Efficient Capacity Provisioning in Delay Constrained IP Networks

Irena Atov, *Member, IEEE,* and Richard J. Harris, *Member, IEEE*

***Abstract*— This paper addresses the problem of capacity provisioning in a QoS enabled IP network, subject to satisfying varying performance requirements for different traffic classes. These performance requirements are viewed in terms of mean end-to-end delays required for the various traffic classes or in terms of random variations of their delays (i.e., jitter) or a combination of both.**

The accuracy of the model is investigated by means of a simulation study over an extended range of test cases for different traffic load characteristics and traffic intensities at the nodes. The results demonstrate the capability of the model in guaranteeing the end-to-end delay requirements for the traffic classes.

***Index Terms*— Multiservice IP networks, capacity provisioning, QoS, QNA.**

I. Introduction

TRAFFIC demands on the Internet today include delay-sensitive traffic (e.g., IP telephony, e-commerce, video, interactive games) that requires better than the standard "best effort" service provided by Internet Protocol (IP) networks. As a result, various new Quality of Service (QoS) mechanisms to support the many different types of traffic - each having varying performance requirements (such as delay bound and/or jitter) - are being developed and implemented (e.g., DiffServ/MPLS) [1], [2]. These new mechanisms are necessary but insufficient to provide any service guarantees. Capacity provisioning is a fundamental requirement and an important precursor to offering service quality. Specifically, determining bandwidth allocations for the various traffic classes on the links is an important design issue as standard Weighted Fair Queueing (WFQ) service disciplines employed in IP QoS networks can only provide *tight* end-to-end delay guarantees for the classes if an adequate level of resources (in terms of bandwidth and buffer space) is allocated along their respective data paths through the network [5], [6].

Generally defined, the capacity provisioning (or capacity assignment (CA)) problem that we consider is the problem of determining bandwidth allocations for the delay-sensitive traffic classes, as well as total bandwidth (capacities) of the links in a QoS enabled IP network so that multiple delay constraints for the traffic classes can be satisfied. The CA problem poses the challenge of performing appropriate link dimensioning in order to balance quality of service against costly overprovisioning. For this problem, an accurate model for describing the characteristics of both the external IP traffic flows and the internal flows (on the links) is required. However, there is a tradeoff between the modelling power of the traffic descriptors used to describe the real traffic and the complexity that they impose when used in network planning functions. In this paper, we present an extended version of a dimensioning model presented in [4] for the solution of the CA problem, which incorporates traffic characterisation procedures that allow burstiness of multi-class IP traffic to be effectively modelled. The selection of a renewal traffic model i.e., GI arrival process, for this purpose, represents a reasonable balance between the accuracy (modelling power) and the efficiency required by a network planning tool, especially, when large-scale networks, as well as large aggregates of individual flows into service classes are considered. In this paper, we describe a detailed implementation of the model discussed in [4] and present an evaluation study of the model for a wide range of test cases.

Based on an analysis of the QoS mechanisms employed in multiservice IP networks and their implications for network planning, an approximate model of QoS-based queueing mechanisms used at the routers was discussed in [4]. This approximate model assumes fixed bandwidth partitioning to split the link capacity between different traffic classes. Such a model allows us to consider the different traffic classes separately and determine their bandwidth allocations on the links independently. Figure 1 shows the network dimensioning process that we consider for the solution of the CA problem. The information required for this process includes:

- Traffic descriptors for the demands of the delay-sensitive classes,
- Routing information for the delay-sensitive classes,
- Local (link) partitions of the global (end-to-end) QoS constraints for the delay-sensitive classes,
- Network topology.

Traffic demands are categorized into classes according to their delay requirements following the ITU-T G. 1010 [3] recommendation. For each class of traffic between each origin-destination (OD) pair, the end-to-end delay requirements are specified in terms of a mean delay and the variance of the delay (i.e., jitter). If there are C classes of traffic, then $C-1$ classes have delay requirements, and the remaining class is regarded as having no delay requirement (e.g., best-effort traffic). The algorithms for the solution of Problem OPQR-G (Optimal QoS Partition and Routing in a multicommodity flow network), discussed in [7], [8], when applied to each delay-sensitive class of traffic independently, will determine the required input for the dimensioning process; that is, the primary routes between the origin-destination (OD) pairs and the QoS partitions on the links for each class of traffic, respectively. We define a *class flow* as a single class of traffic between an OD pair. We model the class flow as a general (GI) arrival process characterised by a mean packet arrival rate and a squared coefficient of variation (SQV) of the packet inter-arrival time. For this renewal process, the coefficient of variation is used to characterise traffic burstiness, i.e., the variability of the arrival stream. IP traffic is bursty in the sense that its squared coefficient of variation is always greater than or equal to unity. In [9], we have presented models, which can be used to translate the OD pair traffic demands for each class of traffic as obtained from traffic measurement data into equivalent class flow traffic descriptors (i.e., GI arrival process parameters) for direct application into the dimensioning procedure.

I. Atov is with the Centre for Advanced Internet Architectures at Swinburne University of Technology, PO Box 218, Vic. 3122, Australia (phone: 613-9214-8089, fax: 613-9925-3748, e-mail: iatov@swin.edu.au).

R. J. Harris is with the Centre for Advanced Technology in Telecommunications at RMIT University, GPO Box 2476V, Vic. 3001, Australia (e-mail: richard@catt.rmit.edu.au).

