Dimensioning Methods for IP Networks under End-to-End QoS Constraints

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

[1]Department of Electronics – Paraná Federal University of Technology
Av. Sete de Setembro 3165, Curitiba (PR), Brazil

[2]Department of Electronics – Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ewille@utfpr.edu.br, {mellia, emilio, ajmone}@polito.it

Abstract. *This paper considers the QoS design of packet–switching networks. 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. Subproblems are derived from a general design problem and a collection of heuristic algorithms are introduced for computing approximate solutions. We illustrate examples of network planning/dimensioning considering Virtual Private Networks (VPNs).*

1. Introduction

The Internet became a very complex global network which represents an opportunity to provide worldwide value-added services requiring a certain level of quality-of-service (QoS) such as QoS-VPN (Virtual Private Network), VoD (Video on Demand), e-learning, e-Commerce, etc. While at the first moment, the objective was to provide more and more bandwidth to attract users and allow them to use freely bandwidth consuming services, the recent results have shown that this approach was not necessarily the more efficient one. It is technically a challenging and complicated problem to deliver multimedia information in a timely, synchronized manner over a decentralized, shared network environment, especially one that was originally designed for best-effort traffic such as the Internet.

Accordingly, a key issue in this area is how to devise reasonable packet–switching network design methodologies that allow the choice of the most adequate set of network resources for the delivery of a given mix of services with the desired level of end-to-end (e2e) QoS and, at the same time, consider the traffic dynamics of today's packet–switching networks.

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 the Transmission Control Protocol (TCP), the reliable transport protocol of the Internet, whose congestion control algorithms dictate the latency of information transfer.

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 [Paxson and Floyd 1995]. However, the traffic flowing in IP networks is known to exhibit Long Range Dependent (LRD) behaviors, which causes

queue dynamics to severely deviate from the traditional (e.g., $M/M/1$ or $M/M/1/B$) model predictions. For these reasons, the usual approach of modeling packet–switching networks as networks of $M/M/1$ queues [Cheng and Lin 1995, Rolland et al. 1999, Gersht and Weihmayer 1990] appears now inadequate for the design of such networks. Recently, in [Fraleigh et al. 2003], the authors for the first time abandon the Markovian assumption in favor of a fractional Brownian motion model, i.e., an LRD traffic model. Unfortunately, it is difficult to extend this approach to consider more general network problems, because the relation among traffic, capacity and queueing delay is not expressed by a closed–form expression.

In this paper, we focus on several types of problems that arise when dealing with packet–switching networks design. We consider the traffic 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. Of course, in any realistic network problem an "optimal design" is an extremely difficult task. In [Wille et al. 2004, Wille et al. 2005] is proposed an IP network design methodology which is based on a "Divide and Conquer" approach, in the sense that it consists of subtasks, which are solved separately. There are two main tasks, which correspond to i) the process of translating QoS specifications between layers of the protocol stack (performed by the *QoS translators*), and ii) the dimensioning process (performed by a suitable *constrained optimization procedure*). Thanks to the QoS translators, all the user-layer QoS constraints are mapped into lower-layer performance constraints, down to the network layer, where performance metrics are typically expressed in terms of average delay and loss probability. The optimization procedure needs as inputs the description of the physical topology, the traffic matrix, and the cost as function of link capacities. The objective of the optimization is to find the minimum cost solution that satisfies the user-layer QoS constraints. A second important point of the proposed methodology is the adoption of a refined TCP/IP traffic modeling technique that is both simple and capable of producing accurate performance estimates for packet–switching networks subject to realistic traffic patterns. The main idea behind this approach corresponds to reproduce the effects of traffic correlations on network queueing elements by means of Markovian queueing models with batch arrivals.

The rest of the paper is organized as follows. Section 2. briefly describes the QoS translation problem. Section 3. outlines the network and queueing models, and provides the formulation of the related optimization problems. It also introduces heuristic algorithms for computing approximate solutions, and discusses numerical and simulation results. Finally, Section 4. summarizes the main results obtained in this research.

2. QoS Translation

The process of translating QoS specifications between different layers of the protocol stack is called QoS translation. According to the Internet protocol architecture, at least two QoS translating procedures should be considered. The *Application-Layer QoS translator* translates the application–layer QoS constraints (e.g., web page transfer latency, data throughput, audio quality, etc) into transport–layer QoS constraints. Given the multitude of Internet applications it is not possible to devise a generic procedure to solve this problem. Hence, ad-hoc solutions depending on the application must be used.

