A Framework of Node Architecture and Traffic Management Algorithms for Achieving QoS Provisioning in Integrated Services Networks

Dapeng Wu[1] Yiwei Thomas Hou[2*] Zhi-Li Zhang[3] H. Jonathan Chao[1]

[1] Polytechnic University, Department of Electrical Engineering, Brooklyn, NY, USA
[2] Fujitsu Laboratories of America, Network Research, Sunnyvale, CA, USA
[3] University of Minnesota, Department of Computer Science, Minneapolis, MN, USA

Abstract

Recent market demand has put quality of service (QoS) support as the key feature in differentiating network products from various vendors. Advances in silicon technology has made it feasible to design per-flow based traffic management algorithms to control QoS with substantially improved performance over traditional class-based approach for the next generation switches and routers. This paper presents a node architecture with the aim of achieving QoS provisioning for the guaranteed service (GS), the controlled-load (CL), and the best-effort (BE) service offerings for the future integrated services networks. Under our node architecture, we propose several novel traffic management algorithms, which includes Adaptive Rate allocation for Controlled-load (ARC) flows, a hybrid model-based and measurement-based admission control algorithm for GS and CL flows, and a Quasi-Pushout Plus (or QPO+) packet discarding mechanism. Simulation results show that, once admitted into the network, our node architecture and traffic management algorithms are capable of providing hard performance guarantees to GS flows under all conditions, consistent (or soft) performance to CL flows under both light load and heavy load conditions, and minimal negative impact to conforming GS, CL and BE traffic should there be any non-conforming behavior from some CL flows. Furthermore, our node architecture and traffic management algorithms resolve some key problems associated with the traditional class-based approach.

Key Words: Integrated services networks (ISN), per-flow queueing, guaranteed service (GS), controlled-load (CL) service, best-effort (BE) service, effective bandwidth, weighted fair queuing (WFQ), admission control, packet discarding, quality of service (QoS)

[*] Please direct all correspondence to Y. T. Hou, Fujitsu Labs of America, Network Research, 595 Lawrence Expressway, Sunnyvale, CA 94086-3922, USA. Tel. (408) 530-4529, Fax (408) 530-4515, Email: thou@fla.fujitsu.com.

1 Introduction

One of the most challenging problems for the next generation Internet is to support diverse multimedia applications with quality of service (QoS) guarantees. To address this challenge, the IETF Integrated Services Working Group has specified three types of services, namely, the *guaranteed service* [12], the *controlled-load service* [15], and the *best-effort service*.

The guaranteed service (GS) guarantees that packets will arrive within the guaranteed delivery time, and will not be discarded due to buffer overflows, provided that the flow's traffic conforms to its specified traffic parameters [12]. This service is intended for applications which need a hard guarantee that a packet will arrive no later than a certain time after it was transmitted by its sender. That is, the GS does not control the minimal or average delay of a packet; it merely controls the maximal queueing delay. Examples that have hard real-time requirements and require guaranteed service include certain audio and video applications which have fixed playback rates. Delay typically consists of two components, namely, fixed delay and queueing delay. The fixed delay is a property of the chosen path, which is not determined by the guaranteed service, but rather, by the setup mechanism. Only queueing delay is determined by the GS.

The controlled-load (CL) service is intended to support a broad class of applications which have been developed for use in today's Internet, but are sensitive to heavy load conditions [15]. Important members of this class are the adaptive real-time applications (e.g., *vat* and *vic*) which are offered by a number of vendors and researchers [7]. These applications have been shown to work well over lightly-loaded Internet environment, but to degrade quickly under heavy load conditions. The controlled-load service does not specify any target QoS parameters. Instead, acceptance of a request for controlled-load service is defined to imply a commitment by the network to provide the requester with a service closely approximating the QoS the same flow would receive under lightly-loaded conditions.

Both the guaranteed service and the controlled-load service are designed to support real-time applications which need different levels of QoS guarantee from the network.

The best-effort (BE) service class offers the same type of service under the current Internet architecture. That is, the network makes effort to deliver data packets but makes no guarantees. This works well for non-real-time applications which can use an end-to-end retransmission strategy (i.e., TCP) to make sure that all packets are delivered correctly. These include most popular applications like Telnet, FTP, email, Web browsing, and so on. All of these applications can work without guarantees of timely delivery of data. Another term for such non-real-time applications is *elastic*, since they are able to stretch gracefully in the face of increased delay. Note that these applications can benefit from shorter-length delays but that they do not become unusable as delays increase.

To support the diverse QoS requirements from the GS, the CL, and the BE services simultaneously within the same network, appropriate network node architecture and traffic management algorithms must be in place. Such architecture and algorithms must meet the following performance evaluation criteria as specified by IETF.

Criterion 1 (C1): For guaranteed service, IETF requires that the architecture and algorithms of each switch must ensure that the delay bounds are never violated and packets are not lost if a sender's traffic conforms to its traffic parameters [12].

Criterion 2 (C2): For controlled-load service, an architecture and algorithms should provide a flow, under all load conditions, with a QoS closely similar to the QoS that the same flow would receive under lightly-loaded network conditions [15].

Criterion 3 (C3): A node architecture and traffic managements algorithms must be capable of controlling non-conforming GS/CL flows by minimizing their negative impact on other conforming GS/CL flows and BE flows [12, 15].

Previous work on integrated services networks has been focused on class-based queueing architecture. A seminar paper by Clark, Shenker, and Zhang in [2] proposed to use a class-based queueing with FIFO+ scheduling for predictive service.[1] In [7], Jamin, *et. al,* employed an architecture to support predictive service using class-based approach with a priority scheduler.[2] However, there are several problems under such class-based approach.

Problem 1 (P1): First, the class-based approach is unable to effectively isolate non-conforming flows and minimize their negative impact on other conforming GS and CL flows (i.e., criterion C3).

Problem 2 (P2): Second, the class-based approach requires to classify all incoming CL flows into a limited set of classes and is unable provide a flexible QoS support for *each individual* CL flow based on its unique traffic behavior and specific QoS requirements.

Problem 3 (P3): Finally, it is impossible for a class-based approach to enforce fair rate allocation for CL flows.

Recent market demand has put QoS support as the key feature in differentiating network products from various vendors. Furthermore, due to advances in silicon technology, hardware implementation of sophisticated per-flow based traffic management algorithms no longer poses any major cost constraint [1]. Such hardware capabilities enable us to design per-flow based traffic management mechanisms to control QoS with substantially improved performance over traditional class-based approach for the next generation switches and routers. This paper presents a node architecture and several traffic management algorithms based on per-flow queueing that not only satisfy the three criteria to support integrated traffic of the GS, the CL, and the BE services, but also resolve the three problems associated with the traditional class-based approach.

Our proposed node architecture strives to offer a good balance between traffic isolation and buffer sharing. We make three separate buffer partitions for the GS, the CL, and the BE flows, and one separate partition for non-conforming GS/CL packets. Per-flow queueing with weighted

[1] Predictive service defined in [2] is not identical to CL service but also has soft QoS requirements.

[2] For GS flows, both [2] and [7] employ per-flow queueing and WFQ scheduling as we have done in this paper.

fair queueing (WFQ) scheduling is employed for GS and CL flows; while shared queueing with FIFO is employed for BE flows and non-conforming GS/CL packets. We propose an Adaptive Rate allocation for Controlled-load (ARC) algorithm to provide soft bandwidth allocation to CL flows while enforcing a guaranteed rate allocation to each GS flow. We present a hybrid call admission control (CAC) algorithm consisting of model-based CAC for GS flows and measurement-based CAC for CL flows. Finally, we design a packet discarding algorithm, called quasi-pushout plus (QPO+), to effectively control non-conforming CL flows. Our simulation results show that once admitted into the network, our architecture offers guaranteed performance to GS flows under all conditions (C1), consistent performance to CL traffic under both light load and heavy load conditions (C2), and minimal negative impact on conforming flows should there be any non-conforming behavior from some CL flows (C3). Furthermore, our node architecture and traffic management algorithms have resolved the three problems associated with the traditional class-based approach.

Our simulation results are based on the assumption that the network employs homogeneous switch architecture and traffic management algorithms presented in this paper. For real world Internet, we would like to stress that even though we cannot require each router/switch use the same architecture and traffic management algorithms, a partial deployment of our architecture and traffic management algorithms can still have clear benefits to Internet traffic. For example, an ISP can put GS and CL services in its backbone and provide GS and CL service between customers (or between POPs (Point of Presence)) [12]. Furthermore, it is entirely feasible to fully deploy our architecture and algorithms within a single administrative domain (e.g., Intranet) and support the GS, the CL, and the BE services.

The remainder of this paper is organized as follows. Section 2 presents our node architecture for supporting the GS, the CL, and the BE services and overviews our traffic management algorithms. In Section 3, we present in detail the key traffic management algorithms under our node architecture. Section 4 uses simulation results to demonstrate the performance of our network architecture and traffic management algorithms under various traffic load conditions and network configurations. Section 5 concludes this paper.

2 Architecture

This section presents our novel implementation architecture using per-flow queueing for supporting QoS provisioning in integrated services networks.

In our model, an integrated services network is constructed by interconnecting switches or routers with a set of links. A flow consists of a sequence of packets within a particular application and traverses one or more links in the network from a sender to a receiver.

We assume that each switch employs output port buffering. Figure 1 shows our architecture for the GS, the CL, and the BE traffic at each output port of a network node. Under our architecture (Fig. 1), we partition each output port buffer pool into four parts: one for GS flows, one for CL flows, one for BE traffic, and one for non-conforming GS or CL packets. Within the same buffer partition for GS or CL flows, we employ per-flow queueing for each individual GS or CL flow.

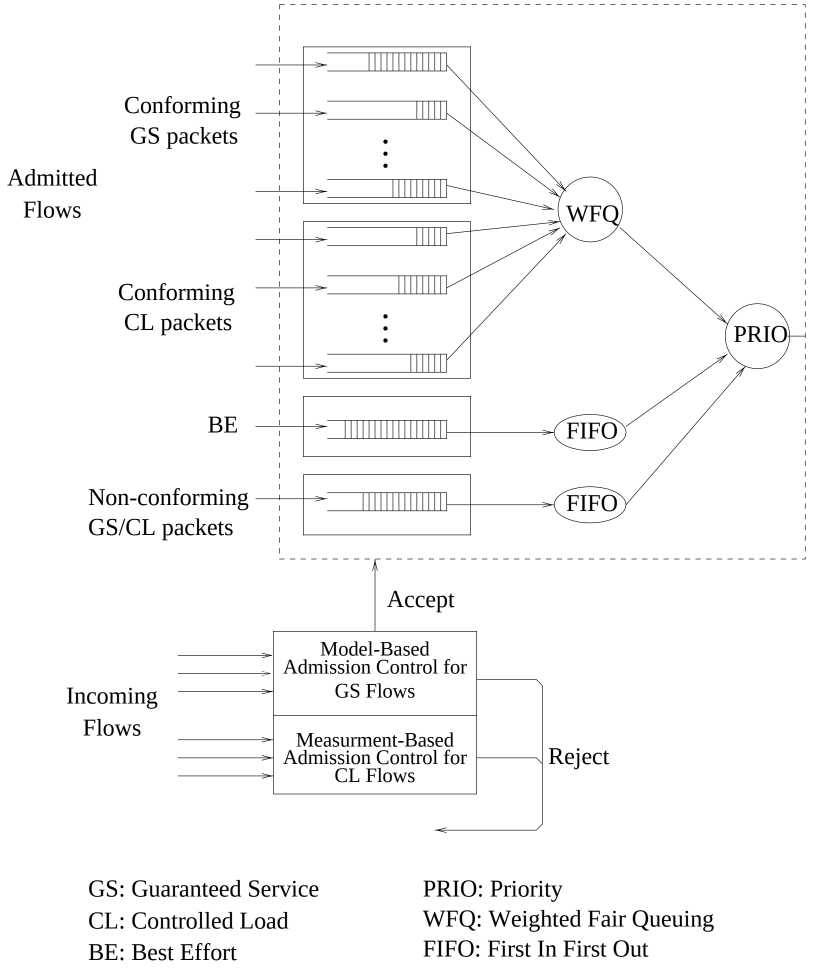

Figure 1: Node architecture for supporting integrated services.

