Deterministic Edge-to-Edge Delay Bounds for a Flow under Latency Rate Scheduling in a DiffServ Network*

Geunhyung KIM and Cheeha KIM, Nonmembers

SUMMARY With the occurrence of new applications such as Voice over IP (VoIP) and multimedia conference, there is an ongoing discussion about realizing QoS in the Internet today. Because of its potential scalability in support of QoS guarantees, the Differentiated Service (DiffServ) architecture with aggregate packet scheduling has recently attracted much attention in the networking community as a feasible solution for providing Internet QoS. Thus, it is important to understand delay bound of an individual flow in the DiffServ architecture in order to provide delay-sensitive applications. In this paper, we study, via both analysis and simulation, the deterministic bound on edge-to-edge delay of a flow in a DiffServ network domain with FIFO aggregation and a class-based Latency Rate (LR) server that provides guaranteed performance with rate reservation for a traffic class. We derive edge-to-edge delay bound for a single flow as a function of allocated service rate for a traffic class, token bucket parameters adopted for flows at the network ingress, and information about joining and leaving flows. We compare the obtained delay bound with previous works using analytic results, and then conduct simulation to confirm the results. The derived bound is less than that of previous studies in all cases.

key words: edge-to-edge delay bounds, DiffServ network, LR server

1. Introduction

To provide QoS guarantees in the Internet, the Internet Engineering Task Force (IETF) has considered a number of architectural extensions, such as Integrated Service (IntServ) architecture [1] and Differentiated Service (DiffServ) architecture [2], to the current Internet. Because of its potential scalability in the Internet, the DiffServ architecture has recently drawn much attention to the networking community and is widely accepted as a feasible solution for providing Internet QoS. Recently, several research efforts [3]–[5] have been made to achieve per-flow bandwidth guarantees without per-flow signaling of core routers in the DiffServ network domain. These researches use aggregate bandwidth reservation or anticipated bandwidth reservation along the path from the ingress router to the egress router. There remains a problem concerning whether it is possible to provide edge-to-edge delay guarantees for a single flow in the case where only aggregate scheduling is implemented in the core, while required bandwidth is ensured by explicit aggregate bandwidth reservation. Unfortunately, there is no result about the evaluation of the edge-to-edge delay for a single flow across a network with aggregate scheduling. As an alternative, delay bound can be considered for it.

Several studies [6], [7], [9] have been conducted to investigate the edge-to-edge delay bound for the DiffServ network domain with FIFO aggregation. Charny and Le Boudec [6] derive delay bound for a network employing class-based strict priority (SP) schedulers, based on the worst-case per-hop delay analysis. Zhang et al. [7] re-establish the result of [6] for a network with FIFO aggregation using a recursive relationship for a packet’s arrival time and departure time to derive upper bound on the worst-case edge-to-edge delay. Jiang [9] derives the edge-to-edge delay bound for a network of Guaranteed Rate (GR) servers [8] with FIFO aggregation and with an arbitrary topology. In [9], Jiang considers more general network than the one considered in [6], since many scheduling disciplines proposed in the literature belong to GR server. All these studies focus on the edge-to-edge delay bound for a network with FIFO aggregation and show that maximum network utilization level must be limited to a small fraction of its link capacities to guarantee edge-to-edge delay, since delay bound for a network with FIFO aggregation depends on the maximum network utilization level and maximum hop count in the network. The edge-to-edge delay bounds of these studies are very conservative and rough, since they are obtained without any knowledge of network topology, routing information, and aggregation information.

Chlamtac et al. [10] and Le Boudec et al. [11] analyze edge-to-edge delay and buffer size needed to guarantee loss-free packet delivery in the connection-oriented networks. In both works, edge-to-edge delay bounds rely on the concept of router interference number (RIN) defined in [10]. The RIN of a given path $P$ is defined as the number of other paths that share a sub-path with $P$. The results of both works are limited to the case where all packets and all links have the same size and the same capacity, respectively. In particular, both works assume that individual flows are shaped at the network entry in such a way that the spacing between packets is at least equal to RIN. Therefore, edge-to-edge delay is bounded by the time to transmit a number of packets equal to RIN, if the processing delay for spacing at the network entry is not considered. The deficiency of this approach is to require complex packet spacing at the network entry to adapt to the variation in the number of flows joining on any output link. Moreover, this approach should be extended for the more generic case where the links have different capacities, packets have different sizes, and different traffic classes.
do exist.

Duan et al. [12] consider mainly class-based edge-to-edge delay guaranteed services for admission control under the bandwidth broker. They define two flows: microflow and macroflow. A macroflow is an aggregation of microflows and a microflow is an individual flow. They investigate the impact of dynamic aggregation of microflows at the network entry on class-based edge-to-edge delay and derive its bound under dynamic flow aggregation and the virtual time reference system (VTRS). In VTRS, each packet has to be modified to carry packet virtual time stamps and the core routers should handle these time stamps. However, they do not consider the impact of dynamic aggregation of microflows at core routers.

