J Netw Syst Manage (2013) 21:474–509
DOI 10.1007/s10922-012-9243-3

QoSPlan: A Measurement Based Quality of Service aware Network Planning Framework

Alan Davy · Brendan Jennings · Dmitri Botvich

Received: 5 April 2011 / Revised: 28 May 2012 / Accepted: 30 May 2012 /
Published online: 10 June 2012

Abstract In this article we present QoSPlan—a measurement based framework for preparing information relevant to Quality of Service (QoS)-aware IP network planning, which aims at reducing a core operational expenditure for the network operator. QoSPlan is designed to reduce the cost of deployment and maintenance of network monitoring systems. The process involves analysis of pre-existing accounting data to estimate a network-wide traffic matrix. Part of this estimation process relates to the generalization of QoS-related effective bandwidth coefficients taken from traffic analyzed on the network. We offer recommendations on how to appropriately realize QoSPlan to maximize its accuracy and effectiveness when applied to different network traffic scenarios. This is achieved through a thorough sensitivity analysis of the methods proposed using real traffic scenarios and indicative network topologies. We also provide an economic analysis of the deployment and maintenance costs associated with QoSPlan in comparison to a direct measurement approach, demonstrating cost savings of up to 60 % given different topology sizes.

Keywords Provisioning · Effective bandwidth · Traffic matrix

Neither the entire paper nor any part of its content has been published or has been accepted for publication elsewhere. It has not been submitted to any other journal.

A. Davy (✉) · B. Jennings · D. Botvich
Telecommunications Software & Systems Group,
Waterford Institute of Technology, Cork Road, Waterford, Ireland
e-mail: adavy@tssg.org

B. Jennings
e-mail: bjennings@tssg.org

D. Botvich
e-mail: dbotvich@tssg.org

1 Introduction

Currently, establishing input for the network planning process relies on the use of dedicated hardware devices collecting large volumes of network traffic data that is then analyzed to identify a network configuration design reflecting estimated demand and specified Quality-of-Service (QoS) requirements. The use of dedicated measurement hardware means the approach is expensive, incurring costs in hardware procurement and maintenance, in addition to significant training and operational costs. We argue that the network planning process can be made more cost-effective, whilst maintaining a sufficiently high degree of accuracy, by reusing alternative sources of information residing within the network accounting system.

In this article we present a QoS-aware network planning framework based on network accounting data as an alternative source of estimating network traffic demands, initially presented in [1, 2]. We also propose a method of capturing a relationship between estimated network demands and required effective bandwidth [3] levels specific to outlined QoS targets through packet trace analysis. We term this relationship as the “effective bandwidth coefficient”. This approach has been proven robust and is agnostic of traffic types, thus requiring no a priori knowledge of traffic model characteristics. Based on an economic analysis comparing deployment and operational costs of our proposed system to a traditional network planning system based on dedicated monitoring equipment, we have shown that relative costs savings can be as high as 60 %. We also present a generalized process network operators can follow to prepare input for QoS-aware IP network planning, as outlined previously in [4].

This paper is organized as follows: Sect. 2 discusses related work in the areas of measurement based QoS-aware network planning and effective bandwidth estimation techniques. Section 3 outlines our measurement based Quality of Service aware network planning framework—QoSPlan and provides an in-depth performance analysis of the QoSPlan framework. Section 5 provides a use case deployment scenario of QoSPlan to evaluate its use on the GÉANT network. Section 6 provides an economic analysis of QoSPlan in comparison to a direct measurement based approach. Section 7 summarizes the paper and outlines areas for future work.

2 Related Work

The network planning process generally requires three sources of input [5]: (1) attributes associated with the current traffic demands on the network which collectively specify their behavioral characteristics; (2) attributes associated with resource constraints on the network topology; and (3) a constraint based routing definition framework which plans routing of traffic subject to (1) and (2). Here we focus on how to estimate, within an acceptable degree of accuracy, attributes relating to the the current traffic demands on the network (which relates to input (1)). We also need to take into consideration the related QoS targets imposed on various services operating over the network, which has a direct impact on the resources required. This section provides an analysis of literature regarding those aspects of QoS aware network planning within a communications network deemed relevant to QoSPlan. Section 2.1 offers a review of approaches proposed to estimate the effective bandwidth of traffic flows. Section 2.2 discusses a number of approaches for estimating the traffic matrix of a network.

2.1 Effective Bandwidth Estimation

The term *effective bandwidth* refers to the minimum amount of bandwidth required by a traffic flow to maintain a specified QoS target. Of particular interest to QoSPlan is the relationship between the effective bandwidth of a traffic flow and the mean throughput of that traffic flow. We term this relationship as the effective bandwidth coefficient. The measurement of effective bandwidth has been a focus of research for some time [3]. A particular issue within a communications network is the effect statistical multiplexing of traffic at the point of aggregation has on the effective bandwidth of aggregated traffic flows. As traffic flows are aggregated, the effective bandwidth requirements of each traffic flow reduces as the level of aggregation increases. This has been observed by Botvich and Duffield [6] and Simonian and Guibert [7]. Effective bandwidth estimation algorithms must be able to capture this effect if traffic performance optimization strategies employing these algorithms are to effectively control QoS of traffic whilst minimizing bandwidth utilization.

In [3] Kelly proposed a theoretical framework for the prediction of effective bandwidth of a defined traffic source. Kelly notes that the theoretical approach he proposes cannot be deployed on an operational network without a complete description of all traffic sources. As this has been shown to be quite a challenging task [8–10], alternative approaches have been developed to use static traffic model assumptions estimated from various traffic metrics as input to the effective bandwidth estimation algorithm. These algorithms include the direct estimator [11] and the block estimator [12]. These approaches are only appropriate for short range dependent traffic and as it has been shown that Internet traffic tends to demonstrate long-range dependence [10], these approaches will not be appropriate in an operational context.

An approach proposed by Guérin [13] recognizes the issue of effective bandwidth for aggregated traffic. To address this issue, Guérin proposes two approaches of estimating effective bandwidth; one is specifically designed to measure effective bandwidth requirements of a single traffic flow, while the other addresses estimation of effective bandwidth for aggregated traffic. The latter approach is based on the premise that as traffic is aggregated at a point, the distribution of the traffic arrival bit rate can be accurately modeled using a Gaussian distribution. Based on this assumption the author proposes to use standard approximations to estimate the tail of the bit rate distribution. It has been shown by Guérin [13] that as traffic flows are aggregated, the arrival bit rate does approach a Gaussian distribution, however this approach does not account for variation in traffic aggregation, impacting on the effective bandwidth estimation at various aggregation levels.

Empirical estimation of effective bandwidth attempts to overcome the limitations of traffic model based effective bandwidth estimation algorithms. The approach is based on the analysis of traffic being replayed through a simulated queue. The approach observes the behavior of the modeled queue buffer as the traffic is processed to measure an effective bandwidth value.

A method proposed by Liu and Baras [14] and Davy et al.[15] involves collecting a packet trace from the network at a point where the effective bandwidth estimation is required. The approach simulates a FIFO queue with an adjustable queue service rate. The packet trace is processed through the FIFO queue at specified service rates, to measure the associated proportion of QoS target violations. The algorithm is based on the observation that as the service rate of the queue increases, the proportion of violating traffic decreases. The FIFO queue service rate for successive iterations is controlled by a search algorithm that decide when an appropriate queue service rate is found; this rate produces an appropriate level of QoS violations from the queue for the processed packet trace. This service rate is then taken as the effective bandwidth.

2.2 Estimating the Traffic Matrix

The traffic matrix is a pair wise, edge-to-edge, matrix of traffic volumes that have traversed the network over a period of time [16]. The network traffic matrix is considered a core element of the network planning process. Traditionally, the traffic matrix is established by analyzing traffic within the network directly, through the use of dedicated network monitoring devices. Such approaches can establish a highly accurate traffic demand estimation across the network. However, as additional hardware for these devices require installation, operation and maintenance, this approach tends to incur high costs to the network operator, increasing both operational and capital expenditure. Core to the ethos of network planning is ensuring cost efficient and timely planning decisions are made in line with network operator objectives.

In [17], Feldmann proposes an approach to estimating a network wide traffic matrix from IP Flow records collected at ingress points in the network. The general approach is as follows. For each flow collected at an ingress point, its destination address is mapped to an egress node at the edge of the network. The approach assumes that no traffic is consumed within the core of the network. Therefore all traffic entering an ingress point has a corresponding egress exit point. The approach also makes the assumption that the volume of traffic within the flow is uniformly distributed from start to finish. Based on this assumption the volume of the flow is divided into equal bins of set durations. The volume within each bin is added into the traffic matrix specifying the volume of traffic between the ingress and egress node for that bin period. Once all flows are processed, the traffic matrix will contain total traffic demand between ingress, egress pairs over each bin period. The authors also state that additional information such as protocol and type of service information held within the flow record can be used to enhance the traffic matrix information.

In [18], Papagiannaki proposes an enhancement to this approach. The work presents a method of distributing the operation of traffic matrix estimation among the ingress router nodes. The approach focuses on distributing two essential functions core to the estimation of the traffic matrix from flow records and makes the following recommendations: (1) Implement a function to map destination network prefixes to egress links or routers within the domain; (2) Modify the definition of the flow record in order to include the result of this mapping. Based on the analysis performed within the paper, the authors found that up to 99 % communications overhead can be reduced if the proposed approach was deployed over a traditional direct measurement approach.

The authors of [19] discuss a stream database for network applications including traffic analysis. The objective of the tool was to develop a network data analysis tool which has the speed and flexibility that network applications require, but which provides a structured querying environment to make complex analysis tractable. The tool may envisage usage for the compilation of a traffic matrix on the fly as the network stream is processed. In [20], the authors focus on the issue of incomplete data in the generation of the network traffic matrix. They introduce a spatio-temporal compressive sensing technique to improve the accuracy in estimating the traffic matrix in the presence of missing values.

3 QoSPlan: Measurement Based Provisioning of QoS

QoSPlan delivers an network traffic matrix of a given network which takes into consideration the QoS requirements of the various classes of traffic being transported. This is essentially achieved by two fundamental processes. Firstly a per traffic class traffic matrix is estimated for traffic traversing the network. This process utilizes pre-existing accounting data within the network management system in for form of IP Flow records. The second element is the estimation of effective bandwidth coefficients of traffic carried over the network, which, when used in conjunction with per traffic class traffic matrices facilitate QoS-aware network planning. These steps will be further discussed in the following sections. The QoSPlan process is outlined in Fig. 1.

3.1 The QoSPlan Process

QoSPlan is broken into three phases: (1) acquisition, (2) analysis and mediation and (3) proposition. We now discuss configuration options at each phase and discuss how important these configurations are in relation to supplying accurate input for network planning.