The *Transport-Layer QoS translator* translates transport–layer QoS constraints

into network–layer QoS constraints, such as *Round Trip Time* (RTT) and *packet loss probability* (P_{loss}). This process is more difficult mainly due to the complexity of the TCP protocol, which implements 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., 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. The approach is based on the numerical inversion of analytic TCP models, taking as input either the file transfer throughput or latency, and obtaining as outputs RTT and P_{loss}. Among the many models of TCP presented in the literature, we used the TCP latency model described in [Cardwell et al. 2000]. We will refer to this model as the CSA model (from authors' name). When considering throughput, we instead exploit the formula in [Padhye et al. 2000], referred as the PFTK model (from authors' name). Here, the numerical inversion is just a root finding procedure. There are at least two parameters that affect TCP performance, 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 fact 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 as a function of file latency and throughput.

In this paper, we consider a mixed traffic scenario where data files are exchanged with the *file size distribution* related in [Mellia et al. 2002]. This distribution, obtained by one–week long measurements, says that 85% of all TCP flows are shorter than 20 packets. Considering this distribution, given the file transfer latency and a fixed throughput it is possible to evaluate the maximum admissible RTT which satisfies the most stringent constraint for different values of P_{loss} [Wille et al. 2004, Wille et al. 2005].

3. Problem Statement

The network infrastructure is represented by a graph $G = (V, E)$ in which V is a set of nodes (with cardinality n) and E is a set of edges (with cardinality m). A node represents a network router and an edge represents a physical link connecting one router to another. The output interfaces of each router is modeled by a queue with finite buffer. For a given link (i, j), the flow f_{ij} is defined as the quantity of information transported by this link, while its capacity C_{ij} is a measure of the maximal quantity of information that it can transmit (both are given in bits per second – bps). Each buffer can accommodate a maximum of B_{ij} packets, and d_{ij} is the link physical length.

According to [Paxson and Floyd 1995], IP packets do not arrive at router buffers following a Poisson process. However, there is a correlation degree, which can be partly due to the TCP control mechanisms. In order to consider the traffic burstiness induced by TCP we choose a specific kind of queue to model each router inside the network topology. Thus, we choose the $M_{[X]}/M/1/\infty$ queue, i.e., a Markovian queue with *batch arrivals* [Chao et al. 1999]. The batch size varies between 1 and W with distribution $[X]$, where W is the maximum TCP window size expressed in segments. Given the flow length distribution, a stochastic model of TCP (described in [Garetto and Towsley 2003]) is used to obtain the batch size distribution $[X]$. A wide range of investigations performed in [Garetto and Towsley 2003] certify the accurate network layer performance estimates

by considering $M_{[X]}/M/1/B$ models. Hence, the *average packet delay* is given by the following expression [Chao et al. 1999] (notice that the subscript (ij) was dropped for simplicity):

$$E[T] = \frac{K}{\mu}\frac{1}{C-f} \qquad \text{with} \qquad K = \frac{m'_{[X]} + m''_{[X]}}{2m'_{[X]}} \tag{1}$$

where $m'_{[X]}$ and $m''_{[X]}$ are the first and second moments of the batch size distribution $[X]$.

We consider that packet lengths are exponentially distributed with mean $1/\mu$ (bits/packet). Additionaly we define the arrival rate $\lambda = \mu.f$ (packets/s), and the link utilization factor $\rho = f/C$.

The average traffic requirements between nodes are represented by a traffic matrix $\hat{\Gamma} = \{\hat{\gamma}_{sd}\}$, where the traffic $\hat{\gamma}_{sd}$ between a node pair (s, d) represents the average number of bps sent from source s to destination d. The traffic routing and the traffic requirements uniquely determine the flow of each link. Thus, a link flow results from the sum of the traffics that are routed on this link. We consider that for each source/destination pair, the traffic is transmitted over exactly one directed path in the network (non-bifurcated routing).

We now can state the general network design problem as follows: consider that we are given the locations of the network routers, the traffic flow requirements, and the link and buffer costs. Our design task is to choose a topology, to select the capacity of the links in this topology, and to design a routing procedure for the traffic from its origins to its destinations, in a way which optimizes an objective function while meeting all the system (QoS and reliability) constraints. As reliability constraint we consider that all traffic must be exchanged even if a single node fails (2-connectivity), and the QoS constrains correspond to maintain the e2e packet delay for each network source/destination pair below a maximum tolerable value. When explicitly considering TCP traffic it is also necessary to tackle the Buffer Assignment (BA) problem, which corresponds to dimension buffer sizes subject to packet loss probability constraints.

The above stated problem is intractable. The number of topologies to consider is too large and, in addition, we have a multicommodity flow problem. Subproblems can be derived from this general problem and solved separately, in a way to obtain feasible solutions to the general problem. Hence, we may now define three subproblems that differ only in the set of permissible design variables. It is important to note that for a given subproblem a specific optimization technique must be applied to solve it.