In this paper, we investigate the impact of dynamic aggregation of flows not only at the network entry but also at core routers on the edge-to-edge delay. Moreover, we deal with the more generic case including packets of different sizes, links of different capacities. To derive edge-to-edge delay bound for a single flow, we combine the notion of Network Calculus [13], a framework for analysis and maintenance of deterministic QoS guarantees in packet switched networks, with routing information at core routers. We derive edge-to-edge delay bound for a single flow that can be applied to arbitrary topology with arrival traffic constraint and show that derived delay bound is tighter than any other results. We believe that the information about joining and leaving flows and the impact of flow aggregation is critical for the tight edge-to-edge delay bound for the target flow. The tight edge-to-edge delay bound is useful to employ admission control for guaranteed service traffic in the DiffServ network domain.

The remainder of this paper is organized as follows. In Sect. 2, we first briefly review the problem of aggregate scheduling. In Sect. 3, we review the definitions of deterministic network calculus, Latency-Rate (LR) server [14], and the relationship between GR server and LR server. We also represent notations, network model, and assumptions used in the rest of the paper. In Sect. 4, we derive deterministic edge-to-edge delay bound for the target flow in a general case. In Sect. 5, a network topology for evaluation is introduced and the analytic and the simulation results for edge-to-edge delay bound for the target flow are presented. Finally, we conclude in Sect. 6.

2. Problem Statement

In the DiffServ network [2], aggregate scheduling is illustrated in Fig. 1. For example, at each output link of every router, flows destined for this link are first aggregated based on the traffic class, and then transmitted. In other words, the same traffic class flows are aggregated, queued in FIFO manner and serviced by a class-based aggregate LR scheduling.

Consequently, flows in the same class sharing a sub-path in the DiffServ network domain must be aggregated and de-aggregated at the start node and the end node of the sub-path, respectively. Figure 2 presents the aggregation nature of flows. It shows that some other flows may join and leave the target flow, of which we want to know the edge-to-edge delay, as in Fig. 2.

Because of the aggregation nature, the profile of an aggregate flow at an output link may be dynamically changed. This dynamic change in the profile of an aggregate flow may cause undesirable effect on the edge-to-edge delay experienced by the target flow [12]. When the potential negative impact of flow aggregation on edge-to-edge delay guaranteed provision is not properly estimated, flow aggregation may cause delay bound violation.

Therefore, to provision per-flow edge-to-edge QoS guarantee in the DiffServ network domain, a limit is required on the number of flows aggregated at each output link. The establishment and tear-down of real-time application sessions lead to aggregation and de-aggregation of flows along the edge-to-edge path. Aggregating several flows in FIFO manner causes an increase in the burstiness of every flow. This increase in the burstiness causes the increase in the edge-to-edge delay of each flow, in turn. Even under the same per-hop behavior (PHB) configuration, the edge-to-edge delay guarantees may be violated according to the degree of aggregation and the routing path.

Namely, since new flow aggregation could affect the edge-to-edge delay of existing flows and the edge-to-edge delay of new flow could be affected by existing flows, the analysis of the impact of an aggregation on the edge-to-edge delay of individual flows is critical to examine whether new flow aggregation violates the edge-to-edge delay bound for existing flows or not.

In the DiffServ network, packet scheduling is done based on traffic class, not individual flows. The DiffServ
architecture is not suitable for hard QoS guarantees without additional mechanisms, such as edge-to-edge signaling or admission control, while it is widely considered as a feasible solution for providing Internet QoS, because of its scalability. In order to determine whether a delay-sensitive service can be supported or not, evaluation of the deterministic edge-to-edge delay or its tight bound for a flow in the DiffServ network with aggregate scheduling is crucial. The impact of dynamic flow aggregation on the edge-to-edge delay must be investigated to obtain the desired results.

3. Network Calculus Preliminaries and Network Model

Before deriving the edge-to-edge delay bound for a single flow, we review some definitions related to deterministic network calculus and represent notations, the network model, and assumptions that will be used throughout the rest of the paper.

3.1 Network Calculus Preliminaries

Network calculus [13] is a framework for analysis of deterministic QoS guarantees in packet switched networks. To provide guaranteed performance to input flow, some specific support from the network is required and the traffic entering into the network needs to be constrained. Typical example of a constraint on input flow is \((\rho, \sigma)\)-constrained, where \(\rho\) is the bucket rate and \(\sigma\) is the burstiness of the flow, defined for the token-bucket scheme. To make an arrival conform to the token-bucket, a token-bucket filter is enforced to each flow at the ingress router. The token-bucket filter is modeled by a two-tuple \((\rho, \sigma)\) where \(\rho\) denotes the bucket rate and \(\sigma\) denotes the burstiness of the flow after filtering at the ingress router.

