Statistical Admission Control Using Delay Distribution Measurements

KARTIK GOPALAN
State University of New York at Binghamton
LAN HUANG
IBM Almaden Research Center
GANG PENG and TZI-CKER CHIUEH
Stony Brook University
and
YOW-JIAN LIN
Telcordia Research

Growth of performance sensitive applications, such as voice and multimedia, has led to widespread adoption of resource virtualization by a variety of service providers (xSPs). For instance, Internet Service Providers (ISPs) increasingly differentiate their offerings by means of customized services, such as virtual private networks (VPN) with Quality of Service (QoS) guarantees or QVPNs. Similarly Storage Service Providers (SSPs) use storage area networks (SAN)/network attached storage (NAS) technology to provision virtual disks with QoS guarantees or QVDs. The key challenge faced by these xSPs is to maximize the number of *virtual resource units* they can support by exploiting the statistical multiplexing nature of the customers' input request load.

While a number of measurement-based admission control algorithms utilize statistical multiplexing along the bandwidth dimension, they do not satisfactorily exploit statistical multiplexing along the delay dimension to guarantee distinct per-virtual-unit delay bounds. This article presents Delay Distribution Measurement (DDM) based admission control algorithm, the first measurement-based approach that effectively exploits statistical multiplexing along the delay dimension. In other words, DDM exploits the well-known fact that the actual delay experienced by most service requests (packets or disk I/O requests) for a virtual unit is usually far smaller than its worst-case delay bound requirement because multiple virtual units rarely send request bursts at the same time. Additionally, DDM supports virtual units with distinct probabilistic delay guarantees—virtual units that can tolerate more delay violations can reserve fewer resources than those that tolerate less, even though they require the same delay bound. Comprehensive trace-driven performance evaluation of QVPNs (using Voice over IP traces) and QVDs (using video stream, TPC-C, and Web search I/O traces) shows that, when compared to deterministic admission control, DDM can potentially increase the number of admitted virtual units (and resource utilization) by up to a factor of 3.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; D.4.2 [**Operating Systems**]: Storage Management

General Terms: Algorithms, Measurement, Performance

Authors' addresses: K. Gopalan, Binghamton University; email: kartik@cs.binghamton.edu; L. Huang, IBM Almaden Research Center; email: lanhuang@us.ibm.com; G. Peng, T.-C. Chiueh, Stony Brook University; email: {gpeng,chiueh}@cs.sunysb.edu, Y.-J. Lin, Telcordia Research; email:yjlin@research.telcordia.com.

1. INTRODUCTION

Performance sensitive applications such as Voice over IP (VoIP), video conferencing, media streaming, and online trading, require dedicated network, storage, and computational resources to meet their stringent delay and throughput requirements. A powerful concept being applied to meet this emerging need is the *virtualization* of physical resources into multiple *virtual units* of resources.

As an example, Internet Service Providers (ISP) provision multiple Virtual Private Networks (VPN) with distinct QoS guarantees (or QVPNs) where each QVPN acts as a traffic trunk carrying performance sensitive aggregated traffic. Technologies such as Multiprotocol Label Switched (MPLS) networks can map each QVPN to a long-term Label Switched Path (LSP). For instance, a QVPN could be a long-term Voice over IP (VoIP) trunk that carries aggregate traffic from several voice sessions rather than just one individual voice session. QVPNs are set up and torn down over longer timescales and carry aggregate traffic that is more stable than short-lived individual connections. Similarly, Storage Service Providers (SSP) increasingly use *storage virtualization* technology to create a set of virtual storage devices from a single physical storage resource such as a Storage Area Network (SAN) or a Network Attached Storage (NAS). Each such *virtual disk* (VD) can have distinct QoS guarantees (QVD) such as capacity, request throughput, and latency bound. QVDs serve as backend storage servers for separate enterprise functions such as Web servers, media servers, or database servers. As in the case of QVPNs, QVDs can bundle multiple virtual units for higher aggregated I/O rates.

The key challenge faced by xSPs is to maximize the utilization efficiency of the physical resource infrastructure and still support the stringent QoS requirements of each virtual unit. Maximizing utilization efficiency calls for an effective admission control algorithm that admits as many virtual units as possible, while allocating the least amount of resources needed to satisfy their QoS requirements. A simple approach of *deterministic* admission control allocates all the resources needed to ensure that the QoS guarantees are never violated. Specifically considering delay guarantees, deterministic admission control ensures that the delay in servicing each request (packet or I/O) never exceeds the worst-case delay bound guaranteed for each virtual unit. On the flip side, however, worst-case delays are rarely encountered in practice. Because deterministic admission control errs on the side of being highly conservative, a large proportion of physical resources remain underutilized. Two specific statistical effects can help to improve the resource usage efficiency of these systems.

(1) *Tolerance to delay violations*. Most real-world delay-sensitive applications can tolerate a small fraction of excess delays in request service times [Wang and Zhu 1998]. For instance, VoIP sessions can tolerate up to 10^{-3} fraction of their packets experiencing excess delays or losses without perceptually affecting audio quality. If 99.9% of the packets are observed to experience at most 50% of their expected worst-case delay, a network admission control algorithm can potentially reserve only half of the resources that deterministic admission control would reserve.

(2) *Statistical multiplexing along delay dimension*. Due to statistical multiplexing, typically not all the virtual units can simultaneously experience their peak request arrival rates. For instance, packet bursts from all QVPNs (or I/O bursts from all QVDs) will usually not arrive exactly at the same time at their service queues and would generally be dispersed over time. Consequently, request service delays rarely approach the worst-case delays bounds that would occur only if all virtual units experience their peak request burst simultaneously. To illustrate this multiplexing effect, we aggregated the ON–OFF packet traces for different numbers of recorded VoIP sessions (details in Section 4). Figure 1 shows the complementary cumulative distribution function of the fraction of VoIP sessions in an aggregate that are simultaneously in their ON state. We observe that half the time, less than 12% of the VoIP sessions are in their ON state simultaneously, and its almost never the case that more than 40% of the sessions are simultaneously active. Similar statistical effects

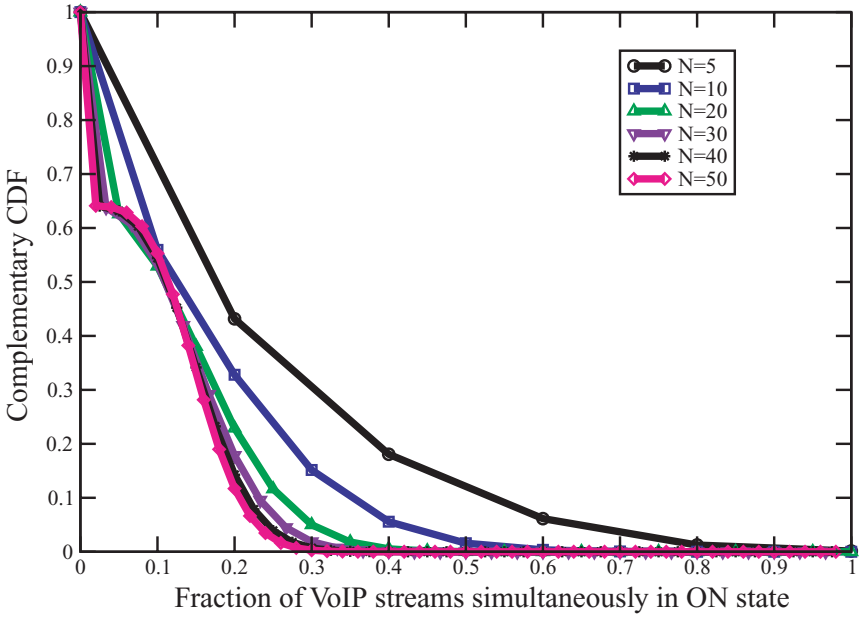

Fig. 1. Complementary CDF of the fraction of VoIP sessions in on state simultaneously as the number of VoIP sessions (N) in aggregate QVPN is varied.

can be expected for other categories of real-time network traffic such as video conferencing and online financial transactions.

This article proposes a practical and efficient measurement-based technique, called *Delay Distribution Measurement* (DDM)-based admission control, that exploits the previous two statistical effects to maximize the number of virtual units admitted with performance guarantees. The QoS parameters that the DDM algorithm supports include delay bound, delay violation probability bound, and the long-term average bandwidth. DDM is the first measurement-based algorithm that simultaneously provides all the following features.

—*Statistical multiplexing along delay dimension*. DDM is the first measurement-based approach which exploits statistical multiplexing along the *delay dimension* to increase resource utilization in comparison to purely deterministic admission control. In contrast, the earlier measurement-based approaches mainly focused on statistical multiplexing along the *bandwidth dimension*, that is, multiplexing due to the fact that virtual units often request rates much below their stated long-term bandwidth requirement.

—*Distinct per-virtual-unit probabilistic delay bounds*. DDM supports virtual units for which a certain percentage of delay bound violations are tolerable. The key difference from prior approaches is DDM's ability to differentiate among virtual units in terms of their tolerance to delay bound violations. Virtual units with higher tolerance to delay bound violations are allocated fewer resources than those with lower tolerance even though they may have the same delay bound requirement.

—*Unified support for probabilistic and deterministic delay bounds.* DDM provides a single admission control framework to support virtual units that may have probabilistic or deterministic delay bounds. Deterministic delay bound requirements simply correspond to zero tolerance to delay violations.

The principal challenge in providing distinct per-virtual-unit probabilistic delay guarantees is to determine the mapping between delay bound, delay violation probability bound, and resource requirements. DDM dynamically measures the service delay of each request, computes the ratio between the actual service delay and the worst-case delay that the request could experience, and derives a delay ratio distribution. This dynamically measured delay ratio distribution is used to derive the bandwidth reservation needed to support a given probabilistic delay bound. Once the DDM algorithm reserves an amount of bandwidth for a virtual unit, a rate-based request scheduler (such as Virtual Clock [Zhang 1991] or WFQ [Parekh and Gallager 1993]) guarantees the assigned bandwidth share.

The DDM algorithm applied to network resource allocation alone was first introduced in our earlier conference article [Gopalan et al. 2004]. In this article, we additionally describe how the concepts of the DDM algorithm are applied to perform efficient storage resource allocation in a multi-dimensional storage virtualization system called *Stonehenge* [Huang et al. 2004]. We also present several additional performance results demonstrating the benefits of DDM for both network and storage resource allocation.