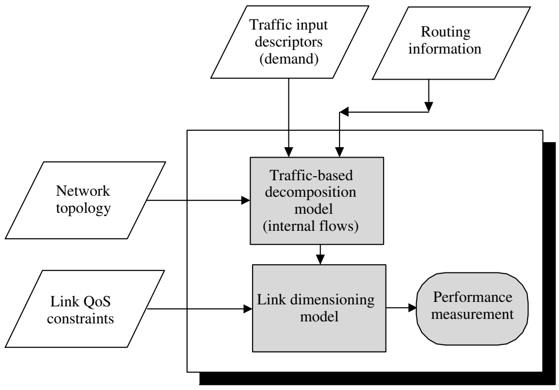

Fig. 1. Multiservice IP network dimensioning process

The dimensioning process can be summarised into three major stages (depicted in Fig. 1 as shaded boxes):

1) *Traffic-based decomposition model*. From the offered class flows to the network and the given routing information, obtain a characterisation of the internal (or internode) class flows.
2) *Link dimensioning model*. On a link by link basis, given the internal class flows and their link QoS constraints, determine the bandwidth allocations required for the delay-sensitive classes, as well as, the total link capacities.
3) *Dimensioning model validation*. Examine the end-to-end QoS performance of the class flows in the capacitated network by using a queueing network simulator.

It is important to emphasize here that, the solution of the CA problem, as obtained from the above dimensioning process, will determine the bandwidth allocations on the links for the delay-sensitive traffic classes. The so-called total link capacities obtained through this process, will be used as *setup* link capacities in a further optimisation process that the network planner has to consider as, in practice, link capacities are limited to a discrete set of values [10].

The rest of this paper is organized as follows. In Section II, renewal-based traffic decomposition models are reviewed. A detailed description is given in Section III of the procedures that combine these models in order to achieve characterisation of the internal class flows. The algorithm for determining the class-based bandwidth allocations on the links subject to satisfying the delay QoS constraints for the classes is described in Section IV. Finally, this paper concludes with a description of a simulation study that validates the dimensioning model used in solving the CA problem.

II. Renewal-based Traffic Decomposition Model

Traffic-based decomposition models encompass procedures required for modelling of the basic network operations of merging, departure and splitting, arising due to the common sharing of resources and routing decisions in the network (Fig. 2). In addition, they deliver approximations for performance measures on both a per-queue and a per-network basis.

For the traffic-based decomposition models that employ GI arrival processes as traffic descriptors, two different solution approaches can be identified: the successive/iterative approach and linearisation. For a comprehensive study on traffic-based decomposition models, see [11]. In the first case, starting from external inputs, the basic network operations are applied successively one at a time for each node of the network in isolation. For each node, the departure (or output) process is approximated. According to probabilistic routing after the node, the output traffic process is split and subsequently merged (or superposed) with other traffic processes in order to obtain the inputs for the nodes further downstream in the network. The internal flows and performance measures can be obtained in a single iteration step i.e., after single analysis of each node, if there are no feedback loops present in the network[1]. Otherwise, additional iterations are required until convergence at the nodes within the feedback loop is reached. In the linearisation solution approach, approximating linear equations are needed to describe the transformations for the parameters of the internal flows, arising from the basic network operations. The parameters of the internal flows are then obtained by solving a system of linear equations.

A. Kühn's and Whitt's decomposition models

In 1979, Kühn published a traffic-based decomposition model for open networks of GI/G/1 queues [12]. The arrival processes and the service-time distributions are partially characterised by their first two moments or, equivalently, by λ_A (the mean arrival rate), c_A^2 (the SQV of the interarrival time), τ_S (the mean service time), and c_S^2 (the SQV of the service time). The performance at each node is described by approximate formulae that depend only on these parameters. This model employs a successive/iterative approach to solve for the internal flows and performance measures at the nodes in the network. Later, in 1983, Whitt extended Kühn's model by adding several new features and he also implemented his decomposition model into a software tool widely known as QNA (Queueing Network Analyzer) [13], [14]. Specifically, he improved and modified the merging operation for GI traffic arrival processes and applied a linearisation approach for the network decomposition.

In view of the two-parameter descriptions of the arrival process (λ_A, c_A^2) and service time (τ_S, c_S^2) at a considered node, the elementary calculus that transforms the two parameters of the internal flows for each of the three basic network operations, as given in [13], is reproduced below:

a) Superposition of GI traffic flows: The superposed process of n individual GI flows, each characterised by λ_j and c_j^2 $(j = 1, \ldots, n)$, as it enters the considered node is approximated by a GI traffic flow with parameters λ_{A} and c_{A}^2, representing the mean arrival rate and SQV of the inter-arrival time of the superposed flow, respectively. By conservation of flow, the mean arrival rate of the superposed flow is given by:

$$\lambda_{\mathrm{A}} = \sum_{j=1}^{n} \lambda_j \ . \tag{1}$$

For the derivation of the SQV of the inter-arrival time of the superposed flow, Whitt combines the stationary-interval method implemented by Kühn with the asymptotic method [15] into a hybrid procedure:

$$c_{\mathrm{A}}^2 = w \sum_{j=1}^{n} \frac{\lambda_j}{\lambda_{\mathrm{A}}} c_j^2 + 1 - w \ , \tag{2}$$

with:

$$w = \frac{1}{1 + 4(1-\rho)^2(\nu - 1)} \ , \qquad \nu = \frac{1}{\sum_{j=1}^{n} \left(\frac{\lambda_j}{\lambda_{\mathrm{A}}} \right)^2}$$

and ρ is the *traffic intensity* or utilisation at the node, defined by $\rho = \lambda_{\mathrm{A}} \tau_S$.

[1]Note that feedback loops do not occur in practical communication networks.

