Performance Comparison between IntServ-Based and DiffServ-Based Networks

Shigeo Shioda∗
Department of Urban Environment & Systems
Chiba University
1-33 Yayoi, Inage, Chiba 263–8522, Japan
Email: shioda@faculty.chiba-u.jp

Kenichi Mase†
Department of Information Engineering
Niigata University
2-8050, Ikarashi, Niigata 950-2181, Japan
Email: mase@ie.niigata-u.ac.jp

Abstract—It is generally believed that IntServ architecture is more suitable for strict QoS guarantees than DiffServ architecture because the former has per-flow-base traffic handling mechanisms but the latter does not. This general belief, however, has not been fully verified yet. In this paper, we theoretically and numerically compare IntServ and DiffServ architectures in terms of x-percentile end-to-end queueing delay as well as the maximum end-to-end queueing delay. We find that, in some network topology, DiffServ architecture makes the maximum queueing delay smaller than IntServ architecture does. We also find that in most cases x-percentile queueing delay in DiffServ-based networks is much smaller than that in IntServ-based networks.

I. Introduction

The emergence of delay- or/and loss-sensitive applications like IP telephony has accelerated the research for the support of quality of service (QoS) in IP (datagram) networks. There are two different approaches for introducing the QoS to IP networks: integrated services (IntServ) [1] and differentiated services (DiffServ) [2]. The IntServ architecture foresees per-flow traffic classification and scheduling algorithm at routers for guaranteeing the QoS to each flow. The DiffServ architecture also has a traffic handling mechanism, but it does not work on a per-flow basis but works with aggregated traffic. In DiffServ-based networks, packets are classified into coarse-grain-QoS classes and the QoS-class information is inserted into the packet header at edge routers. Packets are scheduled and forwarded at each router on a per-class basis.

To be more specific, consider the end-to-end queueing delay of each flow. In IntServ-based networks, each flow reserves network resource and the reserved resource is used by the flow at highest priority. Thus, the maximum end-to-end queueing delay of a flow can be precisely predicted solely from the amount of reserved network resource. In contrast, in DiffServ-based networks, all flows classified into a given QoS class share the common network resource, and thus the end-to-end queueing delay of a given flow should depend on the amount of traffic from other flows belonging to the same class. This observation yields the general belief that the IntServ architecture is superior to the DiffServ architecture in terms of the QoS guarantees.

However, it is not theoretically trivial that the maximum end-to-end queueing delay in an IntServ-based network is smaller than that in a DiffServ-based network when the total network resource and the amount of offered traffic are the same. Moreover, the general belief misses an important advantage of DiffServ: DiffServ can benefit from the statistical multiplexing gain because flows share the common network resource. Thanks to this feature, x-percentile queueing delay in a DiffServ-based network might be much smaller than that in an IntServ-based network.

The aim of this paper is to compare the IntServ-based and DiffServ-based networks in terms of delay performance. For this purpose, using several simple network models, we evaluate the bound of x-percentile queueing delay together with the maximum queueing delay in both types of networks. As a result, we find that, in some network topology, DiffServ architecture makes the maximum queueing delay smaller than IntServ architecture. We also find that in most cases DiffServ makes the bound of x-percentile queueing delay much smaller than that in IntServ-based networks. If both types of networks can guarantee the QoS of flows in progress by limiting the amount of offered traffic, these findings in turn imply that DiffServ-based networks would admit more traffic to guarantee the QoS than IntServ-based networks. Although the analysis conducted in this paper is for extreme situations (e.g., maximum queueing delay), these findings would be counter-examples to the general belief that IntServ architecture is more suitable for strict QoS guarantees than DiffServ architecture.

Note that DiffServ architecture does not require per-flow admission control because providing QoS for individual flows is not primary purpose of DiffServ. Thus, one may claim that the above mentioned findings make sense under a very limited case where a DiffServ-based network has the admission control function. However, the admission control function can be supported even in DiffServ-based networks by employing a bandwidth broker architecture [3]; several important proposals have been made to implement a scalable admission control function in DiffServ-based networks [4], [5]. These studies were mainly motivated by DiffServ’s superior scalability and not by DiffServ’s superiority of statistical multiplexing. Thus, the findings of this paper would motivate further studies on the admission control function in DiffServ-based networks.

