Université de Liège

Faculté des Sciences Appliquées

Réseaux informatiques

Research Unit in Networking

RUN

Prof. Guy Leduc
Institut Montéfiore, Bât. B28
Université de Liège
B-4000 Liège 1 (Belgique)

Attaining per flow QoS with class-based differentiated services

M. Tufail, G. Jennes and G. Leduc
(Université de Liège)

SPIE Symposium on Voice, Video and Data Communications, Conf: Broadband Network, Sept 1999, MA, USA.

RUN-PP-99-03
Public
September 1999

Attaining per flow QoS with class-based differentiated services

Mudassir Tufail, Geoffroy Jennes and Guy Leduc

University of Liege, Belgium

ABSTRACT

The Differentiated Services (DiffServ or DS) framework takes an edge over IntServ[1] because it is scalable and lesser complex. On the other hand, the application level end-to-end quality of service, in DiffServ, may get compromised because: 1) network resources are not allocated at microflow level (a data stream pertaining to a single connection) but at aggregate level (collection of one or more microflows), 2) the DiffServ working group does not specify algorithms for PHBs but their output behaviours and 3) end-to-end quality is function of Service Level Agreements (SLAs) between the adjacent domains transited by the connection and a large diversity in SLAs is quite evident as each DS domain would have different service provision policies. We focus, in this paper, on the first two issues. Our goal is to have DiffServ deployed with all its simplicity and still be able to provide application level end-to-end quality of service. For that, we study a PHB for AF classes. A PHB comprises a packet scheduler and a packet accept/discard algorithm.

For packet scheduler, we use the Extended-VirtualClock (Ex-VC)[2] algorithm. Ex-VC performs delay-based service differentiation among the aggregates while selecting a packet for service. The reasons for having delay-based definition for service differentiation are: it is adaptable to load per aggregate and it does not need to be microflow aware. Other definitions like bandwidth and loss may also be used but the former requires microflow aware management and the latter lacks in simplicity.[2]

For packet accept/discard algorithm, we use RED when all packets have the same drop precedence level and DI-RO (Deterministic for In-RED for Out) when packets are policed at the ingress DS node and packets violating the Service Level Agreement (SLA) are marked OUT. In DI-RO, IN packets are always accepted (except buffer overflow) whereas OUT packets are accepted probabilistically.

Keywords: Differentiated Services (DiffServ), application level QoS, delay based DiffServ, adaptable scheduling, Assured Forwarding

1. INTRODUCTION TO DIFFSERV AND MOTIVATION

The Differentiated Services (DiffServ or DS)[3] framework relies on its two important elements while allocating the network resources: 1) service differentiation is performed at aggregate level rather than at microflow level (it renders the framework scalable) and 2) service differentiation is ensured by employing appropriate packet accept/discard and forwarding mechanisms called Per Hop Behaviours (PHB) at the nodes*.

The DiffServ working group has defined three main classes: Expedited Forwarding (EF),[4] Assured Forwarding (AF)[5] and Best Effort (BE). EF can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS domains. The AF class is allocated a certain amount of forwarding resources (buffer and/or bandwidth) in each DS node and encompasses "qualitative" to "relative quantification" services.[6] The level of forwarding assurance, for an AF class, however depends on 1) the allocated resources, 2) the current load of the AF class and 3) the congestion level within the class. The AF class is further subdivided into four AF classes: AF1, AF2, AF3 and AF4.[5] Each AF subclass may have packets belonging to three drop precedences which eventually makes 12 levels of service differentiation under the AF PHB group. The precedence of a packet defines how much it is prone to be discarded in case of congestion.

The definition of a set of services supported by DiffServ is still an unresolved issue as it requires the DS domain

Further author information: {mtufail, jennes}@run.montefiore.ulg.ac.be, leduc@montefiore.ulg.ac.be

This work was supported by the Flemish Institute for promotion of Scientific and Technical Research in the Industry under the IWT project for which University of Liege and Alcatel Alsthom CRC (Antwerp) are the two partners.

