End-to-End Absolute Differentiated Services for Real-Time Traffic

Jinmei Yang, Changcheng Huang
System and Computer Engineering, Carleton University at Ottawa, Canada
Ottawa, ON, Canada K1S 5B6
{jinmei_y, huang}@sce.carleton.ca

Abstract — For many real-time applications (i.e., VoIP), the absolute quality of service (QoS) requirements should be guaranteed. In this paper, we propose a novel dynamic congestion control framework for scalable support of absolute end-to-end QoS in differentiated services (DiffServ) networks. With dynamic measured feedback control and traffic regulation at edges and relative DiffServ support at core routers, the proposed framework provides quantitative per-class deadline violation probability (DVP) guarantee. More specifically, we extend relative DVP from a single node to an end-to-end path by developing a bound for end-to-end relative DVP. In order to support the framework better, we present a unified scheduler, the Dynamic Deficit Round Robin (DDRR) scheduler, which combines generalized processor sharing (GPS) with relative DiffServ. Simulations are carried out to illustrate the performance of our proposed framework and scheduler under both single-node and multi-node cases.

Keywords — QoS, DiffServ, relative DVP, absolute DVP, DDRR scheduler, traffic engineering, end-to-end

I. INTRODUCTION

To satisfy a diverse set of Quality of Service (QoS) requirements, IETF has proposed Differentiated Services (DiffServ) [1]. DiffServ model supports service differentiation for class-based traffic in a scalable manner. Absolute and relative differentiated services are two approaches in DiffServ model. Absolute DiffServ aims at providing IntServ type of end-to-end absolute performance at a granularity of class level in the network core [2]. Relative DiffServ aims at providing relative services [3]. In this model, the performance of a higher priority class is better than those of a lower priority class. Let the performance metric of class $i$ be denoted by $q_i$, the relative differentiation between class $i$ and class $j$ can be expressed as $q_i/q_j = r$.

Many real-time applications such as Voice over IP (VoIP) require that a packet is delivered within a predefined end-to-end deadline. Packets delivered beyond these end-to-end deadlines are considered useless. In order to have an end-to-end delay guarantee, real-time applications rely on absolute differentiated services. The challenge for DiffServ to provide absolute end-to-end service is to develop solutions that can deliver suitable network control mechanisms with scalable and efficient network state management.

In this paper, we propose a dynamic congestion control framework in order to provide absolute end-to-end differentiated services. The framework is the combination of relative DiffServ mechanism in the network core and traffic regulation at edges. With dynamic measurement-based feedback control and traffic shaping at edges, the proposed framework implements quantitative per-class deadline violation probability (DVP) guarantee. The relative DiffServ in the network core makes our approach different from other works, in which we can guarantee all classes QoS performance by maintaining only one or several classes’ state information. This reduces overhead at edges and leaves the network core as simple as possible. In order to support the framework efficiently, we also introduce a unified scheduler, the Dynamic Deficit Round Robin (DDRR) scheduler, which combines generalized processor
sharing (GPS) with relative differentiated services. The DDRR scheduler achieves relative DVP among different classes at each core node. Based on the single node result, a bound for the end-to-end proportional DVP ratio is developed.

The rest of the paper is organized as follows: In section II, we describe some related research work. In section III, we introduce the DDRR scheduler; develop a bound for end-to-end proportional DVP and propose the framework for converting relative DVP to absolute DVP. In section IV, we illustrate the simulation results and the performance analysis. Finally, section V summarizes this study.

II. RELATED WORK

The relative differentiation model can only offer the relative service among different classes. Most works so far end with studying the single node case [3] [4] [5]. This is clearly not good enough. The extension from a single node to an end-to-end path is definitely necessary but certainly more challengeable. For some specific applications, an end-to-end absolute QoS is required in order to meet the stringent delay and loss requirements [6]. In [7] and [8], users have to adjust dynamically the class of a flow in order to match its end-to-end absolute QoS requirements. Nevertheless, under heavy load condition of any flow, the absolute performance of all classes will become worse simultaneously even they are still proportional. When users with the smallest delay requirements are unsatisfied in the highest class, both approaches could not provide a satisfactory solution. Furthermore, changing the class of a traffic flow can introduce out of order packets.