This paper is organized as follows. In Section II, we briefly introduce an analytical framework to evaluate feed-
forward networks with upper-constrained inputs, which has been proposed in [6]. Using this framework, in Section III, we derive the maximum end-to-end queueing delay and the bound of end-to-end queueing-delay distribution in a simple model for an IntServ-based network. In Section IV, we derive the maximum end-to-end queueing delay and the bound of end-to-end queueing-delay distribution in two different models for Diffserv-based networks. In Section V, we compare these models in terms of delay performance. In Section VI, we conclude the paper with a few remarks.

II. ANALYSIS OF FEEDFORWARD QUEUEING NETWORKS WITH UPPER CONSTRAINED INPUTS

In this section, we briefly describe a framework for analyzing feedforward queueing networks with upper constrained inputs, which has been proposed in [7], [6]. The performance of queues fed by various upper constrained inputs has been extensively studied in recent years, and the similar and related results are also found in several literatures [8], [9], [10].

A. Network model

Consider a feedforward network with $N$ infinite-capacity queues and $I$ flows of customers. All queues in the network are indexed by $n = 1, \cdots, N$. The route followed by customers of flow $i$ ($i = 1, \cdots, I$) is

$$i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{N(i)}$$

where $N(i)$ is the number of queues followed by customers of flow $i$. That is, customers of flow $i$ enter the network at queue $i_1$ and leave the network after service completion at queue $i_{N(i)}$. After service completion at queue $i_k$, a customer goes only to one of queues whose index is greater than $i_k$: that is

$$i_1 < i_2 < \cdots < i_{N(i)}.$$

We define

$$l(i, n) = \begin{cases} 1 & n \in \{i_1, \cdots, i_{N(i)}\}, \\ 0 & \text{otherwise}. \end{cases}$$

When $l(i, n) = 1$, denote by $Pre(i, n)$ the preceding queue of queue $n$ on the route followed by customers of flow $i$. According to the definition

$$Pre(i, i_2) = i_1, \cdots, Pre(i, i_{N(i)}) = i_{N(i) - 1}.$$

For simplicity, we let $Pre(i, i_1) = 0$. Customers of all flows are served in the order of arrival at each queue.

B. Traffic characteristics

We use the term the amount of traffic for referring to the sum of the sizes of jobs brought by customers. Denote by $A_i(t; n)$ ($D_i(t; n)$) the amount of traffic of flow $i$ arriving at (leaving from) queue $n$ during $(0, t]$ and define

$$A_i(t; n) \defeq \sum_{i=1}^{I} A_i(t; n).$$

We assume that the amount of traffic of each flow has the following characteristics:

1. For each $i$ ($i = 1, \cdots, I$), there exists an increasing and subadditive function $\alpha_i(\tau)$ such that

$$A_i(t + \tau; i_1) - A_i(t; i_1) \leq \alpha_i(\tau),$$

where $\alpha_i(\tau)$ is called subadditive envelope of flow $i$ [11], [12].

2. For each $i$ ($i = 1, \cdots, I$) and $k (k = 1, \cdots, N(i))$, $A_i(t; i_k)$ is stationary.

Let $B_n(s, t)$ be the amount of traffic that can be served at queue $n$ during $(s, t]$. We assume the following characteristics of $B_n(s, t)$:

1. There exist a superadditive increasing function $\beta_n(\tau)$ and a subadditive increasing function $\tilde{\beta}_n(\tau)$ such that

$$\beta_n(\tau) \leq B_n(s, t + \tau) \leq \tilde{\beta}_n(\tau).$$

Usually, $\beta_n(\tau)$ and $\tilde{\beta}_n(\tau)$ are respectively referred to as the minimum service curve and the maximum service curve [13].

