In this paper, we propose a dimensioning model that takes account of the new technologies that can provide QoS (i.e., DiffServ/MPLS) and allows for multiple delay constraints so that guaranteed performance can be achieved for each of the traffic classes. The proposed model incorporates procedures that allow correlations and burstiness of IP traffic to be effectively modelled. The accuracy of the model has been investigated by means of a simulation study over a range of test cases. The results demonstrate the capability of the model in guaranteeing the end-to-end delay requirements for the traffic classes.

1 Introduction

The mechanisms for QoS provisioning, which are embodied in the DiffServ/MPLS approach, are necessary but insufficient to provide any service guarantees [1]. Resource provisioning is a fundamental requirement and important precursor to offering service quality. This requires the application of a capacity planning tool to determine the amount of resources to be allocated in the network so that the performance requirements for the various traffic classes can be satisfied. It is traditionally difficult to provide network capacity to meet varying QoS constraints. In addition, the traditional methods for capacity provisioning consider multislot circuit-switched networks (e.g., ATM networks) and use a loss-based approach [2]. That is, dimensioning of the network is achieved so that the blocking (loss) probabilities of various types of traffic (classes) remain below a specified threshold. However, these multirate loss models do not present suitable tools for capacity provisioning of IP-based broadband networks. In IP-based broadband networks (i.e., DiffServ/MPLS), due to the concept of class aggregates and static resource reservation, capacity provisioning is primarily concerned with developing performance guarantees in terms of packet delay or packet delay variation for the various service types (i.e., class aggregates).

Recently in [3] we studied the issues that surround IP network design with QoS and proposed a new methodology for capacity planning of multiservice IP networks. The proposed network design methodology takes account of the new technologies and mechanisms that enable QoS and allows for varying delay (QoS) constraints so that guaranteed performance can be achieved for each of the traffic classes. This methodology exploits a framework in which capacity provisioning is based on the partitioning of the end-to-end delay QoS requirements of each class-based traffic demand between each Origin-Destination (OD) pair into local QoS requirements at each link in the network. More precisely, the capacity planning problem is solved by dividing it into the following two main subproblems. First, for each delay-sensitive class determine a primary path between every OD pair together with a local partition of the global (end-to-end) QoS constraint (i.e., delay bound and/or jitter) by solving the combined routing and QoS partitioning optimization problem (OPQR-G problem). Subsequently, based on the routing information and local delay QoS constraints determine bandwidth allocations for the delay-sensitive traffic classes, as well as, the total bandwidth of the links by solving the capacity allocation problem (CA problem).

In this context, one needs to first consider the combined problem of optimal QoS partition and routing (OPQR-G problem) for a QoS framework in which a performance dependent cost function is associated with each network element and the QoS metric is additive. In this optimisation problem one must assign each link as large a delay as possible, while ensuring that the total delay on each route traversing the link is less than a specified delay constraint. This problem is NP-hard and we have developed an efficient heuristic algorithm that provides a pseudo-polynomial time solution for this problem in our previous work [4].

In this paper we address the second part of the problem. The capacity provisioning problem poses the challenge of performing appropriate link dimen-
sioning 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. Traditionally, for ease of tractability, simplistic traffic models were used (e.g., renewal), which completely neglect the correlation structure of the external and internal flows. However, it is very important that the chosen traffic descriptor for the dimensioning process be able to capture burstiness and correlations, as real IP traffic exhibits these characteristics. In this paper, we propose a model for capacity provisioning that incorporates procedures that allow correlations and burstiness of IP traffic to be effectively modelled. For this purpose Markovian arrival processes (MAPs) are employed to represent the external and internal traffic flows. The motivation for this choice is the fact that complex arrival patterns, such as those appearing in the Internet can best be described using MAPs [8].

The rest of the paper is organized as follows: In section 2 we outline our network modelling assumptions and discuss the required inputs and objectives of the considered capacity provisioning problem. In Section 3 and Section 4 we give an overview of the proposed dimensioning model and discuss each of its main steps and algorithms in turn. Evaluation of the proposed model is presented in Section 5 and, finally, Section 6 concludes the paper.