*The boundary nodes have an additional role of traffic conditioning (metering, marking, shaping and discarding) and they might perform microflow level traffic classification.

administrators to agree on some specific Service Level Agreements[†] (SLAs) encompassing certain services types. However three types of services have been proposed[6]:

- **Better that Best Effort (BBE)** is a qualitative service which promises to carry specific traffic, say web server traffic, at a higher priority than competing best-effort traffic. This service offers relatively loose (not quantifiable) performance from a given ingress to any egress point of a DS domain.
- **Quantitative assured media playback** is a relative quantification service and promises to deliver traffic with high degree of reliability and with variable but bounded latency, up to a negotiated rate. This service is particularly suitable for video or audio playback which are non-interactive and thus makes them delay tolerant.
- **Leased line emulation** is a purely quantitative service and emulates traditional leased line service. This service is based on ingress-egress pair based SLAs. An example of such service is IP telephony.

The first two service types would be constructed over AF PHB whereas the third one would employ EF PHB in DS domain.

Motivation: The DiffServ framework takes an edge over IntServ[1] because it is scalable and lesser complex. On the other hand, the application level end-to-end quality of service is at risk because: 1) network resource are not allocated at microflow level but at aggregate level, 2) the DiffServ working group does not specify algorithms for PHBs but their output behaviours and 3) end-to-end quality is function of SLAs between the adjacent domains transited by the connection and a large diversity in SLAs is quite evident as each DS domain would have different service provision policies. These issues need to be resolved so that the DiffServ is deployed with all its simplicity and still provides application level end-to-end quality of service. For that, we investigate a PHB for the AF class[‡] comprising a packet scheduler algorithm named as Ex-VC[2] and a RED/DI-RO packet accept/discard algorithm and concentrate on the first two issues.

Few words about our previous work: In our previous work,[2] we investigated two important issues concerning DiffServ deployment:

- how would the service differentiation, which is performed at aggregate level, be at microflow level?
- how would the network resource allocation among the aggregates adapt with load so that qualitative (or relative) service differentiation is preserved at all loads?

The article[2] analyses three quality metrics (bandwidth, loss and delay) which might be used for service differentiation among AF aggregates. The analysis aimed at resolving the above mentioned two issues in DiffServ. The bandwidth-based service differentiation requires the microflow aware resource allocating mechanism, hence scalability problem. The loss-based service differentiation does not require the algorithm to be microflow aware, but is rendered tedious when combined with packet precedence levels within an aggregate. It would need a complex algorithm which manages all the thresholds (for aggregates & precedences) not only to respect the relative quality of services among aggregates, but also to ensure the relative quality of services among packets of different drop precedences within an aggregate. Delay is a parameter which provides numerous advantages. The delay metric itself is microflow independent as ensuring better delays at an aggregate level also means ensuring better delays for all the included microflows. Additionally we found that delay-based service differentiation can easily adapt with varying load and we termed this property as *self-regulation*. We concluded that a delay-based service differentiation resolves the above mentioned two issues. We then defined formally a delay-based service differentiation model and developed a delay-based scheduling algorithm named as Extended VirtualClock (Ex-VC).

[†]It is service contract between a customer and its service provider that specifies the forwarding service a customer should receive. A customer may be a user or another DS domain.

[‡]The AF class provides elastic service to aggregates (eventually to applications) and is therefore more complex to provision and to develop than EF class which provides a deterministic quality of service. We, hence, focus on AF class so that the complex issue get investigated.