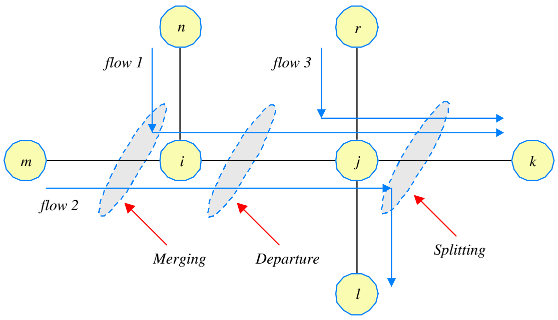

Fig. 2. Basic network operations

b) Departure flow from a queue: The departure flow from a queue is approximated as a GI traffic flow, characterised by $\lambda_{\rm D}$ and $c_{\rm D}^2$, representing the mean arrival rate and SQV of the inter-arrival time of the departure flow, respectively. Under equilibrium conditions, the mean flow entering a queue is always equal to the mean flow exiting the queue (by conservation of flow) and consequently $\lambda_{\rm D} = \lambda_{\rm A}$. The SQV of the inter-arrival time of the departure flow is given by:

$$c_{\rm D}^2 = \rho^2 c_s^2 + (1-\rho^2) c_{\rm A}^2 \ . \tag{3}$$

Whitt obtains the above result by applying a linear approximation to the stationary-interval method implemented by Kühn.

c) Splitting a GI flow Probabilistically: If a GI flow with parameters λ, and c^2 is split into n flows, each selected independently with probability q_i, the parameters for the i-th flow will be given by:

$$\lambda_i = q_i \lambda \ , \tag{4}$$
$$c_i^2 = q_i c^2 + (1-q_i) \ . \tag{5}$$

Note that, in reality, in this type of network, probabilistic splitting does not occur. Instead, the splitting is based on a predetermined static routing matrix (resulting from the fixed routing assumption) and therefore is deterministic. Whitt discusses this in the context of "deterministic flows" and suggests an equivalent probabilistic routing interpretation for deterministic (fixed) routing. We shall use this probabilistic routing approximation subsequently.

Despite the comprehensivness of QNA, the algorithmic procedure that we adopt for the solution of the internal flows in the network, which we discuss next, is based on the successive solution approach introduced by Kühn. Whitt's improved results for the basic network operations (as described above) are, however, considered in the procedure. In doing so, our intention is to develop a unique framework for implementation of traffic-based decomposition that would also be employed when more complex traffic descriptors are considered.

III. Modelling Internal Class Flows

After the preliminaries discussed in the previous section, we devote our attention to the actual algorithmic procedure used in the computation of the internal flows. From the input class flows and the given routing information, one can obtain a characterisation of the total traffic flows for each class $c\,(c = 1, \ldots, C)$ on every link in the network, which are also referred to as *internal* class flows. Each input class c flow between an OD pair $(u, v) \in \Pi$ is assigned to a fixed route r_c^{uv} ($r_c^{uv} \in \boldsymbol{r_c}$), where $\boldsymbol{r_c}$ is the set of routes for class c. We characterise an input class flow with the following set of parameters $\{\lambda_c,\ c_c^2,\ \overline{X_c},\ \overline{X_c^2}\}_{r_c^{uv} \in \boldsymbol{r_c}}$. The first two parameters denote the mean packet arrival rate and the SQV of the packet inter-arrival times of the class c flow ($c = 1, 2, \ldots C$), which has been assigned a route r_c^{uv}. The third and fourth parameters denote the first two moments of the packet size of the class c flow, respectively. We shall use the following notation:

- λ_{cl} - mean packet arrival rate of class c flow into the queue (reserved for class c) associated with the link l,
- c_{cl}^2 - SQV of the packet interarrival time of class c flow entering the queue (for class c) on the link l,
- $\rho_{cl} = \lambda_{cl} \tau_{cl}$ - traffic intensity of class c on link l.

According to the successive approach for network decomposition, the computation of the internal traffic descriptors (λ_{cl}, c_{cl}^2) starts with the links that are incident to the nodes where the external flows enter the network, as the mean rate and the SQV parameter for the external flows are given. Then, by performing the calculus for split, merge and departure operations for each flow, the algorithm proceeds until the destination for each of the flows is reached. In this case, the ordering in which the queues are selected for processing plays a crucial role. This may be difficult to implement, and therefore, we employ the following iterative procedure.

Figure 3 depicts the iterative procedure that we employ in determining the parameters of the internal flows for each service class. The algorithm executes only if the condition that all queues in the network are stable is satisfied (a queue is stable if its traffic intensity is less than unity). The algorithm starts with an initialisation step in which the values of the variability parameters of all flows before and after entering the queues associated with the links in the network, are set to zero, except for the links which represent first traversing links in the paths through the network of the external flows. For clarity, in the flow chart (Fig 3) the queues (associated with the links in the network) are indexed by numbers from 0 to $E-1$, with E being the total number of links in the network. The algorithm proceeds by performing the calculus for the three basic operations required for each queue independently. The order in which the algorithm selects the queues for processing, in general, may be arbitrary. After each queue in the network has been considered, the algorithm will execute another iteration if a suitable convergence criterion has not been met. Convergence is reached when the maximum relative error of the variability parameters of the flows on all links in the network between two consecutive iterations becomes less than or equal to a predefined sufficiently small value. That is, for a given class c the stopping criteria is defined by $\max\left\{\left|\frac{c_{cl}^2 - \widehat{c_{cl}^2}}{c_{cl}^2}\right|\right\}_{l \in E} \leq \epsilon$ with c_{cl}^2 and $\widehat{c_{cl}^2}$ denoting the variability parameters of the internal class c flow on link l, as computed in the last and second last iteration, respectively, and $\epsilon = 0.0001$ (for example).

It can be noted that the traffic descriptors of the internal flows depend on the service capacities that are allocated on the links for the classes, which are not known and need to be determined in the CA procedure. In order to deal with this interdependence, we calculate the internal flows for the classes, as well as, their bandwidth allocations on the links, iteratively as part of the dimensioning procedure discussed in the next section.