3.1.1 Phase 1: Acquisition

QoSPlan depends on the acquisition of two forms of data, namely accounting data in the form of flow records, and short packet traces. In the collection of flow records, packet sampling plays a major role in the accuracy of demand estimation form accounting data. As discussed in the IETF IPFIX architecture [21], PSAMP is employed for packet sampling by IPFIX in the creation of flow records. We analyze

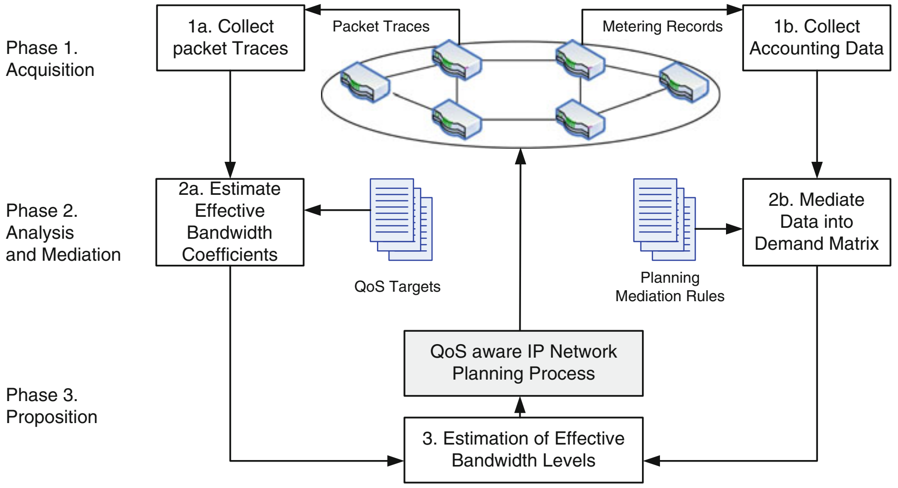

Fig. 1 QoSPlan process

the effect different sample settings have on the estimation of network demand from accounting records in Sect. 4.3.1.

Packet traces are also acquired for input to the QoSPlan process. They are analyzed to establish a relationship between network traffic behavior and specified QoS targets through the calculation of effective bandwidth coefficients as discussed in Sect. 3.3.3. A relatively large number of packet traces must be collected per traffic class from various edge node locations around the network. It is vital that collection points be distributed evenly around the ingress points of the network if accurate effective bandwidth analysis is it to be carried out. If the distribution of collection points is uneven, effective bandwidth coefficient estimations per traffic class may be biased to particular ingress points. Packet trace collection points should be positioned at the ingress edge of the network, as we require this traffic to be unshaped by the network itself. Packet traces also need to be collected over an appropriate duration. If packet trace durations are too small or large, analysis of the traffic may result in misleading effective bandwidth estimations as demonstrated in Sect. 4.3.1.

3.1.2 Phase 2: Mediation and Analysis

The analysis and mediation phase manages collation of relevant metering data into usable information for QoSPlan. There are two internal steps within this phase, namely the estimation of network demand from mediated accounting data, and the calculation of effective bandwidth coefficients from collected packet traces. Both are considered independent processes, but are required to deliver a QoS enhanced traffic matrix to a network planning process.

The mediation of accounting data into the traffic matrix depends on planning mediation rules, much like accounting record mediation depends on accounting business logic. The planning mediation rules outline how to map accounting record flows to ingress and egress edge nodes on the network. It is important to recognize factors that can affect this mapping such as moving or mobile nodes, or multiple entry points for a particular node. If mappings are not updated appropriately, the traffic matrix may contain incorrect measurements. For preparing input to the network planning process, we assume that the network will remain static for the long term. An additional consideration here is the measurement interval over which network demand is being estimated from accounting data.

In the analysis of collected packet traces, the effective bandwidth algorithm processes each collected packet trace. Effective bandwidth of a packet trace is controlled by a number of factors such as the traffic itself, degree of aggregation of traffic and most importantly the QoS target. The QoS targets are targets set out in Service Level Agreements between the network operator and customers of the network.

3.1.3 Phase 3: Proposition

The final phase prepares a matrix of estimated effective bandwidths per traffic class for input to the network planning process. This is achieved by multiplying the appropriate effective bandwidth coefficient by the estimated network demand between edge node pairs for the particular traffic class. A critical decision here is the choice of an appropriate representative effective bandwidth coefficient from the set of collected coefficients per traffic class. As network planning is predominantly based on provisioning for near peak traffic, we recommend choosing the 95th percentile value of this range to ensure a conservative estimate.

3.2 QoSPlan Framework Algorithms

The QoSPlan framework relies on a number of core algorithms to carry out the above states phases. These algorithms are the estimation of effective bandwidth of an aggregated traffic flow and the follow on calculation of the “effective bandwidth coefficient”, and the estimation of a network wide traffic matrix from accounting flow records. We will now discuss the details of each of these algorithms in the following sections.

3.3 Empirical Estimation of Effective Bandwidth

This section specifies and evaluates a purely empirical approach for estimating the effective bandwidth of aggregated traffic flows. We believe such an approach is suitable for use within a communications network as it can operate independently of traffic model assumptions.

A typical example of a QoS delay target is (0.04 s, 0.001), which means that only 0.1 % of traffic is allowed to be delayed more than 40 ms. As the effective bandwidth depends on the QoS target, for different QoS targets, effective bandwidth estimations could be different. Suppose the QoS delay target is fixed and includes $delay_{max}$ the maximum delay and p_{delay} the proportion of traffic which can exhibit delay more than $delay_{max}$. We define effective bandwidth R_{eff} of a traffic flow for the QoS delay target ($delay_{max}$, p_{delay}) as a minimal link rate such that if we simulate a FIFO queue (with an unlimited buffer and assumed initially empty) the proportion of traffic which will exhibit delay more than $delay_{max}$ will be less than p_{delay}.

To estimate the effective bandwidth of a particular traffic flow, we take a recorded packet trace of that flow from the network. We observe that if we simulate a FIFO queue with the same inputted packet trace $\{T_M\}$ for different queue service rates $R_1 < R_2$ and estimate the proportions p_1 and p_2 of traffic delayed more than $delay_{max}$ for different rates respectively, then $p_1 > p_2$. This means that the proportion of traffic, p, delayed more than $delay_{max}$ is a monotonically decreasing function of service rate R. Using this observation we define, a simple binary search algorithm for a recorded packet trace to find the minimal value of a queue rate such that the proportion of traffic delayed more than $delay_{max}$ is less than p_{delay}.

3.3.1 Algorithm for Estimating Proportion of Violating Traffic using the FIFO Queue

There are two approaches commonly used to model a FIFO queue, the packet model or the continuous model. The packet level FIFO queue models the processing of each packet as a whole, where as a continuous FIFO queue models the processing of packets as a continuous bit stream. The latter approach distinguishes between total and partially delayed packets, including only the volume of traffic delayed in the calculation. The former approach will include the volume of a complete packet into the calculation, even if only partially delayed. With the inclusion of total packet size for partially delayed packets in the calculation of traffic violations, it is safe to assume that a packet model would result in a more conservative estimation of effective bandwidth. For the purpose of this work, we implement a continuous FIFO queue model for a more fine grained estimation of effective bandwidth to be guaranteed.

In Algorithm 1 we define our continuous FIFO queue model for use in the empirical estimation of effective bandwidth. This algorithm is used to calculate the proportion of violating traffic for a particular queue service rate. Table 1 summarizes the FIFO queue algorithm notation. Each packet in the trace consists of a pair of attributes that specify the packet size, denoted x_i in bits and packet arrival time denoted t_i in seconds. Let δ_{max} denote the maximum allowable volume of the queue buffer in bits before traffic experiences delay greater than $delay_{max}$. Let δ_{vol} denote the current volume of the queue buffer. Let $TOTAL_{vol}$ denote the total volume of traffic that has passed through the queue. Let $DELAY_{vol}$ denote the total volume of traffic that has exceeded the allowable bound of δ_{max}. Finally, p denotes the proportion of traffic delayed in respect to total traffic processed.

The algorithm assumes an infinite queue buffer, which is initially empty. The justification for using an infinite buffer is to ensure no packets are lost during the processing of the packet trace through the FIFO queue. To consider QoS targets of packet loss, a limit on the queue buffer would be imposed. The algorithm is passed a specified service rate R to process the packet trace, a specified maximum delay target on traffic $delay_{max}$, and a packet trace $\{T_M\}$ as input. Once the queue is

Algorithm 1 FIFO queue algorithm for estimation of violating traffic

```
Input: delay_max, R, {T_M}
Output: p

Set δ_vol = 0;
Set δ_time = 0;
Set TOTAL_vol = 0;
Set DELAY_vol = 0;

Set δ_max = delay_max R;

forall T(x_i, t_i) in T_M do
    //IF Queue is empty
    if t_i ≥ δ_time then
        δ_vol = 0;
        δ_time = t_i + x_i/R ;
        TOTAL_vol = TOTAL_vol + x_i;

    //Elseif packet is waiting
    elseif t_i < δ_time then
        δ_vol = (δ_time − t_i) R + x_i;
        δ_time = δ_time + x_i/R;
        TOTAL_vol = TOTAL_vol + δ_vol;

    //Traffic in violation
    if δ_vol > δ_max then
        DELAY_vol = DELAY_vol + (δ_vol − δ_max);
end
Set p = DELAY_vol / TOTAL_vol;

return p;
```

initialized, the algorithm sets a specified target queue volume δ_{max} by multiplying the specified service rate R and delay target $delay_{max}$ to calculate the maximum volume the queue can be before traffic is delayed greater than this specified target. If the volume of the queue δ_{vol} exceeds the maximum queue limit, all traffic beyond this limit will experience delay greater than the specified target.

The algorithm processes the packet trace as follows: For each packet, the packet arrival time t_i is compared to the time set after the queue has been emptied, following the previous packet arrival δ_{time}. If the packet arrival time is greater or equal to δ_{time}, then the queue is empty and the packet is processed. The packet is processed by updating the queue time by the time it takes the queue to process the packet at the specified queue service rate. If the packet arrival time is less than δ_{time}, this means the packet must wait to be processed. To calculate the waiting time, we must calculate the difference between the packet arrival time and the current queue time. The queue must process traffic for this duration before processing the arrived packet. At this point, we store the volume of traffic within the queue by adding the volume of traffic ahead of the packet plus the packet itself. If at this stage the queue volume δ_{vol} is greater than the maximum allowable queue

Table 1 Notation for the estimation of effective bandwidth

Notation	Description
$T(x_i, t_i)$	A packet within a trace list where packet size x_i and corresponding arrival times t_i
$\{T_M\}$	The set of all packets contained within trace T_M
M	The total number of packets in trace T_M
$delay_{max}$	Maximum allowable packet delay target in seconds
p_{delay}	Target proportion of traffic allowed to violate $delay_{max}$
R	The service rate of the FIFO queue
R_{mean}, R_{peak}	Measured mean and peak throughout rates of packet trace T_M
p	Corresponding proportion of violating traffic for service rate R, packet trace T_M and delay target $delay_{max}$
β	Identifies a margin of accuracy used to find the effective bandwidth value for a particular QoS violation target as a proportion of the QoS delay target p_{delay}
e	Identifies the actual margin of accuracy to be used relative to p_{delay}
R_{eff}	FIFO queue service rate that meets QoS requirements on proportion of traffic violating the specified packet delay target. We use this value as a measure of Effective Bandwidth
δ_{max}	Maximum queue volume, before traffic is delayed greater than $delay_{max}$
δ_{time}	Time to service the queue at service rate R, from the time of packet arrival
δ_{vol}	Current volume of the FIFO queue buffer
$TOTAL_{vol}$	Total volume of traffic that has been processed through the FIFO queue
$DELAY_{vol}$	Volume of traffic delayed greater than the target $delay_{max}$

volume δ_{max}, the difference is recorded as the volume of traffic in breach of the delay target.

As the algorithm proceeds, the total volume of traffic delayed greater than the specified delay target accumulates in $DELAY_{vol}$. When all packets have been processed through the queue, the algorithm will return the proportion of delayed traffic $DELAY_{vol}$ over the total volume of traffic processed, $TOTAL_{vol}$.