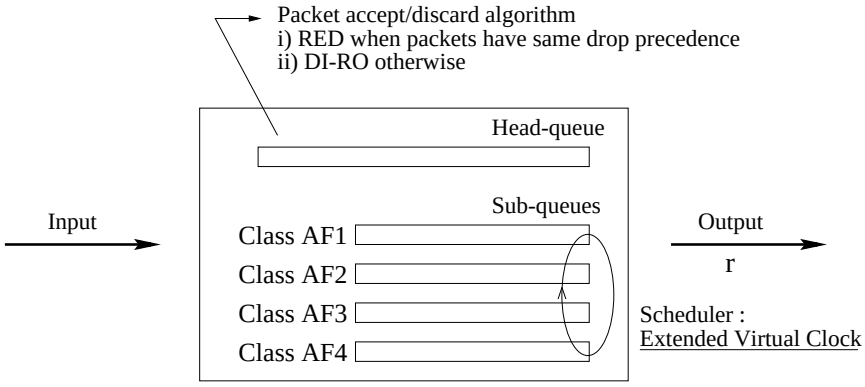

Figure 1. The packet accept/discard and scheduling algorithms

Contribution: The article[2] validated the Ex-VC algorithm with simulations having fluid flows (non-bursty packet arrivals with constant rate) under different types of buffer loading. We took the fluid flows (which are actually non-realistic flows) to verify, a priori, that the algorithm maintains the service differentiation among the aggregates at all loads. In this paper, our goal is to complete the AF PHB implementation by adding RED/DI-RO packet accept/discard algorithm to the Ex-VC packet scheduler and perform the simulations having realistic flows. For that we use the STCP[7] simulator which simulates TCP flows.
Our paper is structured as follows: section 2 presents the AF PHB being investigated i.e. the packet scheduling and packet accept/discard algorithms, section 3 describes the results of simulations carried out in two phases (with and with out SLAs) and finally section 4 concludes the article.

2. AN AF PHB

As said before, a PHB comprises a packet accept/discard algorithm and a scheduler, refer to figure 1. We employ the Ex-VC as packet scheduler and one of the two (RED, DI-RO) as packet accept/discard algorithms depending upon the case. Since the packet accept/discard algorithm is implemented on head queue (i.e. global buffer) and Ex-VC scheduler does not oblige to have separate queues per aggregate[§], the presented AF PHB can also de described with out separate queues per aggregate.

2.1. Ex-VC algorithm

This section presents the Extended VirtualClock (Ex-VC) scheduling algorithm.[2] The Ex-VC algorithm has an additional instruction of self-regulation compared to the traditional VC algorithm,[8] hence the term "extended". Note that the Ex-VC algorithm is not restricted to four aggregates (of DiffServ) only. It may be used with any number of aggregates (or queues). However, the cost of self-regulation increases with the number of aggregates[¶]. q_i represents the quality index associated with an aggregate i. Each packet is stamped at its arrival. The packets are then served in increasing order of the stamp values. $v(t)$ represents the system virtual time at time t and is defined equal to the stamp value of the packet receiving service at time t. $v(t)$ is initially set to zero. The stamp value of the k^{th} packet of the i^{th} aggregate is written as $stamp_i^k$ whereas the packet itself is denoted as p_i^k. s_i^k and f_i^k represents the instants of service-start and service-finish of a packet p_i^k. Each aggregate i maintains two registers $LastStamp_i$ and VS_i (Virtual Spacing). The $LastStamp_i$ registers the stamp value of packet (p_i^k $\forall k$) serviced precedently. It is initially set to zero. The VS_i is updated as $VS_i = \frac{L_i^k}{r_i}$ at each packet (p_i^k $\forall k$) arrival where L_i^k is the size of packet p_i^k and r_i is the service rate of aggregate i. The Ex-VC algorithm works as follows:

[§]For single queue implementation, the Ex-VC would stamp the packets at their arrival and then would insert them in head queue in increasing order of stamp values.

[¶]One may not perform the self-regulation at each packet arrival (i.e. the instance of its stamp calculation). It has been noted that during the stable periods (i.e. fewer burst arrivals) reducing the frequency of self-regulation by 10 does not have a significant effect on algorithm performance.

At an arrival of a packet p_i^k at instant t

- increase b_i by L_i^k
- $r_i = \frac{r b_i q_i}{\sum_{j=1}^{4} b_j q_j}$ /*self-regulation*/
- $VS_i = \frac{L_i^k}{r_i}$
- $stamp_i^k = max(v(t), LastStamp_i) + VS_i$
- $LastStamp_i = stamp_i^k$

At selecting a packet $p_{i'}^{k'}$, having the minimum stamp value, for service at instant t

- $v(t) = stamp_{i'}^{k'}$ where $s_{i'}^{k'} < t \leq f_{i'}^{k'}$

At departure of the packet $p_{i'}^{k'}$

- decrease $b_{i'}$ by $L_{i'}^{k'}$

About existing algorithms: A similar delay-based approach for service differentiation has also been presented precedently.[9] Two schedulers, Backlog-Proportional Rate (BPR) and Waiting Time Priority (WTP), have been studied. The BPR adjusts the service rate (self-regulation property) of an aggregate with its backlog whereas in the WTP, the priority of a packet increases proportionally with its waiting time. The simulation results in[9,10] show that the WTP is significantly better than the BPR. We envisage comparing the Ex-VC with the WTP in our future simulations.