IV. Determining Class-based Bandwidth Allocations on Links

The bandwidth required for each delay-sensitive class c on a link l, which we denote by b_{cl}, can be determined from the traffic characteristics of the total amount of class c traffic on a link l

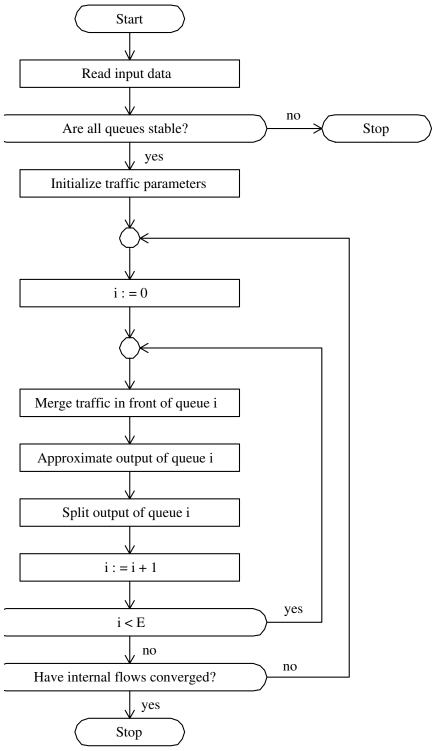

Fig. 3. Algorithm for determining internal flows

and its delay (QoS) constraint for that link q_{cl}, by inverting an approximate delay formula F; that is, find b_{cl} where:

$$q_{c\_l} = F(\lambda_{cl}, c_{cl}^2, b_{cl}) \qquad c = 1, 2, \ldots, \widehat{C} \ .$$

For this purpose, we analyse the GI/G/1 delay performance model. The delay (QoS) constraint for a traffic class on a link, q_{cl}, can be expressed in terms of the mean delay and the variance of the delay, which we denote as d_{cl}, and $\sigma_{d\_cl}^2$, respectively.

The mean delay of a packet from class c on a link l represents the sum of the waiting time of the packet in the queue until it is being serviced, W_{cl}, and the service time of the packet, τ_{cl}:

$$d_{cl} = W_{cl} + \tau_{cl} \ . \tag{6}$$

In a similar way, the variance of the delay for a packet of class c on a link l, is the sum of the variance of the waiting time of the packet in the queue, $\sigma_{W\_cl}^2$, and the variance of the service time of a packet, $\sigma_{\tau\_cl}^2$:

$$\sigma_{d\_cl}^2 = \sigma_{W\_cl}^2 + \sigma_{\tau\_cl}^2 \ . \tag{7}$$

The mean delay for a packet of class c in the queue associated with class c on a link l may be approximated using the QNA approximation (Kramer and Langenbach-Belz approximation) [13] as follows:

$$W_{cl} = \frac{\tau_{cl}\rho_{cl}(c_{cl}^2 + c_{s\_cl}^2)g}{2(1-\rho_{cl})} \tag{8}$$

where $g \equiv g(\rho_{cl}, c_{cl}^2, c_{s\_cl}^2)$ is defined as

$$g(\rho_{cl}, c_{cl}^2, c_{s\_cl}^2) = \begin{cases} \exp\left[-\frac{2(1-\rho_{cl})(1-c_{cl}^2)^2}{3\rho_{cl}(c_{cl}^2+c_{s\_cl}^2)}\right] & \text{for} \quad \mathrm{c}_{\mathrm{cl}}^2 < 1 \ , \\ 1 & \text{for} \quad \mathrm{c}_{\mathrm{cl}}^2 \geq 1 \ , \end{cases} \tag{9}$$

and:

- τ_{cl} is the mean service time of a packet of class c on link l i.e, the time it takes for the mean sized packet of class c to be transmitted on the link which has capacity b_{cl}:

$$\tau_{cl} = \frac{\overline{X_c} s}{b_{cl}} \ , \tag{10}$$

- $c_{s\_cl}^2$ is the SQV of the service time for a packet of class c, which is the variance of the service time divided by the square of its mean and after some algebraic manipulation it can be shown that it is equal to the SQV of the packet size, $\sigma_{X\_c}^2$:

$$c_{s\_cl}^2 = \frac{\overline{X_c^2} - \overline{X_c}^2}{\overline{X_c}^2} = \sigma_{X\_c}^2 \ , \tag{11}$$

- b_{cl} is the service rate or bandwidth allocated to class c (in bps) on link l, and s is a scaling factor ($s = 8$, as 1 byte = 8 bits).

The variance of the queueing delay for packets of class c in the queue may be approximated by using the QNA model result from [13] as follows:

$$\sigma_{W\_cl}^2 = W_{cl}^2 c_{W\_cl}^2 \tag{12}$$

where $c_{W\_cl}^2$ is the SQV of the queueing delay defined by

$$c_{W\_cl}^2 = \frac{c_D^2 + 1 - \chi}{\chi} \tag{13}$$

with χ denoting the probability of delay $\chi = P\{W_q > 0\}$, which is given by

$$\chi \quad = \quad \rho_{cl} + (c_{cl}^2 - 1)\rho_{cl}(1-\rho_{cl})h \ , \tag{14}$$

$$h \quad = \quad \begin{cases} \frac{1+c_{cl}^2+\rho_{cl}c_{s\_cl}^2}{1+\rho_{cl}(c_{s\_cl}^2-1)+\rho_{cl}^2(4c_{cl}^2+c_{s\_cl}^2)} & \text{for} \quad \mathrm{c}_{\mathrm{cl}}^2 < 1 \ , \\ \frac{4\rho_{cl}}{c_{cl}^2+\rho_{cl}^2(4c_{cl}^2+c_{s\_cl}^2)} & \text{for} \quad \mathrm{c}_{\mathrm{cl}}^2 \geq 1 \ , \end{cases} \tag{15}$$

and

$$C_D^2 \quad = \quad 2\rho_{cl} - 1 + \frac{4(1-\rho_{cl})\alpha}{3(c_{s\_cl}^2+1)^2} \ , \tag{16}$$

$$\alpha \quad = \quad \begin{cases} 3c_{s\_cl}^2(1+c_{s\_cl}^2) & \text{for} \quad \mathrm{c}_{\mathrm{s\_cl}}^2 \geq 1 \ , \\ (2c_{s\_cl}^2+1)(c_{s\_cl}^2+1) & \text{for} \quad \mathrm{c}_{\mathrm{s\_cl}}^2 < 1 \ . \end{cases} \tag{17}$$

The variance of the service time of a packet, $\sigma_{\tau\_cl}^2$, is given by: $\sigma_{\tau\_cl}^2 = \tau_{cl}^2 c_{s\_cl}^2$.