Furthermore, a GS (or a CL) flow can share buffer with other GS (or CL) flows within their own buffer partition while there is no buffer sharing across partitions. That is, there is no buffer sharing between GS and CL flows. We believe this approach offers an excellent balance between traffic isolation and buffer sharing.

For BE buffer partition, we employ a common FIFO shared queue. This is because there is no QoS commitment of any kind to each individual BE flow.

For admitted GS or CL flows equipped with policing mechanism, packets not conforming to traffic parameters will be tagged at the network access point [15]. We propose to use one separate buffer for such non-conforming GS or CL packets and give them the lowest service priority so that they will have minimal negative impact on BE traffic [12, 15].

Under the above queueing architecture, we design our per-flow based traffic management algorithms with the aim of achieving the three criteria for the GS, the CL, and the BE services and solve the several problems associated with the class-based approach. For the ease of esposition, we give some highlights of our traffic management algorithms here, all of which will be discussed in detail in Section 3.

The first part of our traffic management is rate and buffer allocation and packet scheduling, which we will present in detail in Section 3.1. For GS traffic, we employ a simple calculation to allocate rate and buffer requirements, which provides a deterministic QoS guarantee (i.e., hard delay bound for each packet and zero packet loss) for each flow. On the other hand, for CL flows, we can choose a much less conservative approach, since it only requires soft QoS guarantees. We show how to estimate the effective bandwidth of a CL flow by measuring the entropy of such flow. To support the link sharing between the GS and the CL flows, we present a novel rate assignment strategy called ARC (short for Adaptive Rate allocation for Controlled-load) to provide hard bandwidth guarantee to GS flows under all conditions and consistent (or soft) bandwidth allocation to CL flows. Also shown in Fig. 1 is a hierarchical packet scheduling architecture where a priority link scheduler is shared among a weighted fair queueing (WFQ) for GS and CL flows,[3] a FIFO for BE flows, and a FIFO for non-conforming GS/CL packets. Service priority is first given to the WFQ scheduler, and then to BE FIFO scheduler. The FIFO scheduler for non-conforming GS/CL packets has the lowest priority in receiving service.

The reason why we use per-flow queueing and WFQ scheduler for CL flows is based on the results in [10] by Lo Presti, Zhang, and Towsley. In [10], it has been shown that GPS (fluid model of WFQ) scheduling is able to provide a flexible QoS support (both loss and delay requirement) and enforce bandwidth allocation for each individual flow. In other words, per-flow queueing with a WFQ scheduler in our architecture solves problems P2 and P3 associated with the class-based approach.

The second part of our traffic management is on admission control, which will be presented in Section 3.2. The objectives of admission control are: (1) to check if the QoS requirements of the new flow will be satisfied should it be admitted into the network, and (2) to check if the QoS commitments of existing (admitted) flows can still be met after admitting the new flow. We design

[3] We leave the specific implementation of WFQ to vendors.

a simple hybrid admission control algorithm which consists of model-based admission control for GS flows and measurement-based admission control for CL flows. Note that there is no admission control for BE traffic and such type of flows are always admitted. Simulation results show that our CAC can achieve high link utilization while meeting the QoS requirements of each admitted flow.

The last part of our traffic management is on buffer management, or more specifically, packet discarding strategy when some buffer partition is full.

For GS buffer partition, since the admission control algorithm for an incoming GS flow includes buffer allocation, an admitted flow will have sufficient buffer space throughout its path. Therefore, there should not be any buffer overflow for GS buffer partition. In the worst case, should the network misbehave, we may employ simple tail-dropping for GS buffer partition.

For CL flows, buffer partition may overflow since the traffic behavior of such flow is unpredictablewe and we do not reserve any buffer space for each CL flow at call setup time. Furthermore, since the network cannot assume that every admitted CL flow is equipped with a policing mechanism at the network access point, some non-conforming CL flow without policing mechanism may keep sending non-conforming packets into the CL buffer partition instead of the non-conforming GS/CL buffering partition. To address this problem, we propose a powerful pushout mechanism, called *quasi-pushout plus* (QPO+) to pushout packets from the quasi-longest queue to non-conforming buffer partitiion whenever the CL buffer partition cannot accommodate a new packet. Our QPO+ extends the classical pushout (PO) mechanism by its ability to handle variable sized packets. Such packet discarding scheme is only possible under per-flow queueing node architecture and can achieve fair buffer sharing among competing flows during congestion. We show that our QPO+ mechanism is capable of protecting the QoS guarantees to conforming flows by isolating and discarding packets from non-conforming flows. The details of QPO+ algorithm will be given in Section 3.3. Note that our QPO+ solves problem P1 associated with the traditional class-based approach.

For BE buffer partition, we use Flow Random Early Drop (FRED) (proposed in [8] to prevent non-adaptive BE flows from harming other TCP-like BE flows) for BE traffic. Finally, we employ simple tail-dropping for non-conforming GS/CL packets buffer partition.

In our simulation results in Section 4, we will show that our node architecture combined with the above traffic management algorithms can achieve the three performance criteria (listed in Section 1) for supporting integrated services.

Remark 1 During our early design phase, we have considered four possible buffering strategies and scheduling algorithms to handle BE flows and non-conforming GS/CL packets.

1. Use one shared buffer partition for both BE flows and non-conforming GS/CL packets and FIFO scheduling. But this mechanism cannot prevent non-conforming GS/CL packets from negatively affecting BE flows.

2. Use one shared buffer partition and employ per-flow queueing for both BE flows and non-conforming GS/CL packets. Under appropriate scheduling, this will provide optimal performance to TCP-type BE flows [13] and the greatest flexibility to serve non-conforming GS/CL

packets.

3. Use two separate buffer partitions, one for BE flows and one for non-conforming GS/CL packets. The scheduling scheme for these two buffer partitions is Weighted Round Robin (WRR) or Round Robin (RR). Under this architecture, non-conforming GS/CL packets can still have negative impact on BE flows when using (W)RR. In particular, it is not clear how to assign appropriate weights in WRR scheduler in order to prevent non-conforming GS/CL packets from harming BE flows.

4. This is the architecture employed in this paper, where we use two separate buffer partitions, one for BE flows and one for non-conforming GS/CL packets. The scheduling scheme is static priority, where BE flows are given higher priority. Regarding buffer management, we use Flow Random Early Drop (FRED) (proposed in [8] to prevent non-adaptive BE flows from harming other TCP-like BE flows) for BE traffic, and use simple tail-dropping for non-conforming GS/CL packets. By giving higher priority to BE flows, non-conforming GS/CL packets cannot have any effect on BE flows. Since the network does not have any commitment to non-conforming GS/CL packets, we can give them the lowest priority.[4] □

In the next section, we discuss the key traffic management algorithms in detail.

3 Traffic Management Algorithms

We organize this section as follows. Section 3.1 discusses rate and buffer allocation for GS flows as well as measurement-based rate estimation for CL. These rates will be used as the basis for our design of WFQ scheduler and hybrid admission control algorithm. Section 3.2 presents our hybrid admission control algorithm. In Section 3.3, we show our quasi-pushout plus (QPO+) packet discarding mechanism.

3.1 Resource Allocation for GS and CL Flows

Model-Based Rate Calculation for GS Flows

According to [12], the end-to-end queueing delay bound for a GS flow j is given by [5]

$$D_j \leq \begin{cases} \frac{\sigma_j - M_j}{p_j - \rho_j} \cdot (\frac{p_j}{R_j} - 1) + \frac{M_j + Ctot_j}{R_j} + Dtot_j & \text{if } \rho_j \leq R_j < p_j; \\ \frac{M_j + Ctot_j}{R_j} + Dtot_j & \text{if } \rho_j \leq p_j \leq R_j. \end{cases} \tag{1}$$

[4] Since non-conforming GS/CL packets are put into lowest service priority, they may arrive out of sequence with respect to the particular GS/CL flow. We assume that the application layer at the receiver side is capable of performing packet re-sequencing in the playback buffer. In the worst case, non-conforming GS/CL packets arriving beyond certain time threshold are subject to discarding.

[5] The end-to-end delay consists of the end-to-end queueing delay and the propagation delay.

where

σ_j: the leaky bucket size for flow j;

ρ_j: the token generating rate for flow j;

p_j: the peak rate of flow j;

R_j: the allocated bandwidth for flow j;

M_j: the maximum pacjet size of flow j;

$Ctot_j$: the rate-dependent error term for flow j;

$Dtot_j$: the rate-independent error term for flow j.

Therefore, for a given delay requirement for a GS flow j, its required rate R_j^{GS} can be easily derived from the above formula, which is a linear function of $\frac{1}{R_j}$. Note that the rate R_j^{GS} for the GS flow j is derived from a leaky bucket model, which is a conservative approach for bandwidth allocation. It is appropriate for the GS since such flows have hard delay requirements [12].

Buffer Allocation for GS

To guarantee zero packet loss for GS, appropriate buffer must be allocated for each GS flow. In [6], Georgiadis, *et. al,* derived an upper bound on the buffer requirement for a GS flow based on Linear Bounded Arrival Process (LBAP) model [3]. In this paper, we use this result on buffer allocation for GS flows. For flow $j \in GS$, the required buffer allocation at the l^{th} switch along the path is given by

$$b_j^{(l)} = M_j + \frac{(p_j - X)(\sigma_j - M_j)}{(p_j - \rho_j)} + \sum_{k=1}^{l}[\frac{C_j^{(k)}}{R_j} + D_j^{(k)}] \cdot X \tag{2}$$

where

$$X = \begin{cases} \rho_j & \text{if } \frac{\sigma_j - M_j}{p_j - \rho_j} \leq \sum_{k=1}^{l}[\frac{C_j^{(k)}}{R_j} + D_j^{(k)}]; \\ \\ R_j & \text{if } \frac{\sigma_j - M_j}{p_j - \rho_j} > \sum_{k=1}^{l}[\frac{C_j^{(k)}}{R_j} + D_j^{(k)}] \text{ and } p_j > R_j; \\ \\ p_j & \text{otherwise.} \end{cases} \tag{3}$$

In the above equations, $C_j^{(k)}$ and $D_j^{(k)}$ are the rate-dependent error term and the rate-independent error term at the k^{th} switch for flow $j \in GS$, respectively. $Ctot_j$ and $Dtot_j$ are the sum of $C_j^{(k)}$ and the sum of $D_j^{(k)}$ along the path of flow $j \in GS$, respectively.

