Configuration Rule and Performance Evaluation for DiffServ Parameters

Shohei Sato*, Kazutomo Kobayashi*, Huanxu Pan*

Sandra Tartarelli, Albert Banchs****

***NEC Corporation**

****NEC Europe Ltd.**

Abstract

The Internet is now widely expected to become an important communication infrastructure of the society, and therefore it is no longer sufficient to simply be able to provide connections. A higher quality of service (QoS) in communications is increasingly being required. As a new framework for providing QoS services, DiffServ is undergoing a speedy standardization process at the IETF. DiffServ not only can offer tiered level of services, but can also provide guaranteed QoS to a certain extent. In this paper, we examine a single DiffServ node model which utilizes token bucket as the policing mechanism, and propose a way of configuring various control parameters in order to accommodate various UDP/TCP traffic. Then, through simulation, we evaluate the throughput characteristics for various cases when TCP and UDP are mixed or separated in different queues, and study the appropriateness of the configurations. The results indicate that the throughputs can be maintained as specified in the Service Level Agreements (SLA) over a considerably wide range of bucket sizes and discard thresholds in the QoS control mechanism.

1. Introduction

The current Internet is basically a "best effort" type of network. As the Internet becomes a communications infrastructure of the society, it is increasingly essential to minimize the impact on the society of any deterioration in the QoS, due to failure, traffic congestion and so on. As a promising mechanism for controlling QoS, DiffServ is undergoing a speedy standardization process at the IETF [1]-[3]. Evaluations of traffic characteristics in DiffServ or related networks have been reported in the literature [4]-[6]. However, to our best knowledge, it has not been systematically discussed as how to configure the various control parameters in a DiffServ implementation.

In this paper, we take up the subject of configuring DiffServ. We propose a practical way of operating a DiffServ network, along with a concrete method for configuring the corresponding control parameters, and evaluate the resultant traffic characteristics based on simulation results. The actual communications network dimensioning rule/configuration rule is determined not only through a study based on the traffic theory, but also by other factors such as an adequate QoS from the users' point of view, the cost for achieving it and the overall economic efficiency. Therefore, engineering experience does play an important role. There have been, however, few real-world examples of applying traffic engineering in the Internet QoS.

In this paper, we attempt to explore appropriate configuration rules for DiffServ networks with a relatively simple model. DiffServ is basically a mechanism for offering tiered level of services, and there are concerns whether it can really provide guaranteed QoS as well. We shall show, however, that DiffServ, when configured appropriately, is able to provide guaranteed QoS to a large extent. In Section 2, we describe the network model considered in this paper, including the general QoS framework with DiffServ and the policing mechanism utilizing "token bucket". Discussions are also given in this section on how the Service Level Agreements (SLA) between the network and users may look like. We will present the method of configuring parameters for the Assured Forwarding (AF) class in Section 3, and for the Expedited Forwarding (EF) class in Section 4. In Section 5, simulation results are presented, particularly on the performance of TCP traffic in the AF class, which is difficult to evaluate analytically. The results show that the throughput performance is maintained as specified in the SLA with our configurations being applied. Section 6 concludes the paper.

2. Network configuration, SLA and policing mechanism

In order to guarantee QoS of a certain level, the traffic load on each part of the network (link, node) through which the traffic passes must be restricted by a certain level. To satisfy this, in this paper we will take up the case where Service Level Agreements (SLA) are contracted between the user and the network which provides DiffServ (hereinafter referred to as "DiffServ domain") concerning EF class and AF class traffic, and the DiffServ domain (Fig.1) has a mechanism to check those SLA at ingress edge routers. SLAs are specified on each user's traffic of each QoS class, and between each pair of ingress and egress nodes. Precisely speaking, the traffic of User "i", having Class "j" and heading from ingress router "k" toward egress router "l", is expressed as "traffic ijkl," and the corresponding SLA is denoted as "SLA_{ijkl}." On the

Internet today, the method generally used for determining the route for traffic between ingress router "i" and egress router "j" is based on routing protocol such as OSPF. However, with that method, proper allocation of network resources to traffic demand cannot be accomplished and it is difficult to guarantee a certain level of QoS. We presume Static Routing in this paper, where the QoS server designates the routes in the DiffServ domain based on the traffic demand indicated in the SLA and allocates those routes to a Label Switched Path (LSP) of Multi Protocol Label Switching (MPLS).