To satisfy end-to-end absolute QoS requirements, we need traffic engineering. Traffic engineering is a difficult task due to the dynamic nature of Internet traffic [9]. Some examples are the range-based model [10] and the hose model [11]. From these models we can see that, intentional over-provisioning is typical in real practice to guarantee a certain level of QoS. In [12], in order to support absolute delay bounds, traffic aggregates of each class are dimensioned at edges by a distributed control mechanism that employ statistics of the entire traffic. In practice, full knowledge of the traffic traversing a network is generally not available. The approach presented in [13] maintains the control states of core routers by a complex bandwidth broker and perform admission control for the whole network. It can support strict QoS guarantee but also lead to more complexity.

Our approach, with measurement-based traffic control at edges and relative DiffServ algorithm inside the network core, bears similarity to the work in [12] but differs in that we provide traffic regulation only for one class rather than all classes discussed in [12]. The QoS requirements for the other classes can be implemented by the relative differentiation in the network core without complex centralized core algorithms in [12]. This allows high utilization can be achieved without the cost of traffic engineering.

Some schedulers have been proposed for realizing relative differentiation of delay, loss and DVP in literature. Link sharing schedulers dynamically change the service rates allocated to classes, such as BPR [3] for relative delay and JoBS [5] for relative loss. Alternatively, priority-based schedulers schedule packets according to their priority, such as WTP [3] and MDP [7] for relative delay, PLR [14] for relative loss, and WEDD [4] for relative DVP.

An important advantage of the link sharing over the priority-based schedulers is to guarantee some minimum bandwidth for each class so that they are not completely distorted by some badly behaved classes. Suppose one real-time traffic class is extremely overloaded under no traffic engineering condition. All other real-time classes under the priority-based scheduler such as WEDD maybe affected so much that they receive too little bandwidth to be useful while trying to maintain certain relative QoS. In the extreme case, some kind of starvation may happen.
In this paper, we propose DDRR, a link sharing scheduler for the network core. It can provide real time traffic with the proportional DVP differentiation and guarantee a minimum bandwidth for each class. Although BPR and JoBS can satisfy a minimum bandwidth requirement, they are trying to achieve relative ratios of delay or loss, not our relative DVP differentiation. We will describe our algorithm in more detail in section III.

III. FROM RELATIVE DIFFERENTIAITON TO ABSOLUTE QOS

In this section, we first describe the DDRR scheduler, then develop the bound for end-to-end relative DVP, at last present our framework to guarantee end-to-end absolute DVP.

3.1 The proposed scheduler: DDRR

Our goal is to provide the proportional DVP among classes in a single node. Fig. 1 shows the system architecture. Traffic flows from different sources are classified into classes by a classifier. We assume that the buffer of each class is infinite in a router. In our model, we first estimate the absolute DVP of each class by the ratio of departure packets with deadline violation and total departure packets, and then calculate the relative DVP ratios between classes. These results will be fed into the weight adjustment process to adjust the weight of each class dynamically based on the difference of the achieved DVP ratio and the target value.

There are different ways to measure the absolute DVP. Here we employ exponential averaging algorithm. For class $i$, assuming $DVP_{mea}$ is the DVP value in a measurement interval, $DVP_{last}$ is the averaged DVP value before the current measurement interval, the average DVP is estimated as: $DVP = (1 - \rho) * DVP_{last} + \rho * DVP_{mea}$. We set the measurement interval same as the weight adjustment interval. $\rho$ is the weight of the measurement interval DVP value in the DVP calculation. As $\rho$ is increased, it will make the averaging process more adaptive to load changes and as $\rho$ is decreased, it will give a smoother average by keeping a longer history.