The rest of the article is organized as follows. In Section 2, we first describe the DDM algorithm in the context of network resource allocation for QVPNs. In Section 3, we describe how the same principles of the DDM algorithm are applied in the context of storage resource allocation to support QVDs with distinct probabilistic delay and bandwidth guarantees. Sections 4 and 5 present performance evaluation of the DDM algorithm for network and storage resource allocation, respectively. In Section 6, we discuss the prior work in statistical admission control in the areas of both network and storage resource allocation. Section 7 summarizes the main research contributions and outlines future research directions.

2. STATISTICAL NETWORK RESOURCE ALLOCATION USING DDM

The primary goal of network resource allocation with DDM is to maximize the number of admitted QVPNs with distinct bandwidth, delay, and delay violation probability bounds. In other words, consider a QVPN F_i that carries aggregate real-time traffic with an average bandwidth of ρ_i^{avg} and burst size σ_i. Assume that F_i traverses a link l having total capacity C_l. It is guaranteed at admission control time that each of F_i's packets will be serviced by the packet scheduler at link l within a delay bound $D_{i,l}$ and with a delay violation probability no greater than $P_{i,l}$. For instance, if $D_{i,l} = 10$ms and $P_{i,l} = 10^{-3}$, it means that no more than a fraction 10^{-3} of packets belonging to the QVPN can experience a delay greater than 10ms.

2.1 Worst-Case Delay Bound

We first review the classical results for deterministic delay bounds using rate-based schedulers. We assume that each QVPN's incoming traffic is regulated by a token bucket with bucket depth σ_i and token rate ρ_i^{avg}. The amount of QVPN F_i traffic arriving at the scheduler in any time interval of length τ is bounded by $(\sigma_i + \rho_i^{avg}\tau)$.

The job of a link scheduler is to prioritize the transmission of packets belonging to different QVPNs over a common link. We assume that packets are serviced by *rate-based link schedulers*, such as WFQ [Parekh and Gallager 1993] or Virtual Clock [Zhang 1991]. It can be shown that the worst-case queuing delay $D_{i,l}^{wc}$ experienced at a link l by any packet belonging to a QVPN F_i under the WFQ or Virtual Clock service discipline is given by the following expression.

$$D_{i,l}^{wc} = \frac{\sigma_i}{\rho_{i,l}} + \frac{L_{max}}{\rho_{i,l}} + \frac{L_{max}}{C_l}, \tag{1}$$

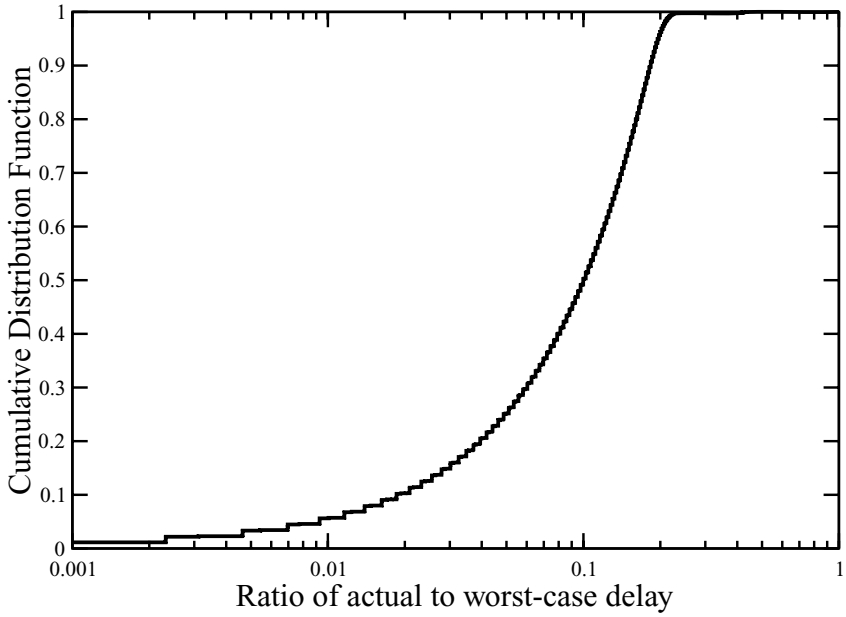

Fig. 2. Example of cumulative distribution function (CDF) of the ratio of actual delay to worst-case delay experienced by packets. X-axis is in log scale to highlight the ratio distribution in the low-ratio range. 39 VoIP QVPNs traverse a 10Mbps link. $\rho_i^{avg} = 256$Kbps. Delay bound=10ms. Delay violation probability $= 10^{-5}$.

where σ_i is F_i's burst size at link l, L_{max} is the maximum packet size, $\rho_{i,l}$ is the reservation for F_i at link l, and C_l is the total capacity of link l. The first component of the delay is fluid fair queuing delay, the second component is the packetization delay, and the third component is scheduler's nonpreemption delay. We are interested in rate-based schedulers since, in their case, the relationship between delay bound and the amount of bandwidth reserved for a QVPN can be explicitly specified. Furthermore, as we will see in Section 2.2, rate-based schedulers enable us to differentiate among QVPNs in terms of their delay violation probability requirements. In contrast, for nonrate-based schedulers, such as Earliest Deadline First (EDF), the resource-delay relationship is difficult to determine, which in turn makes the admission control process more complicated. Hence, even though nonrate-based schedulers can potentially provide higher link utilization, it is difficult to guarantee delay violation probability bound on a per-QVPN basis.

2.2 Delay to Resource Mapping

Probabilistic delay guarantees assist in reducing the bandwidth reservation for each QVPN by exploiting their tolerance to certain level of delay violations. Due to statistical multiplexing, packet bursts from different QVPNs F_i tend to be temporally spread out and rarely occur at the same time. As a result, worst-case delay is rarely experienced by packets traversing a link. Assume that the request for a QVPN F_i specifies its average rate ρ_i^{avg}, burst size σ_i, required delay bound $D_{i,l}$, and delay violation probability $P_{i,l}$ at link l. Each QVPN F_i traversing the link is assigned a bandwidth reservation $\rho_{i,l} \geq \rho_i^{avg}$, which satisfies both the delay requirement $(D_{i,l}, P_{i,l})$ as well as the average rate requirement ρ_i^{avg}. Note that ρ_i^{avg} is the long-term average rate of F_i, whereas the bandwidth reservation $\rho_{i,l}$ is used by the scheduler to determine the runtime preference for F_i's traffic over other QVPNs. In this section, we derive the correlation function that maps F_i's specification $(\rho_i^{avg}, \sigma_i, D_{i,l}, P_{i,l})$ to its bandwidth reservation $\rho_{i,l}$.

2.2.1 *CDF Construction.* Assume that for each packet k, the system tracks the runtime measurement history of the ratio r_k, which is the actual packet delay experienced $D_{i,l}^k$ to the worst-case delay $D_{i,l}^{wc}$, that is, $r_k = D_{i,l}^k / D_{i,l}^{wc}$, where r_k ranges between 0 and 1. We can use these measured samples of ratio r_k to construct a cumulative distribution function (CDF) $Prob(r)$. The distribution $Prob(r)$ gives the probability that the ratio between the actual delay encountered by a packet and its worst-case delay is smaller than or equal to r. Conversely, $Prob^{-1}(p)$ gives the maximum ratio of actual delay to worst-case delay that can be guaranteed with a probability p. Figure 2 shows an example of a CDF constructed in this manner for a specific simulation scenario of 39 VoIP QVPNs. (Simulation details follow in Section 4.)

To construct the CDF in practice, we partition the ratio range from 0 to 1 into a number of subranges, and then, for each subrange, keep updating the count of packets transmitted whose ratio r_k falls within the subrange. The CDF can be constructed by computing the accumulated count of packets from the lowest subrange to each subrange i. The CDF would typically be maintained over a sliding measurement window. The duration of the measurement window partly determines how aggressive the admission control algorithm can be in admitting new QVPNs. The impact of different window sizes on the admission process is evaluated in Section 4.7.

2.2.2 *Resource Mapping.* The CDF curve $Prob(r)$ concisely quantifies the level of statistical multiplexing along the delay dimension. For instance, Figure 2 indicates that most of the packets experience less than 1/4th of their expected worst-case delay. Thus, reserving resources to cover for the worst-case delay is wasteful since it is rarely encountered in practice. In this section, we describe how we can exploit the statistical multiplexing information quantified by $Prob(r)$, in addition to each QVPN's tolerance to delay violations, to reduce the amount of per-QVPN bandwidth reservation.

Given the measured estimate of functions $Prob(r)$ and $Prob^{-1}(p)$, the following expression determines the delay-derived bandwidth reservation $\rho_{i,l}^{delay}$ required to satisfy QVPN F_i's probabilistic delay requirement $(D_{i,l}, P_{i,l})$.

$$D_{i,l} = \left(\frac{\sigma_i + L_{max}}{\rho_{i,l}^{delay}} + \frac{L_{max}}{C_l} \right) \times Prob^{-1}(1 - P_{i,l}). \tag{2}$$

Equation (2) states, that in order to obtain a delay bound of $D_{i,l}$ with a delay violation probability bound of $P_{i,l}$, we need to reserve a minimum bandwidth of $\rho_{i,l}^{delay}$ which can guarantee a worst-case delay of $D_{i,l}^{wc} = D_{i,l} / Prob^{-1}(1 - P_{i,l})$. Conversely, the delay-derived bandwidth requirement $\rho_{i,l}^{delay}$ of a QVPN F_i at link l is

$$\rho_{i,l}^{delay} = \frac{\sigma_i + L_{max}}{\frac{D_{i,l}}{Prob^{-1}(1-P_{i,l})} - \frac{L_{max}}{C_l}}. \tag{3}$$

The actual reservation required to satisfy QVPN F_i's QoS requirement $(\rho_i^{avg}, D_{i,l}, P_{i,l})$ is $\rho_{i,l} = max\{\rho_i^{avg}, \rho_{i,l}^{delay}\}$. In other words, the actual bandwidth reservation for a QVPN is dictated by the tighter of two QoS requirements—one imposed by its average bandwidth requirement ρ_i^{avg}, and the other imposed by its probabilistic delay requirement $(D_{i,l}, P_{i,l})$.

It is worth pointing out once more that this resource mapping function exploits statistical multiplexing along the delay dimension rather than along the bandwidth dimension as in earlier approaches. This is a direct consequence of the fact that DDM measures the distribution of actual to worst-case delay ratio. Specifically, if $\rho_{i,l}^{delay}$ happens to be larger than ρ_i^{avg} for all QVPNs, then the resource allocation will be guided by statistical delay requirements rather than deterministic bandwidth requirements.