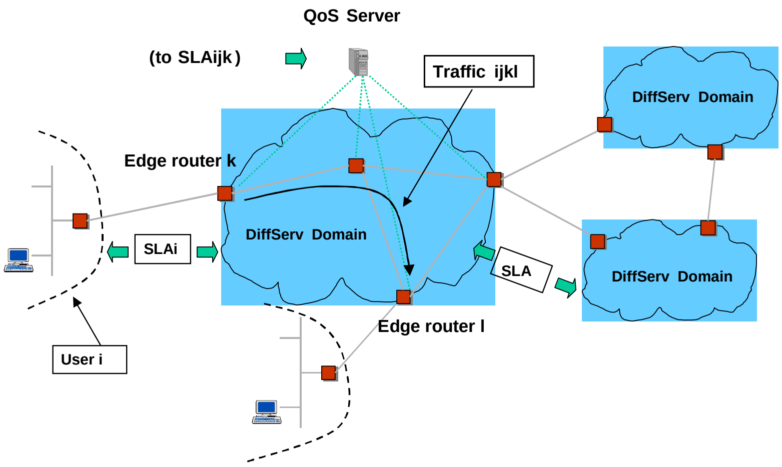


Figure 1 DiffServ Network

In our paper, we will discuss the configuration of parameters and the performance based on those parameters for a queue at a single node (ingress router). One LSP is allocated to a "jkl." We presume Fig.2 as the DiffServ queue configuration at the ingress router. All traffic with SLA is assigned to a class, EF (Class 0),or AF 1-4 (Class 1-4), and is policed by "token bucket" for the SLA.

EF class is a low jitter, low delay class and it is given the highest priority level (Priority Q). The AF class queue is only forwarded when no packets exist in the EF queue, and the forwarding priority among AF classes is provided by means of WRR.

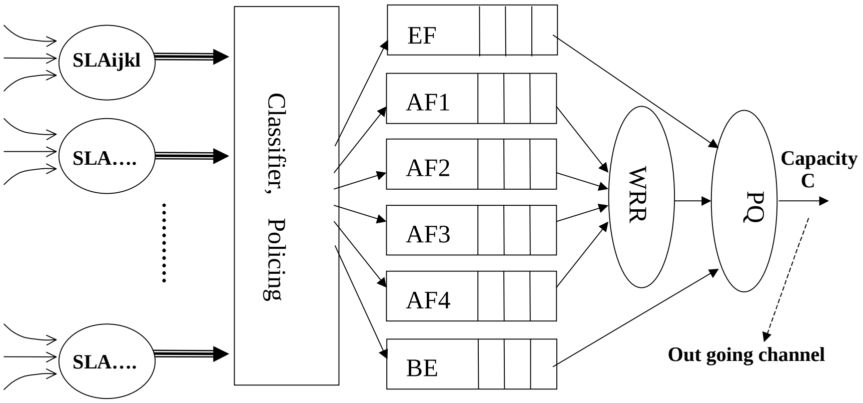


Figure 2 Single Node Model

[Configuration Rule 1]

The allocation of UDP, TCP traffic to AF queue is as follows:

- *Separate SLAs are contracted for UDP and TCP. The UDP traffic and TCP traffic are each assigned to separate AF queues. These are called, respectively, UDP-only case and TCP-only case.*
- *In the case that a user cannot or does not wish to have separate SLAs contracted for UDP and TCP, a mixed UDP+TCP SLA may be contracted (UDP+TCP mixed case). The mixed traffic is then assigned to another dedicated AF queue.*

When the traffic out of a SLA arrives to the EF or AF1 to AF4 queue, the input rate is checked based on the policing mechanism shown in Fig.3. A "Peak Information Rate" (PIR) is defined for EF SLAs and a "Committed Throughput" (CTH) for AF SLAs. Due to the reasons to be explained later, the relationship between the SLA defining parameters PIR, CTH and "Token Rate" , and "Allocated Information Rate" AIR are shown in Table 1 .