3.3.2 Effective Bandwidth Binary Search Algorithm

The effective bandwidth estimation algorithm, as depicted in Algorithm 2, controls the FIFO queue service rate to find a suitable service rate R where the proportion of violating traffic p is equal to the specified violation target p_{delay}. However, as p is a real number, comparisons of equality are difficult, therefore we implement an error region, denoted β. This error region enables us to control the accuracy of the algorithm in estimating the effective bandwidth of a recorded packet trace. We provisionally set this error region at 0.01.

The algorithm, as outlined in Algorithm 2 takes as parameters, a specified traffic delay target, denoted $delay_{max}$, a specified violation target (denoted p_{delay}), a packet trace (denoted $\{T_M\}$), and an error control variable (denoted β). p denotes the corresponding proportion of violating traffic for service rate R using the specified traffic delay target. The objective of the algorithm is to find a service rate R where

Algorithm 2 Estimation of effective bandwidth using a binary search algorithm

Input: $delay_{max}$, p_{delay}, $\{T_M\}$, β
Output: R_{eff}

Set $e = p_{delay}\beta$;
Set $R_{low} = calcMean(\{T_M\})$;
Set $R_{high} = calcPeak(\{T_M\})$;
Set $R_{mid} = R_{low} + \frac{R_{high} - R_{low}}{2}$;
Set $p_{mid} = calcViolations(delay_{max}, R_{mid}, \{T_M\})$;
while $(ABS(p_{mid} - p_{delay}) > e)$ **do**
  if $(p_{mid} < p_{delay})$ **then**
    $R_{high} = R_{mid}$;
  else
    $R_{low} = R_{mid}$;
  Set $R_{mid} = R_{low} + \frac{R_{high} - R_{low}}{2}$;
  Set $p_{mid} = calcViolations(delay_{max}, R_{mid}, \{T_M\})$;
end while
$R_{eff} = R_{mid}$;
return R_{eff};

its corresponding proportion of violating traffic p is equal to p_{delay} ($\pm$ the error region β). The error region is calculated as $\pm$ a specified percentage of p_{delay} dictated by the β attribute.

3.3.3 Effective Bandwidth Coefficient Calculation

The objective is to collect a number of short packet traces from each traffic class at a number of ingress points from around the network. The proposed effective bandwidth estimation algorithm is used to calculate the effective bandwidth of each packet trace collected. This approach is perfectly suited to this scenario as the effective bandwidth algorithm is independent of any traffic model, and only requires packet traces to operate along with supplied QoS targets on packet delay. A method is devised to relate collected effective bandwidth estimations to the estimated mean throughputs within the traffic matrix. The approach taken is to establish a generalised effective bandwidth to mean demand ratio as a method of enhancing the traffic matrix. This is termed the effective bandwidth coefficient and is established as follows: the mean throughput, $mean_i$, is calculated for each packet trace collected, as is the associated effective bandwidth for that packet trace $R_{eff,\,i}$ where i identifies the packet trace being evaluated from the set of collected traces, $i \in \{1, \ldots, I\}$; using these values we estimate the effective bandwidth coefficient k_i as:

$$k_i = \frac{R_{eff,i}}{mean_i} \tag{1}$$

The effective bandwidth coefficient k_i is calculated for all packet traces collected per traffic class. The set of coefficients calculated, allow us to generalise a relationship between estimated network mean demands and effective bandwidth requirements based on supplied QoS targets per traffic class within the network. Further considering a set of I coefficients k_1, …, k_I we first exclude any k_i with too low a mean rate using some appropriate threshold value. The reason for this is that for low levels of traffic aggregation, the effective bandwidth to mean throughput ratio would be quite high in comparison to higher levels of aggregation. This is due to the effect statistical multiplexing has on the effective bandwidth of aggregated traffic flows [15]. The contribution of the low mean throughput to overall network demand, is minimal in respect to the effect its associated coefficient may have on the set of coefficients. We then calculate a suitable representative coefficient of effective bandwidth K as the 95th or 99th percentile of the remaining set of coefficients. We believe that K_{95} (the 95th percentile) is an accurate reflection of the mean to effective bandwidth ratio per traffic class and can be used as a method of enhancing the traffic matrix.

3.3.4 Complexity Analysis

We first focus on the time complexity of the proposed algorithm. As the algorithm relies on the simulation of a FIFO queue model to estimate the proportion of QoS violations that a packet trace incurs for a particular queue service rate, each packet within the packet trace must be processed in succession. This operation happens in time $O(N)$, where N is the number of packets within the packet trace. As the algorithm uses a binary search algorithm to choose appropriate service rates dependent on the associated QoS violations experienced, the algorithm must repeat the previous operation in time $O(\log_2 M)$, where M is the search space of possible QoS violations. M is dependent on both the QoS violation target, p_{delay} and the error resolution parameter β. This error region, calculated as $\frac{p_{delay} * \beta}{2}$, is used by the algorithm to evaluate whether an appropriate QoS violation target has been found. The algorithm evaluates $\pm$ this value of the calculated p against the target p_{delay}. Therefore we can calculate M as $\frac{2}{p_{delay} * \beta}$. As the algorithm employs a binary search strategy to locate the QoS violation target and corresponding queue service rate, the theoretical number of iterations can be found as follows:

$$\log_2\left(\frac{2}{p_{delay} * \beta}\right) \tag{2}$$

The algorithm therefore runs in $O(N \log_2 M)$ time. The smaller the QoS violation target is, and the smaller the error region will be, the larger the search space will be. The performance of the algorithm can therefore be improved by two means; firstly by reducing the number of packets within the collected packet trace to reduce N, and secondly increasing either the QoS violation target or the error resolution parameter β. The first can be achieved by collecting a shorter packet trace or by collecting a trace at a lower resolution of time (e.g. bits per millisecond). In the second case, however, the QoS violation targets tend to be predetermined by the types of traffic and the higher level SLAs and QoS guarantees offered by the network operator. Therefore the controlling parameter to be used is β.

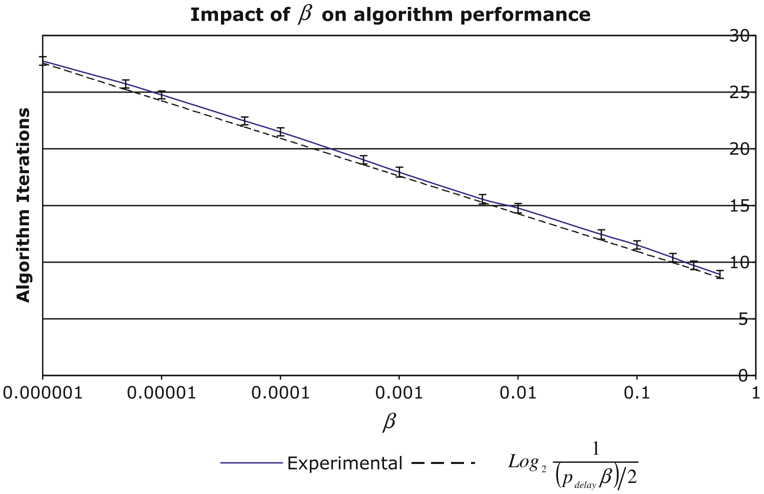

Fig. 2 Affect varying β has on algorithm performance

To evaluate the time complexity of the proposed algorithm we perform the following experiment. We replay a packet trace through the algorithm and vary the β parameter and with a fixed QoS violation target, p_{delay}, of 0.001. We measure the number of iterations of the algorithm for a number of different β values. The results are then plotted against the theoretical binary search function. The packet traces for this experiment were taken from the MOME data-set [22]. Two packet traces of a duration of 6.35 h were processed through the algorithm at 5 min segments. Each segment is processed through the algorithm and the number of iterations are recorded. Figure 2 depicts both the experimental results collected from $\tilde{1}50$ algorithm iterations using different β values and the theoretical binary search of an equivalent search space; as can be seen the algorithm performs in line with the expected theoretical equivalent. Based on the results in Fig. 2, we would recommend using a β value of 0.01 to achieve an acceptable degree of accuracy while ensuring a reasonably fast response from the search algorithm.

With regards to the space complexity of the algorithm, a packet trace is loaded into memory in its entirety once before the algorithm is executed. Therefore the space complexity of the algorithm is O(N), where N is the number of packets within the trace. Efficiency can therefore be increased by reducing the size of the packet trace or by increasing the resolution at which the trace is collected.

3.4 Estimation of the Traffic Matrix from Accounting Data

Based on the type of accounting data records available to the network operator, we propose the following mediation process to produce a traffic matrix for a network planning process. The process assumes that accounting data records capture traffic demands between source and destination nodes across the network in the form of IP

Table 2 Sample accounting data flow record

SrcAddr	DestAddr	DSCP	Size	Packets	Start	End
10.37.2.22	10.34.1.118	EF	332049	1032	12:35.31	12:48.22

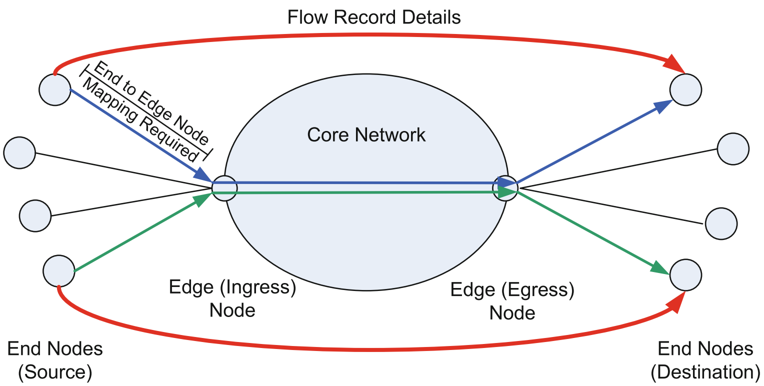

Fig. 3 Mapping end node flow records to corresponding edge nodes

flow records as depicted in Table 2; the record in Table 2 has a format similar to that of the IETF IPFIX [21] and NetFlow 9 [23] accounting records.

The traffic matrix estimation algorithm depends on a number of assumptions on accounting records and traffic within the network. It assumes that an end node device[1] is attached to a single edge node[2] of the network. We assume that all traffic entering the network through an ingress edge node, will exit the core network through an egress edge node, therefore no traffic is generated or consumed within the core network. We assume that accounting flow records are only created for traffic entering the network through an ingress edge node. With this, we can be guaranteed that traffic is only recorded once, and that all traffic generated over the network will be accounted for.

Given the current trends in multi-homing we do recognize the fact that end hosts can generate traffic that has multiple points of entry and exit through the core network. Take for example the Locator Identifier Separation Protocol (LISP) architecture [24]; in this architecture the location (Routing Locator or RLOC) and the identification (Endpoint Identifier or EID) of the end node are separated. This allows an ISP to manage which point of connection to a network the node can use without changing its IP address. As this is a router based approach, the ISP will control what RLOCs are assigned to EIDs.

We believe that this information can be easily integrated into the QoSPlan framework in the sense that the ISP explicitly controls the mapping between end

[1] We consider an end node device to be a source or sink of a traffic flow, i.e. the source or destination nodes attached to the network.

[2] We consider edge nodes to be the point of attachment of an end node device to the core network, i.e. the ingress or egress nodes of the network

node and edge node. The traffic sourced from an edge node will therefore be mapped to its associated Routing Locator (RLOC). Similarly the destination end node will also have an associated RLOC. Assuming that such a system was in place, QoSPlan can interact with the LISP architecture to identify the source and destination RLOCs for associated EIDs of end nodes.