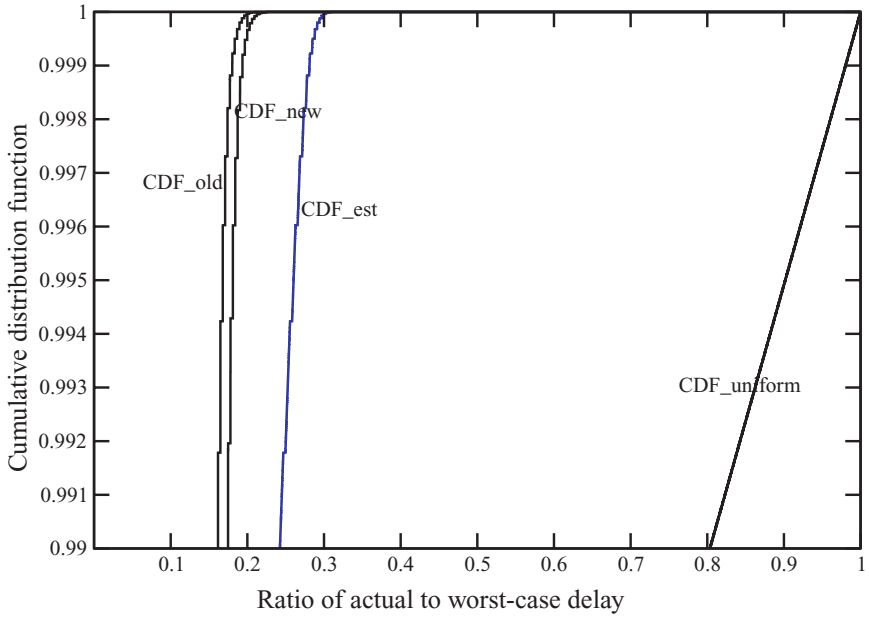

Fig. 3. Example of different CDF curves for one simulation scenario. X-axis is in linear scale to highlight the difference between measured and estimated CDF curves. The Y-axis range shown is from 0.99 to 1.0 which corresponds to the typical tolerance range for delay violations (below 10^{-2}).

2.3 Admission Control Using DDM

In this section, we describe the DDM admission control algorithm for admitting a new QVPN F_N that arrives at a link l on which $N-1$ QVPNs have already been admitted. The principal challenge of admission control lies in estimating the impact of F_N's traffic on the guarantees provided to already admitted QVPNs. If F_N is admitted, it will cause an increase in traffic load carried by the link and consequently larger actual delays experienced by packets from all QVPNs. Specifically, the CDF of actual to worst-case delay ratio will tend to become more conservative by shifting to the right after F_N becomes active. Hence it is important that, even before F_N can be admitted, DDM must estimate and account for the impact of the new QVPN on the delay distribution of existing QVPNs.

The DDM algorithm consists of two phases. The first phase estimates the expected delay distribution assuming QVPN F_N is admitted. The second phase performs the actual admission control using the estimated CDF from the first phase and computes future resource requirements of all QVPNs (including the new one). F_N is admitted only if each QVPN's resource requirement can be satisfied within the available link capacity.

2.3.1 *Significance of CDF Evolution.* If the new QVPN F_N is admitted, the link with a finite capacity C_l has to shoulder the additional traffic load from F_N. As a result, packets, for all QVPNs traversing the link will experience larger delays on average. More specifically, the additional load from F_N could impact the CDF curve shown in Figure 2 by shifting it to the right. In other words, for the same delay violation probability p, if $r_1 = Prob_{old}^{-1}(1-p)$ before admitting F_N and $r_2 = Prob_{new}^{-1}(1-p)$ after admitting F_N, then $r_2 \geq r_1$. Because a larger value of $Prob_{new}^{-1}(1-p)$ translates into-higher bandwidth requirement in Equation (3), CDF_{new} is said to be more conservative than CDF_{old} since CDF_{new} can admit fewer QVPNs than CDF_{old}. Figure 3 provides an example of CDF_{old} and right-shifted CDF_{new}

for one simulation scenario in the Y-axis range from 0.99 to 1.0 (since this range happens to be of most interest).

If we simply use CDF_{old} to derive the bandwidth reservation for F_N, and the actual CDF_{new} turns out to be significantly more conservative than CDF_{old}, F_N may be assigned a much smaller bandwidth than what it actually needs to meet its probabilistic delay requirement. The key research challenge of the DDM algorithm thus lies in how to predict the impact of the new QVPN F_N on the delay distribution of $(N-1)$ existing QVPNs without assuming any apriori traffic model.

The impact of new QVPN F_N on CDF_{old} depends on several factors. In general, tight QoS requirements, such as a small delay requirement $D_{N,l}$, a low tolerance to delay violation $P_{N,l}$, a large average rate ρ_N^{avg}, or a big burst size σ_N, all lead to larger ratio of actual to worst-case delay and a more conservative CDF. Furthermore, the increment from $Prob_{old}^{-1}(1-p)$ to $Prob_{new}^{-1}(1-p)$ could be different for different values of violation probability p. Finally, the magnitude of a new QVPN's relative load contribution to a link's traffic affects the amount of difference between the CDFs before and after the new QVPN is admitted.

2.3.2 *Predicting CDF Evolution.* Given the multitude of factors that influence the evolution of CDF, it is difficult (if not impossible) to exactly predict CDF_{new} using CDF_{old} and QVPN F_N's QoS requirements. The DDM algorithm uses a heuristic approach to approximate CDF_{new}. Let τ be the length of a moving time window over which the delay distribution CDF_{old} of existing $N-1$ QVPNs is measured. Let m be the number of packets generated by $N-1$ QVPNs that traverse the link in duration τ. In a time interval τ, F_N can potentially transmit a maximum of $n = \sigma_N/L_{min} + \rho_N^{avg} * \tau/L_{min}$ number of packets, where L_{min} is the minimum packet size. Assume that these n additional packets experience a uniform distribution of actual to worst-case delay ratio. A uniform distribution is a very conservative estimate of delay distribution (though not the most conservative one) which assumes that packet delays for the new QVPN F_N are expected to be uniformly distributed over the range of ratios from 0 to 1 and that all packets are of size L_{min}. In reality, a large majority of packets experience small packet delays (as shown in Figure 2) and are of size greater than L_{min}.

To characterize CDF_{new}, we first combine the uniform delay ratio distribution for F_N obtained previously with a weight of $\frac{n}{n+m}$ and the delay ratio distribution CDF_{old} with a weight of $\frac{m}{n+m}$ to obtain a distribution called $CDF_{uniform}$, which represents an estimate of the cumulative distribution that would result if F_N were fully loaded and the delay ratio of the packets from F_N were distributed uniformly between 0 and 1. $CDF_{uniform}$ can be constructed using the technique described in Section 2.2, but with the difference that, before computing the accumulated sum for each ratio subrange, we add n/R to the count of ratio samples in each subrange, where R is the number of subranges between 0 and 1. In other words, n delay ratios are assumed to be uniformly distributed over all ratio subranges.

Empirically, $CDF_{uniform}$ is a very conservative estimate of the distribution CDF_{new} because both the uniform delay ratio distribution assumption and the full load assumption are too pessimistic. As a result, CDF_{new} lies somewhere between CDF_{old} and $CDF_{uniform}$ as previously constructed. We further approximate CDF_{new} by constructing CDF_{est}, which in turn is a weighted combination of CDF_{old} and $CDF_{uniform}$. Specifically,

$$Prob_{est}^{-1}(1-p) = \alpha Prob_{uniform}^{-1}(1-p) + (1-\alpha)Prob_{old}^{-1}(1-p). \tag{4}$$

The factor α is the *impact factor* that determines how far the distribution curve CDF_{est} is from $CDF_{uniform}$ and CDF_{old}. For a new QVPN that imposes a relatively large load on the link with respect to an existing load, CDF_{est} should be close to $CDF_{uniform}$ since the latter is more conservative in admitting QVPNs. On the other hand, for a new QVPN that imposes a relatively small load with respect, to an existing load, CDF_{est} should be closer to CDF_{old} since, in this case, the new QVPN has a relatively smaller impact on

CDF_{old}. With this consideration in mind, we define the impact factor as the fraction of new QVPN F_N's load on the total expected load.

$$\alpha = \frac{\rho_{N,l}}{\sum_{i=1}^{N} \rho_{i,l}}. \tag{5}$$

Here $\rho_{i,l}$ is computed using the distribution $CDF_{uniform}$ since it is the only estimate of future delay distribution we have at the time of admitting F_N. Since we are practically interested in only the delay violation probabilities $P_{i,l}$ for existing and new QVPNs, we only need to compute that portion of CDF_{est} which covers these delay violation probabilities of interest; typically the violation probabilities lie in the range 10^{-2} to 10^{-6} which corresponds to a small upper portion of the Y-axis in Figure 2. An example of different CDF curves is illustrated in Figure 3 within the Y-axis range of 0.99 to 1 for one simulation scenario. We see that CDF_{est} is the closest approximation to CDF_{new}, although a bit more conservative. $CDF_{uniform}$ is the most conservative of all.

Note that constructing CDF_{est} involves two levels of weighted combinations, first in constructing $CDF_{uniform}$ from CDF_{old} and a uniform distribution of new QVPN's packets, and second in constructing CDF_{est} from CDF_{old} and $CDF_{uniform}$. The difference is that the $CDF_{uniform}$ provides a first-cut conservative estimate of CDF_{new}, whereas this estimate is further refined by constructing CDF_{est}. In Section 4, we validate that this technique for CDF estimation indeed reliably captures the future delay distribution of admitted QVPNs.

2.3.3 *The Admission Control Algorithm.* With the delay-probability-bandwidth correlation function in place, we now present the DDM admission control algorithm in Figure 1. The algorithm can be invoked either to admit a new QVPN F_N or to periodically recalculate the requirements of already admitted QVPNs. Without loss of generality, the following discussion assumes the first scenario.