In addition, to analyze the service provided by the scheduler, LR server [14] was introduced as the model for analysis of traffic scheduling algorithms. LR server is built on the amount of service received by a flow in its busy period, during which the average arrival rate of input flow remains at or above the reserved rate. LR server is characterized by two service parameters for the aggregate flow it serves; latency \(T\) and service rate \(R\). The latency \(T\) of LR server is seen as the worst-case delay experienced by a packet at the server in the busy period. The latency of a particular scheduling algorithm may depend on its internal parameters, its transmission rate on the outgoing link and the allocated bandwidth. The latencies of many well-known scheduling algorithms are summarized in [14],[15].

Most well-known scheduling algorithms, such as Weight Fair Queueing (WFQ), Packet-level Generalized Processor Sharing (PGPS), and the class-based WFQ (CBWFQ), are LR server [15].

In this paper, we assume that every input flow is \((\rho, \sigma)\)-constrained and that class-based scheduler is LR server. Based on the assumption, we derive the edge-to-edge delay bound for a single flow in the DiffServ network domain in the subsequent sections using the notations given in Table 1.

We need to define the maximum service bandwidth at a node on the sub-path for a flow, which has been aggregated with the target flow, as the largest service rate that is allowed for the flow by the node. Note that we are interested in the delay bound of the target flow which can be evaluated under the condition that available bandwidth is consumed first by the bursty input of the aggregated flow. Thus, when the maximum service bandwidth is fully allocated to serve the aggregated flow, the target flow must experience the longest delay. In a formal way, the maximum service bandwidth of aggregated flow \(i\) at node \(k\) on \(p_i\), \(B_{\{k\}i}\) is defined as \(R_{\{k\}i} - \sum_{m \in F_{\{k\}i} \cup \{i\}} (\rho_k) I_{[k=m]}\), where \(I_E\) denotes the indicator function for statement \(E\), whose value is 1 if \(E\) is true or 0 otherwise, and \(\rho_k\) denotes the average arrival rate of flows. The maximum service bandwidth of aggregated flow \(i\) on \(p_i\), \(B_i\) is defined as \(\min_{k \in \mathbb{P}} B_{\{k\}i}\).

3.2 Network Model

We consider a single DiffServ network domain with DiffServ routers, which are output buffered and implement class-based aggregate LR scheduling, such as PGPS or WFQ. The target flow is transmitted along a single path from an ingress router to an egress router, so called the edge-to-edge path. Other flows may join and leave at any node on the edge-to-edge path. The upper part of Fig. 3 shows an example. Note that it is sufficient to consider one traffic class only since some amount of bandwidth is allocated exclusively to each class.

Given two service parameters of LR server and two parameters of an input flow, the worst-case delay bound experienced by the input flow at the server can be obtained [13],[14],[16],[17]. This relationship between service parameters of LR server and the delay bound experienced by the input flow at the server is defined for the case where LR server performs per-flow scheduling. However, since class-based aggregate LR server has the service rate of a traffic class, not a flow, we cannot use directly service parameters

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Notations used in the paper.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
<td>Meaning</td>
</tr>
<tr>
<td>(P)</td>
<td>the set of nodes on the edge-to-edge path of the target flow</td>
</tr>
<tr>
<td>(F_{{k}})</td>
<td>the set of flows aggregated with target flow at node (k)</td>
</tr>
<tr>
<td>(p_i)</td>
<td>the set of nodes on the sub-path of (P), shared by flow (i)</td>
</tr>
<tr>
<td>(n_i)</td>
<td>the entry node of (p_i)</td>
</tr>
<tr>
<td>(R_{{k}i})</td>
<td>the allocated bandwidth for the traffic class at the output link of node (k)</td>
</tr>
<tr>
<td>(T_{{k}i})</td>
<td>the latency caused by scheduler at node (k)</td>
</tr>
<tr>
<td>(B_{{k}i})</td>
<td>the maximum service bandwidth of flow (i) on (p_i)</td>
</tr>
<tr>
<td>(B_{{k}i})</td>
<td>the maximum service bandwidth of flow (i) at node (k) on (p_i)</td>
</tr>
<tr>
<td>(d_{{k}i})</td>
<td>the nodal delay for flow (i) at node (k)</td>
</tr>
<tr>
<td>(D_i)</td>
<td>the edge-to-edge delay for flow (i) on the edge-to-edge path</td>
</tr>
</tbody>
</table>
of class-based aggregate LR server to derive the delay bound at the LR server or the edge-to-edge delay bound for a flow.