2 Inputs and Objectives of the Capacity Planning Tool

In order to model a multiservice IP core network based on DiffServ/MPLS QoS technologies we regard the network as a collection of links connecting all the nodes together, where each link consists of a separate queueing facility for each class of traffic and a scheduler. In the standard service offering, a class-based WFQ mechanism [12] has been used to enable guaranteed services. It enables the different delay criteria for each class to be met, by allocating a specified proportion of the service capacity to each class. For an analysis of this system, each queueing facility can be treated as an independent FIFO queue, with a fixed capacity equal to its allocation, which enables us to confine the capacity planning problem to a network of single-server queues for each class of traffic, respectively. In this case we choose to disregard the statistical gains in favour of incorporating more sophisticated dimensioning model which takes account of the real IP traffic characteristics.

The purpose of modelling the network in the above fashion is to enable traffic to be categorized into classes based on their sensitivity to the delay performance of the network. In our model, we consider the delay QoS constraints for a traffic class to be specified in terms of the mean delay and the variance of the delay (i.e., jitter). Depending on the type of delay QoS constraint, the following three sets of delay-sensitive classes of traffic are defined: (1) Delay sensitive class (DSC) - contains classes of traffic sensitive to delay i.e., their mean delays are required to be less than or equal to their specified end-to-end delay limits, (2) Jitter sensitive class (JSC) - contains classes of traffic sensitive to variations in the delay, and (3) Delay and jitter sensitive class (DJSC) - contains classes of traffic that are sensitive to both delay and its variation.

Generally defined, the CA problem is the problem of determining bandwidth allocations for the delay-sensitive traffic classes, as well as total bandwidth of the links in the network so that multiple QoS constraints for the traffic classes can be satisfied. The approximate model of QoS-based queueing mechanisms used at the routers 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 considered for the solution of the problem CA.

The algorithms for the solution of problem OPQR-G, discussed in [4], when applied for each delay-sensitive class of traffic independently, will determine the required input for the dimensioning process; that is, the set of primary routes \( r_c \) and the set of QoS partitions on the links \( q_c \) for each class of traffic, respectively (\( c \in C \)). The dimensioning process can be

![Figure 1: Multiservice IP network dimensioning process](image-url)
summarized into three major stages:

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 by using a link dimensioning model.

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. In the following we discuss each of these steps in turn.

## 3 Traffic Characterisation

In this paper, we define a class flow as a single class of traffic between an OD pair. Each class flow is assigned to a fixed route and a class flow is modelled as Markovian arrival process (MAP) [7]. MAP is a process which counts transitions of a finite continuous-time Markov chain (CTMC). Such a CTMC has an infinitesimal generator matrix $Q$ which is defined as $Q = D_0 + D_1$. $D_1$ is a non-negative matrix, with elements that give the transition rates of the observed (or marked) transitions: passing through an observed transition triggers an arrival event. $D_0$ has negative diagonal elements and non-negative off-diagonal elements representing the rates of the hidden transitions. For a MAP the steady state probability vector $\pi$ is defined by $\pi Q = 0$, $\pi e = 1$ (where $e$ is a column vector of ones of appropriate dimension). The mean arrival rate of a MAP is defined as $\lambda = \pi D_1 e$. The marginal distribution of the interarrival time of a MAP is of phase-type (PH) with distribution function $F(t) = 1 - e^\pi D_1 e^D_0 t e$.

The MAPs are able to match correlated and/or bursty arrival processes, which may also have self-similar properties and long-range dependence [8]. Moreover, the MAPs are analytically tractable models, which is manifested in various efficient computational procedures of the matrix-analytic approach for queueing systems. Despite these capabilities, MAPs have not been used previously in the planning process due to the associated state-space problem occurring when the basic network operations of merging and departure from a queue are being modelled. Recently, in [10] a traffic decomposition method for analysis of open queueing networks, where traffic arrivals occur according to a MAP process and service times have a phase-type (PH) distribution, has been developed. However, only finite MAP approximations for the output process of MAP/PH/1(/K) queue have been studied, while exact methodologies for superpositioning and splitting of MAPs have been applied. Using the method of exact superpositioning has limitations, as the computational complexity dramatically increases in practical cases. In order to keep the computational efforts required to a minimum, in analyzing queueing networks using the method of decomposition, one has to use a MAP of small order (e.g., MAP-2) to represent the intermediate node, as well as, the offered traffic inputs in the network. In [5] we developed a model that provides a good approximation of the exact superposed MAP process by a MAP of order two, which, combined with the models presented in [10] provide the necessary ingredients for an efficient traffic decomposition method based on MAPs. This enables us to overcome the state-space problem while capturing the correlation statistics that have impact on the queueing performance, and thus, incorporate these traffic processes into the planning procedure.