Depending on the difference of the achieved DVP ratio and the target value, we adjust the weight of each class in turn by a ratio. The round number $R$ in round robin algorithms is the DVP measurement interval and also as the weight adjustment interval. A larger $R$ will keep the weight more steady and make DVP ratio less synchronized with the target ratio. Weight is updated more frequently given a smaller $R$, and this will increase the complexity. We can also configure each class with a minimum bandwidth guarantee. When the weight adjustment reach a point that the class is receiving bandwidth less than or equal to the guaranteed minimum bandwidth, the adjustment stops. The computational complexity of DDRR depends on the number of traffic classes. In DiffServ networks, traffic flows have been grouped to a very limited number of classes. Therefore, the complexity of DDRR is not a real issue.
3.2 End-to-end relative DVP bound analysis

One key issue for any DVP based algorithms is the setting of the delay deadline. The issue of how to split an end-to-end delay deadline into delay deadlines at each hop is an interesting and complex problem, but it is beyond the scope of this work. In this paper, we assume a simple and conservative approach: an equal split across the hops. This approach is clearly sufficient to guarantee meeting an end-to-end delay deadline and it is conservative because not all the packets that can meet the end-to-end deadline satisfy each node deadline.

Assume that two flows from class \( j \) and \( k \) enter a network and they traverse the same path \( l \). Let \( p_{j}^{l,i} \) denote the absolute DVP of class \( j \) traffic at node \( i \) on path \( l \), \( r_{j,k}^{l} \) denote the relative DVP ratio of class \( k \) to class \( j \) at node \( i \), so \( r_{j,k}^{l,i} = p_{k}^{l,i} / p_{j}^{l,i} \). \( P_{j}^{l,n} \) and \( R_{j,k}^{l,n} \) are the corresponding end-to-end values on path \( l \) with \( n \) nodes and \( R_{j,k}^{l,n} = p_{k}^{l,n} / p_{j}^{l,n} \). Assuming \( p_{k}^{l,i} < p_{j}^{l,i} \) and \( 0 < r_{j,k}^{l,i} < 1 \), we say that class \( k \) has higher priority than class \( j \). The relative DVP ratios at different nodes may be different. In following paragraphs, we address the issue that, given the relative DVP ratio \( r_{j,k}^{l,i} \) at node \( i \), how to calculate the end-to-end relative DVP ratio \( R_{j,k}^{l,n} \).

**Theorem 1:** Assume the events of delay deadline violation are independent across a network. Given two flows from class \( j \) and \( k \) traversing a same path \( l \), the end-to-end relative DVP ratio \( R_{j,k}^{l,n} \) of the specific path \( l \) with \( n \) nodes is bounded by

\[
\begin{align*}
\text{Min} (r_{j,k}^{l,1}, r_{j,k}^{l,2}, \ldots, r_{j,k}^{l,n}) & \leq R_{j,k}^{l,n} \\
& \leq (\sum_{i=1}^{n} (r_{j,k}^{l,i}) - \sum_{i>d}^{n} (r_{j,k}^{l,i} r_{j,k}^{l,i}) + \sum_{i>d}^{n} (r_{j,k}^{l,i} r_{j,k}^{l,i}) - \cdots + (-1)^{n-1} (r_{j,k}^{l,1} r_{j,k}^{l,2} \cdots r_{j,k}^{l,n})) \quad \text{(1)}
\end{align*}
\]

**Proof:**
Given the relative DVP \( r_{j,k}^{l,i} \) at each node, theorem 1 gives the upper and lower bound of the end-to-end relative DVP for an end-to-end path. To prove these results, we first try to get the expression of the end-to-end absolute DVP of each class; then deduce the end-to-end relative DVP from the end-to-end absolute DVP.

Given that all deadline violation events are independent across a network, the end-to-end absolute DVP of class \( j \) traffic \( p_{j} \) can be expressed as:

\[
P_{j}^{l,n} = 1 - \prod_{i} (1 - p_{j}^{l,i}) \quad \text{(2)}
\]

Equation (2) makes no assumption about the topology or traffic pattern of the given network. Therefore, it is quite general. Here we consider only two classes of traffic: class \( j \) and class \( k \) traversing a network following the same path. Fig.2 shows an example network with two nodes. Assume class \( k \) has higher priority than class \( j \), i.e. \( p_{k}^{l,i} < p_{j}^{l,i} \) and \( 0 < r_{j,k}^{l,i} < 1 \).