The objective of the traffic matrix estimation algorithm is to map accounting flow records which only store source and destination edge node information, to their respective edge nodes over which their traffic flowed Fig. 3. Once this relationship is identified, the demand of the flow record can be associated with the corresponding edge node.

In estimating network demand from accounting system records, the algorithm makes a number of assumptions. It assumes that packet inter-arrival times and packet sizes within a collected flow record are uniformly distributed. The reason for this assumption is required is that flow records do not record any information pertaining to the packet size or inter-arrival times of the packets monitored during its creation. Unless clear evidence suggests that another assumption would be appropriate and improve accuracy over the measurement intervals we are interested in, we believe this simplifying assumption will yield appropriate accuracy.

By taking this assumption the proportion of demand within an interval can be easily calculated by multiplying the flows mean rate by the duration of time the flow exists within the current interval. However, this assumption can lead to some inaccuracy in the final estimation process, as throughout the flows duration, packet inter-arrival times and sizes are not normally uniform.

Based on the fact that a flow record contains at the least information such as that in Table 2, each flow record will have a start time (t_{start}) and an end time (t_{end}). The flow's rate $r(f)$ can be calculated from the flow size divided by the flow duration. As the flow record holds the traffic class the record originated from, each traffic matrix can be traffic class specific. The figure shows 4 cases the algorithm captures. The objective of the algorithm is to sum up all demand of all flows that lie within a particular interval $(t, t + t')$.

- Case 1 captures demand of flows that begins before the period and ends during it;

$$d = r(f)(t_{end} - t) \tag{3}$$

- Case 2 captures demand of flows that start within the period and ends after it;

$$d = r(f)((t + t') - t_{start}) \tag{4}$$

- Case 3 captures demand of flows that start and end within the time period, and finally;

$$d = r(f)(t_{end} - t_{start}) \tag{5}$$

- Case 4 captures demand of flows that starts before the period and ends after it.

$$d = r(f)t' \tag{6}$$

Algorithm 3 Traffic matrix estimation from accounting flow records

---

Input: $EdgeRouters_N$, Interval, TotalTime
Output: TrafficMatrix[$IngressRouter, EgressRouter, Interval, flow_{TOS}$]
Init(TrafficMatrix); **foreach** *ingressRouter in EdgeRouters* **do**
  foreach *meteringDevice in ingressRouter* **do**
    foreach *sourceNode of meteringDevice* **do**
      foreach *flow of meteringDevice* **do**
        if $flow_{srcip} == sourceNode$ **then**
          Demand = calculateDemand(timeInterval, interval, flow);
          egressRouter = findEgressRouter($flow_{destip}$);
          TrafficMatrix[ingressRouter, egressRouter, Interval, $flow_{TOS}$] + = Demand;
return *TrafficMatrix*

This process is used to estimate demand of a single flow over a particular period. This process is then repeated for all collected accounting records. The objective is to calculate the demand generated by the flow records between two edge node pairs for a particular traffic class. Algorithm 3 has four nested loops, looping through each ingress router, each metering device, and each end node attached to that metering device, and each flow record within that current metering device. The algorithm matches each flow record to a source node, and estimates the flow's demand within the current measurement interval. The algorithm then matches the destination address of the flow record to a particular egress edge node. This mapping allows us to identify where the traffic is exiting the network.

The findEgressRouter() function in Algorithm 3 is used to return the egress edge node corresponding to where the flow exits the core network. This function is a simple static table lookup of mappings between the destination end node address within an accounting record and its associated egress edge node. Once the egress edge node is found, the demand estimated for that particular accounting record is added to the appropriate dimension within the traffic matrix. Once all records have been processed, the traffic matrix is returned. Table 3 outlines an example static mapping between edge nodes and end nodes. This can be queried to map an accounting data record to an appropriate dimension within the traffic matrix. Based on this table, for example, the record in Table 2 would map to the dimension within the traffic matrix for ingress edge node A and egress edge node D. This static lookup table can be created using methods such as interrogation of routing tables within edge routers [18]. An important point to note is when there are modifications in how end node devices are attached to the network, i.e the change of point of attachment

Table 3 Mapping between edge nodes and end nodes

Edge node	End point mappings
A	10.37.1.*, 10.37.2.*, 10.37.3.*
B	10.36.1.*, 10.36.2.*, 10.36.3.*
C	10.35.1.*, 10.35.2.*, 10.35.3.*
D	10.34.1.*, 10.34.2.*, 10.34.3.*

will require an update to the lookup table. Therefore the mappings between end nodes and points of attachment must be maintained up to date.

4 QoSPlan Framework Evaluation

We now describe a deployment scenario to evaluate the effectiveness of QoSPlan in supplying input to a network planning process. The objective is to demonstrate that QoSPlan can provide this information with adequate accuracy for long term network planning. We use the term long term network planning to denote configuring a network for traffic demand and QoS requirements on time scales of days or weeks. We note that for long term planning the level of accuracy required is minimal as planning on this time scale is limited by human usage patterns, which cannot be accurately predicted.

We evaluate a number of configuration options that can be controlled by QoSPlan to improve accuracy for various traffic conditions, such as elastic and streaming traffic, and monitoring conditions, such as packet sampling. As a basis of comparison, we deploy a direct monitoring system within the network to record actual network demand and effective bandwidth levels per traffic class. The following sections discuss the details of our scenario, simulation topology and traffic settings, and finally a set of experiments to test how configuration settings affect accuracy.

4.1 Simulation Settings

We propose to evaluate QoSPlan under the following scenario: a single domain network operator offering DiffServ controlled services to subscribed end users with guaranteed QoS targets on packet delay. The network operator uses a deployed IPFIX network monitoring system to supply accounting records to its accounting system for billing purposes. We have simulated a network topology using the OPNET™ modeler[3][25]; the network topology consists of four core routers, six edge routers and ten workstations (Fig. 4). The topology is designed in such a manner as to allow all service traffic to cross the core network through at least one core router. Five workstations operate as servers, with the other five workstations operating as consumers. Each workstation is connected to the network by a 10 Mbps Ethernet link. Customers have access to five services; Web browsing, Email, Database, Video on Demand (VoD) and Voice over IP (VoIP). We use the standard OPNET™ application models [25] to model the characteristics of traffic generated by users accessing these services. These traffic models are parameterised to model typical user behavior in the work place as outlined in Table 4. The traffic has been modeled using this approach as we wish to create multiple application interactions across the network between different source and destination pairs; this will ensure

[3] Each OPNET simulation model was build using existing OPNET network models. An IPFIX device and packet probe device was developed as an add-on to the OPNET router models. The implementations were validated within a number of test case scenarios, which demonstrated expected results

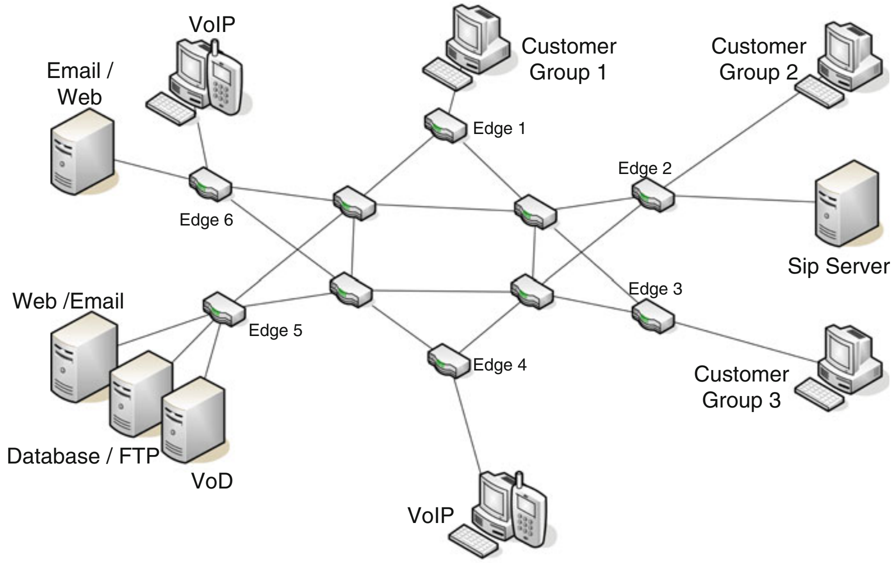

Fig. 4 QoSPlan simulated small network topology

an appropriate mix of traffic types and density of flow records collected by the accounting system.

As different applications require different QoS requirements, two DiffServ traffic classes are deployed within the network, one each for elastic and streaming traffic. Services aggregated into Assured Forwarding (AF) generate elastic traffic and include web and email, and Database. These services specify loose QoS targets on packet delay of (0.04 s, 0.001). Voice over IP (VoIP) and Video on Demand (VOD) applications are sources of streaming traffic and are aggregated into Expedited Forwarding (EF) with a QoS packet delay target of (0.02 s, 0.0001). Traffic is policed and marked at the ingress routers, where packets are assigned appropriate DiffServ Code Points (DSCPs). Within the core network, all core routers are configured with common Per Hop Behavior (PHB) settings to ensure all traffic within a particular traffic class is treated the same by each router.

For collecting accounting records, IPFIX monitoring devices are positioned at the ingress interface of all edge routers. The devices deployed are based on the IPFIX architecture [21]. They can be configured to collect accounting records based on different packet sampling settings. Within the experiments, we evaluate the effect various sampling settings have on the estimation of network demand for elastic and streaming traffic.

For the collection of packet traces, a single monitoring device is modeled as being attached to an ingress interface of an edge router. The device is used to collect packet traces for effective bandwidth analysis. The monitoring device can filter packets from particular DiffServ traffic classes by reading the DSCP within the

Table 4 OPNET traffic model settings

Traffic class	Application	Settings
Assured forwarding	Email	Send/Receive: exp(360 s)
		Mails sent in groups of: 3
		Email size: 2000 bytes
		Users per customer group: 75
	Web	Page inter-arrival: exp(60 s)
		Page size: 1,500 3,000 bytes
		Pages per server: 10
		Users per customer group: 75
	Database	Ratio of queries to other: 100/1
		Transaction inter-arrival: exp(9 s)
		Transaction size: 1,024 bytes
		Users per customer group: 125
Expedited forwarding	VoIP	Codec: G.729 (silence suppression)
		Talk length: exp(0.65 s)
		Silence length : exp(0.325 s)
		Users per customer group: 10
	VoD	Based on Codec: H.264
		24 frames per sec
		Packet size: 550–650 bytes
		Users par customer group: 10

packet header of each monitored packet. This device can be moved between ingress points around the network, and is only operational during collection.

For experimental comparison to QoSPlan, a direct monitoring system is also modeled. This monitoring system will collect every packet that passes an ingress router interface. These packet traces are used to calculate exact network demand, and effective bandwidth estimations within the network.

4.2 Effective Bandwidth Coefficient Selection

We now study the decision process used in selecting an appropriate representative effective bandwidth coefficient. To achieve this we analyze 1000 effective bandwidth coefficients estimated from collected packet traces for both traffic classes and plot a distribution for each. Figs. 5 and 6 depict the distribution of effective bandwidth coefficients collected for AF and EF packet traces respectively. As can be seen, the coefficients approximate a normal distribution.