In this paper, in order to obtain the edge-to-edge delay bound for a flow in the DiffServ architecture with aggregate scheduling, we define a single virtual domain LR server and associate it with its equivalent service parameters as shown in the lower part of Fig. 3. Now we can apply the relationship above to the resulting virtual domain LR server which gives the same service to the input flow that all class-based aggregate LR servers on the edge-to-edge path give to it. That is, the service rate of the virtual domain LR server is equivalent to the latency which the input flow experiences along the edge-to-edge path, excluding the propagation delay between LR servers.

The virtual domain LR server can be obtained by concatenating LR servers in presence of flow aggregation. If we can evaluate the impact of every flow aggregation on the delay of the target flow, it is easy to find two service parameters, service rate $R_i$ and latency $T_i$, of the virtual domain LR server for flow $i$ which in turn, enable us to derive the edge-to-edge delay bound for flow $i$.

According to the concatenation theorem [13], the concatenation of LR servers, performing per-flow scheduling, can be represented as a single LR server whose service rate is equal to the lowest among the service rates of the servers and latency is equal to the sum of delays experienced by the flow at every server. If class-based scheduling is employed instead of per-flow scheduling, we cannot apply the concatenation theorem directly. To find out the virtual domain LR server for a flow, we have to count the effect of flow aggregation at individual nodes on the flow. In the next section, we present the details in obtaining the tight delay bound.

4. Delay Bound for a Single Flow in the DiffServ Network

In this section, we obtain two service parameters of the virtual domain server for the target flow and derive its edge-to-edge delay bound in the DiffServ network, where multiple $(\rho, \sigma)$-constrained flows aggregate at any node. To analyze the impact of flow aggregation, we classify nodes on the edge-to-edge path into three types as shown in Fig. 4: aggregate node, de-aggregate node and transit node. An aggregate node receives several flows from multiple input links and aggregates them to one output link, a de-aggregate node receives aggregate flow from an input link but de-aggregates them to two or more output links, and a transit node receives flow from an input link and forwards it as it is to an output link.

We derive the delay experienced by a packet of a flow in a node in three types. First, we consider an aggregate node where multiple $(\rho, \sigma)$-constrained flows aggregate. When LR server with rate $R$ and latency $T$ serves two flows aggregated in FIFO manner and flow $i$ is $(\rho_i, \sigma_i)$-constrained, $i = 1, 2$. The service rate and latency of the server for flow $i = 1, 2$ are $R - \rho_i$ and $T + \sigma_i$, respectively [13]. In addition, the delay experienced by a packet of flow $1$ in the node is upper bounded by $T + \frac{\rho_1}{R} + \frac{\sigma_1}{R\rho_1}$ [13]. From this result, we can obtain the delay bound for a flow for the general case of aggregating multiple flows in Proposition 1.

**Proposition 1 (Delay bound for a flow in an aggregate node).** Consider an aggregate node $l$ as in Fig. 4(a), which consists of LR server with rate $R$ and latency $T$ and serves $m$ $(\rho_k, \sigma_k)$-constrained flows. If $\sum_{k=1}^{m} \rho_k < R$, the nodal delay $d_{l}(i)$ experienced by a packet of flow $i$ in the node must be less than $T + \frac{\sigma_1}{R - \sum_{k=1}^{m} \rho_k}$, $\sum_{k=1}^{m} \frac{\sigma_k I_{(k,i)}}{R}$.

**Proof.** The aggregate flow of $m - 1$ flows excluding flow $i$ is constrained by $(\sum_{k=1}^{m} \rho_k)$ and $(\sum_{k=1}^{m} \sigma_k I_{(k,i)})$. Now, we can consider the aggregation of $m$ flows as the aggregation of two flows, such as the aggregate flow of $m-1$ flows and flow $i$. Therefore, we can obtain the service rate and latency of LR server for flow $i$ simply. The service rate and latency for flow $i$ in the node are $R - \sum_{k=1}^{m} \rho_k I_{(k,i)}$ and $T + \sum_{k=1}^{m} \frac{\sigma_k I_{(k,i)}}{R}$, respectively.

From these two service parameters for a flow in an aggregate node and $(\rho, \sigma)$-constrained flow $i$, the delay $d_{l}(i)$ experienced by a packet of the flow $i$ in an aggregated node is upper bounded by

$$d_{l}(i) \leq T + \frac{\sigma_i}{R - \sum_{k=1}^{m} \rho_k I_{(k,i)}} + \sum_{k=1}^{m} \frac{\sigma_k I_{(k,i)}}{R}.\tag{1}$$

Next, we consider a de-aggregate node as in Fig. 4(b). The profile of aggregate flow on the output link along the edge-to-edge path changes, since some flows are forwarded to...
other output links. That is, the service rate of the target flow and the maximum service bandwidth of aggregated flows change in a de-aggregate node. With the service rate of the target flow and the maximum service bandwidth of aggregated flows, we can obtain the delay bound for the target flow in Proposition 2.