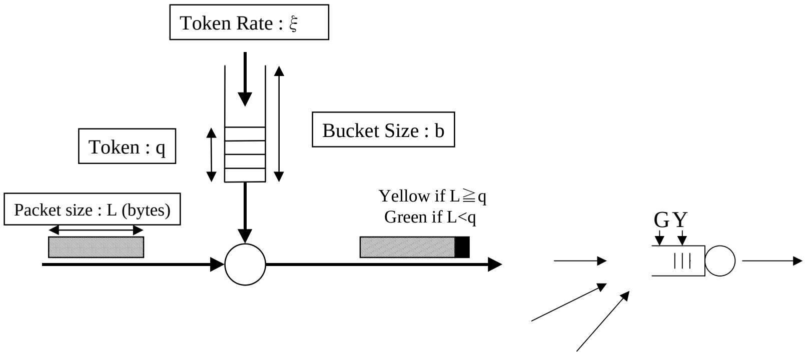


Figure 3 Policing Mechanism (for each SLA)

Table 1 SLA,AIR and Token Rate

		SLA	AIR	Token Rate
EF		PIR	PIR	PIR
AF	UDP-only case	CTH	CTH/ α (α =0.9)	CTH
	TCP-only case	CTH	CTH/ α (α =0.75)	CTH/ α (α =0.75)
	UDP+TCP mixed case	CTH	CTH/ α (α =0.75)	CTH/ α (α =0.75)

3. Assured Forwarding (AF) class

Traffic is assigned to AF class under the following policy. It is well known that when UDP traffic and TCP traffic are assigned to the same queue, throughput fairness cannot be well preserved. Therefore, in case it is possible to contract separate SLAs for UDP and TCP, each traffic shall be assigned to its own separate AF queue (UDP-only case and TCP-only case). Depending on the user, separation may not be possible and in that case, its UDP and TCP traffic will be mixed together in another AF queue (UDP+TCP mixed case). When both UDP and TCP traffic exist in the same AF queue, throughput unfairness between TCP and UDP within the same SLA may be unavoidable as mentioned above, however we must try to prevent serious unfairness from occurring between different SLAs.

3.1. UDP-only case

In this section, first of all we take up policing parameters for an AF queue (AF_j) to which only UDP traffic is to be allocated. "Committed Throughput" (CTH) is defined as the SLA parameter. In this case, incoming packet flow exceeding CTH will be permitted, however, that portion shall be forwarded as "best effort" and the PIR will not be included in the SLA. The CTH is checked by the policing mechanism indicated in Fig. 3. The "Token Rate" is set to the same as CTH (that is, ξ = CTH) and the packet exceeding this rate is marked as "yellow." All other packets are marked as "green." Yellow packets (green packets) shall be discarded at a probability 1, in case the instantaneous queue length at the time of the packet arrival exceeds the threshold value B_{yj} (B_{gj}) (Fig. 4). Here, we have omitted the suffixes indicating ingress node and egress node, and denote an AIR with an SLA of User "i" belonging Class "j" as "AIR_{ij}." The WRR weight W_j that corresponds to Class j (j= 1-4) shall be set as follows:

$$W_j = \frac{\sum_i AIR_{ij}}{\sum_{i,j} AIR_{ij}} \cdots\cdots(1)$$

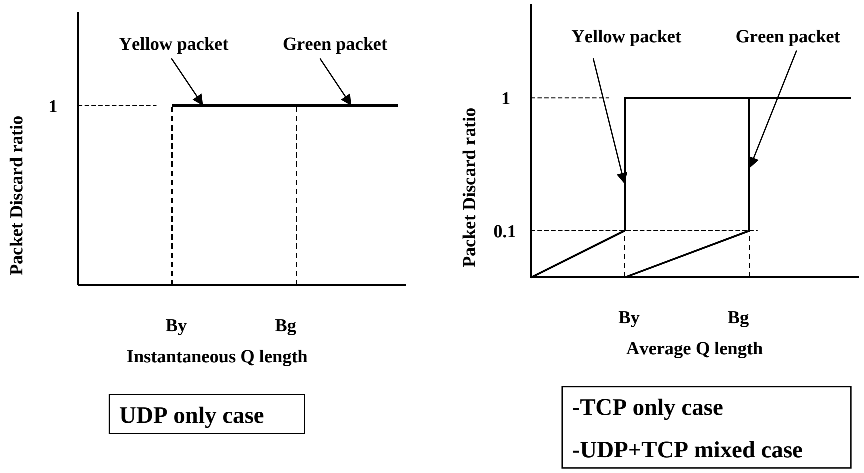


Fig.4 Packet Discard Ratio

UDP shall be applied to applications requiring strict delay, and to ensure the maximum delay time at the queue to be below a certain value, the bucket size b_{ij} shall be determined as follows.