From this distribution, we must then choose an appropriate value to represent the effective bandwidth coefficient of the associated traffic class. This chosen coefficient will be used to enhance the traffic matrix at the final preparation phase of QoSPlan. We recommend choosing the 95th percentile of this range as it represents the relationship between measured mean demand on the network and

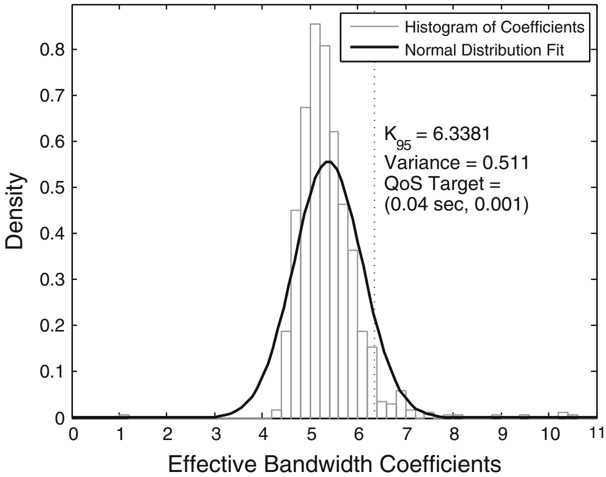

Fig. 5 Effective bandwidth coefficient distribution of AF traffic

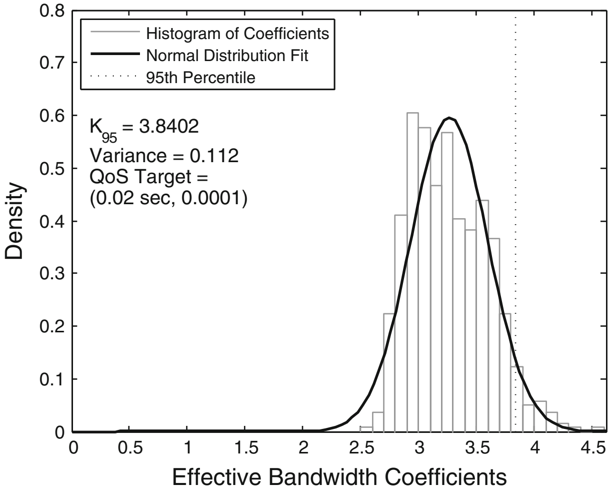

Fig. 6 Effective bandwidth coefficient distribution of EF traffic

near peak required effective bandwidth for that traffic to maintain outlined QoS targets. Were we to select the mean of the coefficient set, vital mean demand throughput to effective bandwidth relationships will be neglected, leading to underestimation of effective bandwidth levels from the traffic matrix. However, in choosing the 95th percentile, we are ensuring that QoSPlan is supplying a network planning process with adequate information to ensure resources are provisioned for the traffic demands on the network. Table 5 depicts the chosen effective bandwidth coefficients that will be used to enhance the traffic matrix in further experiments.

Table 5 Effective bandwidth coefficients used to enhance the traffic class specific traffic matrices

Traffic class	QoS delay target	Coefficient
AF	(0.04 s, 0.001)	6.3381
EF	(0.02 s, 0.0001)	3.8402

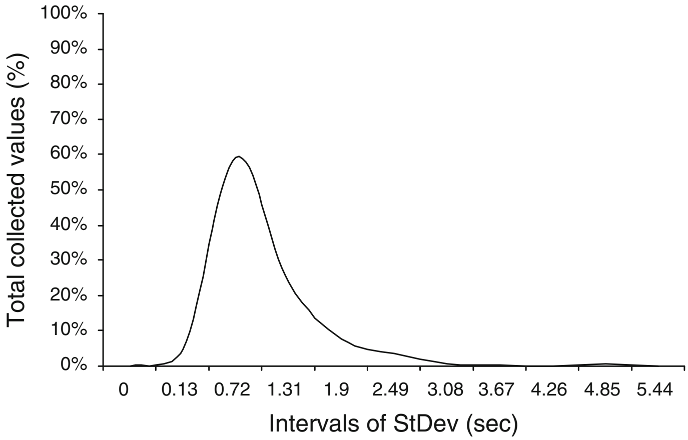

Fig. 7 Flow duration distribution for AF traffic

4.3 Captured Flow Record Characteristics

We analyze the characteristics of accounting data flow records collected for both AF and EF traffic classes to demonstrate the difference in characteristics of flow records generated from elastic and streaming traffic. For this comparison, we analyze the distribution of flow durations of the collected records. The traffic was generated between workstations on the network following the traffic model settings in Table 4. As can be seen in Fig. 7 the AF traffic demonstrates a heavy tailed distribution with a mean of approximately 0.8 s. On the other hand, the EF traffic has a longer mean duration of approximately 90 s with a heavy tail (Fig. 8). The heavy tails of these distributions demonstrate the existence of flow records many times the mean, collected for both traffic classes. The knowledge of these flow duration distributions is vital in choosing appropriate traffic estimation intervals per traffic class. We use these observations in configuring the QoSPlan demand estimation process.

4.3.1 Aggregation and Sampling

This section analyses the effect that deterministic sampling strategies that may be employed by the network accounting system during the collection of accounting data have on the estimation of network demand and estimation of effective bandwidth with associated coefficients. Packet sampling is assumed to take place on the IPFIX device where packets are monitored and processed into flow records. For QoSPlan to use the sampled accounting data to estimate demand comparable to that

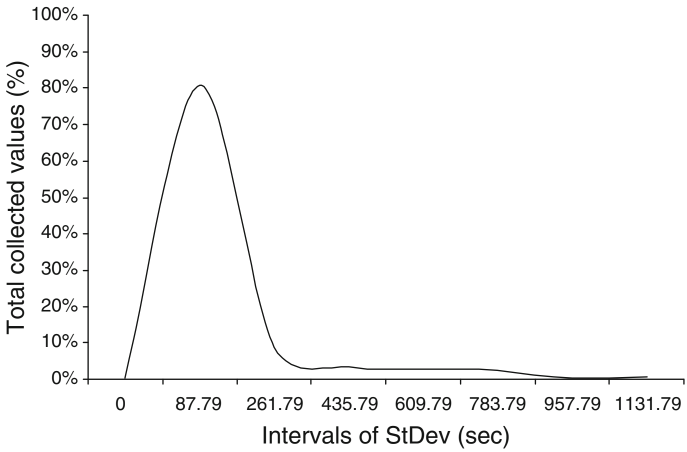

Fig. 8 Flow duration distribution for EF traffic

of direct measurement approaches, the demand estimated is scaled according to the packet-sampling interval.

In the case of a deterministic packet sampling interval n of 1 in 100 packets being employed. If an accounting record was calculated to contain a volume v of 1Mbytes over its duration, as only 1 in 100 packets would get processed into the flow record, the volume is scaled 100 times to calculate an equivalent scaled volume V of 100Mbytes, as if all packets within the monitored traffic flow were collected. Scaling of demand in such a manner depends on the assumption of a uniform distribution of packet rate throughout the recorded traffic flow.

A comparison is performed between directly measured network demand between edge router pairs, and demand estimated from accounting data, subject to sampling, collected between the same pair of edge routers over a set interval of 5 min. We perform this experiment for a range of deterministic sampling values and plot the relative error between demand estimated through direct measurement and using the traffic matrix estimation algorithm with appropriate scaling (Eq. 7).

$$V = n * v \tag{7}$$

Figure 9 demonstrates the variation in accuracy of demand estimation from collected accounting data as sampling intervals increase for different application traffic. This demand was calculated based on accounting flow records created using a number of different sampling intervals, where N represents the size of the set a packet is sampled from, i.e. every Nth packet is collected. Each flow level demand value calculated was compared to a corresponding direct measurement over the same interval for the same edge node pair. From this a relative error between the two values was estimated and plotted. In this case, the relative error between demand directly measured from the network and estimated network demand from collected flow records increases up to 18 % for the AF traffic when a packet-sampling interval $N = 1000$ is used.

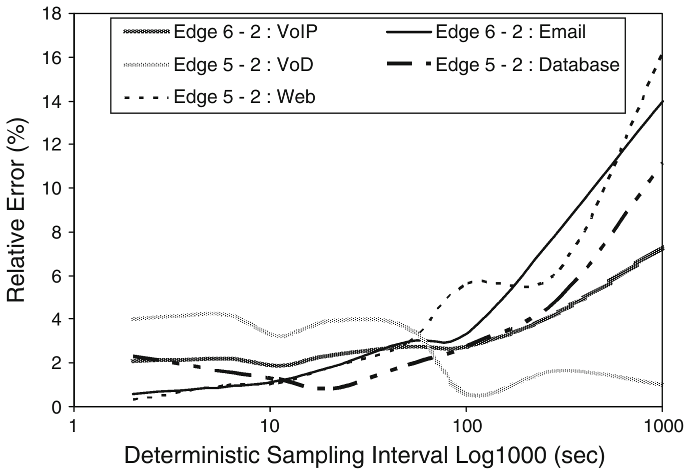

Fig. 9 Relative error in demand estimation from accounting records collected using deterministic sampling

The main reason behind this degradation in accuracy is due to the scaling of demand from collected accounting data. As elastic traffic tends not to follow a uniform distribution of packet size or inter-arrival times, scaling can skew the results of demand estimation. Therefore, for AF traffic, we recommend a reduced sampling interval of up to $N = 100$ to maintain an acceptable level of accuracy. For EF traffic, relative error manages to remain below 8 % even up as far as a packet-sampling interval of 1000. The main reason for this behavior is down to the fact that streaming traffic tends to have a more uniform distribution of packet size and inter arrival times, lending itself well to the scaling process.

The configuration of this attribute can also affect the accuracy with which QoSPlan estimates effective bandwidth. As the effective bandwidth coefficient QoSPlan uses captures a general relationship between mean demand and effective bandwidth requirements, if mean demand is inaccurately estimated the estimated effective bandwidth will be affected.

We now analyze the relative error between directly measured effective bandwidth values, estimated using packet traces collected every 5 min from ingress router interfaces per traffic class, and effective bandwidth estimated with coefficients (in Table 5) to enhance the traffic matrix prepared using different sampling intervals. For each 5 min interval, the directly measured effective bandwidth is compared with the QoSPlan estimated effective bandwidth and a relative error is estimated.

Figure 10 demonstrates a distribution of global relative error between directly measured and coefficient estimated effective bandwidth values, measured for 1,000 different 5-min intervals. We can see that, for AF traffic, as the packet sampling interval increases, the mean global relative error increases from 11 % for N = 100–37 % for N = 2,000. This result demonstrates the effect sampling can have on the estimation of effective bandwidth by QoSPlan using the traffic matrix enhanced by effective bandwidth coefficients.