Measurement-Based Rate Estimation for CL Flows

Unlike GS flows, CL flows do not have hard delay requirements and therefore do not require hard bandwidth and buffer guarantee. Instead, CL flows only require soft bandwidth support from the network for consistent performance under light and heavy load conditions. Therefore, we can adapt more efficient bandwidth allocation based on the measurement of a CL flow's actual traffic behavior (instead of a model based on a rigid parameters).

To accurately estimate the effective bandwidth for CL flows based on the measurement of their traffic behavior, we divide time axis into small fixed constant interval d and denote t_B be the time required to accumulant a total of B bits for a particular CL flow. Clearly, t_B is a variable depending on the particular incoming CL flow traffic behavior. When a flow is inactive or its arrival rate is very small, t_B can stretch over a large time interval in order to accumulant B bits, which makes it difficult to close the measurement window. To address this problem, we introduce a threshold T_{max} to set up an upper bound on the measurement interval. More specifically, we take the minimum of t_B and T_{max} as our measurement window T. That is,

$$T = \min\{t_B, T_{max}\} \ . \tag{4}$$

Let M be the total number of d's within a measurement window T. Then,

$$M = \lceil \frac{T}{d} \rceil \ .$$

Let $A_i^T(k)$, $1 \leq k \leq M$ be the number of bits arrived in the kth measurement interval. We first estimate the scaled cumulant generating function (SCGF) $\Lambda(\delta)$ as follows (see Appendix B).

$$\Lambda^T(\delta_i) = \frac{1}{T} \log \frac{1}{M} \sum_{k=1}^{M} e^{\delta_i A_i^T(k)} \tag{5}$$

where $\delta_i = \frac{-(\log \varepsilon_i - \log \gamma_i)}{b}$, b is the size of the CL buffer partition, ε_i is the packet loss rate requested by sender i, and γ_i is the probability that the queue for flow i is non-empty. Let λ_p^i be the peak rate of flow i. Then, we can obtain the effective bandwidth of CL flow i by

$$\alpha(\delta_i) = \min\{\lambda_p^i, \frac{\Lambda^T(\delta_i)}{\delta_i}\} \ . \tag{6}$$

In our measurement, we only measure the number of packets in bits that have successfully entered the buffer partition, *excluding* discarded packets. This is because that discarded packets will not be served by the scheduler, and thus it is only necessary to consider the packets that have successfully entered the buffer and to allocate appropriate rate for such packets. Furthermore, we find that such measurement technique has the additional advantage of preventing non-conforming flows from unfairly increasing its rate share in the scheduler by sending more packets.

We assume the requirement for packet loss rate is available in order to calculate the required bandwidth. For CL flows, user are not required to explicitly request such QoS parameter. But for engineering purpose (i.e., to estimate required bandwidth), we may assign a value for packet loss rate suitable for the particular CL service.

Rate Assignment for GS and CL Flows

To provide hard rate guarantee to each GS flow and soft rate guarantee to each CL flow, we employ the following weight assignment strategy in the WFQ scheduler. When the sum of guaranteed rates from GS flows (calculated from Eq. (1)) and the estimated rates from CL flows is less than the link capacity, we use these rates directly in the WFQ for the corresponding GS or CL flows and the delay requirement for each GS flow is always guaranteed. On the other hand, if the sum of calculated GS rates and the measured CL rates is greater than the link capacity, to guarantee the hard delay requirements for GS flows, we shall still use the calculated rate for each GS flow as the weight in the WFQ scheduler. But for each CL flow, we use a down-scaled version of the estimated rate (by a factor of remaining capacity divided by the sum of estimated CL rates) as the weight for the corresponding CL flow in the WFQ scheduler. We name this rate assignment *ARC,* for Adaptive Rate assignment for Controlled-load.

Algorithm 1 ARC - Adaptive Rate allocation for Controlled-Load flows
For an admitted CL flow i, its rate R_i^{CL} is given by

$$R_i^{CL} = \begin{cases} \alpha(\delta_i) & \text{if } \sum_{i\in CL}\alpha(\delta_i) + \sum_{j\in GS} R_j^{GS} \le r; \\ \alpha(\delta_i)\cdot\frac{(r-\sum_{j\in GS}R_j^{GS})}{\sum_{i\in CL}\alpha(\delta_i)} & \text{if } \sum_{i\in CL}\alpha(\delta_i) + \sum_{j\in GS} R_j^{GS} > r. \end{cases} \quad (7)$$

where $\alpha(\delta_i)$ is the measured effective bandwidth of the CL flow i, and r is the link rate. □

Once we use the R_j^{GS}, $j \in GS$, and R_i^{CL}, $i \in CL$ as the weight for the respective flow j or i in our WFQ scheduler, we have the following property on rate allocation for GS and CL flows. The rate allocation for each GS flow is no less than its calculated guaranteed rate, which is a hard rate guarantee. On the other hand, the rate allocation for each CL flow may have occasional fluctuations (due to on-line measurement of each CL flow traffic behavior), which is understood to be a soft bandwidth guarantee [15].

3.2 Admission Control Algorithm

For GS flows, we use worst-case deterministic calculations based on the (σ, ρ) parameters of the flow. Therefore, call admission control is relatively easy. But for CL flows, we use soft QoS guarantees and thus call admission control is much more challenging. One obvious question, for example, is: what is the minimum service rate (or bandwidth) a CL flow requires in order to satisfy its QoS guarantees? We resort to the effective bandwidth technique to solve this problem. We refer interested readers to [17] for the details on how to apply the theory of effective bandwidths to call admission control under GPS scheduling.

First, we show the optimal admission control algorithm under GPS (Algorithm 2) by Zhang, *et. al,* in [17].

Suppose we have n sessions sharing a single GPS server with a given GPS assignment, $\{\phi_i\}_{1\le i\le n}$. The session arrival processes are assumed to be independent. For session i, $1 \le i \le n$, $\alpha_i(\theta)$ is the effective bandwidth function of its arrival process.

For each i, let H_i be the maximal partial feasible set of $\mathcal{S}\backslash\{i\}$ with respect to θ_i, and γ_i be the associated delimiting number for H_i. Hence for any session $j \in H_i$, $j \neq i$ if and only if $\alpha_j(\theta_i) < \phi_j \gamma_i$. The optimal admission control algorithm is described as follows [17].

Algorithm 2 Optimal Admission Control Under GPS

```
Upon the arrival of a new flow $j$ requesting connection
    let flow $j$ join set $\mathcal{S}$, $\mathcal{S} := \mathcal{S} \cup \{j\}$;
    for $i \in \mathcal{S}$ {
        if $(\alpha_i(\theta_i) \geq \phi_i \gamma_i)$ then
            reject the connection request for the new flow $j$;
            $\mathcal{S} := \mathcal{S}\backslash\{j\}$;
            exit;
        }
    accept the connection request for the new flow $j$;
    exit.
```

□

The time complexity of the above optimal admission control algorithm is $O(n^2 \log n)$, where n is the number of flows. In many circumstances, a faster and simpler admission control algorithm is desirable, despite the fact that such an algorithm will, in general, be "sub-optimal" in the sense that it rejects calls that may be otherwise admitted under the optimal admission control algorithm. In the following, we present such a sub-optimal algorithm, where the guaranteed bandwidth $g_i = \frac{\phi_i}{\sum_{j \in \mathcal{S}} \phi_j} r$, $i \in \mathcal{S}$ [17].

Algorithm 3 Sub-Optimal Admission Control Under GPS

```
Upon the arrival of a new flow $j$ requesting connection
    let flow $j$ join set $\mathcal{S}$, $\mathcal{S} := \mathcal{S} \cup \{j\}$;
    for $i \in \mathcal{S}$ {
        if $(\alpha_i(\theta_i) \geq g_i)$ then
            reject the connection request for the new flow $j$;
            $\mathcal{S} := \mathcal{S}\backslash\{j\}$;
            exit;
            }
    accept the connection request for the new flow $j$;
    exit.
```

□

Algorithm 3 takes only $O(n)$ time. Note that if we regard $\rho_i = \alpha_i(\theta_i)$ as the "rates" of the session, since $\rho_i < g_i$, the minimum guaranteed bandwidth for session i, the admitted sessions are scheduled according to a RPPS-like GPS policy: each session is guaranteed a minimum bandwidth which exceeds its QoS requirement.

In our architecture, we use RPPS feasibility test to do admission control as in [17]. More specifically, suppose we have n CL flows with aggregate measured rate $\sum_{i=1}^{n} \alpha_i^T(\delta_i)$. We use the effective bandwidth $\alpha_j(\theta_j)$ to describe our admission control algorithm. Since the minimum guaranteed service rate for the new flow $g_j = \frac{\phi_j}{\sum_{i=1}^{n} \phi_i} r = \frac{\alpha_j(\theta_j)}{\sum_{i=1}^{n} \alpha_i^T(\delta_i)+\alpha_j(\theta_j)} r$, the rejection decision $\alpha_j(\theta_j) \geq g_i$ in Algorithm 3 becomes $\sum_{i=1}^{n} \alpha_i^T(\delta_i) + \alpha_j(\theta_j) \geq r$.

The following shows our admission control algorithm for the GS and CL flows, where μ is target utilization and r is the link capacity.

Algorithm 4   Admission Control for GS and CL Flows

```
Upon receiving a new flow requesting the GS
    if ($\sum_{j \in GS} R_j^{GS} + \sum_{i \in CL} R_i^{CL} + R_{new}^{GS} \leq \mu \cdot r$) and ($\sum_{j \in GS} b_j^{GS} + b_{new}^{GS} \leq b^{GS}$)
    /* $b^{GS}$ is the size of GS buffer partition. */
        admit the new GS flow and stop;
    else
        reject the new GS flow and stop;

Upon receiving a new flow requesting the CL service
    if ($\sum_{j \in GS} R_j^{GS} + \sum_{i \in CL} R_i^{CL} + R_{new}^{CL} \leq \mu \cdot r$)
    /* $R_{new}^{CL}$ is the requested rate for the new CL flow rather than measured rate.[6] */
        admit the new CL flow and stop;
    else
        reject the new CL flow and stop.                                □
```

In Algorithm 4, we use the peak rate of a CL for admission control rather than the token generating rate ρ. This is because that our previous experience in [16] has shown that the ρ parameter can be less than the required rate and, therefore, the targeted QoS could be violated if we only reserve a bandwidth of ρ.

3.3 QPO+ Packet Discarding Mechanism

An arriving packet is allowed to enter the particular buffer partition only when there is enough remaining buffer space. Otherwise, we have to either discard the incoming packet or discard some other packet(s) in the buffer in order to make room for the incoming packet.

The pushout (PO) packet discarding scheme allows an incoming packet to enter the buffer by discarding some other packet in the longest logical queue. It has been shown in [14] that the PO mechanism brings the following two key performance advantages: (1) It is fair in the sense that it allows smaller queues to increase in length at the expense of longer queues, and (2) It is efficient in the sense that no packet is dropped before the buffer is full. The only problem associated with PO is that it has $O(N)$ implementation complexity to perform linear comparisons and find out the longest queue, where N is the number of flows in the buffer. The so-called Quasi Pushout (QPO) proposed

[6] We assume that the requested rate for CL flow is its peak rate. If its peak rate is the line rate [15], its peak rate λ_p can be obtained by $\lambda_p = \rho + \sigma/U$, where U is a user-defined averaging period [5].

in [9] is designed to solve this problem by making a tradeoff between the optimal performance of PO and implementation complexity. Basically, instead of performing $O(N)$ comparisons to find the *exact* longest queue, the QPO performs $O(1)$ comparisons to find the quasi-longest queue and has near-optimal performance comparing to the PO scheme.