By Equation (2), the end-to-end relative DVP ratio for the two-node network is,

\[
R_{j,k}^{l,2} = \frac{p_{k}^{l,2}}{p_{j}^{l,2}} = \frac{p_{k}^{l,2} + p_{k}^{l,B} - p_{j}^{l,2} p_{j}^{l,B}}{p_{j}^{l,2} + p_{j}^{l,B} - p_{j}^{l,2} p_{j}^{l,B}} = \frac{r_{j,k}^{l,2} p_{j}^{l,2} + p_{j}^{l,B} - p_{k}^{l,2} r_{j,k}^{l,2} p_{j}^{l,B}}{p_{j}^{l,2} + p_{j}^{l,B} - p_{j}^{l,2} p_{j}^{l,B}}
\]

![Fig.2. The model for end-to-end relative DVP](image-url)
\[
R_{j,k}^{l,n} = \frac{a r_{j,k}^A + b r_{j,k}^B - r_{j,k}^A r_{j,k}^B}{a + b} = r_{j,k}^B + \frac{b(r_{j,k}^A - r_{j,k}^B) + r_{j,k}^A(1 - r_{j,k}^B)}{a + b - 1}
\]  

(3)

\[
= r_{j,k}^B + \frac{a(r_{j,k}^A - r_{j,k}^B) + r_{j,k}^A(1 - r_{j,k}^A)}{a + b - 1}
\]  

(4)

Considering \( r_{j,k}^B > r_{j,k}^A \), \( r_{j,k}^B = r_{j,k}^A \), and \( r_{j,k}^A < r_{j,k}^B \) conditions respectively, from (3) and (4), we have:

\[
\text{Min}(r_{j,k}^A, r_{j,k}^B) \leq R_{j,k}^{l,n} \leq (r_{j,k}^A + r_{j,k}^B - r_{j,k}^A r_{j,k}^B).
\]

(5)

With mathematical induction, we can extend (5) from two nodes to \( n \) nodes, and derive the equation (1). Hence we prove Theorem 1.

The assumption in Theorem 1 is reasonable. It is pointed out in [15] that the single hop delays incurred by a packet are independent random variables, particularly when link utilizations are high, with ON/OFF traffic sources traversing 5 hops in length. Similarly, the independence assumption on the link delays has been referred in [16]. In general, with the occurrence of events of deadline violation, the network will have relatively heavy traffic. Hence, the above conclusion can be applied into our assumption on end-to-end relative DVP.

From Theorem 1, after configuring the single node DVP ratio \( r_{j,k}^i \), we can derive the bound on the end-to-end DVP ratio \( R_{j,k}^{l,n} \). Fig. 3 shows the end-to-end relative DVP bounds for paths with 2, 3 and 4 nodes respectively under each node with equal relative DVP ratios special case. As is increased \( n \), the bounds seem to become looser.

An interesting and very useful special case is that, if we configure all \( r_{j,k}^i \) for the same class \( j \) and \( k \) to be the same across all nodes in the network, the actual values are typically very close to the lower bound, i.e. \( R_{j,k}^{l,n} \approx r_{j,k}^i \).

**Corollary 1:** Assume \( a \) and \( b \gg 1 \) in equation (3), if all \( r_{j,k}^i \) for the same class \( j \) and \( k \) across all nodes on path \( l \) are the same, i.e. \( r_{j,k}^i = r \), then the end-to-end relative DVP ratio \( R_{j,k}^{l,n} \) of the specific path \( l \) is \( R_{j,k}^{l,n} \approx r_{j,k}^i = r \).