3.1. The Capacity Assignment problem

In this subsection we focus on the Capacity Assignment (CA) problem, i.e., the selection of the link capacities. The decision of fixing *a–priori* the loss probability allows us to decouple the CA problem from the BA problem. We first solve the CA problem considering the e2e delay constraints only. Then, we enforce the loss probability to meet the P_{loss} constraints by properly choosing buffer sizes. Different formulations of the CA problem result by selecting i) the cost functions, ii) the routing model, and iii) the capacity constraints. In the VPN case common assumptions are i) linear costs, ii) non-bifurcated routing, and iii) continuous capacities. Given the network topology, the traffic requirements, and the routing, the CA problem corresponds to minimize the network cost subject to the

maximum allowable e2e packet delay.

$$Z_{CA} = min \sum_{i,j} g(d_{ij}, C_{ij}) \tag{2}$$

subject to:

$$K_1 \sum_{i,j} \frac{\delta_{ij}^{sd}}{C_{ij} - f_{ij}} \leq RTT_{sd} - \tau_{sd} - \tau_{ds} \quad \forall\, (s,d) \tag{3}$$

$$f_{ij} = \sum_{s,d} \delta_{ij}^{sd} \hat{\gamma}_{sd} \qquad \forall\, (i,j) \tag{4}$$

$$C_{ij} \geq f_{ij} \geq 0 \qquad \forall\, (i,j) \tag{5}$$

The objective function (2) represents the total link cost, which is a linear function of both the link capacity and the physical length, i.e., $g(d_{ij}, C_{ij}) = d_{ij}C_{ij}$. Equation (3) is the e2e packet delay constraint for each source/destination pair. It says that the total amount of delay experienced by all the flows routed on a path should not exceed the maximum RTT (see section 2.) minus the propagation delay τ of the route. δ_{ij}^{sd} is an indicator function which is one if link (i,j) is in path (s,d) and zero otherwise. Here, $K_1 = K/\mu$. Non–bifurcated routing model is used where the traffic will follow exactly one path from source to destination. Equation (4) defines the average data flow on the link. Constraints (5) are non–negativity constraints.

We notice that the above stated CA problem is a convex optimization problem [Wille et al. 2004], and its global optimal can be found using standard convex programming techniques, for example, the *logarithm barrier method* [Wright 1992]. However, these algorithms are time–consuming. A fast suboptimal solution to this problem can be found using the following heuristic.

3.1.1. Suboptimal solution to the CA problem

The main idea is to decompose the problem into $n \times (n-1)$ single constrained problems (one for each path (s,d)). Let I_{sd} be the set of links which compose path (s,d), and let C_{ij}^{sd} be an auxiliary variable which corresponds to the capacity of the link (i,j) when considering the path (s,d). To solve each single path problem we apply the Lagrangean multiplier method obtaining:

$$L(\psi) = min\Big[\sum_{(i,j)\in I_{sd}} d_{ij} C_{ij}^{sd} + \psi\Big(\sum_{(i,j)\in I_{sd}} \frac{1}{C_{ij}^{sd} - f_{ij}} - b_{sd}\Big)\Big] \tag{6}$$

subject to:

$$C_{ij}^{sd} \geq f_{ij} \geq 0 \qquad \forall\, (i,j),\ \forall\, (s,d) \tag{7}$$

where:

$$b_{sd} = \frac{1}{K_1}(RTT_{sd} - \tau_{sd} - \tau_{ds}) \qquad \forall\, (s,d) \tag{8}$$

The solutions to this problem are given by:

$$C_{ij}^{sd} = f_{ij} + \frac{\sum_{(k,l)\in I_{sd}} \sqrt{d_{kl}}}{b_{sd}\sqrt{d_{ij}}} \tag{9}$$

Knowing the values for the variables C_{ij}^{sd} (in the single path problem) we obtain admissible values for the capacities C_{ij} (in the original CA problem) assigning:

$$C_{ij} = \max_{s,d}\{C_{ij}^{sd}\} \tag{10}$$

3.2. The Buffer Assignment problem

A second step corresponds to dimension buffer sizes, i.e., to solve the following optimization problem:

$$Z_{BA} = \min \sum_{i,j} h(B_{ij}) \tag{11}$$

subject to:

$$\sum_{ij} \delta_{ij}^{sd} p(B_{ij}, C_{ij}, f_{ij}, [X]) \leq P_{loss}, \quad \forall\,(s,d) \tag{12}$$