2.2. Packet accept/discard algorithm

When a packet arrives at a node, it is either accepted or rejected by a packet accept/discard algorithm. Since simulations are performed via the STCP[7] simulator which fragments all the packets into ATM cells, there will be two packet drop precedences (CLP0 and CLP1) instead of three as proposed in the DiffServ framework. Moreover, the packet accept/discard algorithms are AAL5 frame aware. That is to say accepting/discarding a cell means accepting/discarding the entire AAL5 frame. We employ two types of algorithms:

1. RED[11] is used when all packets have the same drop precedence level. This would be the case in our first phase of simulations where there are no SLAs and hence no policer[‖] to mark the packets. Whenever the first cell of an AAL5 frame arrives at the buffer, RED calculates the average queue length based on the total number of cells. If the average value is less than min_th[**] then the cell is accepted along with all the following cells which pertain to the same AAL5 frame. If the average value is above 2*max_th then the cell is discarded along with all the following cells pertaining to the same AAL5 frame. If the average value queue length falls in between the min_th and max_th, then the first AAL5 cell (along with all the following cells of the same frame) are discarded with a probability which increases linearly (from zero at min_th to max_p at max_th) with the average queue length value. If the average value queue length falls in between the max_th and 2*max_th, then the cell (along with all the following cells of the same frame) are discarded with a probability which increases linearly (from max_p at max_th to 1 at 2*max_th) with the average queue length value. We can, precisely, name this implementation of RED as F-RED. A similar approach has also been presented by Rosolen[12] and is named as ATM-RED. In ATM-RED, the decision of accepting/discarding a packet is done at each cell unlikely to our approach where it is done at the arrival of the first cell of AAL5 frame. ATM-RED does not drop cells of an AAL5 frame which has already been partially accepted, instead the next AAL5 frame is dropped completely. Rosolen uses a cell level max_c instead of packet level max_p related as follows: $max_p = 1 - (1 - max_c)^n$ where n is number of cells in AAL5 frame. ATM-RED performs well when implemented per queue. This ensures that the packets of a class will not be discarded due to congestion caused by a precedently accepted packet belonging to another class. In our simulations, we do not implement packet accept/discard algorithm per queue but per global buffer (i.e. head queue) and that is why we do not use ATM-RED.

2. We use DI-RO (Deterministic for In-RED for Out) algorithm when there are SLAs and hence policers at the ingress of the DS domain (ATM backbone in our case) which mark the packets (AAL5 frame) OUT if they exceed the negotiated rate specified in the respective SLA. This represents the second phase of our simulations.

[‖] For the present work, we do not consider marking by the application itself.

[**] The thresholds (min_th and max_th) are expressed in number of cells.

The aggregates will have both IN and OUT packets. The decision of accepting/rejecting a packet will now depend not only upon the buffer utilisation but also upon the packet drop precedence.
Whenever the first out-of-profile cell[††] of an AAL5 frame arrives at the buffer, RED part of DI-RO calculates the average queue length based on the total number of cells (regardless of their CLP values). As for accepting or discarding, DI-RO works in the same way as RED, described above. The only difference is that here the cells are marked as out-of-profile.
Whenever the first in-profile cell of an AAL5 frame arrives at the buffer, it and all the following cells of the same frame would be accepted except if the buffer overflows. The idea is to get the applications in-profile packets transmitted regardless of the density of out-of-profile cells in the buffer whereas the latter would get accepted only if excess resources (buffer) are available. The RED part of DI-RO is parameterised such that we never run into a buffer overflow situation.

How better the approach is? Our implementation of the AF PHB has the property of being simple and efficient. The important points are:

- It does not require buffer management per aggregate (i.e. threshold values per aggregate) and RED/DI-RO is implemented at global buffer (i.e. head queue).
- The DI-RO is better than RIO (RED for In and Out) as the latter may drop (probabilistically) the in-profile packets even if the buffer is not full.
- Each aggregate is associated with a quality index instead of a reserved bandwidth value. This helps the scheduler to adapt the aggregate's service rate with its varying load while ensuring the relative service differentiation at all times and loads.
- The approach is independent of the number of microflows and provides the same QoS at aggregate and at microflow levels.

