic load = 15 Mbps). Notice that, in order to satisfy a very tight constraint of file transfer latency smaller than 0.2s for all flows shorter than 20 segments imposes a utilization close to 20% on some particularly congested links (top plot). Tight constraints mean packet delays with small values and thus greater 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 figure also shows, on the left, a boundary which corresponds to the propagation delay (considering the longest network path) translated to a transport layer performance indicator. On this boundary the packet delay is zero, thus link capacities tend to infinite. On the right, we have a second boundary that divides the figure in two regions. The file transfer latency is the dominant constraint on the left region. The behavior of buffer sizes versus file transfer latency requirements is shown on the bottom plot.

Finally, Fig. 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 = 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 (top plot).

More interesting is the effect of selecting different values of P_{loss} on buffer sizes (bottom 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.

Simulations using ns-2 confirm that the target QoS constraints are met in all cases.

IV. Conclusion

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

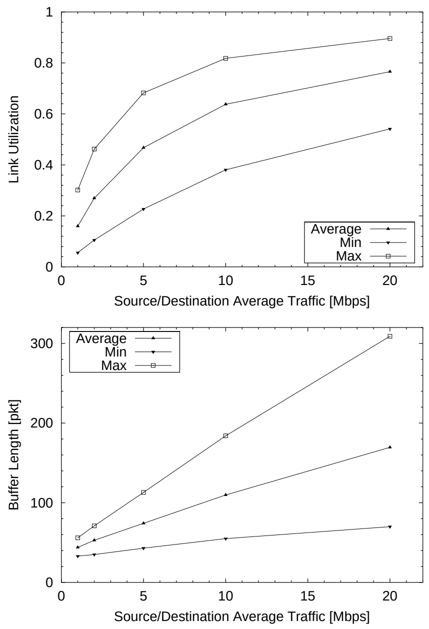


Fig. 7. Link Utilization Factor and Buffer Length for a 10-Node Network (considering different source/destination traffics).

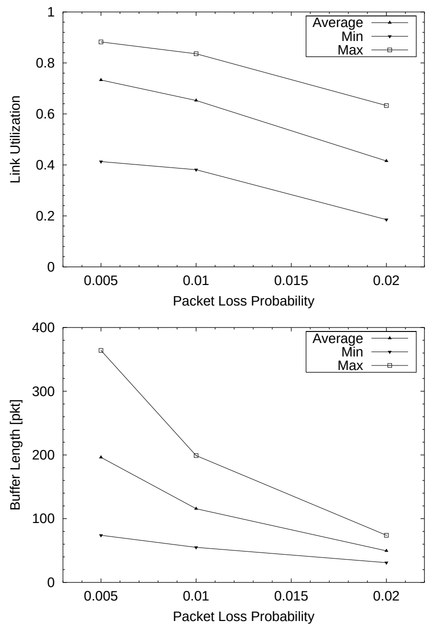


Fig. 9. Link Utilization Factor and Buffer Length for a 10-Node Network (considering different target packet loss probabilities).

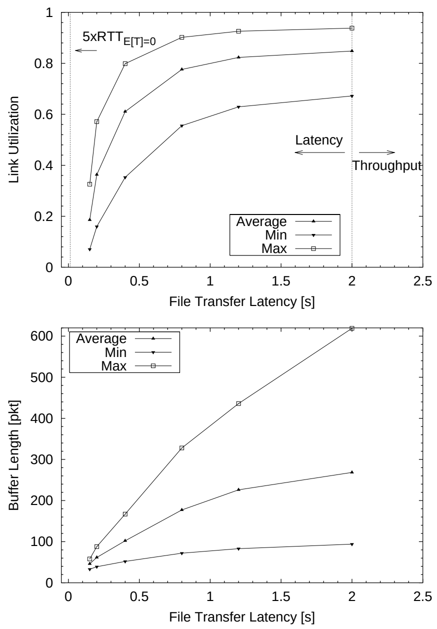


Fig. 8. Link Utilization Factor and Buffer Length for a 10-Node Network (considering different target file transfer latencies).

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, which model a communication network as a Jackson queueing network, thus assuming packet flows to be Poisson, 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 [9].

Examples of application of the proposed design methodology to different networking configurations have shown the effectiveness of our approach.

REFERENCES

[1] M. Gerla, 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 Trans. on Networking*, Vol.3, N.3, Jun. 1995.

[5] 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.

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

[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 Trans. on Networking*, Vol.1, N.4, 1993.

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

[12] J.Padhye, V.Firoiu, D.Towsley, J.Kurose, “Modeling TCP Reno performance: a simple model and its empirical validation”, *I*EEE/ACM Trans. on Networking, Vol.8, N.2, pp.133–145, Apr. 2000.

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

[14] 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.

[15] M.Wright, ”Interior methods for constrained optimization”, *Acta Numerica*, Vol.1, pp.341–407, 1992.