$$B_{ij} \geq 0, \quad \forall\,(i,j) \tag{13}$$

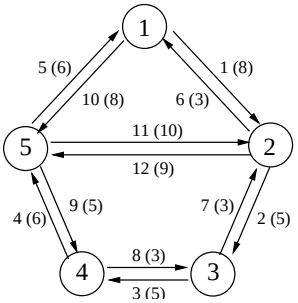
The objective function (11) represents the total buffer cost, which is the sum of the buffer cost functions, $h(B_{ij}) = B_{ij}$. Equation (12) is the loss probability constraint for each source/destination node pair. It says that the total loss probability experienced by all the flows routed on the path (s,d) should not exceed the maximum fixed P_{loss}. Here, $p(B_{ij}, C_{ij}, f_{ij}, [X])$ is the average loss probability for the $M_{[X]}/M/1/B$ queue, which is evaluated by solving its Continuous Time Markov Chain (CTMC). Constraints (13) are non–negativity constraints.

The above stated BA problem is a convex optimization problem [Wille et al. 2004], and its global optimal can be found using standard convex programming techniques.

3.2.1. Numerical Examples and Simulations

We present results obtained considering the mesh network shown in Fig. 1. The network topology comprises 5 nodes and 12 links. In this case, link propagation delays are all equal to 0.5 ms, that correspond to a link length of 150 km. Fig. 1 reports link identifiers, link routing weights (in parentheses), and traffic requirements. Routing weights are chosen in order to have one single path for every source/destination pair. We consider a mixed traffic scenario where the file size (ranging from 1 to 195 packets) follows the distribution related in [Mellia et al. 2002]. We choose, for this case, the following TCP QoS constraints: i) latency $L_t \leq 0.5$ s for files shorter than 20 packets, ii) throughput $T_h \geq 512$ kbps for files longer than 20 packets, and iii) $P_{loss} = 0.01$, using the transport–layer QoS translator we obtain the equivalent network-layer performance constraint $RTT \leq 0.07$ s for all source/destinations node pairs.

We present numerical results, which correspond to the solution of selected CA (and BA) problems (here we used the logarithm barrier method). In order to obtain some comparisons, we also implemented a design procedure using the classical formula which 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. 2, it can be immediately



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 1. 5-Node Network : Topology and Traffic Requirements.

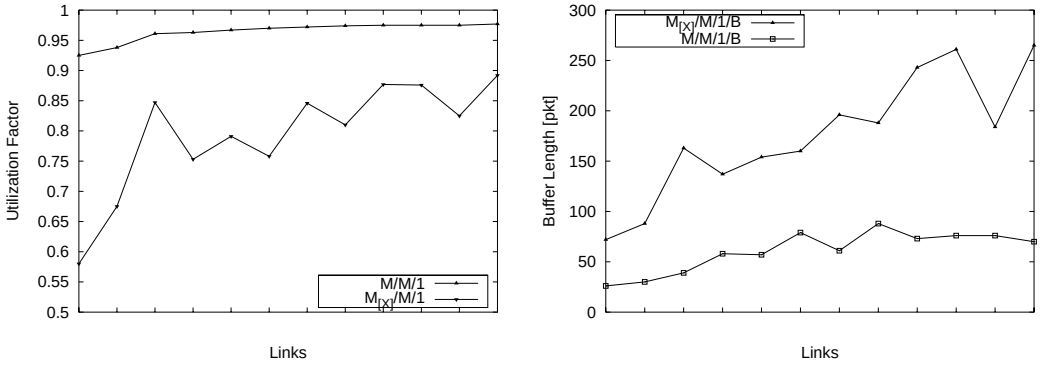


Figure 2. Link Utilization Factor and Buffer Size for a 5-Node Network.

noticed that considering the burstiness of IP traffic radically changes the network design. The link utilization factors have average equal to about $\overline{\rho} = 0.8$, and buffer sizes have average $\overline{B} = 175$, which is about 4 times the average number of packets in the queue (40 packets). Indeed, the link utilizations obtained with our methodology are much smaller than those produced by the classical approach, and buffers are much longer. This is due to the bursty arrival process of IP traffic, which is well captured by the $M_{[X]}/M/1/B$ model. To validate the design methodology, we ran ns-2 [McCanne and Floyd] simulations for droptail and RED [1] buffers. Fig. 3 plots the file transfer latency for all file sizes for a selected source/destination pair (three hops,over links: 8,7,6). The QoS constraint of 0.5 s 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 files shorter than 20 packets. The latency constraint for shorter flows is more stringent than the throughput constraint for longer flows, therefore we obtain a higher value than the minimum desired 512 kbps. Notice that the predicted throughput obtained from the CSA model is a pessimistic estimate. This is due to the limit in the CSA model itself, and not to a mismatch in the network-layer parameters between model and simulation. It is important to observe that a network dimensioned using the classical approach cannot satisfy all the QoS constraints.