The algorithm for determining the bandwidth allocations for the delay-sensitive classes is described in Fig. 4. Based on the given topology, offered class flows, routing information and link delay QoS constraints for each class of traffic, the algorithm returns class-based bandwidth allocations for all links in the network. Initialization of the algorithm involves assigning starting values for the service times at the nodes for each traffic class, respectively. The service times for the classes are specified by the two parameters $(\tau_{cl}, c_{s\_cl}^2)$ and their initial values are computed from the first two moments of the packet size for the classes and their initial fixed capacity shares on the links, $\widehat{b_{cl}}$, by applying (10) and (11), respectively. The fixed capacity shares of each class on the links, $\widehat{b_{cl}}$, are initially set to $\widehat{b_{cl}} \geq \frac{\lambda_{cl}\overline{X_c}s}{\rho_{\mathrm{init}}}$, where ρ_{init} is the value for the initial traffic intensities of all link queues in the network i.e., $\rho_{cl} = \rho_{\mathrm{init}}$ ($l \in E$), which we set to an appropriately small value e.g., 10^{-3}.

Once the initialisation step is completed, the internal class flows are computed c_{cl}^2 ($l \in E, c \in \widehat{C}$), by applying the algorithm described in Fig. 3. After the internal flows have been determined, the algorithm cycles through the classes and, on a link by

Input: $G(V,E), \boldsymbol{r_c} = \{r_c^{uv}\}_{(u,v)\in\Pi}, \boldsymbol{q_c} = \{q_{cl}\}_{l\in E}$,

$\boldsymbol{\Lambda_c} = \{(\lambda_c, c_c^2, \overline{X_c}, \overline{X_c^2})\}_{r_c^{uv}\in \boldsymbol{r_c}}$ for $c \in \widehat{C}$.

Output: $\boldsymbol{b_{cl}} = \{b_{cl}\}_{l\in E}$ for all QoS sensitive classes.

```
1    for class c = 1 to C − 1
2        for all link (l ∈ E)
3            τ_cl ← (X̄_c s)/b̂_cl,  c²_s_cl ← c²_X_c
4    for class c = 1 to C − 1
5    switch (q_c)
6        case q_c = {d_cl}_{l∈E}
6a           for all link (l ∈ E)
6b               Calculate internal flows: {(λ_cl, c²_cl)}_{l∈E},
6c               Invert the delay formula (6) to derive b_cl,
6d               if max{ |(b_cl − b̂_cl)/b̂_cl| }_{l∈E} > ε
6e               b̂_cl ← b_cl for ∀l and repeat steps 3,6.
7        case q_c = {σ²_d_cl}_{l∈E}
7a           The same as step 6, except in 6c, invert the
             variance formula (7) to derive b_cl.
8        case q_c = {d_cl, σ²_d_cl}_{l∈E}
8a           do steps 6 and 7 to get (b_cl)_step6 and (b_cl)_step7 for ∀l,
8b           b_cl ← max{(b_cl)_step6, (b_cl)_step7} for ∀l.
9    return the set of class-based link capacities b_cl.
```

Fig. 4. Algorithm to determine class-based bandwidth allocations

link basis, determines their required bandwidth allocations b_{cl} by numerically inverting a delay formula, a variance of delay formula or both depending on the type of class considered. For a *delay-sensitive* type service class we need to find the value of its bandwidth allocation on a link for which the resulting link delay would be less than or equal to its link delay constraint. Similarly, for a *jitter-sensitive* type service class we need to find the value of its bandwidth allocation on a link such that its jitter link constraint is satisfied. In the case of a *delay- and jitter-sensitive* service class we are given two constraints for each link, one for the delay and one for the jitter, respectively. As before, we need to find the bandwidth allocations which are required to satisfy these constraints individually. The bandwidth allocation for this class would then be the maximum of these two values.

The algorithm calculates the internal flows and the class-based bandwidth allocations iteratively, until the class-based bandwidth allocations on the links converge. The iterative process stops when the maximum relative error of the class-based bandwidth allocations on all links between two consecutive iterations becomes less than or equal to a specified (sufficiently small) value e.g., $\max\left\{\left|\frac{b_{cl}-\widehat{b_{cl}}}{\widehat{b_{cl}}}\right|\right\}_{l\in E} \leq \epsilon$ where $\epsilon = 0.0001$. For inverting the delay and the variance formulae in steps 7c. and 8c., respectively, we have used two different recursive methods. One is based on Newton's method and the other one is based on an algebraic transformation of the given formula. In practice, we have found both methods converge very rapidly, so that either method can be used (for details see [19]).

Having obtained the bandwidth required for each delay-sensitive traffic class on a link, b_{cl} ($c = 1, 2, .., \widehat{C}$), the total capacity of the actual link (i.e., link setup capacity), can be determined as the minimum capacity of all available capacities on the link that is higher than the linear sum of the individual bandwidth allocations for the classes i.e., $\theta_l^{\text{setup}} = mod\left\lceil \sum_{c=1}^{\widehat{C}} b_{cl} \right\rceil$ where $mod\lceil\cdot\rceil$ is a ceiling function. The set of these link setup capacities θ_l^{setup} ($l \in E$) will determine the required input for the solution of the optimisation problem considered in [10].