Both the PO and QPO were introduced for ATM networks where all packets have fixed size. Therefore, they cannot be directly applied to our integrated services network where packet length is assumed to be of variable size. To address this issue, we propose a packet discarding mechanism, called QPO+, which extends the QPO mechanism for variable-sized packets.

In our QPO+ mechanism, a register is used to estimate the longest queue (LQ) in the CL buffer partition and is only updated upon a packet arrival or departure. The queue length of flow i, $QL[i]$, is in the unit of bits. When a packet arrives and the remaining free buffer space cannot accommodate such packet, packets from the quasi-longest queue (LQ) will be transferred to the non-conforming buffer partition (instead of being discarded) and make room for this incoming packet. The following algorithm shows how our QPO+ packet discarding scheme works. We use RB to denote the remaining free buffer space in the CL buffer partition. Note that we consider an output-buffered switch in Algorithm 5.

Algorithm 5 QPO+ Mechanism

```
When a packet of size $P$ from flow $i$ arrives at the output port of a switch,
    if ($RB \geq P$) {
        accept such packet and let it join flow $i$;
        $QL[i] := QL[i] + P$;
        $RB := RB - P$;
        }
    else /* i.e., $RB < P$ */ {
        if ($LQ == i$) or (($QL[LQ] + RB$) $< P$)
            put this incoming packet into the non-conforming buffer partition;
        else {
            pop packets (with a sum of $x$ bits) from the tail of queue $LQ$ to the non-conforming
            buffer until ($RB + x > P$);
            $QL[LQ] := QL[LQ] - x$;    $RB := RB + x$;
            accept the incoming packet and let it join flow $i$;
            $QL[i] := QL[i] + P$;
            $RB := RB - P$;
            }
        }
    if ($QL[LQ] < QL[i]$)
        $LQ := i$;   /* input comparison */

When a packet of size $P$ from flow $j$ departs from the output port of a switch,
    $QL[j] := QL[j] - P$;
    $RB := RB + P$;
```

```
if ($QL[LQ] < QL[j]$)
    $LQ := j$   /* output comparison */
```
□

In contrast to PO, which needs N comparisons to determine the exact longest queue whenever to discard a packet, QPO+ tracks the quasi-longest (or near-longest) queue by using two comparisons only: one at the arrival of an input packet and the other on the departure of a packet. While the quasi-longest queue in QPO+ may not always be the longest, it will be corrected to the true longest queue whenever a packet arrives or departs from the longest queue.

We would like to emphasize the following two points regarding QPO+ packet discarding mechanism. First of all, it should be clear that only under per-flow queueing architecture can we employ such pushout packet discarding mechanism. Secondly, according to [15], network elements must not assume that data senders or upstream elements have taken action to "police" CL flows (i.e., limiting their traffic to conform to the flow's traffic parameters). Therefore, each network element providing CL service must independently ensure that criterion C3 is met in the presence of non-conforming GS/CL traffic. Our simulations have shown that FIFO with tail dropping cannot prevent non-conforming traffic from affecting conforming flows. Only packet discarding mechanism with per-flow control capability such as QPO+ can effectively control non-conforming flows when policing is not available. It has been shown in [8] that FIFO-based RED cannot effectively control non-conforming flows. In the simulation results, we shall further demonstrate that when non-conforming users are present in the network, only QPO+ can minimize the negative impact from such flows on other conforming flow while other packet discarding schemes (e.g., drop-tail) are unable to effectively control such non-conforming flows.[7]

We would like to point out that it is entirely feasible to implement our QPO+ mechanism in hardware for IP router/switch. Since the largest IP packet size is 1500 bytes and the smallest is 64 bytes (under Ethernet), in the worst-case, the incoming packet with the largest packet size will pushout at most 24 packets with the smallest packet size. Unlike ATM where there is a cycle time constraint (e.g., 2.83 μs for OC-3), there is no such cycle time for IP router/switch and the processing time of a packet is basically proportional to the duration of the packet. The longer the packet, the more time there will be available to do pushout. Therefore, our QPO+ scheme will not have a timing constraint bottleneck in hardware implementation.

4 Simulation Results

In this section, we implement our integrated services architecture and traffic management algorithms on our network simulator and perform simulations on various benchmark network configurations and traffic conditions. The purpose of this section is to demonstrate that our architecture and algorithms can meet the three performance evaluation criteria listed in Section 1.

[7] Even in the case that all CL flows are conforming, there are still periods during which the buffer is full. Here, QPO+ can provide fairness in term of buffer sharing among the CL flows while tail-dropping is unable to achieve.

Table 1: Simulation parameters for three types of services.

GS	Peak rate r_p	1.5 Mbps
	Packet size	1K bits
	Delay bound	10 ms
CL	Peak rate r_p	1.5 Mbps
	$E(T_{ON})$	2 ms
	$E(T_{OFF})$	2 ms
	Packet size	1K bits
	Packet loss ratio requirement	10^{-3}
	Delay bound	20 ms
BE (TCP)	Peak rate r_p (light load)	1 Mbps
	Peak rate r_p (heavy load)	10 Mbps
	Mean packet processing delay	300 μs
	Packet processing delay variation	10 μs
	Packet size	1K bits
	Maximum receiver window size	64K bytes
	Default timeout	500 ms
	Timer granularity	500 ms
	TCP version	Reno

4.1 Simulation Settings

The network configurations that we use are the *peer-to-peer* (Fig. 2), the *parking lot* (Fig. 5), and the *chain* (Fig. 8) network configurations. All switches in the simulations are assumed to have output port buffering with internal switching capacity equal to the aggregate rates of its input ports. At each output port of a switch, we implement our architecture and traffic management algorithms.

On the connection level, we assume that a GS or CL flow's inter-arrival times is exponentially distributed with an average of 50 seconds, with the holding time exponentially distributed with an average of 100 seconds.

The simulation parameters for the GS, the CL, and the BE services are shown in Table 1. For GS flows, we use the simple constant bit rate as their traffic pattern. This helps to simplify our simulations without any loss of generality in demonstrating the performance of our architecture and traffic management algorithms. For each BE flow, we use persistent TCP data traffic. For CL flows, we use an exponentially distributed on/off model with average $E(T_{on})$ and $E(T_{off})$ for on and off periods, respectively. During each on period, the packets are generated at a peak rate of r_p. The average bit rate for a CL flow is, therefore, $r_p \cdot \frac{E(T_{on})}{E(T_{on})+E(T_{off})}$. Delay bound is obtained by the ratio of σ over ρ. In our simulations, the requested packet loss ratio ϵ for all the controlled-load service flows is set to 10^{-3}.

Table 2: Simulation parameters at an end system and network components.

End system (Sender or receiver)	GS	σ	15 packets
		ρ	1500 packets/s
		Buffer size	10 packets
	CL	σ	20 packets
		ρ	1000 packets/s
		Buffer size	10 packets
	TCP	Packet processing delay	500 μs
		Buffer size	500 packets
Switch	Buffer Size	Conforming GS	250 packets
		Conforming CL	250 packets
		BE	1000 packets
		Non-conforming GS/CL	1000 packets
	Packet processing delay		4 μs
	Bits required for CL measurement window		100 Kbits
Link	Link speed		10 Mbps
	Distance	End system to switch	1 km
		Switch to switch	1 km

In Table 2, we list the simulation parameters at each end system (i.e., sender and receiver) and network components (i.e., link and switch). Buffer size in Table 2 is the size of the entrance buffer before the leaky bucket. In our simulations, for GS flow j, $Ctot_j$ is assumed to be zero and $Dtot_j$ is only comprised of the packet processing delays at all the switches along its path,[8] i.e., $Dtot_j = Ltot_j \cdot D_j^{(k)} = Ltot_j \cdot 4\mu s$, where $Ltot_j$ is the number of switches along the path for flow j. We assume the propagation delay is $5\mu s$ per kilometer. Therefore, the end-to-end delay bound is determined by the end-to-end queueing delay (Eq. (1)) and the total propagation delay. The leaky bucket parameters (σ, ρ) in Table 2 are chosen based on such requirements: the dropped ratio is zero for GS flows and less than 10^{-3} for CL flows;[9] the ratio $\frac{\sigma}{\rho}$ is equal to the target delay bound.

In our simulations, we set the link capacity to be 10 Mbps and set the target link utilization μ to be 0.90 in order to cushion any traffic fluctuation and measurement error.

We ran our simulator for 300 seconds simulation time and found that 50 simulated seconds are sufficient for our simulator to warm up. In order to obtain 95% confidence interval, we repeated each simulation eight times, each of which with a different seed.

[8] The reason why $Dtot_j$ does not include the propagation delay is that the propagation delay does not contribute to the buffer requirement in Eq. (2).

[9] Dropped ratio is the ratio of the dropped packets at the entrance buffer over the total generated packets.

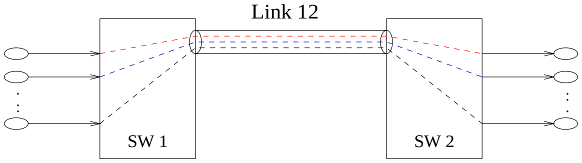

Figure 2: A peer-to-peer network.

Table 3: Number of GS, CL, and BE flows under light and heavy load conditions in the peer-to-peer network.

Load	Number of Flows		
Conditions	GS	CL	BE (TCP)
Light	3	3	3
Heavy	4	8	5

4.2 Simulation Results

We organize our presentation as follows. Section 4.2.1 presents the performance of the GS, the CL, and the BE traffic under light and heavy load conditions and show that criteria C1 and C2 are satisfied. In Section 4.2.2, we show that our architecture and algorithms can effectively control non-conforming flows by minimizing their negative impact on other conforming flows (criterion C3). Section 4.2.3 to 4.2.5 demonstrate the capabilities of each traffic management component. In particular, Section 4.2.3 shows that only our ARC algorithm can enforce hard rate guarantee to GS flows and soft rate allocation to CL flows. Section 4.2.4 compares our QPO+ packet discarding with the tail-dropping mechanism. Section 4.2.5 shows that our measurement-based admission control for CL traffic outperforms model-base admission control.

4.2.1 Performance Under Light and Heavy Load Conditions

We investigate the QoS experienced by the GS, the CL, and the BE traffic under light and heavy load conditions using various benchmark network configurations. The purpose of the simulations is to demonstrate that our network architecture and traffic management algorithm can achieve criteria C1 and C2.

The Peer-to-Peer Network

For this network (Fig. 2), the output port link of SW1 is the only bottleneck link for all flows. Table 3 shows the number of flows under light and heavy load conditions in our simulation. Note that only the GS and the CL flows requires admission control while there is no admission control for BE traffic.