In the following, we assume that an external flow of class $c$ between an OD pair $(u, v) \in \Omega$ is assigned to a single fixed route, $r^u_{cv}$ ($r^u_{cv} \in r^u_c$), and is fully characterised by the two defining matrices of a MAP-2 process, as well as the first two moments of the packet size of class $c$ flow $(\{D_0^{(c)}, D_1^{(c)}\}, \{\lambda_x^{(c)}, \mu_x^{(c)}\})_{r^u_{cv} \in r^u_c}$. Service times at the nodes may be specified by their first two moments or alternatively as continuous PH-type distributions. Thus, the network is assumed to consist of MAP/PH/1 queues, which represent the transmission links in the network.

From the offered class flows to the network and the given routing information, characterization of the total traffic of each class on every link in the network can be obtained by applying the methods for superposition, departure and splitting of MAP traffic arrival processes, as described below.

**Superposition of independent MAPs**. This method will be used for superpositioning individual class flows that are being assigned different routes but are traversing the same link and thus entering the same queue. The superposition of independent MAP arrival processes is also a MAP. Consider $n$ MAP arrival processes, each characterised by $(D_0^{(i)}, D_1^{(i)})$, respectively. Their superposition process is a new MAP $(D_0^{(s)}, D_1^{(s)})$, for which an exact representation is given by [10]:

$$
D_0^{(s)} = D_0^{(1)} \oplus D_0^{(2)} \oplus \ldots \oplus D_0^{(n)},
$$

$$
D_1^{(s)} = D_1^{(1)} \oplus D_1^{(2)} \oplus \ldots \oplus D_1^{(n)}
$$

where $\oplus$ represents the Kronecker-sum. It follows that, the exact superposition of $n$ MAPs, each of order
results in a MAP of order \( k = \Pi_{i=1}^{m_i} m_i \). For practical applications in network design problems, one needs to reduce the complexity of solving queues with a large number of arrival processes. Therefore, the exact superposed process may be approximated by a simpler process that captures important characteristics of the original process as closely as possible e.g., MAP-2. In [5] we have investigated this issue and developed an approximate matching model for evaluation of the exact superposed process as a MAP-2 process, which provides good accuracy across wide range of burstiness parameters for the individual traffic processes.

**Departure flow from a queue** The characterisation of the departure process of MAP/PH/1 queues is required in order to use these as input processes for queues further downstream in the queuing network. In the infinite buffer case, the output process is an infinite MAP, which, due to its infinite size, becomes impractical for further use in network analysis. In order to solve this problem, in [9] truncation techniques for the infinite output MAP of a MAP/PH/1 queue have been studied and a family of MAP approximations to the departure process are proposed. Another approach for modelling the departure process, presented in [10], achieves a more compact MAP representation by choosing the parameters of the output MAP so that it reflects the busy-period behaviour of the queue. The latter approach, due to the more compact MAP representations, is preferred in the analysis of large communication networks.

**Splitting a MAP Probabilistically** The probabilistic splitting of a MAP \((A_0, A_1)\) with probability \(p\), results in two MAPs \((B_0, B_1)\) and \((C_0, C_1)\) defined as [10]:

\[
(B_0, B_1) = (A_0 + (1-p)A_1, pA_1), \quad (C_0, C_1) = (A_0 + pA_1, (1-p)A_1).
\]

This method will be used to characterise the flow being routed from some other node to the queue being examined. It is an approximate model in our case, as we make use of deterministic splitting (due to the fixed routing assumption).

The basic methods described above, provide the means for deriving the internal traffic flows. However, they cannot be derived in a single iteration step, as the internode traffic descriptors of a given service class depend on the service capacity allocated to the same class on the links in the network (as well as on the service time distribution), which on the other hand, needs to be determined in the capacity allocation procedure. In order to deal with this interdependence, we calculate the internal flows for each class of traffic iteratively as part of the capacity allocation procedure.