V. Simulation Results

The most critical components of the dimensioning methodology presented in this paper with respect to accuracy are the approximate models comprising the renewal-based traffic decomposition. In this traffic decomposition model, individual queues are analysed separately after approximately characterising the external and internal flows as renewal processes. The renewal processes are characterised by two parameters representing the arrival rate and SQV of interarrival time, respectively. The arrival rates of the internal flows are computed exactly, whereas the variability parameters are obtained approximately and this represents the major approximation of this model. Computing performance measures for each queue, given its arrival and service parameters, also involves an approximation, but the quality of this approximation is relatively well understood, and is pretty good, as reported in [17]. The greatest difficulty with this decomposition model is determining appropriate variability parameters for the arrival processes to the queues.

In the following, we present a simulation study to assess the capability of the proposed dimensioning methodology in delivering the end-to-end delay QoS guarantees for the traffic classes. The test scenarios used in our simulation study are simple, but sufficiently illustrative, to indicate the quality that can be expected from the proposed dimensioning method across a wide range of traffic input parameters and traffic intensities at the nodes.

For this study, we have used the ns-2 simulation tool [18]. The external arrival processes in the proposed dimensioning model represent class-based aggregates offered to the network and along with their associated static routes they define the set of class flows in the network. In our simulation experiments each class flow is modelled as a hyperexponential process, which represents a suitable model for a bursty renewal traffic since its squared coefficient of variation of inter-arrival times is always greater than unity. Specifically, the interarrival times for the packets are generated according to hyperexponential distribution with two transient states H_2^b, i.e., the mixture of two exponential distributions with balanced means: one with mean m_1 realized with probability p and the other with mean m_2 realized with probability $1-p$, where $pm_1 = (1-p)m_2$.

The specific network model that we analyse, consists of 4 nodes and 3 links (see Fig 5). A single service class is considered and the traffic offered to the network is comprised of a number of traffic streams n, assigned between each OD pair i.e., OD pair $(1,4)$ and $(2,4)$, respectively. We consider twenty different cases for the traffic load in the network involving two values of n, $n = 10$ and 20 and five different types of traffic streams, each having different values for the mean packet arrival rate (λ) and SQV of interarrival times (c^2) as defined in Table I. The packet size for the traffic streams is set to a constant value of 1000 bytes, thus, our model results in a network of GI/D/1 queues. Note that we are testing our model with very bursty traffic streams whose arrival rates vary significantly, as for example, TS 5 has a packet arrival rate that is six times higher than the rate of TS 1. Whitt reported in [14], [16] that the reliability of the approximations used in the QNA decreases when the variability of the external arrival processes increases and when the arrival processes are not in the same time scale (i.e., do not have similar rates). Thus, our intention is to test the dimensioning model with

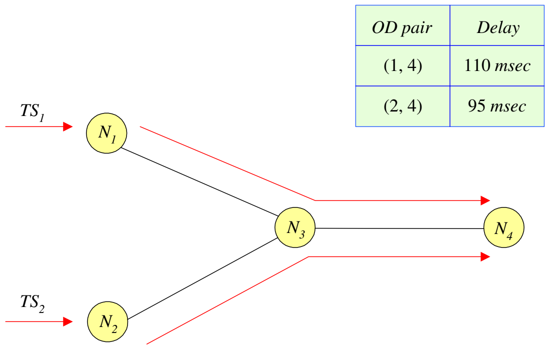

Fig. 5. Simple test network

TABLE I
INPUT TRAFFIC STREAMS

Number	*Mean* - λ	*SQV* - c^2
1	0.5	1.5
2	1.0	2.0
3	1.5	3.5
4	2.0	6.0
5	3.0	8.0

TABLE II
END-TO-END PACKET DELAY RE (%) FOR VARIOUS TRAFFIC INPUTS AND TRAFFIC INTENSITIES AT THE NODES FROM 0.30 TO 0.65

Simulation $d_{(1,4)}$ *[msec]*	*Simulation* $d_{(2,4)}$ *[msec]*	*CA model RE for* $d_{(1,4)}$	*CA model RE for* $d_{(2,4)}$
107.75 (± 3.2e-5)	92.51 (± 2.0e-5)	−2.04 −−	−2.62 −−
107.04 (± 2.9e-4)	89.88 (± 1.3e-5)	−2.69 −−	−5.38 −−
106.05 (± 2.3e-5)	87.31 (± 1.1e-5)	−3.59 −−	−8.08 −−
104.82 (± 4.6e-5)	83.12 (± 2.4e-5)	−4.70 −−	−12.50 −−
105.80 (± 7.5e-5)	88.82 (± 4.7e-5)	−3.81 −−	−6.50 −−
105.01 (± 3.3e-5)	86.89 (± 4.2e-5)	−4.53 −−	−8.53 −−
104.04 (± 6.3e-5)	82.38 (± 5.4e-5)	−5.41 −−	−13.28 −−
102.58 (± 2.8e-5)	85.91 (± 1.9e-4)	−6.74 −−	−9.56 −−
101.99 (± 1.3e-4)	82.27 (± 9.2e-5)	−7.28 −−	−13.40 −−
99.35 (± 4.5e-4)	81.73 (± 5.3e-4)	−9.68 −−	−13.96 −−
−−	−−	5.04	9.37

traffic loads of such characteristics that can potentially expose its worst performance. Note that, although renewal-based traffic inputs are considered in our test scenarios, the actual traffic loads on the links in the network are both bursty and correlated, as superposition of renewal processes and departure process from a queue represent a nonrenewal point process (i.e., the inter-departure times are correlated). It is important to test the model with bursty and correlated traffic loads as real IP traffic exhibit such characteristics.