3.3. The Capacity and Flow Assignment problem

In this problem the goal is to determine a route for the traffic that flows on each source/destination pair and the link capacities in order to minimize the network cost

[1]Optimal values for RED [Floyd and Jacobson 1993] parameters are obtained according to the procedure given in [Wille et al. 2004, Wille et al. 2005].

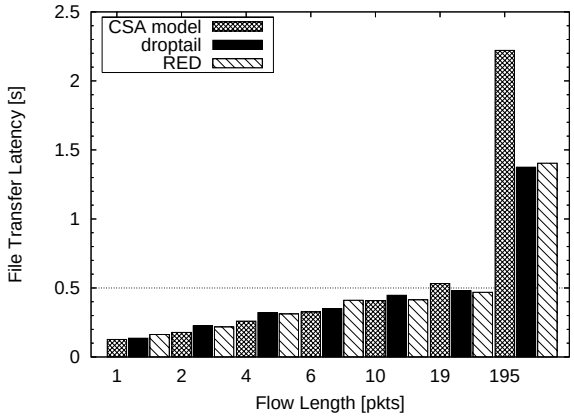


Figure 3. Model and Simulation Results for Latency; 3-Link Path from the 5-Node Network.

subject to the maximum allowable e2e packet delay. Let κ_{ij}^{sd} be a decision variable which is one if link (i, j) is in path (s, d) and zero otherwise. Thus the CFA problem is formulated as the following optimization problem:

$$Z_{CFA} = min \sum_{i,j} g(d_{ij}, C_{ij}) \tag{14}$$

subject to:

$$\sum_{j} \kappa_{ij}^{sd} - \sum_{j} \kappa_{ji}^{sd} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\ (i, s, d) \tag{15}$$

$$K_1 \sum_{i,j} \frac{\kappa_{ij}^{sd}}{C_{ij} - f_{ij}} \leq RTT_{sd} - K_2 \sum_{i,j} \kappa_{ij}^{sd} d_{ij} \qquad \forall\ (s, d) \tag{16}$$

$$f_{ij} = \sum_{s,d} \kappa_{ij}^{sd} \hat{\gamma}_{sd} \qquad \forall\ (i, j) \tag{17}$$

$$C_{ij} \geq f_{ij} \geq 0 \qquad \forall\ (i, j) \tag{18}$$

$$\kappa_{ij}^{sd} \in \{0, 1\} \qquad \forall\ (i, j),\ \forall\ (s, d) \tag{19}$$

The objective function (14) represents total link cost. Constraint set (15) enforce flow conservation, defining a route for the traffic from a source s to a destination d. Equation (16) is the e2e packet delay constraint for each source/destination pair. Equation (17) defines the average data flow on the link. Constraints (18) and (19) are non–negativity and integrality constraints, respectively. Finally, $K_1 = K/\mu$ and K_2 is a constant to convert distance in time.

We notice that this problem is a nonlinear nonconvex mixed–integer programming problem. In [Wille et al. 2005] we proposed a composite upper and lower bounding procedure based on a Lagrangean relaxation of the CFA problem. The purpose is to obtain a relaxed problem, called Lagrangean subproblem, which is easier to solve than the original problem. The objective value from the Lagrangean relaxation problem provides a lower bound (LB), in the case of minimization, for the optimal solution to the original problem. The best lower bound can be derived by solving the Lagrangean dual. To solve the

dual problem we used a subgradient optimization technique [Fisher 1981]. Information obtained from the Lagrangean relaxation is then used by application-dependent heuristics to construct feasible solutions to the original problem, i.e., a primal heuristic (PH). In order to permit some comparisons, we also apply a logarithmic barrier CA solution with minimum-hop routing (MinHop+CA), i.e., we just ignore the routing optimization when solving the CA problem. A new approach is described in the following subsection.

3.4. The Greedy Weight Flow Deviation method

In this section we present a heuristic, based on the classical flow deviation (FD) method, to solve the CFA problem presented in section 3.3.

The main idea is to substitute the *link weights* in the original FD method by $L_{ij} = \frac{d_{ij}C_{ij}}{f_{ij}}$; where the link capacities C_{ij} must be obtained using the CA solver presented in section 3.1.1., in order to enforce e2e QoS delay performance constraints. As our new method relies on the greedy nature of the CA solver algorithm to direct computations toward a local optima, we called it the Greedy Weight Flow Deviation (GWFD) method.

In general the CFA problem admits several local minima. A way to obtain a more accurate estimate of the global minima is restart the procedure using random initial flows. However, we obtained very good results setting as initial trail $L_{ij} = d_{ij}$.