We repeated the simulation eight times to obtain 95% confidence intervals. The 95% confidence

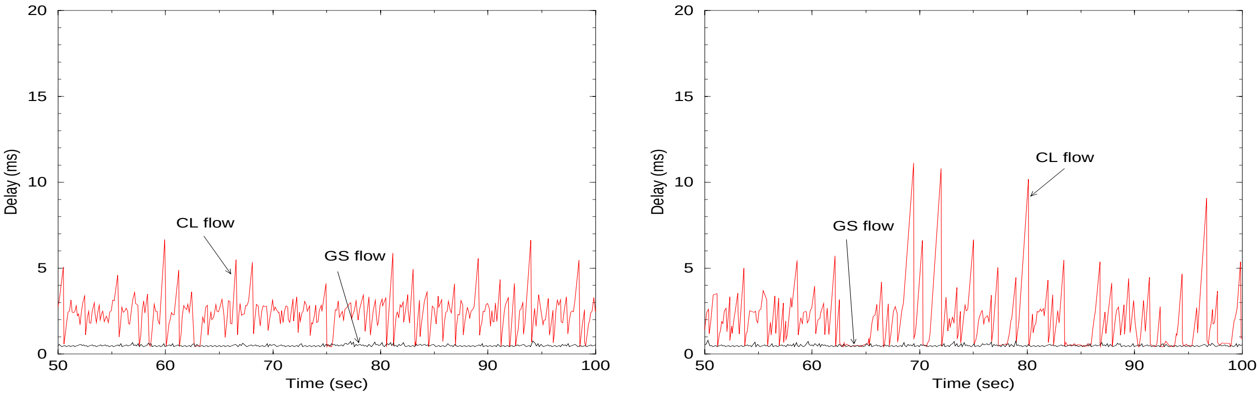

Figure 3: End-to-end delay of a GS flow and a CL flow under light load (left) and heavy load (right) in the peer-to-peer network.

intervals for the maximum end-to-end delay for GS and CL flows under light load are (0.771, 0.832) and (6.46, 6.88), respectively. The 95% confidence intervals for the maximum end-to-end delay for GS and CL flows under heavy load are (0.766, 0.837) and (10.71, 11.53), respectively. We find that the delays experienced by each GS and CL flows are bounded and are much less than the delay requirements for GS and CL flows, respectively. For illustration, we randomly pick up a GS flow and a CL flow among admitted GS and CL flows and plot their delay behavior under light and heavy load conditions in Fig. 3. As shown in Fig. 3, the delay experienced by this GS flow is bounded under both light and heavy load conditions and is much less than its delay bound requirement (10 ms). For the CL flow, its delay is also bounded under both conditions and is less than its delay requirements (20 ms). As expected, there is a delay increase for this CL flow under heavy load than under light load. But such increase is normal and is understood to be under "control" and the specific application supported by this CL flow should operate properly without any significant performance degradation. Figure 4 shows the link utilization on Link12 during the light and heavy load conditions.

Under both the light and heavy load conditions, there is no packet loss from any GS or CL flow.

Table 4 shows the performance of BE flows under light and heavy load. We observe that the throughput of TCP1, TCP2, and TCP3 decrease under heavy load as expected. Unlike GS and CL traffic, there is packet loss for BE traffic under heavy load conditions.

The Parking Lot Network

This configuration and its name is derived from theater parking lots, which consists of several parking areas connected via a single exit path. The specific parking lot network that we use is shown in Fig. 5, where path G1 consists of multiple flows and traverse from the first switch (SW1) to the last switch (SW5), path G2 starts from SW2 and terminates at the last switch (SW5), and so forth. Clearly, Link45 is the potential bottleneck link for all flows.

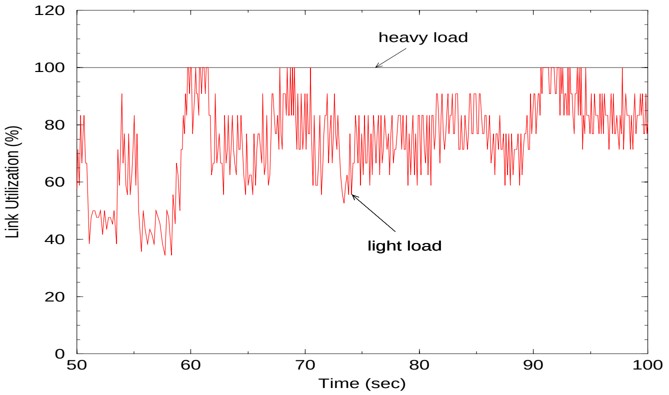

Figure 4: Link12 utilization under light and heavy load in the peer-to-peer network.

Table 4: Performance of BE (TCP) traffic under light and heavy load conditions in the peer-to-peer network.

BE Traffic (TCP flows)		Load Conditions	
		Light	Heavy
Throughput (kbps)	TCP1	303	65.6
	TCP2	305	63.2
	TCP3	298	66.8
	TCP4	–	61.9
	TCP5	–	64.1
Packet loss rate (%)	Link12	0	2.2

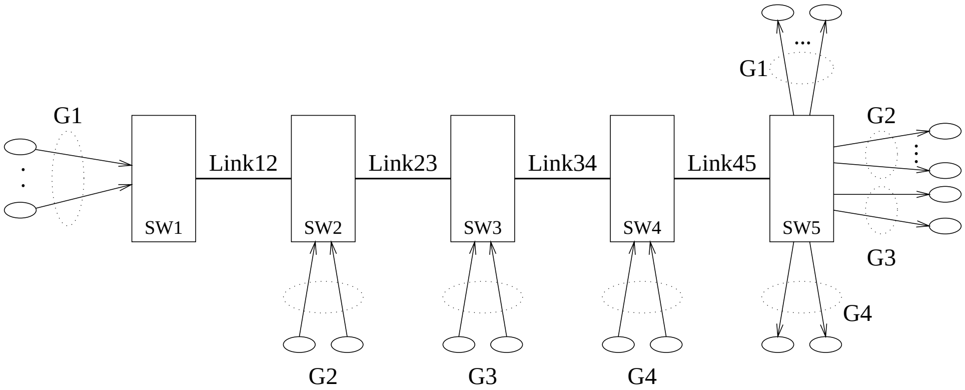

Figure 5: A parking lot network.

Table 5: Number of GS, CL, and BE flows on each path under light and heavy load conditions in the parking lot network.

Path	Traffic Type	Number of Flows	
		Light Load	Heavy Load
G1	GS	1	1
	CL	1	2
	BE (TCP)	1	2
G2	GS	1	1
	CL	1	2
	BE (TCP)	1	2
G3	GS	1	1
	CL	1	2
	BE (TCP)	1	2
G4	GS	1	1
	CL	1	2
	BE (TCP)	1	2

Table 5 shows the number of flows on each path under light and heavy load conditions in our simulation. We repeated the simulation eight times to obtain 95% confidence intervals. The 95% confidence intervals for the maximum end-to-end delay for GS and CL flows under light load are (1.521, 1.566) and (8.58, 9.09), respectively. The 95% confidence intervals for the maximum end-to-end delay for GS and CL flows under heavy load are (1.685, 1.733) and (14.05, 14.57), respectively. We find that the delays experienced by each GS and CL flows are bounded and are much less than the delay requirements for GS and CL flows, respectively. In Fig. 6, we plot the delay experienced by the GS flow and the CL flow traversing SW1 to SW5 (path G1) under light and heavy load. As shown in both figures, the delay experienced by this GS flow is bounded and is much less than its delay bound requirement (10 ms). For the CL flow, its delay is also bounded under both conditions and is less than its delay requirements (20 ms). As expected, there is some occasional delay increase for this CL flow under heavy load than under light load. Again, such increase is normal and is considered satisfying our performance objective for CL flows. Figure 7 shows the link utilization at Link45 during the light and heavy load conditions. Under both the light and heavy load conditions, there is no packet loss from any of the GS or CL flows.

Table 6 shows the performance of BE flows under light and heavy load. We observe that the throughput of TCP1, TCP2, TCP3, and TCP4 all decrease under heavy load as expected. In contrary to GS and CL traffic, there is packet loss for BE traffic under heavy load conditions. We find such loss occurs at Link34 (output port of SW3) and Link45 (output port of SW4), respectively.

The Chain Network

This is one of the benchmark network configurations to examine traffic behavior under the

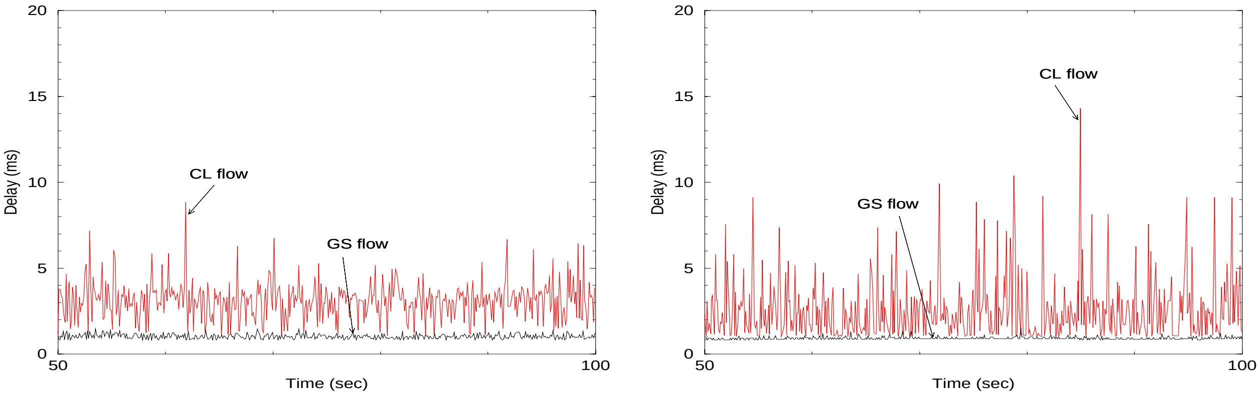

Figure 6: End-to-end delay of a GS flow and a CL flow under light load (left) and heavy load (right) in the parking lot network.

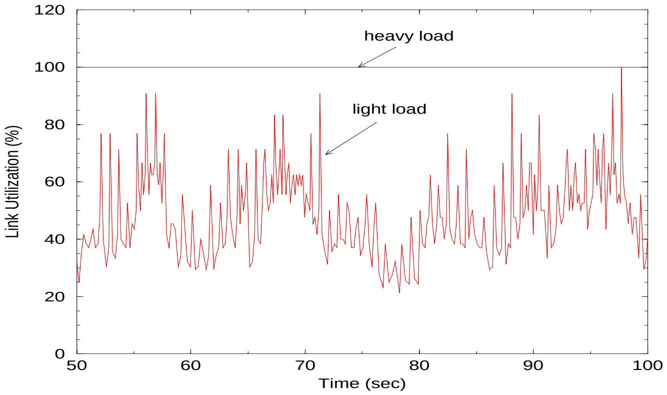

Figure 7: Link utilization of Link45 under light and heavy load in the parking lot network.

Table 6: Performance of BE (TCP) traffic under light and heavy load conditions in the parking lot network.

BE Traffic (TCP flows)		Load Conditions	
		Light	heavy
Throughput (kbps)	TCP1	32.6	10.7
	TCP2	32.5	10.6
	TCP3	32.3	10.4
	TCP4	32.2	10.3
	TCP5	–	10.7
	TCP6	–	10.6
	TCP7	–	10.4
	TCP8	–	10.3
Packet Loss Ratio (%)	Link12	0	0
	Link23	0	0
	Link34	0	1.8
	Link45	0	14.8

impact of traversing interfering traffic. The specific chain configuration that we use is shown in Fig. 8 where path G1 consisting of multiple flows and traverses from the first switch (SW1) to the last switch (SW4), while all the other paths traverse only one hop and "interfere" the flows in G1. The numbers of flows under light load and heavy load are shown in Table 7.