**Proposition 2 (Delay bound for a flow in a de-aggregate node).** Consider a de-aggregate node \( l \) as in Fig. 4(b), which consists of LR server with rate \( R \) and latency \( T \) and serves \( n \) \((\rho_k, \sigma_k)\)-constrained flows. If \( \sum_{k=1}^{n} \rho_k < R \), the nodal delay \( d_{\{i|i\}} \) experienced by a packet of flow \( i \) in the node must be less than \( T + \frac{\sum_{k=1}^{n} \sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}}} \), where \( F_i \) is the set of aggregated flows at the de-aggregate node \( l \) and \( F_{\{i\}} \) is the set of aggregated flows at the entry node of \( n_k \).

**Proof.** Given arrival rate of flows consisting of aggregate flow, we can obtain the service rate of flow \( i \) in the node like that in the aggregate node. Hence, the service rate of flow \( i \) in the node is \( R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}} \). The delay experienced by a packet of flow \( i \) by aggregated flows can be defined as \( T + \sum_{k=1}^{n} \frac{\sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}}} \) by using the maximum service bandwidth.

Hence, the delay \( d_{\{i|i\}} \) experienced by a packet of the flow \( i \) in a de-aggregate node is upper bounded by

\[
d_{\{i|i\}} \leq T + \frac{\sum_{k=1}^{n} \sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}}} + \frac{\sum_{k=1}^{n} \sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}}} \rho_i.
\]

\( \square \)

Finally, we consider a transit node as in Fig. 4(c). We can derive the delay bound for a flow in a transit node in Proposition 3.

**Proposition 3 (Delay bound for a flow in a transit node).** Consider a transit node \( l \) as in Fig. 4(b), which consists of LR server with rate \( R \) and latency \( T \), and serves \( m \) \((\rho_k, \sigma_k)\)-constrained flows. If \( \sum_{k=1}^{m} \rho_k < R \), the nodal delay \( d_{\{i|i\}} \) experienced by a packet of flow \( i \) in the node must be less than \( T + \frac{\sum_{k=1}^{m} \sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{m} \rho_k I_{\{k|i\}}} \), where \( F_i \) is the set of aggregated flows at the transit node \( l \) and \( F_{\{i\}} \) is the set of aggregated flows at the entry node of \( n_k \).

**Proof.** Similar to the de-aggregate node, the delay \( d_{\{i|i\}} \) experienced by a packet of the flow \( i \) in a de-aggregate node is upper bounded by

\[
d_{\{i|i\}} \leq T + \frac{\sum_{k=1}^{n} \sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}}} + \frac{\sum_{k=1}^{n} \sigma_k I_{\{k|i\}}} {R - \sum_{k=1}^{n} \rho_k I_{\{k|i\}}} \rho_i.
\]

\( \square \)

So far, we have derived the delay bound for a flow in a node. Now, based on the propositions, we obtain the service parameters of the virtual domain LR server and the edge-to-edge delay bound for the target flow. Consider the target flow that traverses the edge-to-edge path consisting of a sequence of nodes. All flows including the target flow are \((\rho, \sigma)\)-constrained and node \( k \) consists of LR server whose service parameters for a traffic class are \( R_{\{k\}} \) and \( T_{\{k\}} \). At the output link of node \( k \) on the edge-to-edge path, \( \sum_{i} \rho_i \) is less than \( R_{\{k\}} \).

Based on Proposition 1, 2, and 3, the delay \( d_i \) experienced by a packet of the target flow caused by only flow aggregation on the edge-to-edge path is upper bounded by

\[
d_i \leq \sum_{k \in \mathcal{P}} \sum_{i \in \mathcal{I}_{\{k\}}} \frac{\sigma_i I_{\{k|i\}}}{B_{\{k|i\}}}.
\]

The delay bound for the target flow in Eq. (1) is the sum of delay caused by the burstiness of aggregated flows at every node on the edge-to-edge path. However, the delay of the target flow is affected only once by the aggregating operation of the bursty flow at the aggregating node and is not affected afterwards, since the order in which packets in the aggregate flow are served remains the same beyond the aggregating node due to the FIFO based scheduling. Therefore, Eq. (1) can be rewritten as follows:

\[
d_i \leq \sum_{k \in \mathcal{P}} \sum_{i \in \mathcal{I}_{\{k\}}} \frac{\sigma_i I_{\{k|i\}}}{B_{\{k|i\}}}.
\]

From Eq. (2), we obtain tighter delay bound than Eq. (1). Consequently, the latency \( T_i \) of the virtual domain server for the target flow \( i \) is as follows:

\[
T_i = \sum_{k \in \mathcal{P}} \left( T_{\{k\}} + \sum_{i \in \mathcal{I}_{\{k\}}} \frac{\sigma_i I_{\{k|i\}}}{B_{\{k|i\}}} \right).
\]