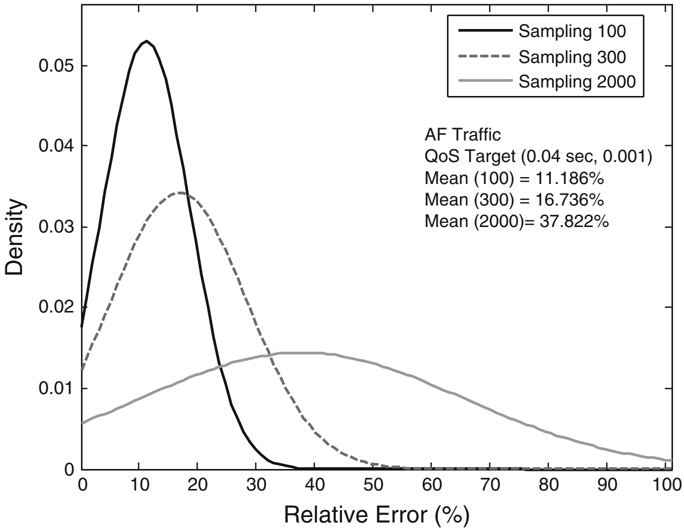

Fig. 10 Directly measured effective bandwidth versus QoSPlan estimated effective bandwidth for AF traffic

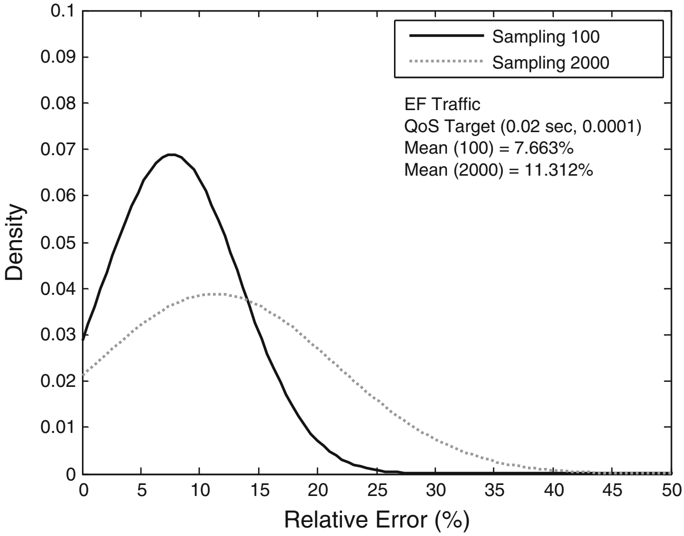

Fig. 11 Directly measured effective bandwidth versus QoSPlan estimated effective bandwidth for EF traffic

Figure 11 demonstrates results for EF traffic following the same procedure as above. We can see that for the EF traffic, sampling has a smaller effect on mean global relative error. For a sampling interval of 100, the mean global relative error is approximately 8 % while for a much higher sampling interval of 2000, global relative error averages around 11 %.

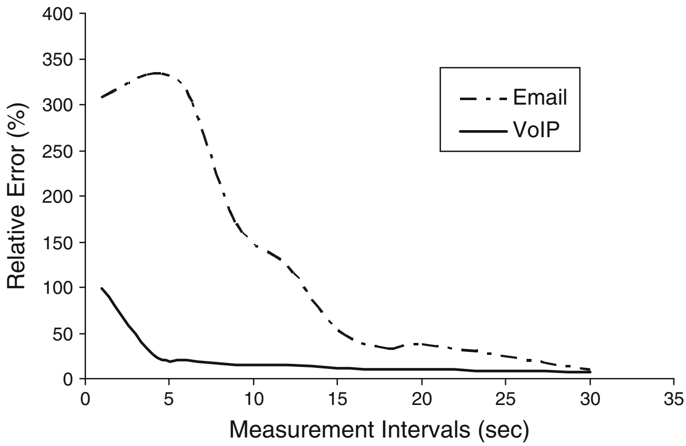

Fig. 12 Varying the demand estimation interval for AF and EF traffic

The previous experiments have been measuring demand over 5 min intervals. We now investigate the effect varying this setting has on the demand estimation and traffic matrix enhancement accuracy of QoSPlan. We firstly evaluate the estimation of network demand from accounting data versus directly measured demand and graph relative error between the two for different demand estimation intervals. This is performed on both AF and EF traffic. The results from this study are depicted in Fig. 12.

For AF traffic, over very short intervals of below 10 s, we see that relative error can be as high as 300 % between the average network demand estimated over the interval from accounting data in comparison to direct demand measurements. This high degree of error is attributed to a number of factors including, the type of traffic being monitored, the assumption our demand estimation algorithm makes regarding the packet distribution within flows being uniform, and the process of flow division between measurement intervals. As the demand estimation algorithm divides flows proportionally between neighboring intervals, this can cause a high degree of error in measuring demand per interval for AF traffic, as packets are generally not uniformly distributed within elastic traffic flows. For EF traffic, the case is different as the traffic tends to be more evenly distributed through a flow. This is down to the fact that applications generating EF traffic maintain a relatively steady stream of traffic throughout the duration of the session, such as a video stream, or voice call. Therefore, the division of EF flows into intervals only results in a relative error of less than 50 % for measurement intervals of under 10 s, with this reducing for larger measurement intervals.

We also notice that the relative error reduces by a considerable amount (to below 10 %) for measurement intervals greater than 30 s. As within QoSPlan, demand from accounting records will be estimated on the scale of minutes, we maintain that such an assumption of uniform packet distributions within flows is an acceptable assumption. We therefore do not investigate efforts of reducing this relative error.

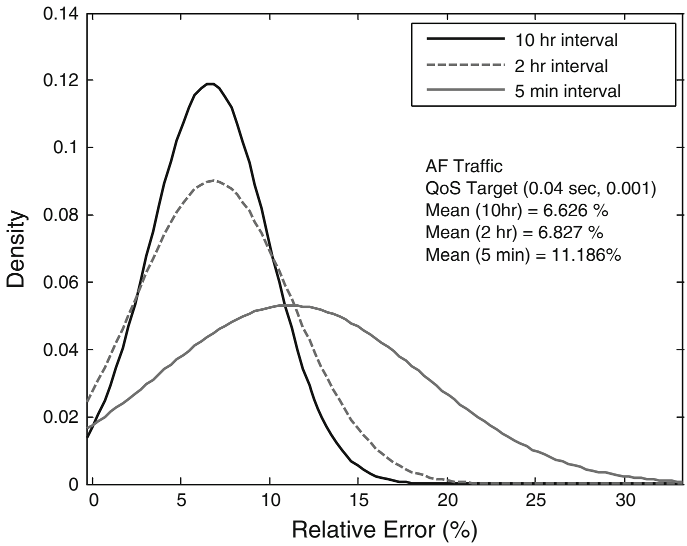

Fig. 13 Variation in global mean relative error for different demand estimation intervals for AF traffic in comparison to directly measured demand

We also demonstrate the effect this setting has on the prediction of effective bandwidth levels. We perform a study measuring global relative error between effective bandwidth calculated from the traffic matrix enhanced with effective bandwidth coefficients and directly measured effective bandwidth levels. The analysis involves comparing the relative error between the two approaches over different measurement intervals. We plot a distribution of relative error for each set of values recorded. We can use this distribution to estimate a mean relative error between the two approaches, for a specified measurement interval. In Fig. 13, we show a mean relative error of around 6 % for AF traffic at a measurement interval of 10 h. The mean remains at just over 6 % for a reduced measurement interval of 2 h. As the measurement interval is reduced to 5 min the distribution of relative error increases to a mean of approximately 11 %. We see from Fig. 14 that for streaming EF traffic there is little variation in relative error over different measurement intervals ranging from approximately 7 % for 1 and 2 h, reaching close to 9.5 % for a measurement interval of 10 h. From this we can see that the longer the measurement interval, the more accurate effective bandwidths can be estimated for both AF and EF traffic.

4.4 Issues of Scalability

When taking scalability into consideration, there are a number of factors that come into play. We now offer a discussion on these factors and offer arguments in favor of using QoSPlan within ISP networks of various sizes and traffic volumes. These factors include: centralized versus a decentralized deployment;

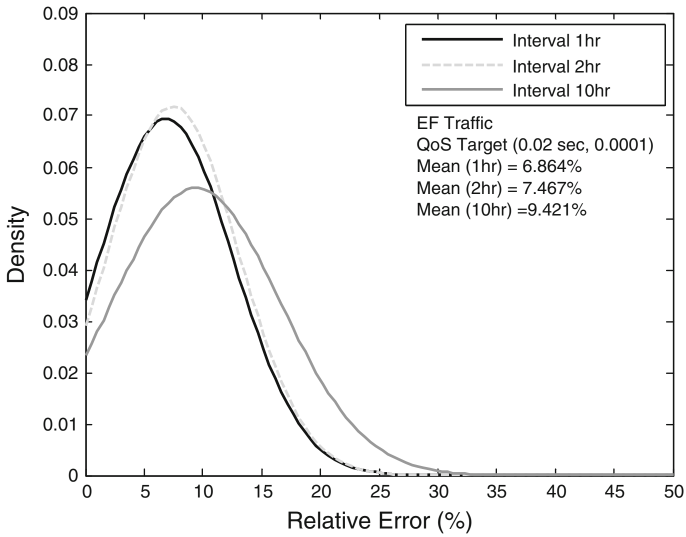

Fig. 14 Variation in global mean relative error for different demand estimation intervals for EF traffic in comparison to directly measured demand

when and where measurements are taken from the network to estimate effective bandwidth coefficients; and available accounting resources for networks of different sizes.

Based on the algorithm specified in Algorithm 3, a centralized accounting system is assumed. This means that all flow records that are created from the monitored traffic are exported to a single collection module[4]. At this collection module the algorithm will process all records within a particular time interval and estimate demand between identified ingress points and their associated egress points. For small ISPs this assumption is generally valid as multiple collection modules may not be required (this is dependent of the volumes of traffic, and in turn the volume flow records collected). However in a large ISP network, there may be multiple collection modules to which flow records are exported to. In this case the algorithm can be easily distributed so that only records on each of the collectors are processed. One criteria for this process to work is that all collectors have access to the same topological information. This is to ensure all analyzed flow records can be associated to an ingress point and an egress point. The resultant traffic matrices can simply be added together to prepare a network wide traffic matrix. Papagiannaki et al. [18] has already proposed a distributed version of such an algorithm that can easily be ported to our purposes. This ensures the scalability of estimating the traffic matrix to large ISP network topologies.

[4] We take the term collection module to mean a centralized location for the storage of accounting records collected from a set of metering devices

5 Case Study: GÉANT Network

To investigate the applicability of QoSPLAN to a realistic network setting, we carried out a case study on the GÉANT network [26]. Our study involved using a real traffic matrix collected from the GÉANT network and applying an effective bandwidth coefficient to this matrix. We begin by analyzing collected packet traces under a range of QoS targets to depict the change in effective bandwidth. We then go on to apply a range of effective bandwidth coefficients to the collected traffic matrix. Through knowledge of GÉANT network link capacities, we calculate the percentage of links that are under-provisioned to ensure specified QoS delay targets are maintained.

GÉANT is a pan-European network connecting universities and research institutes, with a Point of Presence (POP) in each european country. The network topology is composed of 23 routers connected using 38 links as shown in Fig. 15. We focus on a traffic matrix produced from the TOTEM project of the GÉANT network [27]. At the time of collecting the traffic matrix, each link supported a throughout capacity of 155 Mbps.

To build an accurate traffic matrix, Netflow records are collected at a packet sampling rate of 1/1,000 at each POP. The traffic matrix is generated over intervals of 15 min and the estimated network throughput is multiplied by 1,000 to scale the results in line with the sampling rate. The TOTEM toolbox uses this input to generate both a network topology model and traffic matrix model.