Under light traffic load, we repeated the simulation eight times to obtain 95% confidence intervals. The 95% confidence intervals for the maximum end-to-end delay for GS and CL flows are (1.443, 1.491) and (7.21, 7.69), respectively. We find the end-to-end delay experienced by each GS/CL flow is bounded and the packet loss is zero. As an illustration, Fig. 9 (left) shows the delay experienced by the GS and the CL flows traversing from SW1 to SW4 (path G1) under the light load condition shown in Fig. 10 (left).

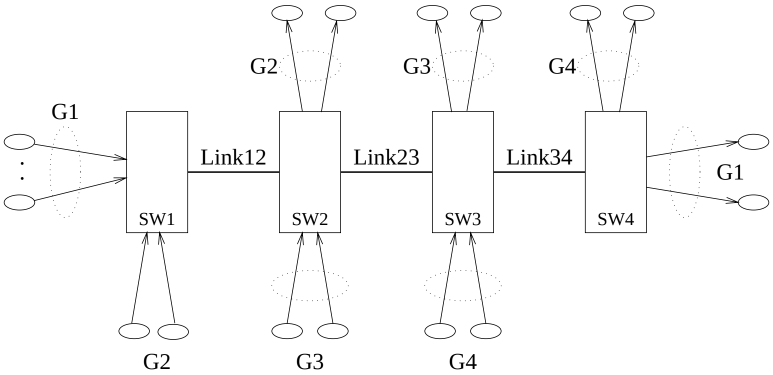

Figure 8: A chain network.

Table 7: Number of GS, CL, and BE flows on each path under light and heavy load conditions in the chain network.

Path	Traffic Type	Number of Flows	
		Light Load	Heavy Load
G1	GS	1	1
	CL	1	2
	BE (TCP)	1	3
G2	GS	0	1
	CL	1	2
	BE (TCP)	1	3
G3	GS	1	1
	CL	1	2
	BE (TCP)	1	3
G4	GS	1	1
	CL	1	2
	BE (TCP)	1	3

Under heavy traffic load, the 95% confidence intervals for the maximum end-to-end delay for GS and CL flows are (1.476, 1.524) and (12.81, 13.32), respectively. Also, the end-to-end delay experienced by each GS/CL flow is bounded and the loss is zero for each GS/CL flow. The delays experienced by the same GS and CL flows traversing path G1 are shown in Fig. 9 (right). Figure 10 (right) shows the link utilization of each link, where Link34 is 100% utilized under heavy load condition. Table 8 lists the throughput and packet loss ratios for BE flows under light and heavy load conditions.

4.2.2 Control of Non-Conforming CL Flows

For GS or CL flows that have policing mechanism, non-conforming flows can be effectively controlled by tagging GS/CL packets and putting them into the non-conforming GS/CL buffer partition.

But according to [15], network elements must not assume that each CL sourceor upstream elements have policing mechanism in place. Under such circumstances, the packets of a non-conforming CL flow may enter the CL buffer partition instead of the non-conforming GS/CL buffer partition. We show that our architecture and algorithms can effectively control such non-conforming CL flows and thus achieve criterion C3.

We use the parking lot configuration under heavy traffic load for demonstration. The non-conforming flow is chosen to be a flow on path G4, which shares the bottleneck link Link45 with all other flows on paths G1, G2, and G3. The non-conforming flow submit a peak rate of 1.5 Mbps as its traffic parameter for admission control but transmits at a peak rate of 10 Mbps. Since there is no policing mechanism for this flow, all packets from this flow enter the CL buffer partition.

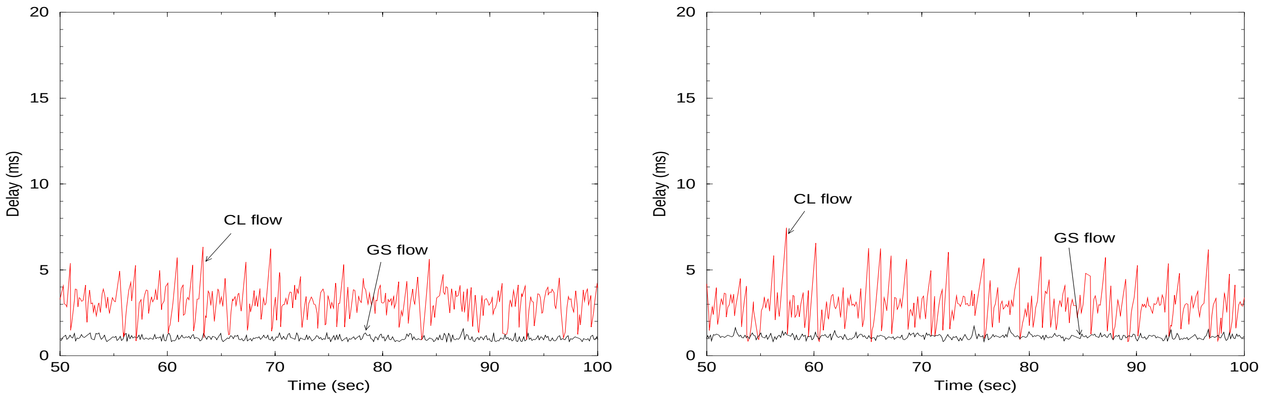

Figure 9: End-to-end delay of a GS flow and a CL flow under light load (left) and heavy load (right) in the chain network.

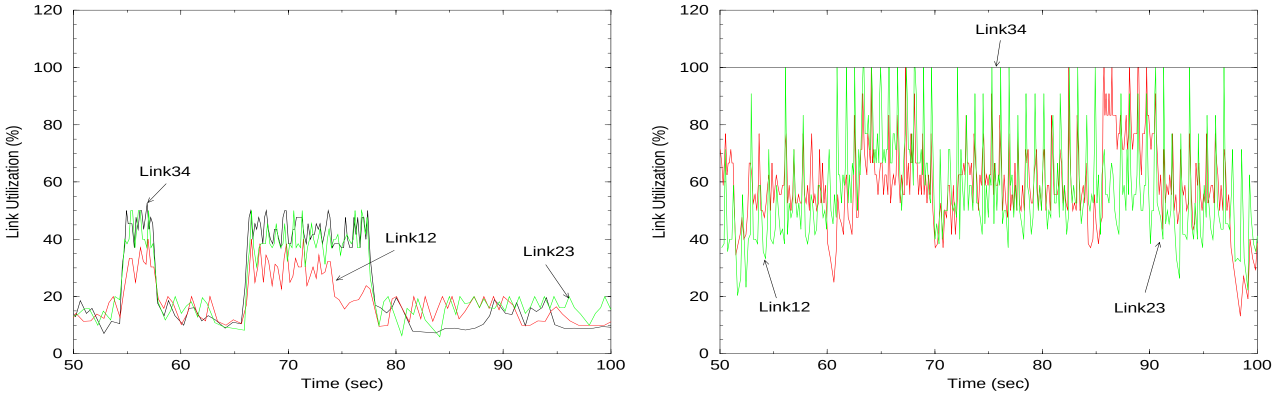

Figure 10: Link utilization under light load (left) and heavy load (right) in the chain network.

Table 8: Performance of BE (TCP) traffic under light and heavy load conditions in the chain network.

BE Traffic (TCP flows)		Load Conditions	
		Light	heavy
Throughput (kbps)	TCP1	32.6	30.1
	TCP2	32.4	29.1
	TCP3	32.5	29.4
	TCP4	32.3	25.8
	TCP5	–	30.1
	TCP6	–	29.1
	TCP7	–	29.4
	TCP8	–	25.8
	TCP9	–	30.1
	TCP10	–	29.1
	TCP11	–	29.4
	TCP12	–	25.8
Packet loss ratio (%)	Link12	0	0
	Link23	0	0.77
	Link34	0	4.05

Our simulations show that in the presence of such non-conforming CL flow, the contracted QoS to those conforming GS/CL flows can still be guaranteed while the non-conforming flow can be effectively isolated (due to per-flow queueing) and suffers from large packet loss rate (due to QPO+ packet discarding). In particular, we plot the delay for the conforming GS and CL flows on path G1 in Fig. 11, which shows that the delays experienced by these conforming GS and CL flows are bounded and are much less than their respective delay requirements. Furthermore the packet loss rate for these conforming flows remains zero during the simulation run. On the other hand, Fig. 12 shows the packet loss ratio experienced by the non-conforming CL flow suffers from heavy packet loss during the simulation.

The throughput of TCP connections (see Table 9) are comparable with those under heavy load in Table 6. So the non-conforming CL flow does not have any significant effect on BE traffic either.

We have just demonstrated that our node architecture and traffic management algorithms are capable of controlling non-conforming flows. Such effectively control are credited mostly to per-flow queueing based ARC and QPO+ algorithms. In the subsequent two subsections, we further examine these two traffic managment algorithms.

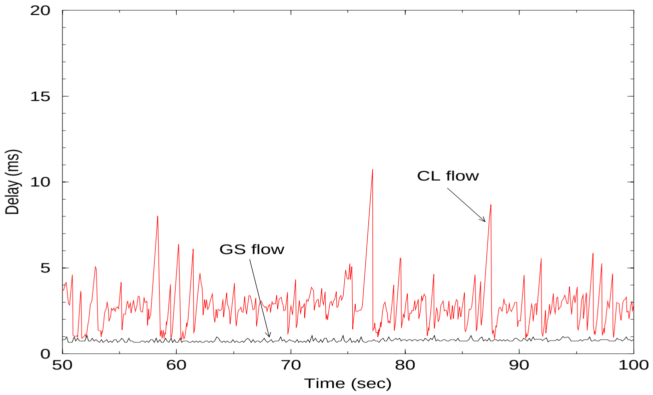

Figure 11: End-to-end delay for conforming GS and CL flows in parking lot network.

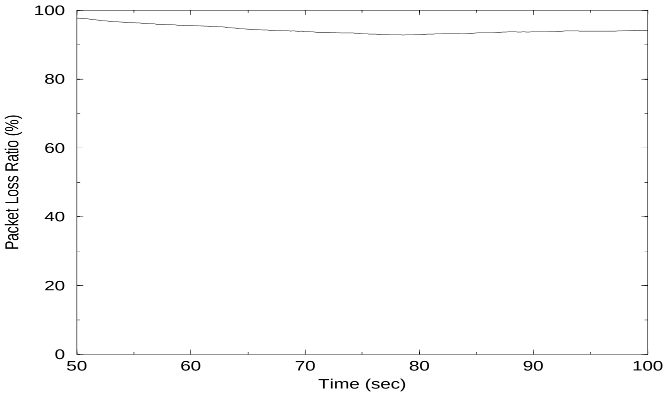

Figure 12: Packet loss ratio for non-conforming CL flows in parking lot network.

Table 9: Throughput of TCP connections under the presence of non-conforming flows in the parking lot configuration.