Assume that $N-1$ QVPNs are currently being served by the scheduler, and F_N arrives for admission. The algorithm first calculates $CDF_{uniform}$ using the measured delay distribution CDF_{old} and QVPN F_N's average rate requirement ρ_N^{avg}. For each of the N QVPNs (including the new one) the algorithm next

Algorithm 1 The DDM algorithm to determine whether a new QVPN F_N can be admitted such that each QVPN F_i, $1 \le i \le N$, can be guaranteed a delay bound $D_{i,l}$, delay violation probability $P_{i,l}$, and average rate ρ_i^{avg}.

```
Input : (a) (D_{i,l}, P_{i,l}, ρ_i^{avg}, σ_i) for each QVPN F_i, 1 ≤ i ≤ N.
        (b) The measured delay ratio distributions.

Compute CDF_old and CDF_uniform from delay ratio distributions.

for i = 1 to N do
    Compute ρ_{i,l}^{delay} = B_l(D_{i,l}, P_{i,l}, σ_i) using Equations (3) and (4).
    ρ_{i,l} = max{ρ_i^{avg}, ρ_{i,l}^{delay}}
end for

/*Perform admission checks*/
if (Σ_{i=1}^{N} ρ_{i,l} > C_l) then
Reject QVPN F_N and exit.
end if

/*QVPN F_N can be admitted*/
for i = 1 to N do
    Reserve bandwidth ρ_{i,l} for F_i.
end for
```

computes the delay-derived bandwidth requirement $\rho_{i,l}^{delay}$ using Equations (3) and (4). The actual bandwidth requirement $\rho_{i,l}$ is the larger of the delay-derived requirement $\rho_{i,l}^{delay}$ and average requirement ρ_i^{avg}. The new QVPN F_N is admitted only if following condition is satisfied.

$$\sum_{i=1}^{N} \rho_{i,l} \leq C_l. \tag{6}$$

Equation (6) states that the sum of bandwidth requirements of all QVPNs under the estimated delay ratio distribution CDF_{est}, should be smaller than C_l. The QVPN F_N is rejected if this condition cannot be satisfied. If the new QVPN is accepted, the algorithm sets the bandwidth reservation for each QVPN to $\rho_{i,l}$ as computed previously.

The robustness of the DDM algorithm, in essence, depends upon the accuracy of estimating CDF_{est} before admitting a new QVPN F_N. This is because the act of admitting F_N results in altering the reservation $\rho_{i,l}$ of already admitted flows F_1 to F_{N-1}. A CDF_{est} that is too conservative can lead to underutilization of a link's resources, whereas one that is overly optimistic can lead to a potential violation of QoS guarantees for all QVPNs at runtime. The principal challenge in the DDM algorithm lies in accurately estimating CDF_{est} before admitting F_N using an appropriate value of the impact factor α in Equation (4) a value that is neither too optimistic nor too conservative. Experiments in Section 4 show that an impact factor given in Equation (5) that equals the fractional load imposed by the new flow provided a good estimate of CDF_{est}.

The admission control algorithm described provides a unified framework to support QVPNs with both probabilistic as well as deterministic delay requirements. Specifically, QVPNs requiring deterministic delay bounds can simply be treated as requiring a violation probability of zero which, in turn, can be easily factored into the calculation of $\rho_{i,l}$ described in Section 2.2.2.

2.3.4 *Time and Space Complexity.* The step for computing CDF_{old} and $CDF_{uniform}$ has $O(R)$ time complexity, where R is the number of subranges in the delay ratio interval from 0 to 1. The subsequent steps in the algorithm have $O(N)$ time complexity, where N is the number of QVPNs being considered. Thus the complexity of the DDM algorithm is $O(N + R)$. In practice, the first step of computing CDF_{old} and $CDF_{uniform}$ is the more dominant of the two components due to the larger number of subranges R. The algorithm itself is invoked quite infrequently, only when either new QVPN requests arrive for admission at the link or existing QVPN reservations need to be periodically recomputed. The runtime computation overhead of maintaining CDFs is also minimal since we only need a few arithmetic operations to record the ratio for each packet transmitted by the link. In terms of space cost, the only significant additional space required is in the order of $O(R)$ (about 400KB with $R = 100$K) for maintaining CDF_{old}, which represents aggregate delay distribution information for all QVPNs. The values for $CDF_{uniform}$ and CDF_{est} can be derived as and when required during admission control. In particular, DDM requires no additional space for per-QVPN state maintenance when compared to any other algorithm that provides per-QVPN QoS. In our context, QVPNs represent a limited number of traffic aggregates (such as LSPs in MPLS), rather than individual TCP/IP connections, which further reduces the space requirement to within practical bounds.

3. STATISTICAL STORAGE RESOURCE ALLOCATION USING DDM

We next describe how the DDM algorithm has been applied in the context of a multidimensional storage virtualization system called *Stonehenge* [Huang et al. 2004] that allows for the creation of multiple QoS-guaranteed virtual disks (QVDs) over a common physical storage infrastructure. Stonehenge effectively isolates the logical storage servers as if they are separate physical storage devices, each having the

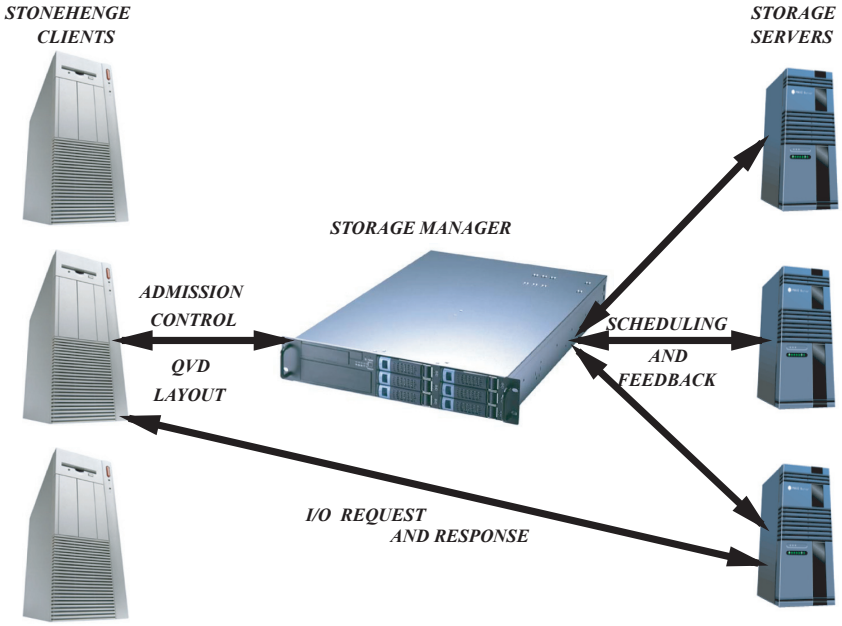

Fig. 4. Stonehenge clients communicate with a centralized management server and a set of storage servers that are connected through a gigabit network.

standard attributes associated with any physical disk such as bandwidth, access latency, capacity, and availability. As a result, QVDs in Stonehenge are as tangible as physical disks but much more flexible and manageable.

Each QVD V_i can be specified in terms of (1) bandwidth ρ_i^{avg} or the number of I/O requests per second $IOPS_i^{avg}$, (2) worst-case delay bound requirement D_i per I/O request, (3) delay violation probability bound P_i, and (4) capacity C_i of the QVD. Given a QVD specification $< \rho_i^{avg}, D_i, P_i, C_i >$, for rate-based QoS-aware disk request schedulers, a correlation function $F(.)$ maps the bandwidth reservation ρ_i^{delay} required to achieve a worst-case delay bound D_i. Given $\rho_i^{delay} = F(D_i)$, one can then further reduce each QVD specification to $< \max(\rho_i^{avg}, \rho_i^{delay}), \infty, P_i, C_i >$.

Figure 4 shows the overall architecture of Stonehenge. Stonehenge is a cluster-based iSCSI storage system that consists of a central management server and a set of storage server nodes connected via a gigabit ethernet network. The central manager server performs admission control and allocates physical disk resources to satisfy each QVD's QoS requirements. At runtime, the management server uses a Virtual Clock scheduler to determine the order in which incoming requests from different QVDs are processed such that each QVD's QoS requirement is satisfied. At the individual storage server nodes, another efficiency and deadline aware Virtual Clock-based disk scheduler is used to decide the actual order in which I/O requests are serviced by physical disks.

In this section, we focus specifically on how Stonehenge applies DDM to convert the latency bound requirement D_i and violation probability requirement P_i to a bandwidth requirement ρ_i^{delay}. While the basic principles behind DDM admission control remain the same for both disk and network resource allocation, important differences arise due to the physical nature of the resources. In the rest of this section, we focus on how the DDM algorithm for admitting QVDs differs in terms of the delay-to-resource correlation function and the manner in which it exploits the runtime load information. Other

major components of Stonehenge, such as a two-level disk scheduling architecture, the disk service time prediction mechanism, and an efficiency conscious real-time disk scheduler are described in Huang et al. [2004] and Gopalan and Chiueh [2001].

3.1 Delay to Resource Mapping

Stonehenge uses a variant of a Virtual Clock scheduler to compute the finish time for each I/O request. Equation (1), which provides the delay bound D_i for a bandwidth reservation ρ_i^{delay}, is appropriate for network resource allocation and scheduling. To convert this network latency bound expression to one appropriate for disk latency bound expression, we need to account for the disk service overhead associated with each request. The resulting delay bound expression as applied to QVDs becomes:

$$D_i \leq (\delta_i + L_{max} + overhead \times C)/\rho_i^{delay} + (L_{max} + avg\_overhead \times C)/C \tag{7}$$
$$where\ \delta_i = max\_pending\_reqs \times avg\_req\_size$$
$$and\ overhead = avg\_overhead \times (max\_pending\_reqs + 1),$$

where, C is the total bandwidth of the underlying system, ρ_i^{delay} is minimum bandwidth reservation required to guarantee a delay bound of D_i, $max\_pending\_reqs$ is the maximum number of requests the queue can hold for a given request size, and $avg\_overhead$ is the average disk access overhead time measured and computed at runtime. Compared with the original delay bound equation, we expand the request size by $(overhead \times C)$ bytes to account for the access overhead for each request. By multiplying the measured average disk access with C, we translate it to the number of bytes that could be transferred during the overhead time.

Seek delay and rotational latency play an increasingly significant part in disk service time. Consequently, disk request size itself becomes relatively unimportant, especially when most requests are small. Therefore, we can further simplify the expression for latency bound as follows:

$$D_i \leq (max\_pending\_reqs + 1)/IOPS_i^{delay} + 1/IOPS_{max}, \tag{8}$$

where, $IOPS_i^{delay}$ is QVD V_i's request throughput (in number of I/O operations per second) required to guarantee a delay bound of D_i. Similarly, $IOPS_{max}$ is the maximum throughput the physical storage system can support. In cases where the assumption about disk request size is invalid, one can always use Equation (7).