To complement the traffic matrix, QoSPlan advocates the collection and analysis of packet traces from the network. This is to produce appropriate effective

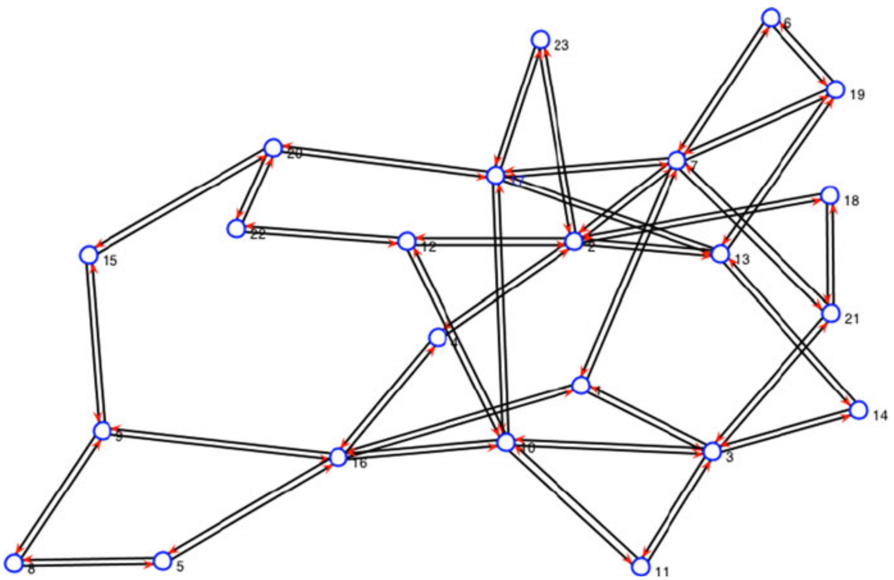

Fig. 15 GÉANT Topology, as generated from the TOTEM network model tool [27] using LocalLayout view

Table 6 Protocol traffic observed from packet traces collected at the CAIDA San Jose passive monitor A from January 19 2012 to February 16 2012

Application	Bits %	Packets %	Tuples %
HTTP	82.14	70.74	43.65
HTTPS	4.34	6.79	8.36
RTMP	4.31	3.83	0.09
UNKNOWN_UDP	3.38	8.65	13.77
UNKNOWN_TCP	3.15	4.30	5.62
SQUID	0.45	0.26	0.09
IPSEC	0.21	0.27	0.00
SSH	0.19	0.16	0.17
QUAKE	0.19	0.50	0.92
SHOUTCAST	0.18	0.11	0.02
ABACAST	0.14	0.35	0.66
SMTP	0.14	0.34	0.39
NOPORTS_UDP	0.05	0.06	0.07
Other	1.14	3.64	26.19

bandwidth coefficients to be used to enhance the traffic matrix. As no packet level traces were available to the authors, indicative effective bandwidth coefficients were chosen to reflect the application traffic mix traversing the GÉANT network. To facilitate estimation of effective bandwidth coefficients of modern day Internet traffic, we use a packet trace collected from the San Jose monitor A equinix node available on the CIADA database [28]. An analysis of the packet trace depicts the protocol breakdown as seen in Table 6. This represents a realistic mix of application traffic.

The packet traces collected were selected between the dates of January 19 2012 and February 16 2012. A total of 30 packet traces were collected, each with a duration of approximately 60 s. Each packet trace was analyzed to estimate its effective bandwidth for a range of QoS targets. Figure 16 depicts a graph of the collected results. As can be seen as the QoS target of both maximum packet delay and probability of target violation reduce the associated effective bandwidth coefficient increases. For example we can see for a QoS target of (0.001 s, 0.001), stating that no more than 0.001 proportion of traffic can be delayed greater than 0.001 s, the effective bandwidth coefficient is 1.45.

The next step in the QoSPlan process is to apply this coefficient to the collected traffic matrix through multiplication. Considering we know the capacity of each link within the network and the traffic demands between each node pair, we can calculate the proportion of node pairs that are currently under provisioned to ensure QoS targets are maintained. For example, if we apply the EB coefficient of 1.45 to the collected traffic matrix, we see that 2.72 % of node pair links are under provisioned. We carried out a further analysis of the impact a range of EB coefficients would have on the GÉANT network. Figure 17 depicts that as the EB coefficient increases, so does the proportion of under-provisioned links. This input is vital for network planning tools from the point of view of capacity planning and traffic engineering.

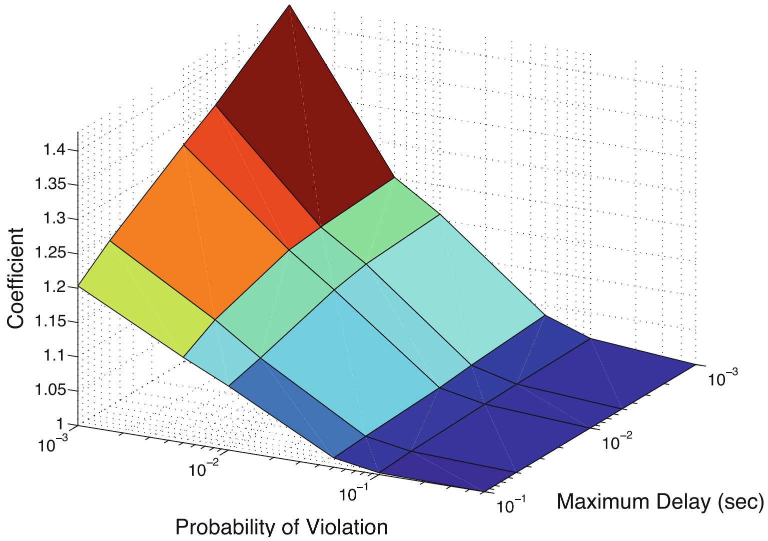

Fig. 16 Effective bandwidth coefficient

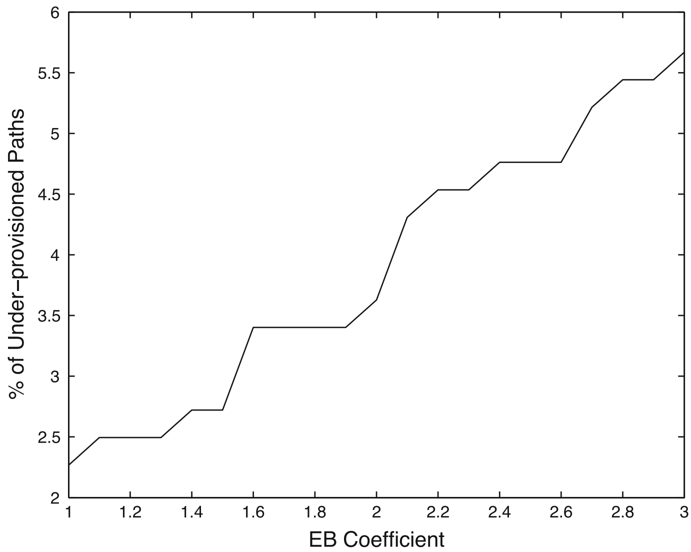

Fig. 17 Percentage of under-provisioned paths as a range of effective bandwidth coefficients are applied to the traffic matrix

6 Economic Analysis

One of the central contentions of this chapter is that a deployment of QoSPlan is significantly more cost effective then a traditional direct monitoring system deployment for supplying input to the network planning process. The previous section demonstrated that QoSPlan can supply QoS related input for long term planning with an acceptable degree of accuracy. We now present the results of a high level comparative economic analysis of the two approaches for three ISPs. Specifically, we compare the costs of extending an existing network accounting system to implement the QoSPlan process with that of a traditional direct monitoring system operating independently of the network accounting system. We proceed by stating our baseline cost assumptions for both deployments. We then state the associated capital and operational costs of both systems, followed by an economic comparison. For a further analysis of cost breakdown with regards ISP expenditure on network management, a recent report [29] has been published discussing the various attributed cost factors. From this report we can see that the ISP spends on average 18 % of all management software expenditure on systems supporting network capacity planning activities. The objective of this study is to demonstrate the possible reductions in cost that can be expected with a deployment of QoSPlan by the ISP. We also note that our study is based on incumbent network installations and do not consider the overall costs associated with green field network deployments.

6.1 Baseline Cost Assumptions

To form the basis of the economic analysis, we first outline common cost assumptions across the two deployments. For the comparison we assume the network operator has a deployed network accounting system for usage based accounting. For the currently deployed network accounting system the network operator is required to pay a number of fees, most significantly database system license fees and accounting system software license fees. The network operator also has to pay for customer support for each of these software systems; in general, these fees are set in line with the software license fees. It is common for network operators to incur hardware related costs, for example rental of hardware storage space, hardware-specific support costs, and costs associated with the network operator's replication policy. A replication policy may state that for every one database live within the network, there must be another two database servers replicating every transaction, for redundancy. We assume for this economic analysis that the network operator does not pay rental on hardware space, and does not employ a replication policy. Of course, license fees are typically kept confidential, so the values we choose are based on anecdote. We assume that license fees increase for larger sized network topologies, which we believe is universally true. Note that our cost model is relatively simplistic; for example we disregard costs such as loss in revenue based on depreciation, down time, data migration.

Given the above assumptions, Table 7 outlines indicative costs for three ISPs relating to the costs associated with management of their network accounting system. We base the network sizes on existing topologies, including that of the Irish national research and education network HEANET [30] as a small ISP, and the European wide research and education network GÉANT as a comparatively large ISP [26].

Table 7 Current operational costs per network

	Small	Medium	Large
Network nodes	5 edge, 3 core	20 edge, 5 core	45 edge , 15 core
Support costs	$20,000	$60,000	$200,000
Data base license fee	$20,000	$60,000	$200,000
Software license fees	$20,000	$60,000	$200,000

Key to our approach is a significant upgrade of the network accounting system to support a QoSPlan deployment. For any significant system upgrade all software, hardware and support costs are likely to increase. We assume that these costs collectively increase by 20 %. In addition to the upgrade, specialised contractors must be hired for tasks such as installation, staff training, and on-site support. We assume a contractor charges a flat rate of $1,500 per day, and that he/she can work at a rate of one unit installation per day. However, as the network size increases, so too does the complexity of the installation; thus, we assume that for our medium and large sized networks, contractor fees raise to $1,750 and $2,000 per day, respectively.

Finally, to support new functionality without degrading system performance, additional servers will typically be purchased and deployed; we assume that purchase and deployment of a single server is $5,000. As well as upgrading the accounting system, a QoSPlan deployment also requires use of a limited number of network monitoring devices for collection of traffic traces used in the effective bandwidth estimation process. Based on the geographical size of the network and the duration of time over which network planning is performed, we also must ensure that it is feasible to move the monitoring devices from location to location within an appropriate period of time. For this reason we assume that a maximum of 3 days is reserved for the movement of a device, and that planning is performed over 30 day cycles.

Based on the specified size of the networks and the assumptions on metering device movement restrictions, a small network operator requires 1 such device, for a medium sized network operator we assume 2, and for the large network operators we assume 4 devices are required. We assume that such devices cost approximately $3,000 each. We base this on list prices quoted for such probe devices as the Network Instruments ethernet probe [31]. Obviously these prices may vary due to volume purchased, however we simplify this calculation by setting a standard rate for all ISPs. In contrast for the measurement-based network planning approach network monitoring devices must be deployed permanently at all edge routers in the network, resulting in a significant cost overhead for larger network operators.

6.2 Comparative Cost Analysis

Based on the cost assumptions outlined above we can now estimate the cost of a QoSPlan and direct monitoring system deployment. We focus on how these

Table 8 Cost of an incremental QoSPlan deployment

	Small	Medium	Large
Support costs	$20,000	$60,000	$200,000
Data base license fees	$0	$0	$0
Software license fees	$4,000	$12,000	$40,000
Server costs	1 × $5,000	2 × $5,000	4 × $5,000
Network monitoring equipment	1 × $3,000	2 × $3,000	4 × $3,000
Contractor fees	2 × $1,500	4 × $1,750	8 × $2,000
Total cost	$35,000	$95,000	$288,000

deployments affect cost of customer support fees, software license fees, specialised contractor fees, and hardware fees.