Fig. 3 The end-to-end relative DVP bounds for paths with 2, 3 and 4 nodes respectively under each node with equal relative DVP ratios \( r_{j,k}^i = r \) and \( n=2 \) \( p_{j,k}^i = 10\% \) special case
Proof: Consider an example path with two nodes (more general case can be proved using induction), from equation (3), we have
\[
R_{j,k}^{i,2} = \frac{ar + br - r \ast r}{a + b - 1} = \frac{r(1 - r)}{a + b - 1}
\]  
(6)

The absolute error from the lower bound will be
\[
\Delta = R_{j,k}^{i,2} - r = \frac{r(1 - r)}{a + b - 1}
\]  
(7)

We define variables \(a\) and \(b\) as the reciprocals of the absolute DVP. Typically, the absolute DVP is much smaller than 1, so we have \(a \text{ and } b \gg 1\), also \(0 < r < 1\), then the absolute error \(\Delta \ll r\). Therefore, we have \(R_{j,k}^{i,2} \approx r_{j,k}^{i} = r\).

For example, assume that the absolute DVP \(p_{j}^{i}\) of each node is 10% and \(a = b = 10\) then \(r_{e} \in (0, 0.013)\) as shown in Figure 3.

Corollary 1 with the equal configuration of relative ratio for all nodes is important and complementary to Theorem 1. The results derived from Corollary 1 are much tighter compared with equation 1 in Theorem 1 even when the number of hops is large. Hence, we can apply the Corollary 1 directly to the real network. Therefore, we recommend the ratios \(r_{j,k}^{i}\) for the same class \(j\) and \(k\) should be set to the same value across a network wherever it is feasible.

3.3 A framework: from the relative DVP to the absolute DVP

In our framework, we implement the traffic shaping and measurement of the end-to-end absolute DVP at edge nodes. We use leaky buckets for traffic shaping. The DDRR scheduler is used to implement relative differentiation at core nodes. The basic operation performed at edge nodes in our framework is to adjust the token rate for class \(i\) based on the feedback of the measured end-to-end absolute DVP of a base class, until the end-to-end absolute DVP of the base class meets its target DVP. We mark the packets with deadline violation at each internal node and calculate the end-to-end absolute DVP by a ratio of marked packets to the total packets at egress nodes. Because a relative end-to-end QoS can be achieved among different classes using DDRR scheduler as discussed in the last section, if the base class can meet its target absolute DVP, all other classes can meet too as long as the ratios are designed based on the ratios of their absolute QoS.

It is important to note that typically we do not need shaping all traffic classes for meeting QoS requirements under our proposed framework. We can choose adjusting token rates of either one (typically the base class) or several traffic classes (typically with lower priorities) depending on whether the target absolute DVP for the base class is easy to achieve or not (when it is the case of high priority overloading, we also need to shape the high priority traffic). Shaping lower traffic class is a good choice because it has the minimum impact on revenue. Usually there is a large mount of lower class traffic in the network. Achieving absolute QoS by shaping only the lower priority traffic is one of the benefits of our proposed framework.

Choosing the base class and measuring its absolute end-to-end DVP is the critical part of our proposed framework. The traffic of the base class should travel the same path as other classes in order to maintain the relative QoS. If the various classes traffic at an ingress node travels to different destinations, a base class flow for each destination is necessary. In the worst case, a mesh of all ingress nodes to all egress nodes may be necessary. This seems to raise scalability issue. However, we argue that this is not a big problem for the following reasons:

1. The way we measure the DVP of the base class is very similar to the way a TCP session reacts to congestion. Today millions of TCP sessions run over the Internet at the same time without big scalability problem due to its intelligence at edge policy. Our measurements are also conducted at the edges of a network. This allows it to scale to a large network.

2. Our measurements are only conducted for the base class traffic. We can choose the base
class to be the lowest traffic class to minimize the impact on the QoS seen by end users. If no such low priority traffic flow exists for a particular path, we can inject a small amount of low traffic into the path we are interested simply for the measurement purpose. This certainly can cause a little bit overhead in terms of utilization. However, compared with the over-provisioning of the traditional traffic engineering approach, this overhead is a very small cost to pay simply because the overhead is in low priority.