Throughput (kbps)	TCP1	10.1
	TCP2	9.9
	TCP3	9.9
	TCP4	9.7
	TCP5	10.1
	TCP6	9.9
	TCP7	9.9
	TCP8	9.7

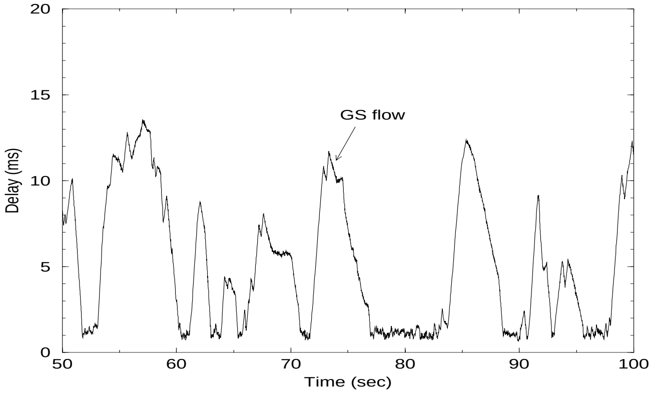

Figure 13: End-to-end delay for a GS Flow in parking lot network when ARC is not used.

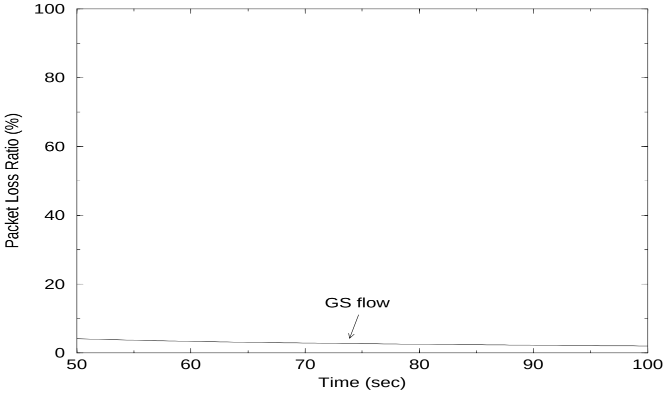

Figure 14: Packet loss ratio for a GS Flow in parking lot network when ARC is not used.

4.2.3 ARC or No ARC

To demonstrate the significance of our Adaptive Rate allocation for Controlled-Load flows (ARC) as described in Algorithm 1, we use the same simulation settings in Section 4.2.2. Here, instead of using ARC, we use calculated rate R_j, $j \in GS$ and measured rate $R_i = \alpha(\delta_i)$, $i \in CL$ directly as the weight in the WFQ scheduler.

Figures 13 and 14 shows delay and loss of the same GS flow on path G1. Here, the delay bound of 10 ms is violated and there is also packet loss for this GS flow, while the delay for the same GS is bounded (see Fig. 11) with zero packet loss when ARC is employed.

4.2.4 QPO+ vs. Tail-dropping

We compare the performance of QPO+ with tail-dropping packet discarding scheme. Again, we use the same simulation settings in Section 4.2.2, except we discard the incoming packet when the buffer partition is full (tail-dropping) instead of QPO+. Poisson call arrival is not used, and instead, we just run the simulation for 300 seconds.

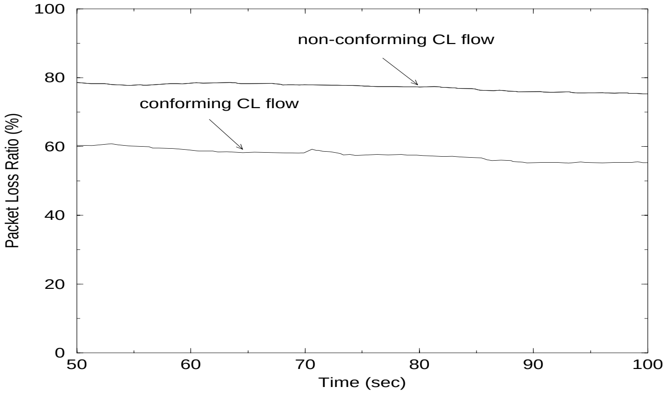

Figure 15: Packet loss ratio for conforming and non-conforming CL flows in the parking lot network under tail-dropping packet discarding mechanism.

Figure 15 shows that under tail-dropping, even conforming CL flow experiences large packet loss, while the same conforming CL flow experienced zero packet loss under QPO+ in Section 4.2.2.

4.2.5 Measurement-Based vs. Model-Based CAC for CL Flows

We would like to compare our measurement-based CAC algorithm (Algorithm 4) with a model-based CAC algorithm. We give a simple model-based CAC algorithm for CL traffic as follows.

Algorithm 6 Model-Based Admission Control for CL Flows

```
if a new flow requests the CL service
    if ($\sum_{i \in CL} R_i^{CL} + R_{new}^{CL} \leq \mu \cdot r$)
    /* $R_i^{CL}$ and $R_{new}^{CL}$ are calculated rates rather than measured rates. */
        admit the new CL flow and stop;
    else
        reject the new CL flow and stop.
```

□

Since packetized GPS (PGPS) scheduler is used, the delay bound D_i for flow i is given by the following expression [11].

$$D_i \leq \frac{\sigma_i + (K-1)L_i}{R_i} + \sum_{m=1}^{K} \frac{L_m^{max}}{r_m} \qquad (8)$$

where

σ_i: the leaky bucket size for flow i;

K: the number of switch nodes traversed;

L_i: the maximum packet size of flow i;

R_i: the allocated bandwidth for flow i;

Table 10: Parameters for CL flows

σ_i (packets)	K	L_i (Kbits)	L_m^{max}	r_m (Mbps)	R_i^{CL} (Mbps)	Target Rate (Mbps)
20	2	1	1	10	1.061	9

L_m^{max}: the maximum packet size among all the flows on link m;

r_m: the bandwidth capacity on link m.

By using the parameters for CL in Tables 1 and 2, we can obtain the related parameters for Algorithm 6 as shown in Table 10.

To compare measurement-based CAC and model-based CAC for CL traffic, we use only the CL traffic in our simulation. We use the peer-to-peer configuration and set K to two. Based on Eq. (8), the required bandwidth R_i^{CL} for CL flow i can be obtained (see Table 10). The maximum admissible number of flows N is given by

$$N = \lceil \frac{Target\ Rate}{R_i^{CL}} \rceil \ .$$

Thus, N is eight under Algorithm 6. On the other hand, we find that the maximum admissible number under our measurement-based admission control algorithm is twelve. Thus, measurement-based CAC achieves higher network utilization than model-based CAC for CL flows.

4.3 Summary of Simulation Results

Our simulation results in this section have demonstrated the following properties of our node architecture and traffic management algorithms for supporting integrated traffic of GS, CL, and BE services.

- Our framework is able to achieve the hard delay and loss guarantee to GS flows under all conditions (C1).
- Our framework provides consistent (soft) delay and loss performance for CL flows under both light and heavy load conditions (C2).
- Our framework can effectively control non-conforming flows by minimizing their negative impact on conforming flows (C3) (resolves P1 in the class-based approach).

5 Concluding Remarks

This paper presents a framework of network node architecture and traffic management algorithms, which has been demonstrated, to offer QoS provisioning for integrated traffic of the guaranteed service, the controlled-load, and the best-effort services for the future integrated services networks. The main contributions of this work is that our proposed node architecture and traffic management algorithms not only meet the three criteria for integrated services networks, but also solve the three problems associated with the traditional class-based approach. The highlights of our implementation architecture are listed as follows.

- To support the GS, the CL, and the BE services simultaneously within the same network, we proposed a queueing architecture for network node, where separate buffer partition was used for each service. Per-flow queueing architecture was employed within the GS and CL partitions for each admitted flow. Our buffering architecture offers an excellent balance between buffer sharing and traffic isolation among the three types of services.
- The use of per-flow queuing enabled us to design powerful traffic management algorithms that can provide QoS provisioning for the GS, CL, and BE traffic that was once not achievable under the traditional class-based queueing discipline.
  - Under per-flow queueing, we can employ WFQ scheduling algorithm that can guarantee hard bandwidth and delay constraints for the GS flows and soft bandwidth and delay requirements for the CL flows. Furthermore, under per-flow queueing, we can offer flexible QoS support for each individual flow based on its unique traffic behavior and QoS requirements. Such QoS capability and flexibility are not possbile under the traditional class-based approach.
  - We presented a measurement scheme to measure incoming CL traffic behavior and showed how to apply entropy theory to estimate effective bandwidth for each CL flow based on this measurement. Furthermore, we presented an adaptive rate allocation scheme (ARC) for our WFQ scheduler to serve GS and CL flows, which has the property of providing hard bandwidth guarantee to GS flows under all conditions and soft bandwidth allocation to CL flows.
  - We designed a simple hybrid admission control algorithm consisting of model-based admission control for GS flows and measurement-based admission control for CL flows.
  - We designed a packet discarding mechanism, called QPO+, which extends the quasi-pushout (QPO) to accommodate variable sized packets. Such packet discarding mechanism is only possible under our per-flow queueing architecture for the GS and the CL traffic. We showed that QPO+ was capable of controlling non-conforming flows and minimizing their negative impact on other conforming flows. Class-based packet discarding scheme such as tail dropping and RED are unable to effectively control non-conforming flows.

Simulation results showed that our proposed node architecture and traffic management algorithms provide guaranteed performance for GS flows under all conditions, consistent (soft) performance for CL traffic under both light load and heavy load conditions, and minimal negative impact on conforming flows when there are some non-conforming traffic flows. Furthermore, our simulation results show that while meeting QoS requirements of each GS/CL flow, our measurement-based admission control algorithm for CL is able to achieve consistent high link utilization under all simulation scenarios.

Acknowledgements

Prof. Zhi-Li Zhang's work has been supported in part by an NSF CAREER Award grant NCR-9734428.

Appendix A: Theory of Effective Bandwidth

Consider an arrival process $\{A(t),\, t \geq 0\}$ where $A(t)$ represents the amount of arrivals over the time interval $[0, t)$. We assume that sample paths of $A(t)$ are right continuous with left limits. For any $0 \leq \tau \leq t$, let $A(\tau, t) = A(t) - A(\tau)$. Hence $A(\tau, t)$ denotes the cumulative arrivals over the time interval $[\tau, t)$. Suppose that the asymptotic log-moment generating function of A, defined as

$$\Lambda(\theta) = \lim_{t\to\infty} \sup_{t} \frac{1}{t} \sup_{s\geq 0} \log E[e^{\theta A(s,s+t)}] \tag{9}$$

exists for all $\theta > 0$, then the *effective bandwidth* of A is defined as

$$\alpha(\theta) = \frac{\Lambda(\theta)}{\theta} \tag{10}$$

for all $\theta > 0$.

From (10), it is clear that effective bandwidth has the following sub-additive property: let $\{A_i(t),\, t \geq 0\}$, $1 \leq i \leq n$, be n independent arrival processes with effective bandwidths $\alpha_i(\theta)$, and let $\{A(t) = \sum_{i=1}^{n} A_i(t),\, t \geq 0\}$ be the aggregate arrival process, then the effective bandwidth $\alpha(\theta)$ of the aggregate process A is bounded above by $\sum_{i=1}^{n} \alpha_i(\theta)$, i.e., $\alpha(\theta) \leq \sum_{i=1}^{n} \alpha_i(\theta)$.