Setting the output channel capacity at "C" as in Fig. 2, and prescribing $C_j=W_j$ (C- $\Sigma_i AIR_{i0}$),

$$\sum_{i,j} \mathrm{AIR}_{ij} \le \mathrm{C}_j \cdots\cdots(2) \qquad\qquad \sum_{i} \mathrm{b}_{ij} + \mathrm{B}_{yj} \le \mathrm{B}_{gj} \cdots\cdots(3)$$

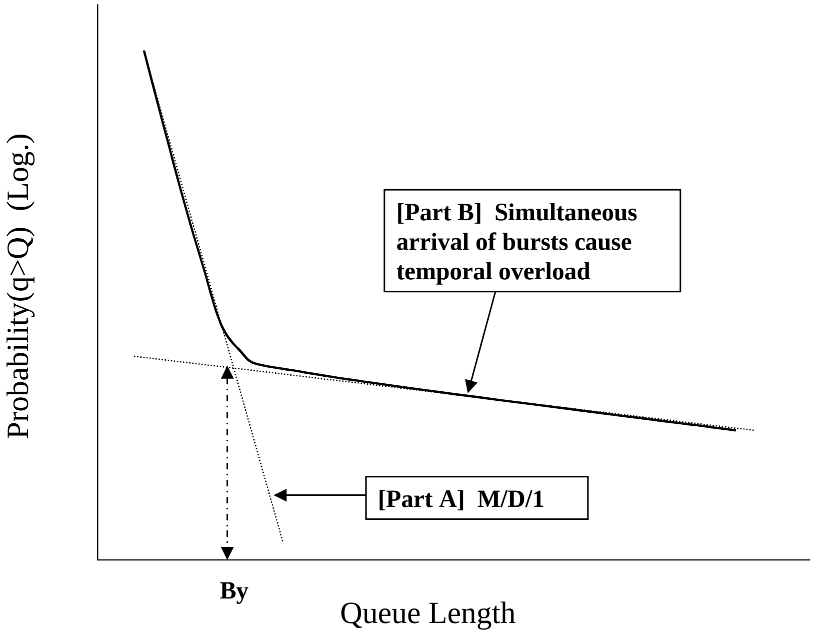
Note that it is possible to ensure no loss of green packets if the parameters satisfy above conditions. Next, if the ratio of b_{ij}/AIR_{ij} is set equally to the fixed value dj which is determined for this class (that is, $d_j = b_{ij}/AIR_{ij}$), then you will note that the maximum delay of this class dmaxj can be confined by the following [7]:

$$\mathrm{dmax}_{\mathrm{j}} \le \frac{\sum_i \mathrm{b}_{ij} + \mathrm{B}_{yj}}{\mathrm{C}_{\mathrm{j}}} \le \frac{\sum_i \mathrm{b}_{ij}}{\sum_i \mathrm{AIR}_{ij}} + \frac{\mathrm{B}_{yj}}{\mathrm{C}_{\mathrm{j}}} = \mathrm{d}j + \frac{\mathrm{B}_{yj}}{\mathrm{C}_{\mathrm{j}}} \cdots\cdots(4)$$

If the maximum permitted delay of Class j "Dmaxj" (e.g. 10ms, 100ms) is set, and half of the value is allocated to d_j, b_{ij} can be determined from these conditions and $AIR_{ij.}$ Fixed value d_j not only reflects the maximum delay time, but also reflects the CTH policing time scale. From the standpoint of maximum delay time, the smallest possible value is advantageous for d_j, however, considering policing time scale , too small d_j leads overly sensitive judgment and this will become a weak point. Accordingly, it is not good for this value to be excessively small, and it shall be set to a level which corresponds to the maximum delay time permissible (that is, $d_j=Dmax_j/2$).