Whether we use low priority user traffic or injected low priority test traffic as the base class, the measurement results at the egress nodes need to be sent back to the corresponding ingress nodes. The ingress nodes then make decisions on how to shape ingress traffic to maintain absolute end-to-end QoS.

IV. PERFORMANCE EVALUATION

In this section, we start with evaluating the effectiveness of the proposed DDRR scheduler for a single node network. After applying it to a multi-node network, we illustrate the achieved end-to-end relative DVP results and compare them with our analytical bounds. The efficiency of the proposed framework, in terms of the end-to-end absolute DVP, is then assessed. All the simulations are executed in OPNET 8.0 platform.

4.1 DDRR scheduler at the single node

The traffic source in our simulation is aggregate ON/OFF traffic. Each user has exponential on and off periods. Average ON and OFF periods are 100 ms and 158 ms respectively. The simulated topology is a single bottleneck link with capacity OC-3 155 Mbps. Packet distribution is a Poisson distribution with the mean size 1024 bits. The delay bound of packets is 10ms.

As we mentioned above, our GPS-based DDRR can provide each class with the minimum bandwidth share and priority-based WEDD have the starvation problem when one class is overloaded. In this scenario, class 1 generates 150 Mbps traffic and class 2 is 15 Mbps. The traffic load ratio is 10:1. The target DVP differentiation ratio is 2. In DDRR, when we adjust the weight distributed to each class, we set the minimum bandwidth 1/10 of total link bandwidth. As shown in Fig. 4(a), when class 1 is overloaded, it results in the starvation of class 2 in WEDD. The measured utilization for class 1 is almost 100% and class 2 is starved after t = 12s. On the other hand, in DDRR, the minimum utilization for each class can be guaranteed to at least 1/10 of total capacity in Fig. 4(b) and the achieved DVP ratio is approximate to the target ratio.

4.2 End-to-end relative DVP ratio

In order to evaluate the end-to-end relative DVP, we consider the scenario with two tandem

![Fig.4. Comparison of utilization under WEDD and DDRR](image-url)
nodes as shown in Fig. 5. All traffic sources are similar to the single node case. We set sources 1 and 2 as having 20 users with 7.95 packets/ms sending rate in ON periods, and sources 3, 4, 5, and 6 as having 10 users with 0.88 packets/ms sending rate. Sources 1, 3, and 5 have class 1 and sources 2, 4, and 6 have class 2 traffic. Sources 1 and 2 traffic traverse node 1 and node 2 to destination 1, sources 3 and 4 to destination 2, sources 5 and 6 to destination 1. The link capacity is 155 Mbps. The link utilization of node 1 to node 2 and node 2 to destination 1 are 85% and 90% respectively.

The goal of the simulation is to measure the end-to-end relative DVP given each node target relative DVP ratio, and compare with our analytical bound. We measure the achieved relative DVP ratio at each node and the end-to-end relative DVP ratio at destination 1 respectively. The target ratio is the DVP differentiation parameter of class 2 to class 1 traffic. Class 2 priority is set higher than class 1. The results are summarized in table 1. The analytical bounds are listed in the same table. As shown in table 1, the achieved end-to-end relative DVP ratios are consistent with our analytical bounds in all five cases. According to Corollary 1, under the special cases of equal node relative DVP configuration (0.75, 0.5, 0.25, 0.125 for both nodes respectively); the end-to-end relative DVP can achieve the similar results (0.7409, 0.4929, 0.242, 0.121 for ETE).

4.3 End-to-end absolute DVP

To guarantee the end-to-end absolute DVP, we consider the scenario similar to the relative DVP scenario in Fig. 5 except we add a leaky bucket at source 1. The bucket depth is 50 packets. The token rate is adjusted dynamically. We set the class 1 as Poisson traffic with average interarrival time 0.015 ms and class 2 is ON/OFF traffic. The link utilizations of node 1 to node 2 and node 2 to destination 1 are 90% and 92% respectively. The other simulation setup is the same as the relative DVP scenario.