The following is a description in pseudo–code of the GWFD method:

Greedy Weight Flow Deviation method:

Given: feasible f^0 and C^0; $f^* = f^0$; $C^* = C^0$; $p = 0$
    Repeat
        (1) Compute link weights L^p
        (2) Compute minimum–weight paths
        (3) Compute flows f^{p+1}
        (4) Solve CA problem and obtain C^{p+1}
        (5) If $D(C^{p+1}) \geq D(C^p)$ Stop
            Else
              (a) $f^* = f^{p+1}$; $C^* = C^{p+1}$
              (b) $p = p + 1$
            End Else
    End Repeat
End

It must be noted that the problem represented by the formulation (14)-(19) and the problem addressed by the GWFD algorithm are not exactly the same. In fact, the traffic routing solutions resulting from the GWFD algorithm are minimum–weight paths, and those resulting from the CFA formulation are not necessarily minimum–weight paths.

3.4.1. Numerical Examples

In this section we present results obtained considering ten fixed topologies (40-node, 160-link each), which have been generated using the BRITE topology generator [Medina et al. 2001] with the router level option. We consider the same mixed traffic scenario where the file size follows the distribution shown in [Mellia et al. 2002]. Link propagation delays are uniformly distributed between 0.5 ms and 1.5 ms, i.e., link lengths

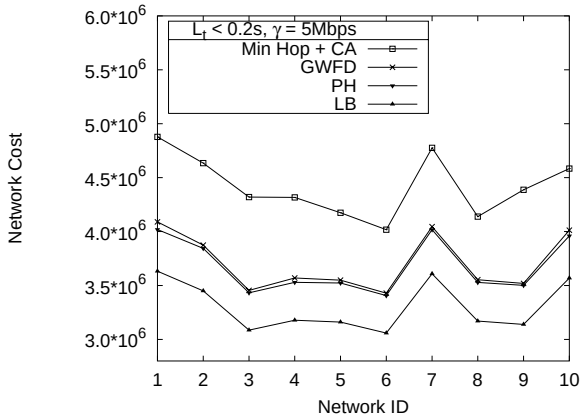


Figure 4. Network Cost for 40-node Network Random Topologies

vary between 100 km and 300 km. Random traffic matrices were generated by picking the traffic intensity of each source/destination pair from a uniform distribution. The average source/destination traffic requirement was set to $\hat{\gamma}_{sd} = 5$ Mbps. For all source/destination pairs, the target QoS constraints are: i) latency $L_t \leq 0.2$ s for files shorter than 20 segments, ii) throughput $T_h \geq 512$ kbps for files longer than 20 segments, and iii) $P_{loss} = 0.001$. Using the transport–layer QoS translator (section 2.), we obtain the equivalent network–layer performance constraint $RTT \leq 0.032$ s for all source/destinations node pairs.

In Fig. 4 the GWFD solutions are compared to solutions from other three techniques (LB, PH, and MinHop + CA) [Wille et al. 2005]. We can observe that the GWFD solutions, for all considered topologies, always fall rather close to the lower bound (LB). The gap between GWFD and LB is about 13%. In addition, the GWFD algorithm is faster than the primal heuristic approach (PH) – only 5 seconds of CPU time are needed to solve an instance with 40 nodes – while it obtains very similar results. Avoiding to optimize the flow assignment subproblem results in more expensive solutions, as shown by the "Min Hop" routing associated with an optimized CA problem. This underlines the need to solve the CFA problem rather then a simpler CA problem.

3.5. The Topology, Capacity and Flow Assignment problem

The Topological, Capacity and Flow Assignment (TCFA) problem can be formulated as follows: given the geographical location of the network nodes on the territory, the traffic matrix, the capacity costs; minimize the total link cost, by choosing the network topology and selecting link flows and capacities, subject to QoS and reliability constraints. As reliability constraint we consider that all traffic must be exchanged even if a single node fails (2-connectivity). There is a tradeoff between reliability and network cost; we note that more links between nodes imply more routes between each node pair, and consequently the network is more reliable; on the other hand, the network is more expensive. Finally, the QoS constrains correspond to maintain the e2e packet delay for each network source/destination pair below a maximum tolerable value. This is a complex combinatorial optimization problem, which can be classified as NP–complete [Garey and Johnson 1979]. Polynomial algorithms which can find the optimal solution for this problem are not known. Therefore, thanks to its good trade-off between solutions

quality and time, in this work we apply genetic algorithms (GAs) to find solutions for the problem. GAs are heuristic search procedures which apply natural genetic ideas such as natural selection, mutations and survival of the fittest.