### 4 Capacity Allocation

The bandwidth required for each delay-sensitive class on a link, can be determined from the total amount of traffic of each class traversing the link and their respective link delay (QoS) constraints by inverting an appropriate delay formula. For this purpose, we analyze the MAP/PH/1 delay performance model. The underlying infinite Markov chain of MAP/PH/1 queue can be considered as a special case of a quasi-birth-and-death process (QBD) where each level of the QBD state space corresponds to a specific number of customers in the queuing system. For this system, the matrix \( R \) plays a central role, as the queue length distribution \( \Psi = [\pi_0, \pi_1, \pi_2, \ldots] \) can be computed from it. The matrix \( R \) can be derived by using matrix-geometric solution methods, like the LR-approach [11]. Having obtained \( \Psi \), many performance measures of the queue can be computed easily. In our model, we make use of the first two moments of the waiting time, which formulae have been derived in [10] and are given below:

\[
E[W] = \frac{1}{\lambda} \pi_1 (I - R)^{-1} A_0 (e \otimes (T)^{-1} e) + \frac{1}{\lambda} E[S] \pi_1 (I - R)^{-2} R A_0 e,
\]

\[
E[W^2] = \frac{2}{\lambda^2} \pi_1 (I - R)^{-1} A_0 (e \otimes (T)^{-2} e) + \frac{1}{\lambda} E[S] \pi_1 (I - R)^{-2} R A_0 (e \otimes (T)^{-1} e) + \frac{1}{2} E[S^2] \pi_1 (I - R)^{-2} R^2 A_0 e + E[S^2] \pi_1 (I - R)^{-3} R^3 A_0 e
\]

where the moments of the PH-type service, \( E[S] \) and \( E[S^2] \), are computed as follows [10]:

\[
E[S^2] = (-1)^i i! \alpha T^{-i}
\]

We adopt the following notation. The delay (QoS) constraint for a traffic class \( c \) on a link \( l \), \( d_{cl} \), can be expressed in terms of the mean delay and/or the variance of the delay, denoted as \( d_{cl} \) and \( \sigma_{d_{cl}}^2 \), respectively, depending on the type of class considered (e.g., DSC, JSC, DJSC). The mean delay of a packet from class \( c \) on a link \( l \), \( d_{cl} \), represents the total delay of the packet comprising of the waiting time in the queue until it is being serviced and the packet service time \( d_{cl} = W_{cl} + \tau_{cl} \) where \( W_{cl} = E[W] \) and \( \tau_{cl} = E[S] \) are computed from (3) and (5), respectively. Furthermore, the variance of the total delay of the packet of class \( c \) while traversing the link \( l \) is the sum of the variances of the waiting time and the service time for the packet \( \sigma_{d_{cl}}^2 = \sigma_{W_{cl}}^2 + \sigma_{\tau_{cl}}^2 \) where \( \sigma_{W_{cl}}^2 = E[W^2] - E[W]^2 \).
and \( \sigma_{2,cl}^2 = E[S^2] - E[S]^2 \) are computed from (3), (4) and (5), respectively. Note that, \( \tau_{cl} \) represents the time it takes for the mean sized packet of class \( c \) to be transmitted on the link which has capacity \( b_{cl} \):
\[
\tau_{cl} = \frac{X_c}{b_{cl}} \quad (6)
\]
where \( b_{cl} \) 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 squared coefficient of variation (SQV) of the service time for a packet of class \( c \), \( \sigma_{c}^2 \), is defined as 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, \( \sigma_{X_c}^2 \):
\[
\sigma_{c}^2 = \frac{X_c^2 - X_c^2}{X_c^2} = \sigma_{X_c}^2. \quad (7)
\]

The capacity allocation algorithm based on the given topology \( G(V,E) \) (where \( V \) and \( E \) denote the number of nodes and links in the network, respectively), offered class flows, routing information and link delay QoS constraints for each class of traffic, 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 \( (\tau_{cl}, \sigma_{c}^2) \) 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, \( \hat{b}_{cl} \), by applying (6) and (7), respectively. The fixed capacity shares of each class on the links, \( b_{cl} \), are initially set to sufficiently large values in the following way: \( \hat{b}_{cl} \geq \frac{\lambda_c}{\rho_{\text{min}}} \), where \( \rho_{\text{min}} \) is the value for the initial traffic intensities of all link queues in the network i.e., \( \rho_{cl} = \rho_{\text{init}} \ (l \in E) \), which we set to an appropriately small value e.g., \( 10^{-3} \). The mean arrival rates of the internal class flows \( \lambda_{cl} \ (l \in E, c \in \hat{C}) \) are computed as discussed in Section 3. Subsequently, from the specified moments, a PH-type fitting for the service distribution at the nodes is performed by applying the fitting models presented in [10].