Firstly, we address the costs incurred in deploying QoSPlan over an existing accounting system. Additional servers will be required to host the new upgrades to the accounting system. As accounting system records are already stored within a deployed database system, there is no need to upgrade the database system. License fees and support costs will be increased by 20 % as these are upgraded to an existing system. Depending on network size between 1 and 4 network monitoring devices must be purchased for packet trace collection, and specialised contractors will be required to configure and install them. Table 8 shows the cost of an upgrade to QoSPlan for the three network operator types.

Secondly, we address the costs incurred in implementing a traditional direct measurement based approach. The network monitoring system will include the installation of a larger number of network monitoring devices, each monitoring traffic at a single edge router. The deployment will require an extension to the existing database server, as a larger amount of new data will be collected and stored for subsequent analysis; hence database license fees will increase by 20 %. Additional servers will be required to host the network monitoring services and applications, which will themselves incur new license and support fees. Finally, installation of the new system and hardware will require specialised contractors. Table 9 outlines the cost to the ISP of this approach.

Table 9 Cost of direct network monitoring systems deployment

	Small	Medium	Large
Support costs	$20,000	$60,000	$200,000
Data base license fees	$4,000	$12,000	$40,000
Software license fees	$20,000	$60,000	$200,000
Server costs	1 × $5,000	2 × $5,000	4 × $5,000
Network monitoring equipment	5 × $3,000	22 × $3,000	49 × $3,000
Contractor fees	6 × $1,500	22 × $1,750	49 × $2,000
Total cost	$73,000	$246,500	$705,000

Table 10 Comparison of deployment costs

	Small	Medium	Large
QoSPlan deployment	\$35,000	\$95,000	\$288,000
Direct monitoring deployment	\$73,000	\$246,500	\$705,000
Difference	\$54,000	\$193,000	\$565,000
(Relative savings)	48 %	61 %	60 %

Based on our outlined set of assumptions, Table 10 shows a clear difference in the cost of deploying both approaches for network of different sizes, with the QoSPlan deployment incurring significantly less costs, particularly as network size increases. This is a result of the greater level of reuse of existing systems in the QoSPlan deployment and the requirement for installation of significantly more hardware in the direct monitoring system deployment.

7 Conclusions

We have presented QoSPlan; a measurement based framework for preparation of input for QoS-aware IP network planning based on mediation of pre-existing accounting data and analysis of a limited number of representative packet traces collected from the network. QoSPlan will output a matrix of estimated effective bandwidths per traffic class between node pairs, which can subsequently be used for network planning, and indeed other purposes. We presented a through experimental analysis to demonstrate the sensitivity of the framework to various settings such as packet sampling and time aggregation of traffic flows to estimate network traffic demands. We found that by carefully choosing the setting over which traffic flow records are collected and analyzed, the accuracy of QoSPlan can be equivalent to a direct measurement approach, with the loss of some accuracy.

We contend that adoption of QoSPlan has the potential to greatly reduce the cost of network planning of up to 60 % depending on the topology in question, as it allows the service provider replace their costly dedicated device metering architecture. The QoSPlan process can be easily adapted to real network of large deployments. To demonstrate this we have carried out a case study of the application of QoSPlan to the GÉANT network. We demonstrate that given certain QoS targets, an analysis of whether there is sufficient capacity provisioned within the network to meet the Effective Bandwidth requirements of traffic demands on the network. To facilitate a gradual changeover it would be straight-forward to modify the process to utilize a hybrid architecture incorporating data collected from dedicated metering devices and data from mediated accounting data.

Acknowledgments This work has received support from Science Foundation Ireland via grant numbers 03/CE3/I405 (Autonomic Management of Communications Networks and Services) and 08/SRC/I1403 (Federated, Autonomic Management of End-to-End Communications Services) and the Irish Research Council for Science, Engineering and Technology, co-funded by Marie Curie Actions under FP7.

References

1. Davy, A., Botvich, D., Jennings, B.: An efficient process for estimation of network demand for QoS-aware IP networking planning. In: Proceedings of 6th IEEE International Workshop on IP Operations and Management, IPOM, vol. LNCS 4268. Springer, Berlin, pp. 120–131 (2006)
2. Davy, A., Botvich, D., Jennings, B.: On the use of accounting data for QoS-Aware IP network planning. In: Mason L., and Drwiega T. (eds.) Proceedings of 20th International Teletraffic Congress (ITC-20), vol. LNCS 4516. Springer Berlin, pp. 348–360 (2007)
3. Kelly, F.P., Zachary, S., Zeidins, I. (eds): Notes on Effective Bandwidth, vol. 4. Oxford University Press, Oxford (1996)
4. Davy, A., Jennings, B., Botvich, D.: QoSPlan: process for QoS-Aware IP network planning using accounting data and effective bandwidth estimation. In: Proceedings of 2007 IEEE Global Telecommunications Conference (GLOBECOM), pp. 2690–2695 (2007)
5. Awduche, D., Malcolm, J., Agogbua, J., O'Dell, J., McManus, J.: Requirements for traffic engineering over MPLS. IETF Network Working Group RFC 2702, 1999, available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc2702.txt
6. Botvich D., Duffield, N.: Large deviations, the shape of the loss curve, and economies of scale in large multiplexers. Queueing Syst. **20**(3), 293–320 (1995)
7. Simonian, A., Guibert, J.: Large deviations approximation for fluid queues fed by a large number of on/off sources. IEEE J. Sel. Areas Commun. **13**(6), 1017–1027 (1995)
8. Leland, W.E., Taqqu, M.S., Willinger, W., Wilson, D.V.: On the self-similar nature of Ethernet traffic (extended version). In: IEEE/ACM Transactions on Networking, vol. 2, no. 1. IEEE Press, Piscataway, NJ (1994), pp. 1–15
9. Mansfield, G., Roy, T., Shiratori, N.: Self-Similar and fractal nature of Internet traffic data. In Proceedings of the 15th International Conference on Information Networking (ICOIN), pp. 227–231 (2001)
10. Gong, W.-B., Liu, Y., Misra, V., Towsley, D.: Self-similarity and long range dependence on the Internet: a second look at the evidence, origins and implications. Comput. Netw. **48**(3), 377–399 (2005)
11. Gibbens R. (1996) Traffic characterisation and effective bandwidths for broadband network traces. In: Kelly, F.P., Zachary, S., Ziedins, I.B. (eds) Stochastic Networks: Theory and Applications, Royal Statistical Society Lecture Note Series 4. Oxford University Press, pp. 169–179
12. Duffield, N., Lewis, J.T., O'Connell, N., Toomey, F.: Entropy of ATM traffic streams: a tool for estimating QoS parameters. IEEE J. Sel. Areas Commun. **13**(6), 981–990 (1995)
13. Guérin, R., Ahmadi, H., Naghshineh, M.: Equivalent capacity and its application to bandwidth allocation in high-speed networks. IEEE J. Sel. Areas Commun. **9**(7), 968–981 (1991)
14. Liu N.X. and Baras, J.S.: Measurement and simulation based effective bandwidth estimation. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), vol. 4, pp. 2108–2112 (2004)
15. Davy, A., Botvich, D., Jennings, B.: Revenue optimized IPTV admission control using empirical effective bandwidth estimation. IEEE Trans. Broadcast. **54**(3 part 2), 599–611 (2008)
16. Medina, A., Fraleigh, C., Taft, N., Battacharryya, S., Diot, C. A taxonomy of IP traffic matrices. In: SPIE Proceedings Series, pp. 200–213 (2002)
17. Feldmann, A., Greenberg, A., Lund, C., Reingold, N., Rexford, J., True, F.: Deriving traffic demands for operational IP networks: methodology and experience. IEEE/ACM Trans. Netw. **9**(3), 265–279 (2001)
18. Papagiannaki, K., Taft, N., Lakhina, A.: A distributed approach to measure IP traffic matrices. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, (2004), pp. 161–174
19. Cranor, C., Johnson, T., Spataschek, O., Shkapenyuk, V.: Gigascope: a stream database for network applications. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of data, ser. SIGMOD '03. ACM, New York, NY (2003), pp. 647–651. [Online]. doi: 10.1145/872757.872838
20. Zhang, Y., Roughan, M., Willinger, W., Qiu, L.: Spatio-temporal compressive sensing and internet traffic matrices. In: Proceedings of the ACM SIGCOMM 2009 Conference on Data communication, ser. SIGCOMM '09. plus 0.5em minus 0.4em. ACM, New York, NY, (2009), pp. 267–278. [Online]. doi:10.1145/1592568.1592600

21. Sadasivan, G., Brownlee, N., Claise, B., Quittek, J.: Architecture for IP flow information export. IETF IP flow information export WG—internet-draft, 2006, available: 08-10-2008, online at: http://www.ietf.org/internet-drafts/drafts-ietf-ipfix-architecture-12.txt
22. MOME: IST MOME—Cluster of European Projects aimed at Monitoring and Measurement, 2008, available: 08-10-2008, online at: http://www.ist-mome.org/database. [Online]. Available: http://www.ist-mome.org/database
23. Claise, B.: Cisco Systems NetFlow Service Export Version 9. IETF Network Working Group RFC 3954, (2004), available: 08-10-2008, online at: http://www.ietf.org/rfc/rfc3954.txt
24. Farinacci, D., Fuller, V., Meyer, D., Lewis, D.: Locator/id separation protocol (lisp). IETF Network Working Group Internet Draft, February 2012
25. OPNET: Discrete event smulation model library. OPNET Modeler™, 2011. [Online]. Available: http://www.opnet.com/
26. GÉANT: 2012, available: 01-02-2012, online:http://www.geanet.net/
27. Uhlig, S., Quoitin, B., Lepropre, J., Balon, S.: Providing public interdomain traffic matrices to the research community. ACM SIGCOMM Comput. Commun. Rev. **36**(1), 83–86 (2006)
28. CAIDA: CAIDA Anonymized 2012 Internet Traces dataset. 2012, available: 01-02-2012, online at: https://data.caida.org/datasets/passive-2012/
29. Grabani J.-P., Mendel T. (2008) Market Overview: The IT Management Software Market In 2008. Forrester Research, Inc.
30. HEANET, Higher Education Authority Network: available: 08-10-2008, online: http://www.heanet.ie/ (2008)
31. Network Instruments: 10/100/1000 Ethernet Probes, (2008) available: 01-10-08, online:http://www.networkinstruments.com/products/probes/

Author Biographies

Alan Davy received his BSc in Applied Computing and PhD degrees from Waterford Institute of Technology, Ireland in 2002 and 2008 respectively. He received a Marie Curie CoFUND Postdoctoral fellowship in 2010 where he worked at the Universitat Politècnica de Catalunya in Barcelona, Spain for 18 Months. He now works at the Waterford Institute of Technology, as a member of the BioNETS research group of the TSSG. His current areas of interest are wireless network management, traffic measurement and analysis, and bio inspired systems.

Brendan Jennings was awarded BEng in Electronic Engineering and PhD degrees from Dublin City University, Ireland, in 1993 and 2001 respectively. He is a Senior Research Fellow with the Telecommunications Software & System Group (TSSG) at Waterford Institute of Technology, Ireland, currently working in the areas of network and service management. He has published over 90 peer reviewed papers in international journals and conference proceedings and has participated in the work of standards bodies ETSI, FIPA, TM Forum, and ACF.

Dmitri Botvich is currently a Chief Scientist at the Telecommunication Software & Systems Group, Waterford Institute of Technology. He received his Bachelors (Mathematics) degree and PhD (Mathematics) from Moscow State University, Faculty of Mechanics and Mathematics (Russia), in 1980 and 1984, respectively. His research interests include bio-inspired autonomic network management, security, trust management, sensor and ad hoc networking, queuing theory, and mathematical physics.