The following inequality holds:

$$\lim_{t \to \infty} \frac{1}{t} A_i(t; n) < \lim_{t \to \infty} \frac{\beta_n(t)}{t}.$$
is assigned to each QoS-aware flow. (QoS-aware flows

\[ \frac{x(k, T)}{y(k, T, d)} \leq \frac{1 - x(k, T)}{1 - y(k, T, d)} \]

where

\[ k_{\text{max}}(T, d) \overset{\text{def}}{=} \sup \{ k : I_0(tkT; 1) > \beta_1((k - 1)T) + \beta_2(d) \}, \]

\[ x(k, T) = \frac{kT}{\alpha(t),} \quad y(k, T, d) = \frac{\beta_1((k - 1)T) + \beta_2(d)}{I_0(tkT)} \]

3) Output process: The output process of flow \( i \) has the subadditive envelope \( \delta_i(t) \), which is given by

\[ \delta_i(t) = \begin{cases} \min \{ \inf_{s \geq 0} \{ \alpha_i(s + d_{\text{max}}) + \beta_i(t - s) \}, \beta_i(t) \} & t \geq 0, \\ 0 & t < 0. \end{cases} \]

D. Analysis of downstream queues

The maximum waiting time and the bounds of virtual-waiting-time distribution in downstream queues can be iteratively evaluated. To see this, first note that the amount of flow \( i \) traffic arriving at queue \( n \) has a subadditive envelope \( \alpha_i(t; n) \), which is expressed as

\[ \alpha_i(t; n) = \begin{cases} \alpha_i(t) & \text{Pre}(i; n) = 0, \\ \delta_i(t; \text{Pre}(i; n)) & \text{Pre}(i; n) \neq 0. \end{cases} \]

Thus, the maximum waiting time at queue \( n \) can be evaluated using (1) by replacing \( \alpha_i(t) \) with \( \alpha_i(t; n) \) and \( \beta_i(t) \) with \( \beta(n) \). More precisely,

\[ d_{\text{max}}^{(n)} = \beta_i^{-1}(\sup_{t \geq 0} \{ \alpha(t; n) - \beta(n) \}), \]

\[ \alpha(t; n) \overset{\text{def}}{=} \sum_{l: \text{Pre}(l; n) = 0} l(i, n) \alpha_l(t) + \sum_{l: \text{Pre}(l; n) > 0} l(i, n) \delta_l(t; \text{Pre}(l; n))). \]

The bound of virtual waiting time distribution in queue \( n \) can also be evaluated using (2) by replacing \( \alpha_i(t) \) with \( \alpha_i(t; n) \) and \( \beta_i(t) \) with \( \beta(n) \).

The output process of flow \( i \) from queue \( n \) also has the subadditive envelope and its expression by

\[ \delta_i(t; n) = \begin{cases} \min \{ \inf_{s \geq 0} \{ \alpha_i(s + d_{\text{max}}) + \beta_i(t - s) \}, \beta(n) \} & \text{Pre}(i; n) = 0, \\ \min \{ \inf_{s \geq 0} \{ \delta_i(s + d_{\text{max}}; \text{Pre}(i; n)) + \beta(n) \}, \beta(n) \} & \text{Pre}(i; n) \neq 0, \end{cases} \]

which is used for the analysis of downstream queues.

III. DELAY CHARACTERISTICS OF INTSERV MODEL

The framework explained in previous section allows us to evaluate the delay performance of IntServ- and DiffServ-based networks. In this section, based on the proposed framework, we evaluate the delay performance of a simple model for an IntServ-based network.

A. Network model

We consider a network model (Model I) depicted in Fig. 1. This network model is a series of nodes where QoS-aware and best-effort traffic flows are multiplexed together. The number of nodes is \( N \) and the number of QoS-aware flows is \( I \). In each node, a buffer is dedicated to each QoS-aware flow and, through some scheduling algorithm, a minimum service rate \( c \) is assigned to each QoS-aware flow. (QoS-aware flows are assumed to be classified into “guaranteed service” class.) The best-effort traffic flows share a common buffer in each node. For simplicity, we assume that all nodes use the same scheduling algorithm and the capacities of links between nodes are all equal to \( C(> Ic) \). We use the following notation:

- \( L_{\text{be}} \): maximum packet size of best effort traffic
- \( L_{\text{qs}} \): maximum packet size of QoS-aware traffic
- \( L_{\text{max}} \overset{\text{def}}{=} \max\{L_{\text{be}}, L_{\text{qs}}\} \): maximum packet size

![Fig. 1. IntServ network model (Model I)](image)

Best effort

[Diagram of a network model with nodes labeled 1 to N, flows marked as 'intserv flows' and 'qos-aware flows']

We assume that all QoS-aware flows have the same subadditive envelope \( \alpha(t) \). Let \( B_{i \text{q}}^{(n)(t, t + \tau)} \) denote the amount of traffic of QoS-aware flow \( i \) that can be served at node \( n \) during \( (t, t + \tau) \). We assume that \( B_{i \text{q}}^{(n)(t, t + \tau)} \) has the following upper and lower bounds, which depends on the scheduling algorithm:

\[ \beta_{i \text{q}}(\tau) \leq B_{i \text{q}}^{(n)(t, t + \tau)} \leq \bar{\beta}_{i \text{q}}(\tau). \]

For later use, the functional forms of \( \beta_{i \text{q}}(\tau) \) and \( \bar{\beta}_{i \text{q}}(\tau) \) of some typical scheduling algorithms are listed below [13]:

1) Deficit Round Robin (DRR)

\[ \beta_{i \text{q}}(\tau) = \max \left\{ c(\tau - \frac{3 - 2c}{c} F), 0 \right\}, \quad \bar{\beta}_{i \text{q}}(\tau) = c\tau, \]

\[ F \overset{\text{def}}{=} L_{\text{qs}} I + L_{\text{be}}. \]

2) Generalized Processor Sharing (GPS)

\[ \beta_{i \text{q}}(\tau) = c\tau, \quad \bar{\beta}_{i \text{q}}(\tau) = C\tau. \]

3) Packetized Generalized Processor Sharing (PGPS)

\[ \beta_{i \text{q}}(\tau) = \max \left\{ c(\tau - \frac{L_{\text{qs}}}{c} + L_{\text{max}} C), 0 \right\}, \quad \bar{\beta}_{i \text{q}}(\tau) = C\tau. \]

B. Analysis

In this subsection, we derive the maximum end-to-end queuing delay and the bound of complementary cumulative distribution function (CCDF) of end-to-end queuing delay of a QoS-aware flow.
Let $D_{sv}$ denote the end-to-end queuing delay of a QoS-aware flow and $d_{sv}^{\max}$ denote its maximum. Define
\[
\beta_{sv}^{(N)}(\tau) \overset{\text{def}}{=} \underbrace{\beta_{sv} \ast \cdots \ast \beta_{sv}}_{N}(\tau),
\]
where
\[
\beta_1 \ast \beta_2(t) \overset{\text{def}}{=} \inf_{0 \leq s \leq t} [\beta_1(s) + \beta_2(t - s)].
\]
From a viewpoint of a specific QoS-aware flow, a series of queues of Model I is equivalent to a single-stage queue whose minimum and maximum service curve are respectively given by $\beta_{sv}^{(N)}(\tau)$ and $\beta(\tau)$ [14]. Thus, we can readily evaluate $d_{sv}^{\max}$ and the bound of CCDF of $D_{sv}$ considering this single-stage queue. We summarize the results below:

- **[Maximum end-to-end queueing delay]**
  \[
  d_{sv}^{\max} = \beta_{sv}^{(N-1)}(\sup_{t \geq 0}[\alpha(t) - \beta_{sv}^{(N)}(t)]).
  \]

- **[Bound of CCDF of end-to-end queueing delay]**
  \[
  P[D_{sv} > d] \leq \beta_{sv}^{(N)}(d) = \sup_{k \geq 1} \left( \frac{1 - q_{sv}(k,T,d)}{1 + q_{sv}(k,T,d)} \right),
  \]
  where
  \[
  q_{sv}(k,T,d) = \sum_{t=1}^{k_{sv}^{\max}(T,d)} \frac{1 - y_{sv}(k,T,d)}{1 + y_{sv}(k,T,d)},
  \]
  \[
  k_{sv}^{\max}(T,d) = \sup\{k: \alpha(kT) > \beta_{sv}^{(N)}((k-1)T) + \beta_{sv}^{(N)}(d)\},
  \]
  \[
  y_{sv}(k,T,d) = \frac{\beta_{sv}^{(N)}((k-1)T) + \beta_{sv}^{(N)}(d)}{\alpha(kT)},
  \]
  \[
  \rho = \lim_{t \to \infty} \frac{\alpha(t)}{t}.
  \]