Once, the service times at the nodes are initialized \( \{\alpha^{(c)}, \mathbf{T}^{(c)}\} \), the algorithm successively applies the models for the three basic operations (i.e., merging, splitting and departure) for each node in the network in order to derive the two defining matrices of the internal MAP flows for each service class \( \{\mathbf{D}_{0}^{(c)}, \mathbf{D}_{1}^{(c)}\} \). Having computed the internal MAP flows, the algorithm cycles through the classes and, on a link by link basis, determines their required bandwidth allocations by numerically inverting a delay formula, a variance of delay formula or both depending on the type of class considered. The algorithm calculates the internal flows and the class-based bandwidths iteratively, until the class-based bandwidth allocations on the links converge. The stopping criterion for the algorithm is satisfied 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., \( \max \left\{ \frac{|b_{cl} - \hat{b}_{cl}|}{b_{cl}} \right\} \leq \epsilon \), where \( \epsilon = 0.0001 \).

Having obtained the bandwidth required for each delay-sensitive traffic class on a link, the total capacity of the actual link, can be easily determined by summing up the individual bandwidth allocations for the classes and by allowing a certain link utilization

---

**Figure 2: Algorithm for determining class-based bandwidth allocations**

<table>
<thead>
<tr>
<th>Input:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G(V,E), \mathbf{r}<em>c = {r^{(u)} }</em>{(u,v) \in \mathbb{E}}, \mathbf{q}<em>c = {q</em>{cl} }_{l \in E}, )</td>
</tr>
<tr>
<td>( \mathbf{A}<em>c = {{\mathbf{D}</em>{0}^{(c)}, \mathbf{D}<em>{1}^{(c)}}, X</em>{c}, X_{c}^2}_c ) for ( c \in \hat{C}. )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b_{cl} = {b_{cl} }_{l \in E} ) for all QoS sensitive classes.</td>
</tr>
</tbody>
</table>

1. for class \( c = 1 \) to \( C - 1 \)
2. for all link \( (l \in E) \)
3. calculate mean arrival rate \( \{\lambda_{cl} \}_{l \in E} \)
4. for all link \( (l \in E) \)
5. calculate internal flows: \( \{\{\mathbf{D}_{0}^{(c)}, \mathbf{D}_{1}^{(c)}\}\} \)
6. run algorithm for inverting link delay formula to derive \( b_{cl} \)
7. do steps 5, 6 to get \( (\hat{b}_{cl}) \) for all \( l \) in \( E \)
8. switch \( (\mathbf{q}_c) \)
9. return the set of class-based link capacities \( b_{cl}. \)
constraint to be satisfied. The bandwidth allocation type queueing mechanisms (e.g., WFQ) are normally configured in such a way, that the sum of all bandwidth allocation to the delay-sensitive classes cannot exceed a specified utilization value of the link bandwidth. The spare bandwidth is used for the best-effort traffic.

5 Numerical Results

In order to simulate a network with correlated traffic inputs we implemented an MMPP (a special case of MAP) traffic generator in ns-2. The test scenario (see Fig 3) used in this case study is simple, but sufficiently illustrative, to indicate the quality that can be expected from the proposed dimensioning model across a wide range of traffic input parameters and traffic intensities at the nodes. A single service class in the network is considered and the traffic load is comprised of two traffic streams, each assigned to the OD pair (1,4) and (2,4), respectively. We consider 10 different cases for the traffic load in the network involving five traffic streams. The cases we use have been set up to test the proposed method over a fairly extreme set of circumstances i.e., over extremes of combinations of arrival rate and variability. The combinations of arrival rate which have been considered for the two component processes start from equal rates for the two processes to combinations where one process has two to twenty times the mean arrival rate of the other. The combinations of SQV of the interarrival times that we consider are all combinations of different pairs from \{1.5, 2.5, 5, 8.5, 10\}. The packets were generated according to the MMPP traffic model and the size for the packets was generated according to the hypoexponential Erlang-2 distribution with a mean of 1000 bytes and SQV of 0.5. Each combination of the traffic streams was tested for various end-to-end delay constraints between the two node pairs (in seconds); specifically, \(d(TS_1, TS_2) = \{(0.18, 0.19), (0.24, 0.25), (0.30, 0.35), (0.60, 0.75), (1.20, 1.40), (2.40, 2.60)\}\), which represent the sum of the link delay constraints \(d_{link,1} = \{0.06, 0.08, 0.10, 0.20, 0.50, 1.20\}\), \(d_{link,2} = \{0.07, 0.09, 0.115, 0.35, 0.70, 1.40\}\) and \(d_{link,3} = \{0.12, 0.16, 0.20, 0.40, 0.70, 1.20\}\), respectively. By varying the target delay requirements and keeping the traffic load fixed we achieve an increase in the traffic intensities at the nodes. The target delays are chosen such that the resulting traffic intensities at the nodes in the capacitated network fall across the range of interest e.g., \(0.3 < \rho < 0.8\).