3.2 Exploiting Load Information in Admission Control

As with network resource allocation, the DDM algorithm in Stonehenge exploits statistical multiplexing along the delay dimension to increase the total number of QVDs that can be admitted into a physical storage system. Equation (8) converts a delay bound to its equivalent throughput requirement based on the worst-case delay bound associated with the Virtual Clock scheduler. In practice, this proves to be too conservative because not every disk request experiences the worst-case delay. Therefore, Stonehenge also measures the CDF $Prob(r)$, that is, the cumulative probability distribution of the ratio between the actual delay experienced by a request and the worst-case delay of the QVD with which the request is associated. $Prob(r)$ depends on the number of QVDs in the system because the delay a request actually experiences depends on the actual load in the system which is correlated with the number of QVDs. With $Prob(r)$, the delay bound expression used to decide whether to admit the Nth QVD becomes:

$$D_N \leq ((max\_pending\_reqs + 1)/IOPS_N^{delay} + 1/IOPS_{max}) * (Prob^{-1}(1 - P_N) + s), \tag{9}$$

where $Prob^{-1}(.)$ is the inverse function of $Prob(r)$, P_N is the probability bound that the Nth QVD's delay bound could be violated, and s is an adjustment factor that accounts for the impact of the new QVD on

the delay behavior of existing QVDs. When the system is lightly loaded or N is small, $Prob^{-1}(1-P)$, with P equal to 0.05, for example, can be as low as 10%, which means 95% of the requests experience a delay that is smaller than 10% of the worst-case delay. In contrast, a deterministic admission control algorithm will assume 100% of the requests experience the full worst-case delay. For a given P, $Prob^{-1}(1-P)$ grows closer to 1 with increasing N. The value of s is largely workload-dependent and is 0.2 in Stonehenge. However, if the system is stable enough, the measurement-based feedback is able to detect a relatively stable s value. In this case, Equation (9) can be used. Otherwise, Equation (8) should be used if the workload is highly unpredictable.

The DDM admission control algorithm for QVDs is similar in operation to the algorithm for QVPNs described in Section 2.3.3. Assume that QVD V_N with requirement $(IOPS_N^{avg}, C_N, D_N, P_N)$ arrives for admission where $(N-1)$ QVDs have already been admitted. DDM first calculates $IOPS_i = \max(IOPS_i^{avg}, IOPS_i^{delay})$, $1 \leq i \leq N$, where $IOPS_i^{delay}$ is calculated using Equation (9). The QVD V_N is admitted only if $\sum_{i=1}^{N} IOPS_i \leq C$.

4. PERFORMANCE OF DDM IN NETWORK RESOURCE ALLOCATION

In this section, we study the performance of the DDM algorithm for admitting QVPNs in comparison to deterministic admission control. We use the deterministic approach as a baseline instead of one of the earlier approaches for the following reasons. First, earlier measurement-based approaches mainly address multiplexing along the bandwidth dimension, that is, multiplexing due to the fact that QVPNs typically transmit at rates much below their stated long-term bandwidth requirement. In contrast, DDM exploits multiplexing along the orthogonal delay dimension which occurs even when individual QVPNs transmit at their stated bandwidth, that is, multiplexing due to the fact that different QVPNs transmit their traffic bursts at different times. Second, to the best of our knowledge, earlier analytical approaches that address probabilistic delay guarantees either assume a fluid traffic model (as opposed to a packetized model) or do not support distinct per-QVPN probabilistic delay bounds, but rather provide shared guarantees such as by multiplexing QVPN traffic in a shared buffer. Thus the problem addressed by DDM is fundamentally different from earlier approaches and leaves deterministic admission control as the baseline for comparison.

The real traffic traces used in our simulations are principally composed of VoIP sources. However, a note regarding applicability of DDM to heterogeneous real-time traffic is in order. Unlike voice, video conferencing applications have relatively higher and more variable data rates (due to quantization via motion vectors and prediction algorithms), though with similar latency requirements. Online trading applications, on the other hand, have much lower data rates with tighter latency requirements. In the presence of different categories of real-time traffic, we still expect significant potential gains in link utilization with varying degrees of statistical multiplexing. However, the DDM algorithm is equally applicable to mixes of all categories of real-time traffic and nothing in the algorithm precludes any specific traffic category.

4.1 Evaluation Setup

Using the *ns-2* simulator, we configured a single link at 10Mbps, and packets arriving at the link were served by a WFQ scheduler. Traffic for each QVPN was generated using aggregated traffic traces of recorded VoIP conversations used in Jiang and Schulzrinne [1996] in which spurt-gap distributions were obtained using a G.729 voice activity detector. In other words, packet sizes and interpacket arrival durations within each QVPN followed the exact pattern as in real traffic traces. Each VoIP stream had an average data rate of around 13Kbps, peak data rate of 34Kbps, and packet size of $L_{max} = 128$ bytes.

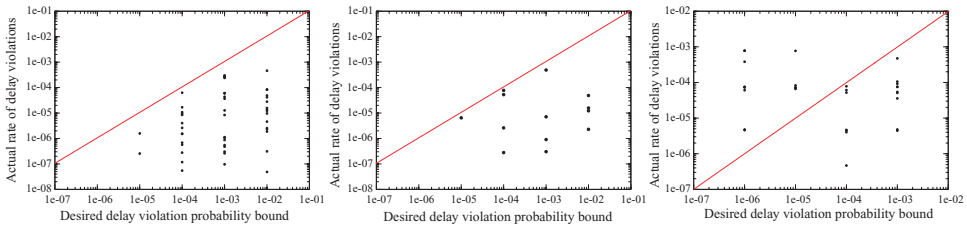

Fig. 5. The DDM algorithm satisfies distinct per-QVPN delay violation guarantees when (a) all other requirements are the same and (b) constituent QVPNs have dissimilar delay bound, data rate, and burstiness requirements. Plot (c) shows that deterministic admission control, with a pure oversubscription of link capacity by a factor of 2, cannot satisfy distinct per-QVPN delay violation guarantees. All plots include data points from 5 simulation runs with different random seeds.

We temporally interleaved the 20 VoIP streams to generate aggregate traffic trace for each QVPN with an aggregate data rate of $\rho_i^{avg} = 256$Kbps.

Each aggregated VoIP trace was 8073 seconds long. Every QVPN in our simulations sent traffic for the entire lifetime of the simulation with the aggregate traffic trace repeated over its lifetime. Traffic from each admitted QVPN passed a token bucket with bucket depth of 1280 bytes (10 packets) and token rate of 256Kbps. Each new QVPN required a guarantee on a delay bound and a delay violation probability. The admission control algorithm decided whether to admit or reject the QVPN and how much bandwidth to reserve according to the algorithm in Figure 1. Each QVPN was generated with a periodic interarrival time of 10, 000 seconds. The reason we selected periodic instead of exponential interarrival times (as in other works) is that our QVPNs are long-lived and are expected to arrive fairly infrequently so that the measured CDF can stabilize before being used to admit another QVPN. Hence the request arrival pattern does not significantly impact the admission control decisions. The CDF was measured over a time interval of 10, 000 seconds between QVPN arrivals. Each simulation run lasted for 1000, 000 seconds.

For simulations, we recorded the ratio of actual to worst-case delay of every packet traversing the link within the current CDF window (although in a realistic scenario, an intelligent sampling mechanism would be more desirable). The observed ratios are accumulated into a histogram. The actual CDF is computed from the histogram only when making admission decisions or recalculating existing reservations.

4.2 Per-QVPN Probability Bounds

We start by validating that the DDM algorithm can indeed provide distinct guarantees on heterogeneous delay violation probabilities for a mix of different traffic types. In the first experiment, we consider a traffic mix in which all QVPNs request the same delay bound of 20ms, the same average rate of 256Kbps, and the same burst size of 10 packets, but require different guarantees on delay violation probability since the requirement is uniformly distributed among the four values 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}. Figure 5(a) plots the actual fraction of packets exceeding their delay bound of 10ms against the desired violation probability for each QVPN that experiences any excess delay. The figure includes data points from 5 simulation runs with different random seeds, and each data point represents the rate of delay violation experienced by one QVPN. Figure 5(b) plots the same data when the constituent QVPNs have heterogeneous delay bounds (10ms–30ms), data rates (256Kbps–2Mbps), and burst-sizes (10–40 packets), in addition to heterogeneous violation probability requirements (10^{-2}–10^{-5}). The line through the graph marks the limit above which the actual rate of delay violations would exceed the desired delay violation probability. The fact that all data points are below the line indicates that the actual delay violation rate is smaller than the maximum permissible for each QVPN. Furthermore the

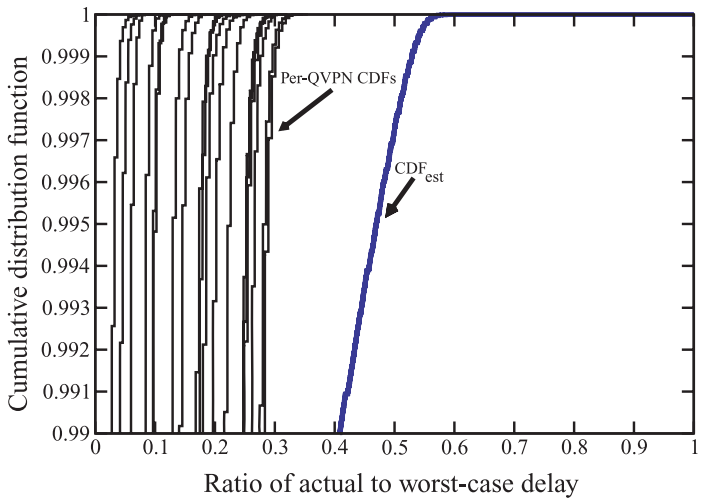

Fig. 6. The predicted CDF_{est} (the rightmost curve) provides a reliable bound on future delay ratio distribution for each admitted QVPN (all other curves). The Y-axis range shown (from 0.99 to 1.0) corresponds to the tolerance range below 10^{-2}.

figure shows that QVPNs that have a higher tolerance to delay violations are more likely to experience a higher rate of violation than QVPNs with lower tolerance. The DDM algorithm is able to distinguish among QVPNs in terms of delay violation rates because it assigns service bandwidth $\rho_{i,l}$ to QVPNs in the inverse proportion to their tolerance to delay violations. This translates to higher dynamic preference for packets belonging to QVPNs with low delay tolerance and vice versa.