3. SIMULATION

The simulations were performed with the STCP simulator.[7] In all scenarios, the TCP workstation are connected to IP routers which are then interconnected by an ATM backbone. Each IP router is dedicated to a certain AF class, i.e. all the microflows routed through a given router are aggregated to the same AF class. TCP sources transmit short files of 200Kbytes without inter file delay. The slow time-out is 200ms (the maximum delay between two consecutive delayed ACKs) whereas fast time-out is 50ms (instead of traditionally used 500ms timer). The Maximum Segment Size (MSS) is 1460 and maximum window size is 64Kbytes. The Selective ACKnowledgement (SACK) option is enabled. The simulation duration is fixed to 102 seconds.
We carry out simulations in two phases. Phase 1 contains packets with the same drop precedence and hence employ RED as packet accept/discard algorithm with min_th=4000, max_th=9500 (cells), $max_p = 0.02$ and $wq = 0.002$. In phase 2, we introduce SLAs. These SLAs are enforced by markers and packets will get marked OUT if they violate the respective SLAs. In this phase, we use DI-RO as packet accept/discard algorithm and RED part of DI-RO is configured with min_th=1000, max_th=7000 (cells), $max_p = 0.02$ and $wq = 0.002$. DI-RO accepts in-profile packets deterministically and requires a hard admission of out-of-profile packets hence more severe RED thresholds. The total buffer capacity is 20000 cells for both packet accept/discard algorithms. As for packet scheduler, it is Ex-VC in all cases.

3.1. Phase 1: Basic simulation scenario

In this scenario, as shown in figure 2, 200 pairs of TCP sources are interconnected via four routers and two ATM switches forming the backbone. Each of the four routers is dedicated to a certain AF class as shown. The backbone is at 50Mbps and has a transmission delay of 10ms. The ATM switch performs AF PHB (RED + Ex-VC) at aggregates in accordance with their respective quality indexes. We perform three types of loadings per aggregate:

[††]The F-GCRA policer, used in our simulations, is AAL5 frame aware. It means that an out-of-profile AAL5 frame will have all its cells tagged.

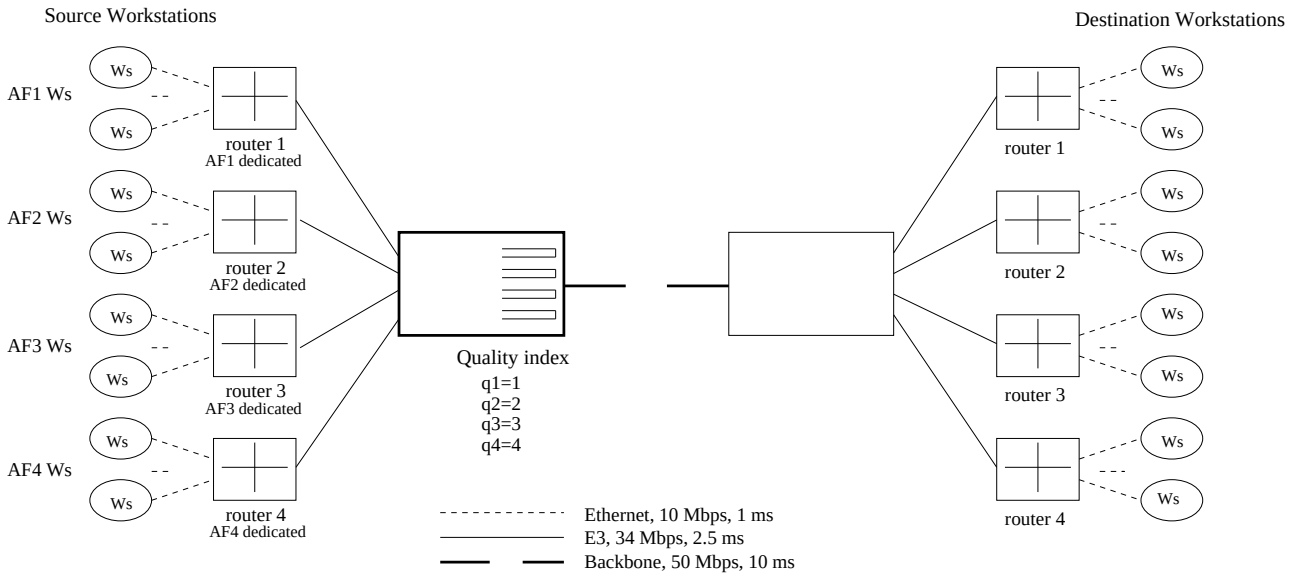

Figure 2. The basic simulation scenario.