<table>
<thead>
<tr>
<th>Case</th>
<th>Target DVP ratio</th>
<th>Achieved DVP ratio</th>
<th>ETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.495</td>
<td>0.4929</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>0.125</td>
<td>0.125</td>
<td>0.125</td>
</tr>
<tr>
<td></td>
<td>0.125</td>
<td>0.1310</td>
<td>0.121</td>
</tr>
<tr>
<td>5</td>
<td>0.5</td>
<td>0.125</td>
<td>0.125−0.5625</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.1329</td>
<td>0.408</td>
</tr>
</tbody>
</table>
Table 2 The end-to-end absolute DVP

<table>
<thead>
<tr>
<th></th>
<th>Node 1</th>
<th>Node 2</th>
<th>ETE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Without leaky bucket</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target DVP ratio</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Achieved DVP ratio</td>
<td>0.5031</td>
<td>0.5024</td>
<td>0.5073</td>
</tr>
<tr>
<td>Class 1 absolute DVP</td>
<td>0.0948</td>
<td>0.0811</td>
<td>0.1565</td>
</tr>
<tr>
<td>Class 2 absolute DVP</td>
<td>0.0477</td>
<td>0.0307</td>
<td>0.0794</td>
</tr>
<tr>
<td><strong>With leaky bucket</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target DVP ratio</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Achieved DVP ratio</td>
<td>0.5036</td>
<td>0.5032</td>
<td>0.5055</td>
</tr>
<tr>
<td>Class 1 absolute DVP</td>
<td>0.0415</td>
<td>0.0311</td>
<td>0.0720</td>
</tr>
<tr>
<td>Class 2 absolute DVP</td>
<td>0.0209</td>
<td>0.0156</td>
<td>0.0364</td>
</tr>
</tbody>
</table>

The goal of this simulation is to find out, how to adjust the token rate in order to meet the end-to-end absolute DVP requirement on the node relative differentiation basis. We first measure the end-to-end absolute DVP of class 1 traffic at destination 1, then create a probe packet with this information and send it to the source node. The target end-to-end absolute DVP of class 1 is set to 7.5%. The DVP differentiation parameter of node 1 and node 2 is configured to 0.5. According to Corollary 1, the end-to-end relative DVP ratio should approximate to 0.5.

The achieved results are summarized in Table 2. Without the leaky bucket to control the class 1 traffic access rate, the end-to-end absolute DVP of class 1 and class 2 can achieve 15.65% and 7.94% respectively. After using the proposed framework, the end-to-end absolute DVP of class 1 traffic is 7.2%, approximately equal to our target value. The absolute DVP of class 2 is 3.64%, which is very close to its target value. Fig.6 shows the sample path of the achieved end-to-end absolute DVP. At \( t = 70 \) second, we turn on the leaky bucket control mechanism in Figure 6(b). We can see that the simulation results behave as we expect. That is, when we shape class 1 traffic, the absolute QoS of both classes improve at the same time while maintaining the relative QoS between the two classes unchanged.

V. CONCLUSIONS

In this paper, we have presented a novel dynamic congestion control framework for providing absolute end-to-end differentiated service for real-time applications. By maintaining the state information for only one class, this approach reduces overhead at network and hence, is scalable to large systems. A general end-to-end relative DVP ratio bound is developed given the relative DVP ratio of all nodes in an end-to-end path. According to Theorem 1, the end-to-end relative DVP ratio is dependent on the relative DVP ratios of all nodes on the path. When we configure the relative DVP ratio of all nodes equally, we can get a much tighter bound for the end-to-end DVP ratio.

Fig.6. Providing the end-to-end absolute DVP
In future work, the research will be focused on fairness issues. The shapers in the framework are provided based on per source and per destination. When egress nodes feedback the results to the different sources contributing to congestion, a fair solution is needed to regulate the shapers and further allocate resources to each user according to the demands.

REFERENCES