Our solution approach is based on the exploration of the solution space (i.e., 2--connected topologies) using GA algorithms. As the goal is to design a network that remains connected despite one node failure, for each topology evaluation, actually, we construct n different topologies, that are obtained from the topology under evaluation by the failure of a node each time, and then for each topology we solve its related CFA problem (using the GWFD method). Link capacities are set to the maximum capacity value found so far considering the set of topologies. Using the obtained capacities, the objective function (network cost) is obtained.

3.6. Applying GAs to network design

In the following paragraphs we describe the techniques that were employed in the GA algorithm for topological optimization.

Encoding Scheme: A network topology is represented by an $n \times n$ binary matrix, where n is the number of nodes. A "1" in row i and column j of the matrix stands for an arc from node i to node j and a "0" represents that node i and node j are not connected.

Fitness Evaluation: In this paper, a fitness function (an estimation of the goodness of the solution for the topological design problem) is inversely proportional to the objective function value (cost).

Parent Selection: Parent selection emulates the survival–of–the–fittest mechanism in nature. In this paper *tournament selection* is used, where pairs of individuals are picked at random and the one with the higher fitness (the one which "wins the tournament") is used as one parent. The tournament selection is then repeated on a second pair of individuals to find the other parent from which to breed.

Genetic Operations: *Crossover* is a recombination operator used to produce offsprings. In this paper *single-point* crossover is used. Given two parents a crossover point is randomly selected and the portions of the two chromosomes beyond this point are exchanged to form the offspring. However, in many problems, simply concatenating two substrings of feasible solutions do not produce feasible solutions. In this case, the parents are considered as crossover outputs. *Mutation* is used in order to avoid the convergence of the solutions to "bad" local optima. In our experiments good results were obtained using a mutation operator that simply changes one bit, picket at random, for each produced offspring.

Replacement Strategies: In order to generate a new population we used an *elitist strategy* where once the sons' population has been generated, it is merged with the parents' population according to the following rule: only the best individuals present in both sons' population and parents' population enter the new population.

3.6.1. Numerical Examples and Simulations

In this section we present numerical results considering network designs obtained with the GA approach. We consider the dimensioning of VPN network over a given 10-node,

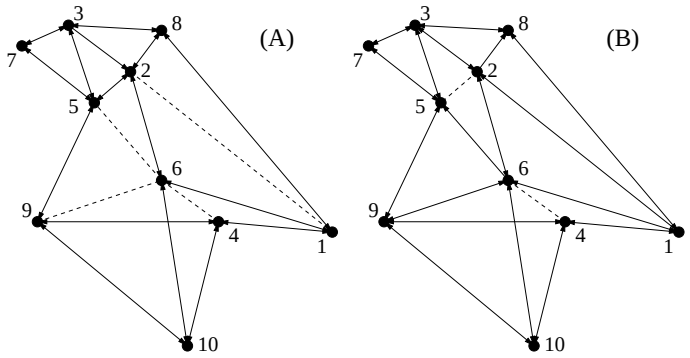


Figure 5. Network Topologies for two Different Traffics Scenarios; (A) uniform distribution, (B) non–uniform distribution

40-link physical topology. The traffic scenario is that one where the file size follows the distribution shown in [Mellia et al. 2002]. Link lengths vary between 140 km and 760 km (average = 380 km). The target QoS constraints for all source/destination pairs are: i) file latency $L_t \leq 1$ s for files shorter than 20 segments, ii) throughput $T_h \geq 512$ kbps for files longer than 20 segments. Selecting $P_{loss} = 0.01$, we obtain a network–level design constraint equal to $RTT \leq 0.15$ s for all source–destination pairs. We analyze the impact of two different traffic scenarios on the obtained network topology.

In the first scenario, source/destination traffic are randomly generated from an uniform distribution with average value $\hat{\gamma}_{sd} = 1$ Mbps. Using the GA approach, the final topology is shown in Fig. 5 (A). Solid lines correspond to the chosen links that synthesize the VPN network topology (dashed lines are existing links, but they are not chosen for the VPN topology). We notice that several connections are needed to guarantee the network 2–connectivity. In the second case, traffic relations are set as follows: two nodes offer an average aggregated traffic equal to 5 Mbps (nodes 3 and 6 in Fig. 5), one node offers 2 Mbps (node 4), and the rest offer traffic equal to 1 Mbps. From Fig. 5 (B) we see that three new links were added in order to drain off the increased traffic from nodes 3 and 6; while a link between nodes 2 and 5 was dismissed.

In this case, we have completed the design of the network shown Fig. 5 (A) by solving its associated BA problem with droptail buffers. Table 1 reports the optimal values for capacities and buffer sizes; it also shows link flows values for two working scenarios: i) normal network operation, and ii) failure of network node 5 (in this case some links must transport an increased traffic flow, f'). We notice that, in this example, the path from nodes 10 to 3 is the same for node 5 working/failure case.