B_{yj} is determined by allocating half of the remaining maximum permissible delay in the 2nd term of above inequality, however in that case, as C_j becomes larger, there is the possibility that B_{yj} will become too large. Another method to determine the upper limit of B_{yj} is to consider the discard ratio of yellow packets. According to study on ATM traffic [8] [9], in the case where incoming bursty traffic flows like ON-OFF model are multiplexed, it is known that the complementary distribution of the queue length is composed (as shown in Fig. 5) of the M/D/1 portion (Part A) and the portion which corresponds to the event that the flow rate temporarily exceeds the server capacity due to overlapping bursts (Part B). Part B, speaking for the case of this paper, corresponds to the portion of the incoming flow that exceeds CTH, and packet discarding at this portion does not cause much troubles. Also, as you will discern from Fig. 5, even if the buffer size increases, Part B's discard ratio will not proportionately decrease. Considering the above, the threshold value for yellow packet discard shall be set so as to achieve a sufficiently low level packet discard ratio in Part A's domain. . The average input rate is CTH, but in order to have a sufficiently low packet discard ratio even when the incoming packet rate fluctuate randomly, the allocated bandwidth AIR should be set such that the CTH is 90% of it (AIR=CTH/0.9). The packet length distribution will not be easily identified as it depends on the application, etc. For simplicity, if we assume it to be exponential distribution (M/M/1) and the discard ratio is set to 10^{-5}, the required buffer size will be approximately 100 packets (That is, B_y=100 packet).

Figure 5 Queue distribution in ATM multiplexer



[Configuration Rule 2]

In conclusion: the allocated bandwidth in the UDP-only case is AIR=CTH/0.9. When the maximum permitted delay for that class is set to D_{maxj} (e.g. 10ms, 100ms), the bucket size shall be set to b= D_{maxj}AIR/2. The token rate ξ is ξ=CTH. B_y=min(D_{maxj} C_j/2, 100packets). B_g is set based on inequality (3).

3.2. TCP-only case

The weight derived from inequality (1) shall be allocated to the AF queue that will accommodate TCP. α =0.75 shall be assigned in the case of TCP (AIR=CTH/0.75). The reason for this is as follows. In congestion control of TCP, if there is no packet loss during transmission, the window size gradually increases, and when a non-consecutive packet loss occurs, the window size is halved (Reno version of TCP). From that point, the window size is again gradually expanded. A yellow mark will be attached to the incoming packet when the flow rate exceeds the token rate ξ allocated to that flow, and a marked packet may get lost causing another window size shrinking. Because of this, the incoming flow rate is presumed to oscillate between the allocated bandwidth and half of that at a cycle of several ten times the round trip time (RTT). Therefore, if we consider a time period longer than the oscillation cycle, the average throughput should be about 0.75 ξ . Thus the token rate should be set such that CTH=0.75 ξ , and in addition, the bandwidth to be allocated to this class shall be set according to the token rate (That is, AIR= ξ). In the UDP-only case, it was CTH = ξ , however, the difference here arises from the fact that the congestion control mechanism is implemented in TCP.

Whether yellow packets are discarded or not is determined based on the average queue length at the time of arrival and the discard ratio curve shown in Fig. 4. "Bucket size" b_{ij} is used for the purpose of deciding by what approximate time scale should be made on whether or not the incoming flow rate has exceeded the allocated bandwidth. It has been found that TCP flow throughput, if not particularly controlled, is inversely proportional to RTT [10]. Accordingly, in case bucket size is set to a value which is extremely large compared to RTT (e.g. a value corresponding to 1 second), it will not be possible to eliminate throughput unfairness caused by the difference in RTT. If the bucket size is excessively small, the bucket will too easily become empty, causing too frequent yellow-markings and packet losses, even before it has reached the window size set at that TCP session. RTT fluctuates in every session, so it is not feasible to set one RTT as a premise and determine the most suitable value in advance. In this paper, we will perform a simulation using bucket sizes in a range corresponding to 20ms to 250ms and compare the performance characteristics. This range was decided considering the RTT range. RTT=250ms comes from the assumption of a total delay time of 50ms at routers, etc. plus 200ms of propagation time for a round trip over 20,000 Km of optical fiber, and is estimated to be practically the maximum case of RTT. Conversely, RTT= 20ms is estimated to be the minimum RTT level. The next step is to determine B_y and B_g. B_y is set using the same methodology as the UDP-only case at B_y= 100packet, and B_g is likewise set using inequality (3). We will perform a simulation varying the B_y value, and compare the throughput characteristics.