The packet delay requirements for the traffic aggregates between the OD pairs is 110 msec and 95 msec, respectively (see Fig 5). The link delay requirements are as follows: $d_{1-3} = 85$ msec, $d_{2-3} = 70$ and $d_{3-4} = 25$ msec, where the subscript denotes the connecting nodes of the link[2]. The first test we performed considered $n = 10$ traffic streams of type i between the OD pair $(1, 4)$ and 10 traffic streams of type j between the OD pair $(2, 4)$, where $i, j = \{1, 2, 3, 4, 5\}$ and $i \neq j$. In this case, all combinations for (i, j) were considered, which results in 10 different traffic load scenarios. The link bandwidth allocations obtained from the CA algorithm for all traffic load scenarios considered in this test resulted in traffic intensities at the nodes (links) that fall in the range from 0.30 to 0.65.

For each test scenario (i.e., (i, j) pair) network simulation experiment was set up based on the link capacities obtained from the dimensioning model. For each experiment, we performed nine simulations with different seeds for the random number generator. The estimates of the mean packet delays were computed based on the replication method and 95-percent confidence intervals were obtained assuming a Student-t distribution. The results for all combinations of (i, j) traffic streams and $n = 10$ are summarised in Table II. Each row in the table indicates the traffic load scenario considered in the test. For example, the test scenario in the first row consists of 10 traffic streams of type 1 assigned between the OD pair $(1, 4)$ and 10 traffic streams of type 2 between the OD pair $(2, 4)$. The first and second columns show the results obtained from the simulations for the mean packet delay of the traffic aggregates between the node pairs $(1, 4)$ and $(2, 4)$, respectively, in the capacitated network. The confidence limits are given in parentheses below the simulation estimates. Columns three and four show the relative percentage errors for the mean packet delays between the two node pairs with respect to the target delays, respectively, which are defined as

$$\text{RE} = \frac{\text{Simulation delay} - \text{Target delay}}{\text{Target delay}} \cdot 100\% \; .$$

At the bottom of the table are the average absolute relative percent errors (ARE) for each OD pair, respectively, which are defined as $\text{ARE} = \sum_{\text{i=1}}^{10} |\text{RE}_\text{i}|/10$. The table shows the difference between the end-to-end packet delays in the capacitated network and the delay requirements given at input for various traffic loads for which the resulting traffic intensities at the nodes fall in the range $(0.30 - 0.65)$. It can be seen that, for this range of traffic intensities, the dimensioning model performs well with ARE for the two OD pairs of 7.20%.

The second test we performed considered $n = 20$ traffic streams of type i between the OD pair $(1, 4)$ and 20 traffic streams of type j between the OD pair $(2, 4)$, where $i, j = \{1, 2, 3, 4, 5\}$ and $i \neq j$. Similarly to the previous test, all combinations for (i, j) were considered, which results in 10 different traffic load scenarios. Also, the same packet delay requirements for the traffic aggregates between the two OD pairs, as well as, link delay requirements are considered. Based on the given traffic demands for this set of test scenarios and the delay requirements, the link bandwidths computed by the CA algorithm resulted in traffic intensities at the nodes, which fall in the range from $\rho_i = 0.50$ to $\rho_i = 0.90$ $(i = 1, 2, 3)$. The results for all combinations of (i, j) traffic streams and $n = 20$ are summarised in Table III.

The results in Table III show that for the case of various traffic loads, which result in traffic intensities at the nodes in the range $(0.50 - 0.90)$, the dimensioning model performs well with ARE for the node pairs delay requirements of 8.97%. These results together with the results from Table II lead us to conclude that the dimensioning model performs well with respect to guaranteeing the end-to-end delay requirements for the traffic demands in the

[2] Note that the optimisation problem of optimal partitioning of end-to-end delay constraints into link constraints is addressed in [8].

TABLE III

END-TO-END PACKET DELAY RE (%) FOR VARIOUS TRAFFIC INPUTS AND TRAFFIC INTENSITIES AT THE NODES FROM 0.50 TO 0.90

Simulation $d_{(1,4)}$ *[msec]*	*Simulation* $d_{(2,4)}$ *[msec]*	*CA model RE for* $d_{(1,4)}$	*CA model RE for* $d_{(2,4)}$
105.95 ($\pm$ 8.1e-5)	90.46 ($\pm$ 9.3e-5)	-3.68 $--$	-4.77 $--$
104.20 ($\pm$ 8.7e-5)	87.15 ($\pm$ 8.2e-5)	-5.27 $--$	-8.26 $--$
102.88 ($\pm$ 1.0e-4)	85.95 ($\pm$ 1.2e-4)	-6.47 $--$	-9.52 $--$
100.61 ($\pm$ 2.2e-4)	81.79 ($\pm$ 2.5e-4)	-8.53 $--$	-13.90 $--$
103.40 ($\pm$ 3.0e-5)	86.04 ($\pm$ 3.6e-4)	-6.00 $--$	-9.43 $--$
101.73 ($\pm$ 7.3e-4)	85.18 ($\pm$ 6.2e-4)	-7.51 $--$	-10.33 $--$
101.08 ($\pm$ 4.9e-5)	81.34 ($\pm$ 5.3e-4)	-8.10 $--$	-14.37 $--$
101.12 ($\pm$ 3.7e-4)	84.59 ($\pm$ 4.2e-4)	-8.07 $--$	-10.95 $--$
99.58 ($\pm$ 5.0e-4)	82.60 ($\pm$ 5.2e-4)	-9.47 $--$	-13.05 $--$
99.95 ($\pm$ 6.3e-4)	83.01 ($\pm$ 6.8e-4)	-9.13 $--$	-12.62 $--$
$--$	$--$	7.22	10.72

case of various traffic loads and traffic intensities at the nodes in the range of $(0.30-0.90)$. In our analysis we did not consider the case where traffic intensities at the nodes fall out of this range as such cases are not of practical interest when considering a well planned functional network.