- **Symmetrical loading:** The aggregates are loaded proportionally to their (relative) delay guarantees. That is to say that aggregate AF4 will receive packets, on the average, at rate half of that at which aggregate AF2 would receive the packets. Remember that aggregate AF4 experiences half the delay of AF2. The symmetric loading is simulated by having 80 workstations (1 TCP flow per workstation) for AF1 (i.e. attached to router 1), 60 for AF2, 40 for AF3 and 20 for AF4. This yields a buffer loading where Ex-VC algorithm self-regulates at a rather easy-going pace.
- **Equal loading:** All aggregates are loaded with equal rates regardless of their quality indexes. Here, the basic scenario contains 50 workstations for each AF class.
- **Asymmetrical loading:** It tests the Ex-VC algorithm in tending-to-worst buffer loading configuration and algorithm self-regulates at a hard-going pace. Queues are loaded inversely proportionally to their delay guarantees. In other words, the aggregate AF4 will receive packets, on the average, at rate double of that at which aggregate AF2 would receive. The class AF1 is fed with 20 TCP flows, AF2 with 40, AF3 with 60 and AF4 with 80 flows.

The purpose of having three different buffer loadings in basic scenario is to verify the following three points:

1. The service differentiation among the aggregates is respected at all loads.
2. The service at microflow level is independent of the number of microflows in the aggregate.
3. The packet loss ratio is the same for all the aggregates as RED is implemented at head queue, not on individual queues.

The simulation results presented in table 1 proves the first point. The table shows the two types of mean delays: local delay at switch which comprises the local queueing delay only and end-to-end delay which comprises the queueing delay at all traversed nodes plus transmission delay (a constant factor). The delay differentiation among the aggregates is respected under all loading configurations. This differentiation follows very closely the aggregate's quality indexes as far as the local mean delay values are concerned. For end-to-end delay values, the differentiation is, though, not in the same magnitude order as that of aggregate's quality indexes (due to the addition of constant transmission delay), the relative delay differentiation among aggregates is still preserved. The values of mean global buffer (sum of four AF aggregates) occupancy, at switch, in symmetrical, equal and asymmetrical loading configurations are

Table 1. Basic scenario results under different loading configurations (delay).

		Class AF1	Class AF2	Class AF3	Class AF4
Mean delay at switch (msec)	Symmetrical	119.8	61.5	42.6	34.6
	Equal	154.6	77.7	53.0	40.5
	Asymmetrical	201.2	99.4	66.7	50.0
End-to-end delay (msec)	Symmetrical	131.2	78.1	63.2	53.6
	Equal	168.7	95.9	71.5	60.0
	Asymmetrical	228.3	113.6	84.9	70.4

(in cells): 9.06K, 9.08K and 9.05K respectively. Since the mean buffer occupancy at switch is the same for all the configurations and the server is work conserving, the average delay per packet per global buffer is the same for all the configurations. However, we notice in table 1 that aggregates suffer, on the average, individually more delays as we move from symmetrical to asymmetrical buffer loading.
The table 2 presents two important results: mean goodput per workstation and mean Packet Loss Ratio (PLR) per aggregate. The goodput represents the amount of useful data transmitted (excluding TCP/IP/ATM headers) per sec. All transmitted packets are counted once i.e. retransmissions are not counted. The goodput values, showed in simulation results, are calculated using the following formula:

$$goodput_{workstation} = \frac{TCPACKnowledgedBytes_{workstation}}{simulation\_time} \quad (1)$$

Considering an aggregate say AF4 in table 2, the value of mean goodput per workstation (i.e. per microflow) is shown under three loading configuration which are 0.26, 0.27 and 0.25. These values are very close and prove the point # 2 that the service at microflow level is independent of aggregate load (number of microflows). Now look at PLR values of table 2, these values are approximately the same for all the aggregates under a given loading configuration. For example under equal load, the PLRs are 7.5, 7.6, 7.8 and 7.7 for AF1, AF2, AF3 and AF4 respectively. The aggregates face a proportional loss in case of congestion and this is attained by having packet accept/discard algorithm (RED) on the head queue. The PLR results support our third claim (point # 3).