In the next experiment, we show that pure oversubscription of link capacity cannot provide distinct guarantees on heterogeneous delay violation probabilities. We use the same parameters as in Figure 5(a) except that, instead of using the DDM algorithm, we use deterministic admission control and oversubscribe the link capacity by a factor of 2.0 in order to admit the same number of QVPNs as the DDM algorithm (i.e., 35 QVPNs) with no oversubscription. Figure 5(c) shows that regardless of the desired delay violation bounds, all QVPNs experience similar rates of actual delay violations. In fact, QVPNs with low tolerance (10^{-5}) to delay violations can experience an order of magnitude higher delay violations than their actual tolerance. This is because pure oversubscription does not correlate to delay violation bound requirements for a QVPN with its bandwidth reservation. We need more than just bandwidth oversubscription, specifically, a delay-probability-bandwidth correlation function. such as in Equation (2), to guarantee distinct per-QVPN probabilistic guarantees.

4.3 Validating the CDF Estimation Technique

Next we validate that the technique for predicting the future delay ratio distribution CDF_{est} in Section 2.3 indeed reliably bounds the delay ratio distribution of admitted QVPNs. Validating the CDF estimation technique is important in establishing that the DDM algorithm does not underestimate the resource requirements for individual QVPNs, resulting in excess delay violations in the long-term. Figure 6 shows a representative simulation scenario in which 19 constituent QVPNs are admitted with heterogeneous delay bound, data rate, and burstiness requirements. The rightmost curve marked CDF_{est} shows the delay ratio distribution estimated by DDM before admitting the 19th QVPN, where the curves on the left represent the stable per-QVPN distributions at the end of the simulation lifetime. The figure demonstrates the fact that the CDF_{est} distribution used at admission control time still remains more conservative than individual QVPN distributions in the long-term. Thus CDF estimation

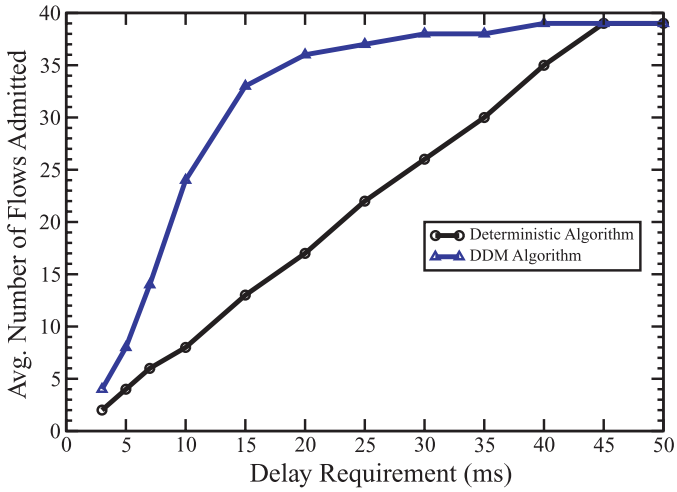

Fig. 7. Number of admitted QVPN vs. delay bound. Delay violation probability = 10^{-5}, burst size = 10pkts, link capacity = 10Mbps.

technique can effectively reduce each QVPN's resource requirement to suit their individual tolerance to delay violations without risking, underestimation of true requirements.

4.4 Delay Bound Variation

Next we compare the performance of the DDM algorithm against deterministic admission control as the delay bound requirement varies. With the DDM the algorithm, the delay violation probability for each QVPN is 10^{-5}, where deterministic admission control considers a zero delay violation probability. Figure 7 plots the number of QVPNs admitted as the delay bound requirement is varied from 3 to 50ms. The maximum number of QVPNs that can be admitted on the 10Mbps link is limited to 39 QVPNs due to the average rate requirement of 256Kbps for each QVPN. Figure 7 shows that, for small delay bound requirements, the DDM algorithm admits around 3.0 times more QVPNs than deterministic admission control when the delay violation probability as small as 10^{-5} is allowed. As the delay bound requirement becomes less stringent, the DDM algorithm still admits more QVPNs and achieves better link utilization than the deterministic algorithm but with smaller improvements. Beyond a 45ms-delay requirement, both algorithms are limited to admitting 39 QVPNs due to the average rate the requirement of 256Kbps for each QVPN. The gain for the DDM algorithm comes from the fact that the large majority of packets experience just 1% to 3% of the worst-case delay dictated by their reserved bandwidth. This statistic gets reflected in the CDF which, in turn, helps to reduce the resource requirement for each QVPN.

4.5 Burst Size Variation

Figure 8 compares the DDM algorithm against deterministic admission control as the burst size σ_i for each QVPN is increased from 1 to 100 packets. Larger burst sizes have the effect of increasing the average time a packet spends waiting in queue to be serviced by the link scheduler. Up to burst sizes of 40 packets, the DDM algorithm admits a significantly larger number of QVPNs than the deterministic algorithm. This is because the deterministic algorithm operates on the worst-case scenario that bursts from all QVPNs arrive at the link simultaneously. On the other hand, DDM successfully exploits the statistical multiplexing effect, that is, bursts from different QVPNs are temporally dispersed and rarely

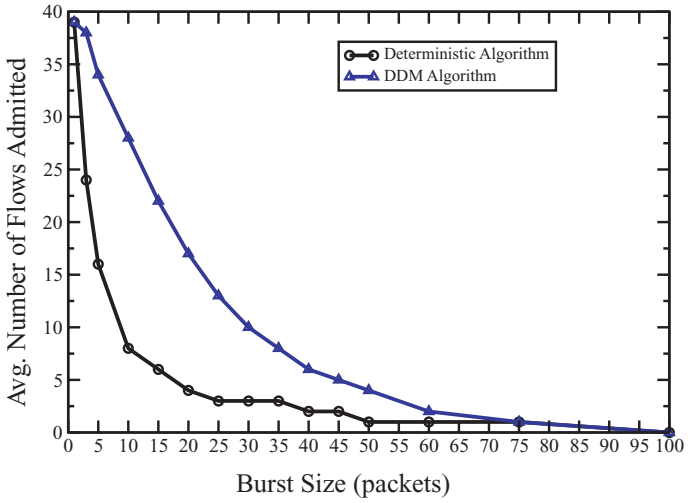

Fig. 8. Number of admitted QVPNs vs. burst size. Delay bound = 10ms, violation probability = 10^{-5}, link capacity = 10Mbps.

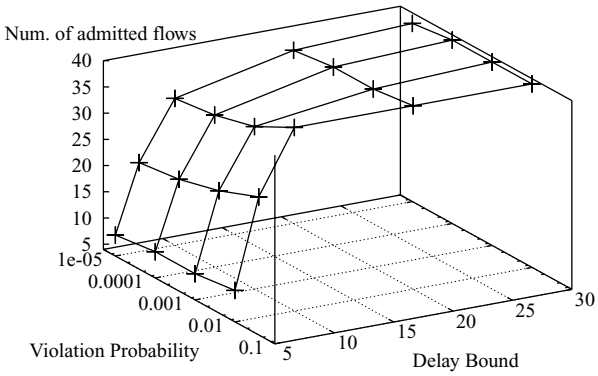

Fig. 9. Admission region for various combinations of delay and delay violation probability. Link capacity = 10Mbps, burst size = 10pkts, average rate = 256Kbps.

occur at the same time. With larger burst sizes, the delay-derived bandwidth requirement increases due to the diminishing impact of statistical multiplexing.

4.6 Admission Region

Figure 9 shows the admission region for various combinations of delay and delay violation probability. As the delay bound and delay violation probability requirements become less stringent, the number of admitted QVPNs increases. Note that, even with a low violation probability of 10^{-5} at 10ms delay, the DDM algorithm can admit up to 24 QVPNs, which is 3 times more than in the deterministic case of 8 QVPNs. Thus even a small tolerance to delay violation can produce large gains in resource utilization efficiency.

4.7 Effect of the CDF Measurement Window

Another factor influencing the performance of the DDM algorithm is the CDF measurement window. Figure 10 shows that a large measurement window leads to a more conservative admission process,

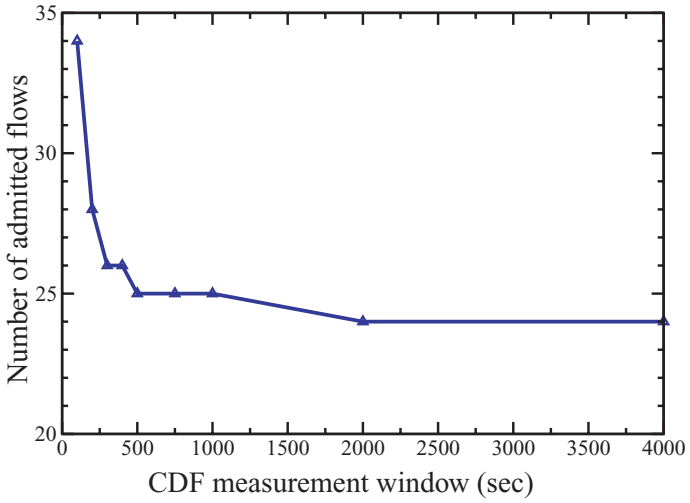

Fig. 10. Number of admitted QVPNs with different CDF measurement windows, Admission control becomes more conservative with larger measurement windows, delay bound = 10ms, violation probability = 10^{-5}, burst size =10 pkts.

that is, a large measurement window admits fewer QVPN requests than a small window over the same interval of time. The reason for this behavior can be traced back to Figure 1. Typically bursts from different QVPNs tend to be temporally spread out and multiple QVPNs rarely burst simultaneously. However, such events do occur and small window sizes are more likely to miss out on such rare simultaneous traffic bursts, whereas large window sizes are more likely to capture these. Consequently, larger measurement windows produce more representative CDF curves than small windows. Admission decisions based on small measurement windows could thus be overoptimistic, leading to a greater number of QVPNs being admitted quickly. With large window sizes, the DDM algorithm is slower in reacting to changes in traffic patterns and thus admits fewer QVPNs as traffic load increases. While a very small window size can result in overoptimistic admissions, an extremely large window size could also lead to inaccurate admission decisions since it might include history that could be too old for consideration.