In all cases, the dimensioning model slightly overprovisions the network, as the mean delays obtained in the capacitated network are always less than the specified end-to-end delay constraints. Thus, the performance of the network is guaranteed to be better than what is required by the delay-sensitive traffic classes. The results also indicate, as expected, that the reliability of the approximations used in QNA and, thus the accuracy of the dimensioning model, decreases when the variability of the external arrival processes increases (e.g., $c^2 > 6$) and when the arrival processes have rates that vary significantly.

VI. CONCLUSIONS

In this paper, we presented a dimensioning model for determining bandwidth allocations for the delay-sensitive traffic classes, as well as the total continuous bandwidth of the links by taking into account the varying delay requirements (QoS) of the traffic classes. This model allows the burstiness of IP traffic to some extent to be modelled, however, it cannot handle correlations often observed in real input traffic. Although it lacks modelling accuracy in describing the characteristics of the external and internal IP traffic flows, this model is very computationally efficient and, therefore, is appropriate to be used as a solution method for problem CA when large size networks are being considered. Not only should this model be used in this case, due to its efficiency, but also because it provides a reasonable solution method when large networks, as well as large aggregates of individual flows into service classes are considered. This is due to the multiplexing which occurs on a very large scale in this case and, as a result, the correlations significantly reduce due to the inter-mixing of packets from different traffic streams.

The proposed algorithm for the solution of the CA problem consists of two main algorithmic steps. In the first step, from the offered class flows to the network and the given routing information characterisation of the internal class flows is achieved by applying the methods for superposition, departure and splitting of GI traffic arrival processes as provided by the famous QNA approach. For this purpose, we have incorporated the QNA procedures in an efficient algorithm for computation of the variability parameters of the internal flows. Having determined the internal class flows, the second step employs an algorithm, which on a link by link basis determines the class-based bandwidth allocations by numerically inverting a delay formula, a variance of delay formula or both depending on the type of class considered.

Although, the required building blocks for consideration of a service class which is sensitive to the variations of the delay (i.e., jitter) are developed, they haven not been implemented into the dimensioning tool. Therefore, in this paper we have evaluated the capability of the dimensioning model in guaranteeing the end-to-end delays only for the service classes. The simulation study validated the modelling approach and confirmed that in the case of bursty and correlated traffic loads the dimensioning model performs well with respect to guaranteeing the end-to-end delay requirements for the traffic demands. The accuracy of the model has been demonstrated for various burstiness characteristics of the traffic load and for a wide range of the traffic intensities at the nodes.

REFERENCES

[1] S. Blake, D. Black, E. Carlson, et al., *An Architecture for Differentiated Services*, IETF RFC 2475, 1999.

[2] E. Rosen, A. Viswanathan, and R. Collon, *Multiprotocol Label Switching Architecture*, IETF RFC 3031, 2001.

[3] ITU-T Recommendation G. 1010, *End-user Multimedia QoS Categories*, November 2001.

[4] I. Atov and R. J. Harris, "Dimensioning Method for Multiservice IP Networks to Satisfy Delay QoS Constraints", *Proceedings of IFIP-TC6 Interworking 2002*, Perth, Australia, October 2002, pp. 13-25.

[5] R. Guérin and V. Peris, "Quality-of-Service in Packet Networks Basic Mechanisms and Directions", *Computer Networks*, Vol. 31, No. 3, February 1999, pp. 169-179.

[6] S. Floyd and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", *ACM Transactions on Networking*, Vol. 3, 1995.

[7] I. Atov, H.T. Tran and R.J. Harris, "OPQR-G: Algorithm for Efficient QoS Partition and Routing in Multiservice IP Networks", *Elsevier Journal of Computer Communications*. To appear.

[8] I. Atov, H.T. Tran and R.J. Harris, "Efficient QoS Partition and Routing in Multiservice IP Networks", *Proceedings of 22nd IEEE International Performance, Computing and Communications Conference (IPCCC)*, Phoenix, Arizona, April 2003, pp. 435-443.

[9] I. Atov and R.J. Harris, "Characterization of Class-Based Traffic Flows in Multiservice IP Networks", *Proceedings of 8th IEEE International Conference on Communication Systems 2002 (ICCS)*, Singapore, November 2002.

[10] I. Atov and R.J. Harris, "LP-based Algorithm for Optimization in Multiservice IP Networks", *WSEAS Transactions on Communications*, Vol. 2, No. 4, 2003, pp. 432-438.

[11] A. Heindl, *Traffic-Based Decomposition of General Queueing Networks with Correlated Input Processes*, PhD Thesis, Technische Universität Berlin, 2001.

[12] P. J. Kühn, "Approximate Analysis of General Queueing Networks by Decomposition", *IEEE Transactions on Communications*, Vol. 27, No. 1, pp. 113-126, 1979.

[13] W. Whitt, "The Queueing Network Analyzer", *Bell System Technical Journal*, Vol. 62, No. 9, pp. 2799-2815, 1983.

[14] W. Whitt, "Performance of the Queueing Network Analyzer", *Bell System Technical Journal*, Vol. 62, No. 9, pp. 2817-2843, 1983.

[15] W. Whitt, "Approximating a point process by a renewal process, I. Two basic methods", *Operations Research*, Vol. 30, No.1, pp.125-147, 1982.

[16] K. Shiram and W. Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data", *IEEE Journal on Selected Areas in Communications*, Vol. 4, No. 6, 1986, pp. 833-846.

[17] W. Whitt, "Approximations for the $GI/G/m$ Queue", *Productions and Operations Management*, Vol. 2, 1993, pp. 114-161.

[18] S. McCanne and S. FLoyd, UCB/LBNL/VINT Network Simulator - ns (version 2.26), March 2003. http://www.isi.edu/nsnam/ns/

[19] I. Atov, "QNA Inverse Model for Capacity Provisioning in Delay Constrained IP Networks", *Technical Report 040611A*, Swinburne University of Technology, June 2004.