3.3. UDP+TCP mixed case

In this case, we assume a situation in which SLAs cannot be separately contracted for UDP and TCP. Which method should be used for configuring DiffServ parameters, UDP-only or TCP-only method? In the mixed case, confining the UDP delay time below a certain value as in the UDP-only case can not be the objective. Rather, the aim here is to be able to guarantee throughput for the entire SLA, that is to say, the same target as the TCP-only case. Therefore, we will examine the characteristics based on a simulation applying the same configuration method as for the TCP-only case.

4. Expedited Forwarding (EF) Class

The SLA for EF traffic is defined using Peak Information Rate (PIR). When incoming flow exceeds PIR, the packets will be discarded by the policing mechanism (Fig.3). When determining the token bucket size b_{i0}, it is necessary to take into account the fact that the arrival rate of incoming packet will instantaneously exceed PIR due to fluctuation of the transfer time up to arrival at the ingress router. When determining that specific value, influence of the delay fluctuation value of incoming traffic flow should be evaluated considering the ingress router's position on the network. Generally speaking, a few packets should be appropriate.

Packets (yellow) which are judged to have exceeded PIR will be discarded, so B_y= 0. If B_g is set such that it satisfies the conditions of equation (3), green packets will not be discarded. Also, the maximum delay time $dmax_0$ can be guaranteed by the inequality shown below, in the same manner as (4)

$$\mathrm{dmax}_{\mathrm{j}} \leq \frac{\sum_{i} \mathrm{b}_{ij}}{\mathrm{C}_{\mathrm{j}}} \cdots\cdots (5)$$

5. Simulation and Discussion

In this section, we will conduct simulation on two cases, TCP-only and UDP+TCP mixed case. For the former case, we will verify the appropriate bucket size value for policing. For the latter, we will investigate the impact of interference between the UDP+TCP SLAs. The TCP utilized is the Reno version. The Packet size is 1Kbyte.

5.1. TCP-only Case (Fig.6)

In this case, there are two SLAs, one having a CTH of 1Mbps and one with a CTH of 5Mbps. Each SLA contains three flow's with different RTT's, that is 20ms, 100ms and 250 ms respectively. We are conducting this simulation on three cases of dj which defines bucket size, 20ms, 100ms and 250ms, equivalent to the assumed RTTs, and on three cases of B_y, 50packets, 100 packets and 200 packets. The throughput achieved is indicated in Table 2.

Viewing the result, a throughput of approximately 1.3 to 1.6 times of the committed throughput is achieved for all of the 9 combinations of d_j and B_y. Since AIR=CTH/0.75, the realized throughput corresponds to 0.98 to 1.2 times of the AIR. We observe that the situation is more advantageous to SLAs with a smaller CTH (1Mbps) than to those with a larger CTH (5Mbps), but all the throughput requirements are satisfied.

Regarding the achievement of throughput of 1.3 to 1.6 times of the CTH, viewing this alone, it may be perceived as over-provisioning. When we consider a network with many hops, however, a packet marked yellow could be dropped at any of the hops along its path, which may not be a too small probability if any part of the network is moderately or heavily congested. Since any drop of a TCP packet will cause a slowdown in its flow rate, this “over-provisioning” may become necessary. A more quantitative study on this issue is needed.

In all cases, no discard of green packets is observed. The figures shown in [ ] type brackets in Table 2 represent the ratio of maximum throughput to minimum throughput of the three flows included in one SLA. The fairness control between flows within one SLA is outside of the control mechanism presumed in this paper, but we have indicated these ratios for information. That ratio decreases as the B_y increases because the packet queueing delay becomes larger due to By's expansion, and consequently impacts more on the flows with a smaller RTT as the queueing delay contributes more significantly to the entire RTT for these flows.

Apart from this ratio, concerning the d_j and B_y values, no remarkable variation in performance characteristics was observed in the range of this simulation, and we believe that the values set were appropriate in the case of this scope.