Hence, one needs to strike a correct balance in selecting a measurement window size that yields optimal performance. A possible recommendation for selecting the CDF measurement window could be the duration between successive QVPN arrivals coupled with lower bound on the measurement window size. Since the arrival of successive QVPNs is expected to be over sufficiently long timescales, the traffic characteristics between successive arrivals can be expected to be largely stable and indicative of true load imposed by currently active QVPNs.

4.8 Statistical Multiplexing Gain from Underutilization

Finally, we vary the number of streams per QVPN to determine the extent of gain we obtain by underutilizing the aggregate QVPN's reserved capacity. At full capacity, each aggregate QVPN can carry 20 VoIP streams. Figure 11 shows that the number of admitted QVPNs decreases from 35 to 24 as the level of aggregation in each QVPN increases from 2 to 20 VoIP streams. Thus, the DDM algorithm can successfully exploit additional statistical multiplexing due to a smaller level of aggregation in each QVPN. In this case, the maximum gain is limited by the average rate requirement of 256Kbps for each QVPN and link capacity of 10Mbps. This is because the DDM algorithm exploits the statistical multiplexing

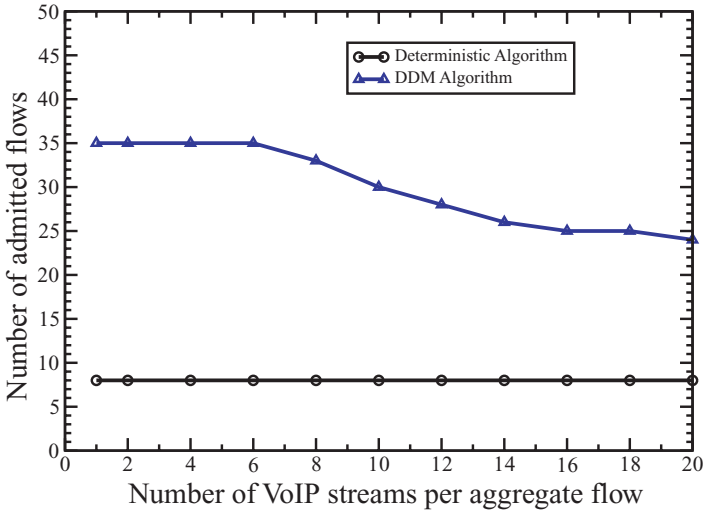

Fig. 11. Number of admitted QVPNs with variation in number of VoIP streams per aggregate QVPN, Delay bound = 10ms, violation probability = 10^{-5}, burst size = 10 pkts.

Table I. Specifications of the Three QVDs in Stonehenge

QVD	Trace	Capacity (GB)	Throughput (IOPS)	Latency	I/O Rate Scaling
0	TPC-C	9	N/A	N/A	N/A
1	Financial	15	128	120 msec	1.25
2	Web Search	96	675	N/A	2

The performance requirements of QVD 0 (best-effort) are not guaranteed. The delay bound and throughput requirements of QVD 1 are guaranteed, and the throughput of QVD 2 is guaranteed. The I/O Rate Speed-up column shows the speed-up of the corresponding trace when it is run. This speed-up ensures that the system is fully loaded.

effect only along the delay dimension but not along the bandwidth dimension. Multiplexing gains could be higher if the latter dimension could also be accounted for in the DDM algorithm.

5. PERFORMANCE OF DDM IN STORAGE RESOURCE ALLOCATION

In this section, we evaluate the performance of the DDM algorithm in admitting QVDs in the Stonehenge system. The performance evaluation study is carried out on the following Stonehenge testbed. The storage server nodes are Pentium III 1GHz machines with 512MB of memory, 64-bit 66MHz PCI bus, two Promise 66MHz dual-channel IDE controllers, an Intel Gigabit network card (Intel Pro-1000 XT), and a RAID0 array of four IDE drives (IBM DTLA-307075) connected to the IDE controllers. Each disk array has a total capacity of 300GB, and uses a 64KB stripe unit. The client and central manager machines have the same hardware configuration except the disk array. The operating system is RedHat Linux with kernel version 2.4.18. The read-ahead cache on the disk is turned on while the write-back cache is turned off to ensure data integrity.

Test programs running on the clients read requests from trace files and send them to the storage servers through iSCSI protocol. Four traces are used in this study: TPC-C trace is a transaction processing trace with 14 warehouses, video stream trace is a trace that sequentially accesses data with a 64KB request size, Web search trace is a disk access trace collected from a Web search engine, and Financial trace is a trace collected from enterprise financial applications. Table I shows the specifications of the

Table II. A Comparison of the Maximum Number of QVDs that Deterministic or Measurement-Based Admission Controller Can Accept with the Same System Resource

	QVD Type	(1-P)	Deterministic	DDM	Oracle
Run 1	Financial	0.95	7	20	22
Run 2	Mixed	0.95	7	14	14
Run 3	Mixed	0.85	7	17	17

In the case of mixed QVD type, 50% of the QVDs are running financial trace and 50% are running Web search trace.

Table III. Resource Reservation of the DDM and Deterministic Admission Control Algorithms When Admitting a Sequence of QVDs with Mixed Types

Number of QVDs	7	8	9	10	11	12	13	14	15
$Prob^{-1}(0.95)$	11%	14%	15%	19%	24%	30%	37%	49%	—
DDM Resource Reservation	N/A	34%	38%	43%	47%	51%	55%	67%	95%
Deterministic Resource Reservation	90%	—	—	—	—	—	—	—	—

The table also shows the evolution of the parameter $Prob^{-1}(r)$ used in Equation 9.

three QVDs used in this study. Unless otherwise stated, each of these QVDs is served by three identical storage server nodes and the system is fully loaded in all the experiments.

To demonstrate the efficiency of the measurement-based admission control (DDM) algorithm, we run three experiments each using a distinct sequence of QVD requests as input. We show the maximum number of QVDs accepted in Table II under three different admission control schemes: deterministic, DDM, and the Oracle scheme. The Oracle scheme assumes no limit of system resources and keeps admitting QVD requests until some admitted QVDs' QoS guarantees are violated. The number of QVDs admitted by the Oracle scheme represents the upper bound of the number of QVDs a system can sustain while satisfying all QVDs' QoS requirements. The table shows that the DDM can double or even triple the maximum number of QVDs that can be admitted by the deterministic approach. In most cases, DDM admits almost as many QVDs as the Oracle scheme can. Due to the conservative value of s we used in run 1, DDM performs slightly worse than Oracle. Also, fewer QVDs are admitted in run 2 compared to run 3 because the QVDs in run 2 have more stringent delay violation probability.

To understand why DDM can admit more QVDs than the deterministic scheme, we compare the resource reservation that the DDM and deterministic approach actually make as the number of QVD requests increases in Table III. It shows that one can significantly reduce the bandwidth requirement for a given delay bound when using Equation (9), and thus increase the number of QVDs to be admitted. For example, when there are seven QVDs, the deterministic admission control already reserves close to 90% of the disk bandwidth while DDM reserves less than 40% of the resources. Despite this advantage, as $Prob(r)$, and thus $Prob^{-1}(r)$, already accurately capture the runtime load and dynamic disk access pattern, DDM's decision to admit more QVDs rarely leads to violation of the QoS guarantees of existing QVDs. Table III also shows that $Prob^{-1}(0.95)$ grows faster than linearly with the increasing if N. Adding one more QVD usually increases $Prob^{-1}(1 - P)$ by 0.2, which we choose as a conservative factor s in Equation (9).

Table IV shows the effect of QoS guarantee probability on the number of QVDs that can be admitted. In this test, the QVDs are latency-bound and are either running financial trace or Web search trace. However, the capacity, latency, and throughput requirements have been scaled down to allow more QVDs to be accepted in Stonehenge. As expected, the number of QVDs admitted increases with the less stringent violation probability. The improvement of DDM over the deterministic approach increases from 1.4 fold when $(1 - P) = 0.95$ to 1.9 fold when $(1 - P) = 0.80$.

Table IV. The Impact of Delay Violation Probability P on the Admission Efficiency of DDM

Probability $(1 - P)$	0.95	0.90	0.85	0.80
DDM	17	18	19	20
Deterministic	7	7	7	7
Oracle	17	20	21	22

With a more relaxed requirement on QoS guarantee, DDM can accept more QVDs with a fixed amount of resources. When the (1-P) reaches 0.70, the QVDs transform from latency-bound QVD to throughput-bound QVD. At this point, the number of QVDs accepted depends on the throughput requirements and available resources.

6. RELATED WORK

The principal features that distinguish DDM from earlier works are (1) its ability to exploit statistical multiplexing along the delay dimension, as opposed to the bandwidth dimension, and (2) its ability to provide a distinct probabilistic delay guarantee to each virtual unit (QVPN or QVD) in contrast to the shared guarantees in some of the earlier approaches. The literature on exploiting statistical multiplexing is extensive, especially in the context of network resource allocation. We discuss the ones most relevant to this work, first in the area of network resource allocation followed by storage resource allocation.

6.1 Statistical Network Resource Allocation

Knightly and Shroff [1999] provide an excellent overview of admission control approaches for link-level statistical QoS. Kurose [1992] derived probabilistic bounds on delay and buffer occupancy of QVPNs using the concept of stochastic ordering for network nodes that use FIFO scheduling. Unlike FIFO schedulers that inherently cannot differentiate between performance requirements of different QVPNs, we are interested in real-time traffic schedulers that can provide per-QVPN delay and bandwidth guarantees. Reisslein et al. [2002] have derived statistical delay bounds for traffic in a single link and network settings using a fluid traffic model. Their work approximates the loss probability at a link using independent Bernoulli random variables. All QVPNs share a common buffer with traffic loss assumed to be split among QVPNs in proportion to their input rates. In contrast, we assume a packet-based model, an independent buffer space for each QVPN, and permit explicit specification of a delay violation probability bound for each QVPN. Elwalid and Mitra [1999] have proposed a scheme to provide statistical QoS guarantees in the GPS service discipline for two guaranteed traffic classes and one best-effort class. Again a fluid traffic model was considered. Boudec and Vojnovic [2002] consider stochastic delay guarantees in expedited forwarding (EF) networks with aggregate scheduling. Their work operates under the Diffserv framework in which EF traffic is marked at the network ingress. Each forwarding node in the network interior is abstracted by a service curve and provides a common stochastic rate and latency guarantee to all transiting EF traffic. Several analytical approaches [Kesidis and Konstantopoulos 2000; C-S. Chang et al. 2001; Cruz 1991; Guillemin et al. 2002] have also considered the performance of multiplexing with a shared buffer for independent regulated inputs. In contrast, we consider distinct per-QVPN probabilistic delay bounds with independent buffers for independent regulated inputs. Schemes for providing probabilistic QoS in networks using Earliest Deadline First (EDF) scheduling were proposed in Andrews [2000], Sivaraman and Chiussi [2000], and Boorstyn et al. [2000]. Unlike the rate-based schedulers considered in our work, EDF decouples rate and delay guarantees at the expense of admission control complexity. Additionally, it is difficult to guarantee distinct per-QVPN delay violation probabilities with EDF due to strong interactions among QVPNs sharing a link. In contrast, rate-based schedulers, such as the one we use, provide explicit

performance isolation among QVPNs and are especially suited to guarantee QVPN-specific delay violation probabilities.