Since the service rate of the virtual domain server for the target flow is the lowest among the service rates for the target flow at every LR server, the service rate \( R_i \) of the virtual domain server for the target flow \( i \) is

\[
R_i = \min_{k \in \mathcal{P}} \left( R_{\{k\}} - \sum_{i \in \mathcal{I}_{\{k\}}} \rho_i \right).
\]

Based on \( R_i \) and \( T_i \) of the virtual domain server for the \((\rho_i, \sigma_i)\)-constrained target flow, the edge-to-edge delay \( D_i \) for the target flow is upper bounded by

\[
D_i \leq \sum_{k \in \mathcal{P}} \left( T_{\{k\}} + \sum_{i \in \mathcal{I}_{\{k\}}} \frac{\sigma_i I_{\{k|i\}}}{B_{\{k|i\}}} \right).
\]

Equation (3) shows the edge-to-edge delay bound for the target flow in the DiffServ network with aggregate schedulers. This bound considers the impact of dynamic flow aggregation not only at the network entry but also at
core routers.

As a simple example, consider the case as in Fig. 5. Flow \( f_i \) is \((\rho_i, \sigma_i)\)-constrained, node \( k \) consists of LR server characterized by \((R_{ik}, T_{ik})\) and the target flow is \( f_1 \).

According to Proposition 1, 2, and 3, the delay bound for \( f_1 \) caused by \( f_2 \) at node 1 is \( \sigma_2/R_{11} \) and the delay bounds for \( f_1 \) caused by \( f_3 \) at node 2 and node 3 are \( \sigma_3/R_{21} \) and \( \sigma_3/R_{31} \), respectively. Since \( f_3 \) aggregates with \( f_1 \) at node 2 and aggregate flow of \( f_1 \) and \( f_3 \) passes node 3 in the order, we can obtain the tight delay of \( f_1 \) caused by \( f_3 \) on the sub-path, which consists of node 2 and 3. The tight delay of \( f_1 \) caused by \( f_3 \) on the sub-path is \( \sigma_3/B_3 \), where \( B_3 = \text{min}(R_{21}, R_{31}) \). Consequently, the latency \( T_1 \) of the virtual domain LR server for \( f_1 \) is \( \sum_{k=1}^{3} T_{ik} + \frac{\sigma_2}{R_{11}} + \frac{\sigma_3}{B_3} \).

In addition, the service rate \( R_1 \) for \( f_1 \) is \( \text{min}(R_{11} - \rho_2, R_{21} - \rho_3, R_{31} - \rho_3) \). Hence, the edge-to-edge delay \( D_1 \) for \( f_1 \) in Fig. 5 is upper bounded by

\[
D_1 \leq \sum_{k=1}^{3} T_{ik} + \frac{\sigma_2}{R_{11}} + \frac{\sigma_3}{B_3} + \text{min}(R_{11} - \rho_2, R_{21} - \rho_3, R_{31} - \rho_3).
\]

5. Performance Evaluation

In this section, we present the analytic and simulation results of the edge-to-edge delay bound for the target flow in the network topology of Fig. 6. Here, the adopted topology is a linear multi-hop topology widely used in the previous works, as in [18], [19].

In Fig. 6, \( f_i \) is the target flow and the other flows are cross flows. The nodes \( N_1, N_2, \ldots, N_n \) are the routers, which employ a class-based WFQ (CBWFQ) scheduler. For simplicity, we assume that packet size for all flows is fixed to \( L \) and all flows including the target flow have token-bucket parameters \((\rho_i, \sigma_i)\). We also assume that the core routers are allocated for the traffic class at the edge links, whose capacity is \( C_e \), and the the core links, whose capacity is \( C_c \), respectively.

5.1 Analytic Results

This section presents analytic edge-to-edge delay bounds for the target flow in the network topology of Fig. 6. For the analytic comparison, we assume that each link between node \( N_k \) and \( N_{k+1} \) has propagation delay \( d_k \), for \( k = 1, 2, \ldots, n-1 \).

\( A_l \) and \( R_l \) denote the set of all flows constituting the aggregate flow and the allocated capacity for the class at the link \( l \), respectively. Then, the maximum link utilization level \( u \) and the maximum burstiness factor \( b \) for the network are defined as \( \max_f \sum_{l \in A_f} \rho_f/R_l \) and \( \max_f \sum_{l \in A_f} \sigma_f/R_l \), respectively [7].

The edge-to-edge delay derived by Jiang et al. [7] is bounded by a function of network diameter \( h \), maximum hop count of any flow, maximum link utilization level \( u \), and maximum burstiness factor \( b \) as follows:

\[
D \leq h \frac{u b + \zeta}{1 - (h - 1) u}
\]

if \( u \) satisfies the condition \( u < 1/(h - 1) \).