IV. DELAY CHARACTERISTICS OF DIFFSERV MODEL

A. Network model

Next we analyze delay characteristics of DiffServ-based networks using two network models, which are respectively shown in Figs. 2 and 3. (QoS-aware flows are assumed to be classified into “expedited forwarding” class.) The network model shown in Fig. 2 (Model II) is a series of $N$ nodes, where QoS-aware and best-effort traffic flows are multiplexed together. Each node has two buffers: one is for QoS-aware traffic flows and the other is for best-effort traffic flows. Each node assigns a minimum service rate equal to $cl$ to the aggregation of QoS-aware flows through some scheduling mechanism. As the IntServ network model (Model I), the number of QoS-aware flows is $l$ and QoS-aware flows have the same subadditive envelope $\alpha(\tau)$.

The network model of Fig. 3 (Model III) focuses on a specific QoS-aware flow. In this model, only one QoS-aware flow (QoS-aware flow 1) passes through all nodes. In each node, this QoS-aware flow shares the buffer with different $(I - 1)$ QoS-aware flows. For example, in node 1, QoS-aware flow 1 shares the buffer with QoS-aware flows numbered from 2 to $I$ and, in node 2, shares the buffer with QoS-aware flows numbered from $I + 2$ to $2I$. For simplicity, all QoS-aware flows except QoS-aware flow 1 pass through only one node. For example, packets of QoS-aware flows numbered from 2 to $I$ enter the network at node 1 and leave the network after service completion at node 1. In both models, the amounts of traffic from different QoS-aware flows are assumed to be statistically independent from each other.

For both models, let $B^{(n)}(t, t + \tau)$ denote the amount of traffic of QoS-aware flows that can be served at node $n$ during $(t, t + \tau)$. As in the IntServ network model, $B^{(n)}(t, t + \tau)$ has the following deterministic upper and lower bounds.

\[
\beta_{sv}(\tau) \leq B^{(n)}(t, t + \tau) \leq \tilde{\beta}_{sv}(\tau).
\]

For later use, we show the functional forms of $\beta_{sv}(\tau)$ and $\tilde{\beta}_{sv}(\tau)$ for some typical scheduling algorithms [13]:

1. **(1) DRR:**
   \[
   \beta_{sv}(\tau) = \max \left\{ cl(\tau - \frac{3 - 2cl}{C} F), 0 \right\},
   \tilde{\beta}_{sv}(\tau) = clF, \quad F \overset{\text{def}}{=} L_{sv}^{\min} + L_{be}.
   \]
2. **(2) GPS:**
   \[
   \beta_{sv}(\tau) = cl\tau, \quad \tilde{\beta}_{sv}(\tau) = C\tau.
   \]
3. **(3) PGPS:**
   \[
   \beta_{sv}(\tau) = \left( cl(\tau - \frac{L_{sv}^{\min}}{cl} + \frac{L_{max}}{C} F) \right), \quad \tilde{\beta}_{sv}(\tau) = C\tau.
   \]

B. Analysis

1) **Model II:** Denote by $D_{sv}$ the end-to-end queuing delay of QoS-aware flow 1 and by $d_{sv}^{\max}$ its maximum. Define
\[
\beta_{sv}^{(N)}(\tau) \overset{\text{def}}{=} \underbrace{\beta_{sv} \ast \cdots \ast \beta_{sv}}_{N}(\tau).
\]

By a similar consideration with Model I, from a viewpoint of QoS-aware traffic flows, the series of queues in Model II is found to be equivalent to the single-stage queue whose minimum and maximum service curves are respectively given by $\beta_{sv}^{(N)}(\tau)$ and $\tilde{\beta}_{sv}(\tau)$. Thus, $d_{sv}^{\max}$ and the bound of CCDF of $D_{sv}$ are readily evaluated based on the single-stage queue. We summarize the results below:

---

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.
[Maximum end-to-end queueing delay]
\[ d_{max}^{dv} = \beta_{dv}^{N-1} \left( \sup_{\tau \geq 0} [I(a(t) - \beta_{dv}^{N}(t))] \right). \]

[Bound of CCDF of end-to-end queueing delay]
\[
P[D_{dv} > d] \leq P_{bound}^{dv}(d) = \sum_{k=1}^{k_{dv}(T,d)} \left\{ \left( \frac{x(k, T)}{y_{dv}(k, T, d)} \right)^{1-y_{dv}(k, T, d)} \times \frac{1-x(k, T)}{1-y_{dv}(k, T, d)} \right\}^{l-1},
\]
where \( x(k, T) \) is defined as in (3) and
\[
k_{dv}(T, d) = \sup_{\tau \geq 0} [I(a(kT) > \beta_{dv}^{N}(k(k-1)T) + \beta_{dv}^{N}(d))],
\]
\[
y_{dv}(k, T, d) = \frac{\beta_{dv}^{N}(k(k-1)T) + \beta_{dv}^{N}(d)}{I(a(kT))}.
\]

2) Model III: In contrast to Model II, Model III does not have any equivalent single-stage queueing models. Thus, we need to directly analyze the queueing delay at each node by applying the analysis explained in Section II and, based on the results, estimate the end-to-end delay characteristics. Denote by \( D_{dv}^{(n)} \) the queueing delay of QoS-aware flow \( I \) at node \( n \) and by \( d_{max}^{dv(n)} \) its maximum. We iteratively define the followings:
\[
z^{(1)} = \beta_{dv}^{-1}(\sup_{\tau \geq 0} [I(a(t) - \beta_{dv}(t))]),
\]
\[
z^{(n)} = \beta_{dv}^{-1}(\sup_{\tau \geq 0} [I(a(t) - \beta_{dv}(t)) + \sum_{m=1}^{n-1} \alpha(z^{(m)})]) \quad (n \geq 2),
\]
With the above definitions, we can easily show that \( d_{max}^{dv(n)} \leq z^{(n)} \). Thus, the end-to-end queueing delay is bounded by \( \sum_{k=1}^{N} z^{(k)} \) from above. If \( D_{dv}^{(k)} \) is statistically independent from each other, then we have the following bound:
\[
P[D_{dv}^{(1)} + \cdots + D_{dv}^{(N)} > d] \leq N P_{bound}^{dv}(d/N),
\]
where
\[
P_{bound}^{dv}(d) = \sum_{k=1}^{k_{dv}(T,d)} \left\{ \left( \frac{x(k, T)}{y_{dv}(k, T, d)} \right)^{1-y_{dv}(k, T, d)} \times \frac{1-x(k, T)}{1-y_{dv}(k, T, d)} \right\}^{l-1},
\]
In the above equation, \( k_{dv}(T, d) \), \( x(k, T) \), and \( y_{dv}(k, T, d) \) are defined in the similar manner as those in (3), and
\[
\bar{x}(k, T) = \frac{\alpha(kT)}{\alpha(kT) + \sum_{m=1}^{N-1} \alpha(z^{(m)})},
\]
\[
\bar{y}_{dv}(k, T, d) = \frac{\beta_{dv}(k(k-1)T) + \beta_{dv}(d)}{\alpha(kT) + \sum_{m=1}^{N-1} \alpha(z^{(m)})}.
\]

V. Delay performance comparison between IntServ- and DiffServ-based networks

A. Comparison between Models I and II
One may conjecture that, thanks to the statistical multiplexing effect, the maximum and \( x \)-percentile end-to-end queueing delay in DiffServ model (Model I) should be smaller than that in IntServ model (Model II). This conjecture is verified by the following theorem. For simplicity, we assume that both models use the same scheduling algorithm. It easy to see that the following theorem holds [6]:

Theorem I: For any scheduling algorithm among DRR, GPS, PGPS, we have
\[
d_{max}^{dv} \leq d_{max}^{dv(n)}, \quad P_{bound}^{dv}(d) \leq P_{bound}^{dv(n)}.
\]

B. Numerical examples
Next, we numerically compare three network models (Models I, II, and III) in terms of end-to-end queueing delay performance. We assume that a large number of IP-telephony sources coded by G.729 are multiplexed as QoS-aware flows in networks. The bit rate of G.729 coding is 8 kbps. Note that, in G.729 coding, a ten-byte frame is generated every 10 ms during a talkspurt while a two-byte frame is generated every 20 ms during a silence period. Since an IP packet is constructed from two frames of G.729 with the IP/UDP/RTP header during a talkspurt, the total size of IP packet is 60 byte during a talkspurt. The IP packet during a silence period is made of a two-byte frame of G.729 and the IP/UDP/RTP header, so its size is 42 byte. Now, let \( \delta \) denote talkspurt activity. The average bit rate of a single IP-telephony source is then given by \( \rho(\delta) = \delta \times 24 + (1 - \delta) \times 16.8 \) kbps because bit rate in a talkspurt is \( 60 \times 8/20 = 24 \) kbps and that in a silence period is \( 42 \times 8/20 = 16.8 \) kbps. We assume the traffic from a IP-telephony source of G.729 is constrained by a dual token bucket whose peak bit rate is 9.6 kbps, average bit rate is \( \rho(\delta) \), and bucket size is
\[
\sigma = (24 - (\gamma \times 24 + (1 - \gamma) \times 16.8)) \times 10 \text{ kbit}.
\]
The bucket whose size is \( \sigma(\delta) \) can store the data when the talkspurt lasts 10 seconds. Since the talkspurt duration usually ranges from 100 ms to 500 ms, most of IP telephony sources should be transparent to the dual token bucket explained above.

Table I compares the maximum end-to-end queueing delays of IP-telephony sources between three models when \( N = 10 \), \( c = 23 \text{ kbps} \), \( C = 100 \text{ Mbps} \), \( I = 1000 \) and, \( \gamma = 0.4 \). The maximum end-to-end queue delay in Model I is smaller than that in Model III. This result supports the general belief that IntServ is superior to DiffServ in term of hard-QoS guarantee. The maximum end-to-end queueing delay in Model I is, however, larger than that in Model II as indicated by Theorem 1. This implies that applying IntServ architecture does not always yields smaller maximum end-to-end queue delay.

Table II compares 99.9-percentile end-to-end queueing delays of IP-telephony sources between three models. As indicated by Theorem 1, 99.9-percentile end-to-end queueing delay in Model I is larger than that in Model II. In addition to
this, 99.9-percentile end-to-end queueing delays in Model I is also larger than that in Model III. This implies that applying DiffServ architecture makes x-percentile end-to-end queueing delay much smaller than those in an IntServ-based network. In IntServ model (Model I), the maximum end-to-end queueing delay is almost same as 99.9-percentile delay because the CCDF of end-to-end queueing delay steeply decreases around the maximum value. In contrast, in DiffServ models (Models II and III) the maximum end-to-end queueing delay is much larger than 99.9-percentile delay because of the statistical multiplexing effect.

In Figs. 4 and 5, we compare the CCDFs of end-to-end queueing delay in Models I, II, and III when DRR or GPS is used as the scheduling algorithm. These figures directly verify that x-percentile end-to-end queueing delay in DiffServ-based networks (Models II and III) is much smaller than that in IntServ-based networks (Model I).

VI. CONCLUSION

In this paper, we show that DiffServ-based networks could provide lower end-to-end queueing delay than IntServ-based networks. As mentioned in introduction, if both types of networks have the admission control function, this finding implies that DiffServ-based networks would admit more traffic to guarantee the QoS than IntServ-based networks. In a sense, this result is not surprising because the point of IntServ is its ability to enforce a traffic control with a loss of efficiency. We believe that our study would motivate further research for introducing the efficient traffic enforcement mechanism (admission control) in DiffServ-based networks.

REFERENCES