Several existing measurement-based admission control algorithms (MBAC) address QoS requirements along the dimensions of the bandwidth or aggregate loss rate. The notion of Effective Bandwidth [Kelly 1996] is an important concept in MBAC algorithms that provides a measure of bandwidth resource usage by flows relative to their peak and mean usage. Breslau et al. [2000] performed a comparative study of several MBAC algorithms [Qiu and Knightly 2001; Jamin et al. 1997; Floyd 1996; Gibbens and Kelly 1997; Crosby et al. 1997] under FIFO service discipline and concluded that none of them accurately achieve loss targets. Qiu and Knightly [2001] proposed an MBAC scheme that measures maximal rate envelopes of aggregate traffic to exploit statistical multiplexing along the bandwidth dimension. Their scheme provides aggregate loss rate guarantees, but does not differentiate among flows that may have different tolerance to delay violations.

In contrast to the existing MBAC schemes, an important difference of our DDM algorithm is that it is capable of differentiating among multiple QVPNs that have distinct tolerance to delay violations even if they have the same delay bound requirement.

6.2 Statistical Storage Resource Allocation

The Façade [Lumb et al. 2003] system has similar overall design goals as our Stonehenge system. However, its underlying implementation has important differences. Since Façade uses EDF as the disk scheduling algorithm, it is difficult to correlate delay bound guarantee with bandwidth resource requirement. As a result, it does not include an admission control algorithm, let alone a statistical admission control algorithm that can exploit statistical multiplexing while providing bandwidth and delay guarantees.

Urgaonkar et al. [2002] studied resource overbooking in shared hosting platforms for CPU and network resources to achieve high resource utilization while guaranteeing QoS. To overbook resources in a controlled fashion, their approach does capacity profiling (either CPU or network bandwidth). In comparison, DDM focuses on request latency profiling as we need to guarantee distinct per-QVD latency and violation probability bounds. In addition, Stonehenge performs runtime aggregate profiling of applications to better exploit the multiplexing effect of multiple streams. On the other hand, Urgaonkar et al. [2002] utilize offline individual profiling of each application.

Vin et al. [1994] discussed statistical admission control algorithms for media servers and found that three times the number of streams can be admitted compared to the deterministic approach if up to 3% of the playback cycles are allowed to overflow. Vernick et al. [1996] reported empirical measurements from actual implementations of statistical admission control algorithms in a fully operational disk-array video server. The DDM algorithm not only handles media stream playback workload but can also provide distinct statistical QoS guarantees for a heterogeneous mix of workloads that do not have as predictable request patterns as media stream playback.

7. CONCLUSIONS

In this article, we proposed the Delay Distribution Measurement (DDM)-based admission control approach. DDM can effectively take advantage of the statistical multiplexing effect along the delay dimension, and at the same time, provide each virtual resource share with a distinct probabilistic delay guarantee, that is, a bound on both delay as well as delay violation probability. We have applied DDM as a general approach to measurement-based admission control in (1) network resource allocation to admit QVPNs and (2) storage resource allocation to admit QVDs. By dynamically measuring the distribution of the ratios between actual request servicing delay and the worst-case delay bound, DDM is able to significantly lower the resource requirement of virtual units (QVPNs or QVDs) that have a

small tolerance to delay violations. DDM also provides a unified framework to support QVPNs requiring deterministic or probabilistic delay bounds. Through detailed trace-driven performance evaluation of QVPNs (using Voice over IP traces) and QVDs (using video stream, TPC-C, and Web search I/O traces), we have shown that DDM can potentially increase the number of admitted virtual units (and resource utilization) by up to a factor of 3.0 over deterministic admission control approaches.

The framework of the DDM algorithm could also be extended to include simultaneous multiplexing along the bandwidth dimension to yield potentially greater link utilization. We are also interested in using DDM as a building block to exploit statistical network resource multiplexing in the end-to-end scenario where QVPNs traverse multiple network links. Beyond network and storage, another interesting application of DDM is in managing a mix of heterogeneous resources across shared server platforms, such as application hosting clusters, where each subscriber receives distinct service rate, latency, and tolerance guarantees.

ACKNOWLEDGMENTS

We would like to thank Henning Schulzrinne and Wenyu Jiang for providing the VoIP traces used in this article.

REFERENCES

ANDREWS, M. 2000. Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In *Proceedings of IEEE INFOCOM* (March).

BOORSTYN, R., BURCHARD, A., LEIBEHERR, J., AND OOTTAMAKORN, C. 2000. Statistical service assurances for traffic scheduling algorithms. *IEEE J. Select. Areas Comm. 18*, 13, 2651–2664.

BOUDEC, J.-Y. L. AND VOJNOVIC, M. 2002. Stochastic analysis of some expedited forwarding networks. In *IEEE Infocom* (June).

BRESLAU, L., JAMIN, S., AND SHENKER, S. 2000. Comments on performance of measurement-based admission control algorithms. In *Proceedings of IEEE INFOCOM* (March) .

C-S. CHANG, CHIU, Y., AND SONG, W. 2001. On the performance of multiplexing independent regulated inputs. In *ACM Sigmetrics 2001/Performance 2001*. 184–193.

CROSBY, S., LESLIE, I., MCGURK, B., LEWIS, J., RUSSELL, R., AND TOOMEY, F. June 1997. Statistical properties of a near-optimal measurement-based admission CAC algorithm. In *Proceedings of IEEE ATM*.

CRUZ, R. 1991. A calculus for network delay, Part I: Network elements in isolation. *IEEE Trans. Inform. Theory 37*, 1, 114–131.

ELWALID, A. AND MITRA, D. 1999. Design of generalized processor sharing schedulers which statistically multiplex heterogeneous QoS classes. In *Proceedings of IEEE INFOCOM* (March). 1220–1230.

FLOYD, S. 1996. Comments on measurement-based admission control for controlled load services. Tech. rep., Lawrence Berkeley Laboratory (July).

GIBBENS, R. AND KELLY, F. 1997. Measurement-based connection admission control. In *Proceedings of 15th International Teletraffic Conference* (June).

GOPALAN, K. AND CHIUEH, T. 2001. Real-time disk scheduling using deadline sensitive scan. Tech. rep. ECSL-TR-92, Experimental Computer Systems Lab, Stony Brook University.

GOPALAN, K., CHIUEH, T., AND LIN, Y. 2004. Probabilistic delay guarantees using delay distribution measurements. In *Proceedings of ACM Multimedia*, New York, NY.

GUILLEMIN, F. M., LIKHANOV, N., MAZUMDAR, R. R., AND ROSENBERG, C. 2002. Extremal traffic and bounds for the mean delay of multiplexed regulated traffic streams. In *Proceedings of IEEE INFOCOM*, New York, NY. (June).

HUANG, L., PENG, G., AND CHIUEH, T. 2004. Multi-dimensional storage virtualization. In *Proceedings of ACM Sigmetrics/Performance*, New York, NY.

JAMIN, S., DANZIG, P., SHENKER, S., AND ZHANG, L. 1997. A measurement-based admission control algorithm for integrated services packet networks. *IEEE/ACM Trans. Network. 5*, 1, 56–70.

JIANG, W. AND SCHULZRINNE, H. 1996. Analysis of On-Off patterns in VoIP and their effect on voice traffic aggregation. In *Proceedings of ICCCN* (March).

KELLY, F. 1996. Notes on effective bandwidths. *In Stochastic Networks: Theory and Applications 4*, 141–168.

KESIDIS, G. AND KONSTANTOPOULOS, T. 2000. Worst-case performance of a buffer with independent shaped arrival processes. *IEEE Comm. Lett. 4*, 1, 26–28.

KNIGHTLY, E. AND SHROFF, N. B. 1999. Admission control for statistical QoS. *IEEE Network 13*, 2, 20–29.

KUROSE, J. 1992. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proceedings of ACM Sigmetrics*. 128–139.

LUMB, C. R., MERCHANT, A., AND ALVAREZ, G. A. 2003. Façade: Virtual storage devices with performance guarantees. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, San Francisco, CA.

PAREKH, A. AND GALLAGER, R. 1993. A generalized processor sharing approach to flow control in integrated services networks: The single-node case. *IEEE/ACM Trans. Network. 1*, 3, 344–357.

QIU, J. AND KNIGHTLY, E. 2001. Measurement-based admission control with aggregate traffic envelopes. *IEEE/ACM Trans. Network. 9*, 2, 199–210.

REISSLEIN, M., ROSS, K., AND RAJAGOPAL, S. 2002. A framework for guaranteeing statistical QoS. *IEEE/ACM Trans. Network. 10*, 1, 27–42.

SIVARAMAN, V. AND CHIUSSI, F. 2000. Providing end-to-end statistical delay guarantees with earliest deadline first scheduling and per-hop traffic shaping. In *Proceedings of IEEE INFOCOM* (March) .

URGAONKAR, B., SHENOY, P., AND ROSCOE, T. 2002. Resource overbooking and application profiling in shared hosting platforms. In *Proceedings of Symposium on Operating Systems Design and Implementation* (Dec.) Boston, MA.

VERNICK, M., VENKATRAMANI, C., AND CHIUEH, T. 1996. Adventures in building the stony brook video server. In *Proceedings of ACM Multimedia*.

VIN, H. M., GOYAL, P., AND GOYAL, A. 1994. A statistical admission control algorithm for multimedia servers. In *Proceedings of ACM Multimedia*.

WANG, Y. AND ZHU, Q. 1998. Error control and concealment for video communication: A review. *Proceedings of IEEE 86*, 5, 974–997.

ZHANG, L. 1991. Virtual Clock: A new traffic control algorithm for packet-switched networks. *ACM Trans. Comput. Syst. 9*, 2, 101–124.

Received June 2005; revised April 2006; accepted July 2006