For each test scenario 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 this case study are summarized in Table 1. The results are the average over each set of traffic inputs and delay requirements of the absolute relative percentage error for the mean end-to-end delay for the two node pairs. Each row in the table represent a set of tests performed for all combinations of traffic streams between the two node pairs and for a given pair of the target delays for OD pairs (1,4) and (2,4), respectively. The relative percentage error for the mean packet delays between the two node pairs in the capacitated network with respect to the target delays is defined as \(RE = \frac{100 \cdot \text{Simulation delay} - \text{Target delay}}{\text{Target delay}}\).

It can be seen that the results show a very good performance of the dimensioning model in providing the target end-to-end delay guarantees. In addition, we have compared the model with a renewal-based dimensioning model (i.e., it completely neglects the correlation structure of the traffic), discussed in [6],

![Figure 3: Simple test network](image)

Table 1: Averaged absolute RE (%) for the mean end-to-end delay for various traffic inputs and target delays.

<table>
<thead>
<tr>
<th></th>
<th>MAP based CA model</th>
<th>MAP based CA model</th>
<th>QNA based CA model</th>
<th>QNA based CA model</th>
</tr>
</thead>
<tbody>
<tr>
<td>RE for (d_{TS1})</td>
<td>3.202</td>
<td>5.423</td>
<td>37.9</td>
<td>40.24</td>
</tr>
<tr>
<td>RE for (d_{TS2})</td>
<td>4.051</td>
<td>7.192</td>
<td>45.14</td>
<td>53.12</td>
</tr>
<tr>
<td>RE for (d_{TS1})</td>
<td>8.370</td>
<td>11.44</td>
<td>50.62</td>
<td>72.60</td>
</tr>
<tr>
<td>RE for (d_{TS2})</td>
<td>6.810</td>
<td>9.890</td>
<td>43.11</td>
<td>42.10</td>
</tr>
<tr>
<td>RE for (d_{TS1})</td>
<td>10.98</td>
<td>14.72</td>
<td>48.23</td>
<td>43.78</td>
</tr>
<tr>
<td>RE for (d_{TS2})</td>
<td>13.72</td>
<td>17.53</td>
<td>50.60</td>
<td>57.69</td>
</tr>
<tr>
<td>RE for (d_{TS1})</td>
<td>7.954</td>
<td>10.79</td>
<td>45.91</td>
<td>51.13</td>
</tr>
</tbody>
</table>
which is based on the inversion of the famous QNA performance model for analysis of networks of GI/G/1 queues [13]. It is worth noting that in all cases the mean delays obtained in the capacitated network were always less than the specified target delay values. This suggests that both models will overprovision the network. However, the renewal based model will grossly overprovision the network (up to 60 %) and this example demonstrates significant advantages for incorporating traffic processes, which can convey the correlation structure of the arrival processes, such as MAPs, into network planning procedures.

6 Conclusion

In this paper, we have provided an overview of a novel framework for capacity provisioning in multi-service IP networks. With the increased demand for delay-critical IP traffic, the capacity planning tool has to allow for multiple delay constraints so that guaranteed performance can be achieved for the traffic classes. The proposed framework in this paper takes account of the multiple delay QoS constraints for the classes, which can be specified in terms of delay bound or variance of the delay (jitter) or a combination of both. In addition, it incorporates procedures for effective modelling of correlations and burstiness associated with the IP traffic.

We have proposed an algorithm for determining bandwidth allocations on the links for the delay-sensitive traffic classes, as well as total capacities on the links, so that the varying delay constraints for the classes can be satisfied. The proposed algorithm efficiently incorporates iterative procedures for inversion of the MAP/PH/1 delay performance model and for the computation of the internal flows into a unified framework for network dimensioning. The presented simulation study demonstrates the capability of the model in guaranteeing the end-to-end delay requirements for the traffic classes.

Acknowledgments

The authors wish to acknowledge Prof. Peter Taylor of University of Melbourne for his valuable discussions and comments in relation to the modelling of MAPs.

References