Since the results of Charny and Le Boudec [6] are for the case where the aggregate flow is served at strict priority, the extension of their results for CBWFQ is in Eq. (4).

If \( u < \frac{\text{min}(\Gamma_l)}{R_l} \) where \( \Gamma_l \) is the sum of the bit rate all incoming flows at link \( l \), then

\[
D \leq h \frac{u b + \zeta}{1 - (h - 1) u}
\]

where \( r = \max_f(\Gamma_l/R_l)/(\Gamma_l - u R_l) \), \( \zeta = \max_f[l/R_l + L_{\text{max}}/C_c] \).

Jiang derives the edge-to-edge delay bound for a network of GR servers in [9] as shown in Eq. (5) and his result is similar to Eq. (4). Moreover, Jiang et al. [19] derive another edge-to-edge delay bound for the network of Fig. 6. By applying specific parameter settings for Fig. 6, if the link utilization level \( u \) satisfies

\[
u \leq \frac{R_e + R_c}{(h - 1) R_e + R_c}
\]

then

\[
D \leq h(\frac{rb + E}{1 - (h - 1) u})
\]

where \( r = R_e/(R_e + (1 - \gamma)R_c) \), and \( E = 2(1 - r)L/R_c \).

Since CBWFQ is a class based LR server in the network, the latency term caused by CBWFQ should be considered. The latency of CBWFQ scheduler is \( L/C_e + L/R_e \) [14], [15] and the delay caused by the burstiness of cross flow is \( \sigma_c/R_c \). Hence, the tight edge-to-edge delay of the target flow, excluding the node delay of the last node, in Fig. 6 is upper bounded by (see Eq. (3))
Table 2  Parameter values used in comparison.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$C_e$, $C_c$</th>
<th>$R_e$</th>
<th>$R_c$</th>
<th>$N$</th>
<th>$L$</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>10 Mb/s</td>
<td>10 Mb/s</td>
<td>2 Mb/s</td>
<td>5–10</td>
<td>1 KB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\rho$, $\sigma$, $\rho_c$, $\sigma_c$, $d_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>0.2 Mb/s, 2 KB, 0.2 Mb/s, 2–10 KB, 1 ms</td>
</tr>
</tbody>
</table>

$$D \leq \left(\frac{\sigma_c}{R_c - \rho_c}\right) + \sum_{k=1}^{N} \left[\frac{\sigma_c + L}{R_c} + \frac{L}{C_c}\right] + \sum_{k=1}^{N-1} d_k.$$

The comparison of the edge-to-edge delay bounds is performed using the parameter values in Table 2.

To investigate the impact of burstiness of cross flows on delay of the target flow, we set $h$ and $N$ to 5 and vary the burstiness of cross flows $\sigma_c$ from 1,000 bytes to 10,000 bytes at intervals of 1,000 bytes. The impact of burstiness of cross flows is shown in Fig. 7. All delay bounds increase linearly as the burstiness of cross flows increases. Our derived delay bound increases in proportion to $h/R_c$, while other delay bounds increase in proportion to higher constant values than $h/R_c$. The gap between our bound and others increases especially when the burstiness of cross flows is high and our tight delay bound is more robust to the burstiness of cross flows than others. This result is based on the fact that our delay bound counts the impact of bursty aggregated flow once at the aggregating node while others count it at every node on the sub-path.

In addition, to see the impact of the number of hops on the edge-to-edge path on the delay, we change the number of hops ($N$), the number of nodes that flow aggregation occurs, from 5 to 10 and we set $R_e$ to 4 Mb/s to lessen link utilization and set $\sigma_c$ to 2000 bytes. As shown in Fig. 8, our tight delay bound increases linearly, while others increase more rapidly when the edge-to-edge path of the target flow is getting longer. We can observe that our delay bound affected by the aggregating operation is independent of the number of hops on the path. This is not true for others. This observation is also justified the fact that ours is affected once by the aggregating operation while others are not.

Finally, Fig. 9 shows the delay bounds as the link utilization increases. In the result, our tight delay bound outperforms any other, as the link utilization increases. In the previous works, the edge-to-edge delay bounds are no longer useful, when the link utilization is higher than 30%. Their results show that the link utilization is saturated much earlier since their approaches overestimate the impact of aggregation too much. On the contrary, our tight edge-to-edge delay bound is still applicable regardless of the link utilization.

5.2 Simulation Results

In this section, we conduct simulations to confirm our analytic results. In the simulations, we use the network simulator ns-2 [20] and the network configuration in Fig. 10.

In these simulations, we consider two traffic classes, such as EF traffic class and best-effort traffic class defined in the DiffServ architecture and investigate the impact of flow aggregation of EF traffic class on the delay of the target flow of EF traffic class. The flows generated from $s_i$ and $b_i$; des-tine for $d_i$ and $bd_i$, respectively. The $s_i$ is the traffic source of EF traffic class and the $b_i$ is the traffic source of best-effort traffic class. The flow generated in $s_0$ is the target flow and
Fig. 10  Network configuration for simulation.