In order to validate the network design, we compare the target performance parameters against the performance measured from very detailed simulation experiments (using the ns–2 simulator). We performed packet–level simulations to check whether the e2e QoS constraints are actually met. As one example, we did *path simulations* considering the path that connects nodes 10, 6, 2, 8 and 3. The results are in perfect agreement with specifications. We notice also that in the case of normal network operation the file transfer latency has smaller values, resulting from the greater gap between link capacities and flows.

Table 1. Dimensioning for a 10-Node Network (values for a 5-link path)

$Link$	f [Mbps]	f' [Mbps]	C [Mbps]	B [pkt]	d [Km]
10–6	35.6	44.7	47.3	656	485
6–2	82.1	67.3	88.6	222	380
2–8	57.3	62.4	67.2	587	155
8–3	43.7	82.6	87.0	756	270

4. Conclusion

In this paper, we have considered the QoS and reliability design of packet networks, presenting mathematical formulations and introducing a collection of heuristic algorithms for computing approximate solutions. Two important elements are considered in our approach: (a) the mapping of the e2e QoS constraints into transport–layer performance constraints first, and then into network–layer performance constraints; and (b) a refined TCP/IP traffic modeling technique that is both simple and capable of producing accurate performance estimates for general–topology packet–switching networks loaded by realistic traffic patterns. By explicitly considering TCP traffic, we also need to consider the impact of finite buffers, therefore facing the Buffer Assignment problem. To the best of our knowledge, no previous work solves packet network design problems accounting for user layer e2e QoS constraints considering more realistic traffic models.

The numerical results have shown that the burstiness of IP traffic radically changes the network design. Indeed, the link utilizations obtained with our approach are much smaller than those produced by the classical approach, and the buffer values are much longer. This is due to the bursty arrival process of IP traffic, which is well captured by the $M_{[X]}/M/1/B$ model. On the other hand, the capacity assignment using the classical approach cannot satisfy all the QoS constraints.

In addition, network costs can be reduced by the jointly optimization of routing and link capacities since they are closely interrelated. For this scope, we have proposed a new CFA algorithm, called GWFD, that is capable of assigning flow and capacities under e2e QoS constraints. The proposed GWFD method is particularly interesting. It can solve CFA instances in a fast and quite accurate way. Based on the GWFD method we have proposed a practical, useful way to solve the topological design problem with e2e QoS and reliability constraints. This approach, which considers the GA metaheuristic, while not necessarily an original idea, represents a pragmatic solution to the problem. In order to validate the proposed methodology, we have compared results against detailed simulation experiments (using the ns-2 software) in terms of network performances. The target QoS performances are met in all cases.

References

V. Paxson, S. Floyd, “Wide-Area Traffic: The Failure of Poisson Modeling,” *IEEE/ACM Transactions on Networking*, Vol. 3, No. 3, pp. 226-244, Jun. 1995.

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

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

A. Gersht, R. Weihmayer, "Joint optimization of data network design and facility selection," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 9, pp. 1667-1681, Dec. 1990.

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

E. Wille, M. Garetto, M. Mellia, E. Leonardi, M. Ajmone Marsan, "Considering End-to-End QoS in IP Network Design," *NETWORKS 2004*, Vienna, Austria, June 13-16, 2004.

E. Wille, M. Garetto, M. Mellia, E. Leonardi, M. Ajmone Marsan, "IP Network Design with End-to-End QoS Constraints: The VPN Case," *19th International Teletraffic Congress*, Beijing, China, August/September 2005.

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

J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Reno performance: a simple model and its empirical validation," *IEEE-ACM Transactions on Networking*, Vol. 8, No. 2, pp. 133-145, Apr. 2000.

M. Mellia, A. Carpani, R. Lo Cigno, "Measuring IP and TCP behavior on Edge Nodes," *Proceedings of IEEE Globecom 2002*, Taipei, Taiwan, Nov. 2002.

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

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

M. Wright, "Interior methods for constrained optimization," *Acta Numerica*, Vol. 1, pp. 341-407, 1992.

S. McCanne, S. Floyd, "NS Network Simulator", Available at http://www.isi.edu/nsnam/ns/.

S. Floyd, V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, 1993.

B. Gendron, T.G. Crainic, A. Frangioni, "Multicommodity Capacitated Network Design". B. Sansò and P. Soriano (eds), *Telecommunications Network Planning*, pp. 1-19, Kluwer, Boston, MA, 1998.

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

M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, CA, 1979.

M. L. Fisher, "The Lagrangean relaxation method for solving integer programming problems", *Management Science*, Vol.27, pp. 1–18, 1981.