Table 2. Basic scenario results under different loading configurations (goodput & loss).

		Class AF1	Class AF2	Class AF3	Class AF4
Mean goodput per work station (Mbps)	Symmetrical	0.17	0.25	0.26	0.26
	Equal	0.15	0.22	0.24	0.27
	Asymmetrical	0.12	0.20	0.23	0.25
Mean packet loss ratio (%)	Symmetrical	7.5	7.5	7.7	8.0
	Equal	7.5	7.6	7.8	7.7
	Asymmetrical	7.6	7.5	7.6	7.6

3.2. Phase 2: What if SLA exists?

A Service Level Agreement (SLA) represents the contract between two adjacent DS domains and between a client and its DS compliant service provider. This contract may allocate the resources (bandwidth and/or buffer) either for each aggregate or for all aggregates together. Here we will study the case where each AF aggregate is allocated with a certain amount of resources. In order to ensure that an aggregate does not violate the SLA, we employ rate controller policer at the DS ingress nodes. Since the TCP segments are commuted via ATM cells in our simulations,

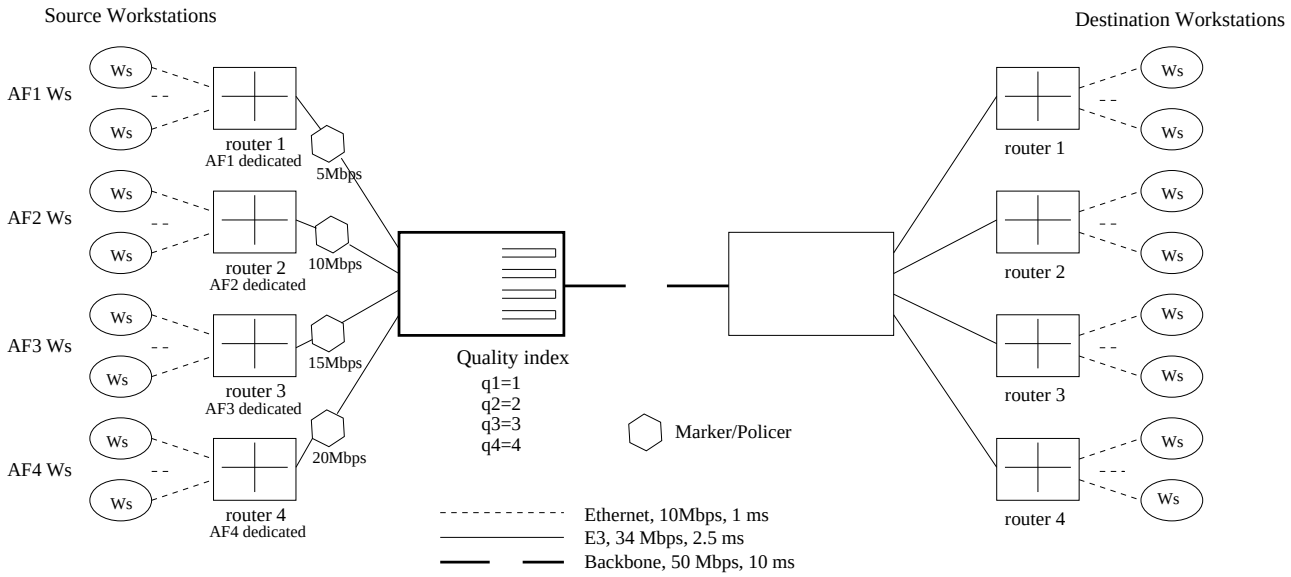

Figure 3. The basic simulation scenario with marker.

we use a frame (AAL5) aware version of Generic Cell Rate Algorithm and denote it as F-GCRA. The F-GCRA is configured with a throughput value, if the aggregate exceeds this value by a certain tolerance τ, its packets (i.e. AAL5 frame) get marked. A marked AAL5 frame will have all its cells with CLP1. It means that there will be cells of different drop precedences at inner nodes (ATM switch in our case) of the DS domain, hence the packet accept/discard algorithm should be CLP aware. For all the following simulations, we employ the DI-RO algorithm instead of RED. As for the packet scheduling algorithm, we employ the Ex-VC with out any modification.