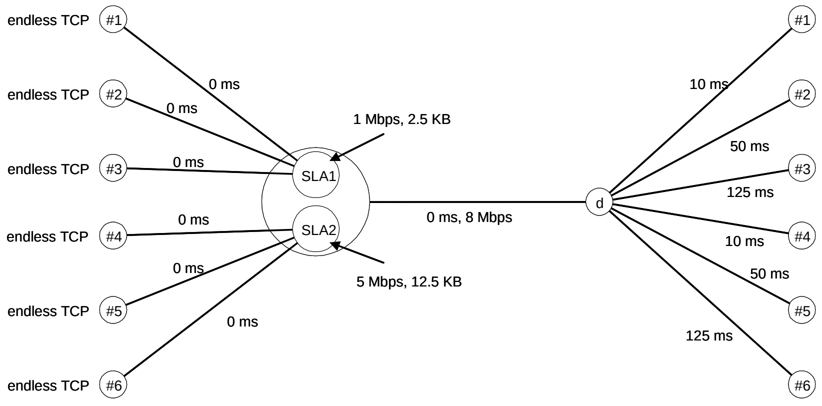


Figure 6 Simulation model (TCP-only case)

Table 2 Achieved Throughput - TCP-only case

dj (ms)	CTH (Mbps)	By(packets)		
		50	100	200
20	1	1.44Mbps(1.44)* [4.5]**	1.50Mbps(1.50) [3.9]	1.53Mbps(1.53) [3.3]
	5	6.46Mbps(1.29) [5.9]	6.46Mbps(1.29) [5.2]	6.47Mbps(1.29) [3.7]
100	1	1.43Mbps(1.43) [4.9]	1.48Mbps(1.48) [3.9]	1.59Mbps(1.59) [3.5]
	5	6.43Mbps(1.29) [6.6]	6.49Mbps(1.30) [5.2]	6.40Mbps(1.28) [4.3]
250	1	1.42Mbps(1.42) [4.6]	1.52Mbps(1.52) [4.1]	1.59Mbps(1.59) [3.2]
	5	6.49Mbps(1.30) [6.3]	6.43Mbps(1.28) [4.3]	6.39Mbps(1.28) [4.1]

* ( ) indicates (achieved throughput)/CTH.

**[ ]indicates (max throughput of flow) / (min throughput of flow) in the SLA

[Configuration Rule 3]
In conclusion, in the TCP-only case, the allocated bandwidth is AIR=CTH/0.75. The token rate is ξ =AIR. The bucket size is b=d·AIR (d= 20 to 250ms). By = 100 packets (50 to 200). B_g is set based on equation (3).

5.2. UDP+TCP Mixed Case

We performed simulation on three cases. In all cases, there were two SLAs and of those, one or both had a mix of UDP+TCP. (Refer to Table 3.) The simulation model for Case 1 is shown in Fig. 7. The TCP flows in each SLA are the same as the TCP-only case, with RTT of 20ms, 100ms, 250ms respectively. If the SLA contains UDP, three flows are included in each SLA and each flow outputs data in compliance with "on-off" model. The "on" period follows Pareto distribution (average 50ms) and the "off" period is exponentially distributed (average 50ms). The data output rates in the "on" period are all 1Mbps.

When we observe the result, for any case the achieved throughput is 1.25 to 1.40 times of the CTH and 0.94 to 1.05 times of the AIR (=CTH/0.75), and fairness between SLAs is reasonably kept. In addition, discard of green packets did not occur in any of the cases. As in Table 2, the figures shown in [ ] type of brackets in Table 3 represent the ratio of maximum throughput to minimum throughput of the three TCP flows included in one SLA. They indicate the same trend as that in Table 2 and we believe that it can be explained in the same way.

[Configuration Rule 4]

In conclusion, in the UDP+TCP mixed case, AIR, ξ, b, B_y, B_g are set by applying the same rules as for the TCP-only case.

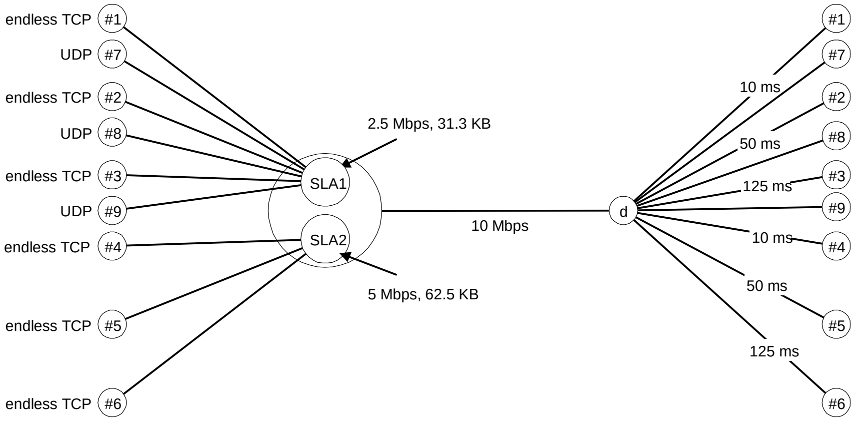


Figure 7 Simulation model (UDP+TCP mixed case)