Fig. 11  Simulation results of edge-to-edge delay vs. burstiness of cross flows.

all other EF traffic flows are cross flows. All flows enter
the network through the edger router (ER) and the C(i)
the core router. The simulation network is configured using
parameter values in Table 2.

The target flow and the cross flows are generated by
using Pareto On/Off process in which the shape parameter
is set to 1.4, burst time to 900 ms, idle time to 100 ms, and
rate to 10 Mbps. The reason why each Pareto On/Off source
generates relatively higher rate than $\rho_i$ is to generate highly
bursty traffic. Nevertheless, the actual traffic entering
the network is shaped by the token bucket at the ingress edge
router (ER). The best-effort flows are generated by using a
CBR model. The edge-to-edge delay of the target flow is
measured as the packet delay between point 1 and point 2
in Fig. 10, since we consider nodes where flow aggregates
to calculate the edge-to-edge delay in analytical results. All
simulations are conducted in a period of 500 seconds and
performed 20 times for each scenario.

Figure 11 shows the simulation and analytic result of
the edge-to-edge delay as the burstiness of the cross flows
increases from 2,000 bytes to 10,000 bytes at intervals of
2,000 bytes. For all cases, the number of core routers be-
tween point 1 and point 2 is fixed to 5. We obtain
the maximum delay experienced by a packet during the simu-
lation. While the maximum delay increases as the bursti-
ness of cross flows increases, there is the gap between the
analytic result and the maximum simulation delay. This
gap is based on two fact that the case where the burstiness of all
cross flows has an influence on the delay of a specific packet
simultaneously is rare. From the simulation result of Fig. 11,
we can conclude that the maximum simulation delay is al-
ways lower than the edge-to-edge delay bound obtained in
the analysis.

Figure 12 shows the simulation and analytic results as
the number of hops between point 1 and point 2 increases.
In this simulation, the link utilization is set to 0.1 to make
the same environment used for obtaining analytic results.
Similar to the results in Fig. 11, the maximum edge-to-edge
delay is always lower than the edge-to-edge delay bound
obtained in the analysis.

Figure 13 shows the simulation and analytic results as
the link utilization increases by increasing the rate of cross
flows. As shown in Fig. 13, the maximum edge-to-edge
delay is always lower than the edge-to-edge delay bound
obtained in the analysis.

An average delay bound error is 20% when the bursti-
ness of cross flows changes and average delay bound error
is 27% in case when the number of hops changes. However, In terms of link utilization, the delay bound error increases from 23% to 36% (especially 50% at 0.9 link utilization) as link utilization increases.

6. Conclusion

Evaluation of edge-to-edge delay bound for a flow in the DiffServ network with aggregate scheduling is critical to provide delay-sensitive service in the Internet. In this paper, we investigated the impact of dynamic aggregation of flows not only at the network entry but also at core routers on edge-to-edge delay for a flow. Moreover, we derived deterministic edge-to-edge delay bound for a flow that can be applied to arbitrary topology with token-bucket constrained flow in a DiffServ network domain consisting of LR servers, such as CBWFQ. We showed that our derived delay bounds is tighter than any other results by analytic comparison. Especially, while the delay bounds of previous results are no longer useful when the link utilization is higher than 30%, our tight edge-to-edge delay bound is still applicable regardless of the link utilization.

We also showed by simulation that our tight delay bound is confirmed. However, there is a gap between the analytic result and the maximum simulation delay. It implies that the case where the burstiness of all cross flows has an influence on the delay of a specific packet simultaneously is very rare.

Based on analytical comparison, we conclude that if input flows are token-bucket constrained and the route information of flows is known in an arbitrary route system, then tight edge to edge delay bound over other bounds can be secured. In addition, with our tight bound, DiffServ network admits more flows and gives higher network utilization than other previous results. We will extend the applicability of our tight delay bound to dynamic resource provisioning with edge-to-edge path reservation for EF flows in the DiffServ network domain.

References


Cheeha Kim received his B.S. and M.S. degrees in electronic engineering from the Sogang University in 1986 and 1988, respectively, and a Ph.D. degree in computer science and engineering from the POSTECH, Korea in 2005. He is presently working as a senior researcher in Korea Telecom. His research interests include computer communications, mobile computing and QoS provisioning in fixed and mobile networks.

Geunhyung Kim received his B.S. and M.S. degrees in electronic engineering from the So-gang University in 1986 and 1988, respectively, and a Ph.D. degree in computer science and engineering from the POSTECH, Korea in 2005. He is presently working as a senior researcher in Korea Telecom. His research interests include ubiquitous networks, wireless sensor networks, computer communications, mobile computing, distributed systems and performance evaluation.