The effective bandwidth function $\alpha(\theta)$ is a key quantity that characterizes the stochastic behavior of the arrival process A in a G/D/1/∞ queueing system. Consider a queue of infinite capacity served by a server of constant service rate r. Suppose there are n independent sessions sharing the queue. The arrival process of session i, $1 \leq i \leq n$, is denoted by $\{A_i(t),\, t \geq 0\}$, and its effective bandwidth is $\alpha_i(\theta)$. Let the service discipline be any work-conserving scheduling policy. Let $A(t) = \sum_{i=1}^{n} A_i(t)$ be the aggregate arrival process, and $\alpha(\theta)$ its effective bandwidth. Then it can be shown that the tail distribution of the backlog process $Q(t) = \sup_{0\leq\tau\leq t}\{A(\tau, t) - r(t-\tau)\}$ satisfies

$$\lim_{b\to\infty} \sup \frac{1}{b} \log Pr\{Q(t) \geq b\} \leq -\theta \qquad \text{if } \alpha(\theta) \leq \sum_{i=1}^{n} \alpha_i(\theta) < r. \tag{11}$$

Eq. (11) demonstrates the importance of effective bandwidth in network call admission control. Consider a network switch which has total bandwidth r and a shared queue of size b. The backlog tail distribution $Pr\{Q(t) \geq b\}$ of an infinite capacity queue with a server of service rate r provides a conservative estimate of the loss probability at the network switch. Let ε be a desired upper bound on the loss probability, then from Eq. (11), for b sufficiently large, the condition $\sum_{i=1}^{n} \alpha_i(\theta) < r$, where $\theta = -\frac{\log \varepsilon}{b}$, implies that $Pr\{Q(t) \geq b\}$ can be asymptotically upper bounded by $e^{-\theta b} = \varepsilon$. Clearly the test $\sum_{i=1}^{n} \alpha_i(\theta) < r$ provides a basis for call admission control for our CL service.

Note that the above method of estimating loss probability may not be valid for small queue size. Moreover, it ignores the potential statistical multiplexing gain when there are a large number of senders. To obtain better estimate of the loss probability, one remedial method is to add the prefactor γ as follows:

$$\lim_{b \to \infty} Pr\{Q(t) \geq b\} \leq \gamma e^{-\theta b} \; . \tag{12}$$

The prefactor γ is the probability that the queue is nonempty. γ is dependent on the number of senders and reflects the gain achieved by multiplexing many senders while keeping the service rate per sender fixed.

Appendix B: Estimation of Effective Bandwidth

The essential part of measurement-based admission control is to measure QoS parameters (i.e., packet loss ratio, delay, bandwidth) and use such measured QoS parameters in admission control [4]. Since the large deviation rate function of the arrivals process is a function of QoS, we try to estimate rate function or entropy of the traffic directly rather than QoS itself. Consider a stationary arrival sequence $\{A_n, n \geq 1\}$, where A_n represents the number of arrivals over the time interval $[t_{n-1}, t_n)$. Suppose we have a single queue with a buffer of size b and constant service rate $r > E(A_1)$. The entropy of the arrival sequence is defined, for $x > E(A_1)$, by

$$I(x) = \lim_{n \to \infty} \frac{1}{n} \log Pr\left(\sum_{k=1}^{n} A_k > nx\right) \tag{13}$$

whenever this limit exists. From the theory of effective bandwidth (see Appendix A), the tails of the queue-length distribution can be bounded by (similar to (12))

$$\lim_{b \to \infty} Pr\{Q(t) \geq b\} \leq Pe^{-\delta b} \tag{14}$$

where

$$\delta = \inf_{w > 0} \frac{I(w+s)}{w} \tag{15}$$

Using the bound in (14), one can estimate the packet loss ratio, packet delay variation, etc. It is easier to estimate the scaled cumulant generating function (SCGF) than the entropy itself. The SCGF Λ is defined by

$$\Lambda(\theta) = \lim_{n \to \infty} \frac{1}{n} \log E\left[exp\left(\theta \sum_{k=1}^{n} A_k\right)\right] \tag{16}$$

whenever this limits exists; it is related to the entropy $I(x)$ by

$$\Lambda(\theta) = \sup_x\{x\theta - I(x)\}. \tag{17}$$

In addition, δ can be calculated directly from the SCGF using the formula

$$\delta = sup\{\theta : \Lambda(\theta) \leq s\theta\} \tag{18}$$

The sufficient conditions for the validity of the theory are stationarity and no long-range dependence.

For target packet loss rate (PLR) less than or equal to $Pe^{-\delta b}$, we estimate the effective bandwidth by

$$\alpha(\delta) = \frac{\Lambda(\delta)}{\delta} \tag{19}$$

where

$$\delta = \frac{-(\log\varepsilon - \log P)}{b} \tag{20}$$

and ε =PLR.

References

[1] H. J. Chao, "A delay-bound guarantee packet scheduler using a RAM-based searching engine," filed for U.S. patent, registration number 36242, Nov. 5, 1997.

[2] D. Clark, S. Shenker and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanism," *Proc. ACM SIGCOMM,* Aug. 1992.

[3] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation," *IEEE Trans. on Information Theory,* vol. 37, no. 1, pp. 114–131, Jan. 1991.

[4] N. G. Duffield, J. T. Lewis, N. O'Connell, R. Russell and F. Toomey, "Entropy of ATM traffic streams: a tool for estimating quality of service parameters," *IEEE Journal on Selected Areas in Communications,* vol. 13, no. 6, pp. 981–990, Aug. 1995.

[5] S. Floyd, "Comments on measurement-based admissions control for controlled-load service", *Technical Report,* July 1996, URL: http://www.aciri.org/floyd/papers/admit.ps.Z.

[6] L. Georgiadis, R. Guerin, V. Peris and R. Rajan, "Efficient support of delay and rate guarantee," *Proc. ACM SIGCOMM'96,* Aug. 1996.

[7] S. Jamin, P. B. Danzig, S. Shenker and L. Zhang, "A measurement-based admission control algorithm for integrated services packet networks," *IEEE/ACM Trans. on Networking,* vol. 5, no. 1, pp. 56–70, Feb. 1997.

[8] D. Lin and R. Morris, "Dynamics of random early detection," *Proc. ACM SIGCOMM'97.*

[9] Y. S. Lin and C. B. Shung, "Quasi-pushout cell discarding," *IEEE Commun.. Letters,* pp. 146–148, Sept. 1997.

[10] F. Lo Presti, Z.-L. Zhang and D. Towsley, "Bounds, approximations and applications for a two-queue GPS system," *Proc. IEEE INFOCOM'96,* pp. 1310–1317, March 1996.

[11] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks - the multiple node case," *IEEE/ACM Trans. on Networking,* vol. 2, no. 2, pp. 137–150, April 1994.

[12] S. Shenker, C. Partridge and R. Guerin, "Specification of guaranteed quality of service," *RFC 2212,* Internet Engineering Task Force, Sept. 1997.

[13] B. Suter, T. V. Lakshman, D. Stiliadis and A. K. Choudhury, "Design considerations for supporting TCP with per-flow queueing," *Proc. IEEE INFOCOM'98,* pp. 299–306, March 1998.

[14] L. Tassiulas, Y. C. Hung and S. S. Panwar, "Optimal buffer control during congestion in an ATM network node," *IEEE/ACM Trans. on Networking,* vol. 2, no. 4, pp. 374–386, Aug. 1994.

[15] J. Wroclawski, "Specification of the controlled-load network element service," *RFC 2211,* Internet Engineering Task Force, Sept. 1997.

[16] D. Wu and H. J. Chao, "Efficient bandwidth allocation and call admission control for VBR service using UPC parameters," *Proc. IEEE INFOCOM'99,* March 1999.

[17] Z.-L. Zhang, Z. Liu, D. Towsley and J. Kurose, "Call admission control schemes under the generalized processor sharing scheduling discipline," *Telecommunication Systems,* vol. 7, no. 1–3, pp. 125–152, July 1997.

Author Biographies

Dapeng Wu received the B.E degree from Huazhong University of Science and Technology, and the M.E. degree from Beijing University of Posts and Telecommunications in 1990 and 1997 respectively, both in Electrical Engineering. Since July 1997, he has been working towards his Ph.D. degree in Electrical Engineering, Polytechnic University, Brooklyn, New York.

During the summer of 1998 and part of 1999, he conducted research at Fujitsu Laboratories of America, Sunnyvale, California, on architectures and traffic management algorithms in integrated services (IntServ) networks and differentiated services (DiffServ) Internet for multimedia applications. His current interests are in the areas of next generation Internet architecture, protocols, implementations for integrated and differentiated services, and rate control and error control for video streaming over the Internet. He is a student member of the IEEE and the ACM.

Yiwei Thomas Hou obtained his B.E. degree (*Summa Cum Laude*) from the City College of New York in 1991, the M.S. degree from Columbia University in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, New York, in 1997, all in Electrical Engineering. He was awarded a National Science Foundation Graduate Research Traineeship for pursuing Ph.D. degree in high speed networking, and was recipient of Alexander Hessel award for outstanding Ph.D. dissertation in 1998 from Polytechnic University.

While a graduate student, he worked at AT&T Bell Labs, Murray Hill, New Jersey, during the summers of 1994 and 1995, on implementations of IP and ATM inter-networking; he also worked at Bell Labs, Lucent Technologies, Holmdel, New Jersey, during the summer of 1996, on network traffic management. Since September 1997, Dr. Hou has been a Research Staff Member at Fujitsu Laboratories of America, Sunnyvale, California. His current research interests are in the areas of next generation Internet architecture, protocols, and implementations for differentiated and integrated services. Dr. Hou is a member of the IEEE, ACM, and Sigma Xi.

Zhi-Li Zhang received the B.S. degree in Computer Science from Nanjing University, China, in 1986 and his M.S. and Ph.D. degrees in Computer Science from the University of Massachusetts, Amherst, in 1992 and 1997, respectively.

In 1997 he joined the Computer Science and Engineering faculty at the University of Minnesota, where he is currently an Assistant Professor. From 1987 to 1990, he conducted research in Computer Science Department at Århus University, Denmark, under a fellowship from the Chinese National Committee for Education. Dr. Zhang's research interests include computer communication and networks, especially the QoS guarantee issues in high-speed networks, multimedia and real-time systems, performance evaluation, queueing theory, applied probability theory, theory of computation and algorithms. He is a member of IEEE, ACM and INFORMS Telecommunication Section. Dr. Zhang received the National Science Foundation CAREER Award in 1997.

H. Jonathan Chao received the B.S.E.E. and M.S.E.E. degrees from National Chiao Tung University, Taiwan, in 1977 and 1980, respectively, and the Ph.D. degree in Electrical Engineering from The Ohio State University, Columbus, OH, in 1985.

He is a Professor in the Department of Electrical Engineering at Polytechnic University, Brooklyn, NY, which he joined in January 1992. His research interests include large-capacity packet switches and routers, packet scheduling and buffer management, and congestion flow control in IP/ATM networks. From 1985 to 1991, he was a Member of Technical Staff at Bellcore, NJ, where he conducted research in the area of SONET/ATM broadband networks. He was involved in architecture designs and ASIC implementations, such as the first SONET-like Framer chip, ATM Layer chip, and Sequencer chip (the first chip handling packet scheduling). He received Bellcore Excellence Award in 1987.

He served as Guest Editor for IEEE Journal on Selected Areas in Communications (JSAC) with special topics on "Advances in ATM Switching Systems for B-ISDN" published in June 1997 and "Next Generation IP Switches and Routers" to be published in 1999. He is currently serving as Editor for IEEE/ACM Transactions on Networking.