Table 3 Achieved Throughput - UDP+TCP mixed case

	CTH (Mbps)	By(packets)		
		50	100	200
Case 1	2.5(UDP+TCP)	3.49Mbps(1.39)* [5.8]**	3.45Mbps(1.38) [3.9]	3.46Mbps(1.38) [2.9]
	5(TCP)	6.28Mbps(1.26) [9.3]	6.42Mbps(1.28) [5.8]	6.53Mbps(1.31) [3.5]
Case 2	1(TCP)	1.43Mbps(1.43) [5.7]	1.48Mbps(1.42) [3.7]	1.39Mbps(1.39) [3.6]
	6.5(UDP+TCP)	8.15Mbps(1.25) [9.2]	8.42Mbps(1.30) [6.1]	8.60Mbps(1.32) [3.5]
Case 3	2.5(UDP+TCP)	3.48Mbps(1.39) [6.4]	3.52Mbps(1.40) [5.3]	3.45Mbps(1.38) [3.5]
	6.5(UDP+TCP)	8.16Mbps(1.26) [9.7]	8.32Mbps(1.28) [6.4]	8.54Mbps(1.31) [3.5]

* ( ) indicates (achieved throughput)/CTH. dj=100ms for all cases.

** [ ] indicates (max throughput of TCP flow) / (min throughput of TCP flow) in the SLA

6. Concluding Remarks

In this paper, we proposed Configuration Rules 1 through 4, based on the DiffServ implementation model as shown in Figures 2 and 3, and then evaluated, through simulation, the traffic characteristics of a TCP-only and a UDP+TCP mixed case to which those rules were applied. The results show that all the service levels as specified in the SLAs are consistently satisfied with our configurations. It is also observed that the throughput performance does not seem to be very sensitive to the token bucket size b and the packet discard threshold By after testing over a considerably wide range of the parameter values for (b/AIR=20 to 250ms and By=50 to 200 packets). It will be necessary to examine how far this “safe” configuration range can be extended to. The total throughput achieved in our simulation surpasses the total committed throughput by about 30%. This is due to the fact that the allocated bandwidth AIR was set to AIR=CTH/0.75 for the TCP-only and UDP+TCP mixed traffic. The effect of the division by this 0.75 may appear to be an over-provisioning in the single node model studied in this paper. When we consider a network with many hops, however, a packet marked yellow could be dropped at any of the hops along its path, which may not be a too small probability if any part of the network is moderately or heavily congested. Since any drop of a TCP packet will cause a slowdown in its flow rate, this “over-provisioning” may become necessary. A more quantitative study on this issue is needed.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, Dec 1998.

[2] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, Jun 1999.

[3] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC 2598, Jun 1999

[4] D.D.Clark and W. Fang, “Explicit allocation of best-effort packet delivery service”, IEEE/ACM Transaction on networking, Vol.6, No.4, pp.362-373, Aug. 1998

[5] N. Seddigh, B. Nandy, P. Pieda, "Bandwidth Assurance Issues for TCPFlows in a Differentiated Services Network", Proc. of GLOBECOM '99, pp.1792-1798, Dec. 1999.

[6] S. Sahu, D. Towsley, J. Kurose, "A Quantitative Study of Differentiated Services for the Internet", Proc. of GLOBECOM '99, pp.1808-1817, Dec. 1999

[7] R.L. Cruz; “A calculus for network delay, part I : network elements in isolation”, IEEE trans. on information theory, Vol.37, No1, pp.114-131, Jan. 1991

[8] Kroner H.,Theimer T.H. and Briem U.:”Queueing models for ATM systems-A comparison”, Proc.ITC7-th seminar, 1990

[9] S.Sato,”An analysis of ATM cell loss probability based on virtual queueing system”, Trans. on comm. IEICE, Vol. J76-B-1 No.1, pp.10-20, Jan. 1993

[10] S.Floyd, “Connections with multiple congested gateway in packet-switched network Part1:One way traffic”, Computer Communications Review, Vol.21,No5, pp.30-47, Oct. 1991,