3.2.1. Basic scenario with marker

This is the same scenario presented in section 3.1 but with addition of F-GCRA markers between the routers and the ATM switch. These markers police the traffic of each router (dedicated to an AF class) and mark the out-of-profile packets. We simulate this scenario also under three loading configurations: symmetrical, equal and asymmetrical (refer to section 3.1). The goal here is to verify that:

- The SLAs are respected at all loads.
- The aggregate's packet loss rates increase with their excess traffic (i.e. beyond SLA value). Recall that in our previous simulation, all aggregates had approximately the same PLR for a given loading configuration.

As per SLA, the class AF1 is allocated with 5Mbps, AF2 with 10 Mbps, AF3 with 15Mbps and AF4 with 20Mbps (together they fill the backbone at 50Mbps). The F-GCRAs are configured respectively as shown in figure 3. The simulation results are summarised in table 3. The router throughput (which is actually the corresponding aggregate's throughput) is policed by F-GCRA at the value dictated by the SLA. The packets exceeding the SLA value are marked. We find that the mean throughput attained by the aggregates correspond to the values specified in their respective SLAs. Moreover, SLAs are respected in all loading configurations, verification of first point. Note that the class AF4 under symmetrical loading does not attain its 20 Mbps (throughput specified in its SLA). Having lesser TCP sources (i.e. 20 workstations) is the main reason for not availing the reserved resources. It manages to reach up to 15.5Mbps. Similar observations have also been described in the article by Ikjun Yoem[13] where TCP connections with larger values of reserved bandwidths were found unable to reach them.

The more is the aggregate's excess traffic (i.e. beyond the SLA value), the more is the PLR. The PLR values of the aggregates in table 3 support the statement. Under symmetrical loading, there are 80 TCP flows in AF1 class whereas it is allocated with only 5Mbps. Naturally, it faces the highest PLR. The same thing is repeated for class AF4 in asymmetrical loading but not at the same intensity (compare AF4 PLR (7.9) in asymmetrical to AF1 PLR

Table 3. The SLA enforcement in basic scenario, under different loading configurations.

		Class AF1	Class AF2	Class AF3	Class AF4
SLA: Bandwidth (Mbps)		**5**	**10**	**15**	**20**
Symmetrical	Aggregate mean throughput (Mbps)	6.9	11.4	14.5	15.5
	Aggregate mean goodput (Mbps)	5.7	10.1	13.5	15.1
	Mean PLR (%) out-of-profile	13.1	8.5	5.5	2.9
Equal	Aggregate mean throughput (Mbps)	5.4	10.2	14.4	18.6
	Aggregate mean goodput (Mbps)	4.7	9.2	13.2	17.2
	Mean PLR (%) out-of-profile	9.4	7.6	6.4	5.9
Asymmetrical	Aggregate mean throughput (Mbps)	4.4	9.6	14.8	20.2
	Aggregate mean goodput (Mbps)	4.2	8.8	13.3	17.9
	Mean PLR (%) out-of-profile	4.5	6.7	7.3	7.9

(13.1) in symmetrical loading). This is because, having 80 TCP for 20 Mbps is not as significant as having them for 5Mbps. We may conclude that aggregates suffer more loss as they exceed more their respective SLAs, verification of the second point.
We have also shown the aggregate's goodput value which is function of its throughput and PLR. Note that SLAs control the throughput value not the goodput values. As the PLR increases the difference between throughput and goodput becomes more significant.

3.2.2. Simulating more complex scenarios

Till now we have simulated as if there was only one client for a DS domain providing the service via four AF classes. A DS domain will most probably serve more than one client and traffic from different clients will be multiplexed and then forwarded. We will now simulate two organisations A and B having their sites interconnected by a DS domain. Each of these clients generates the traffic pertaining to four AF classes. The aggregates are policed by

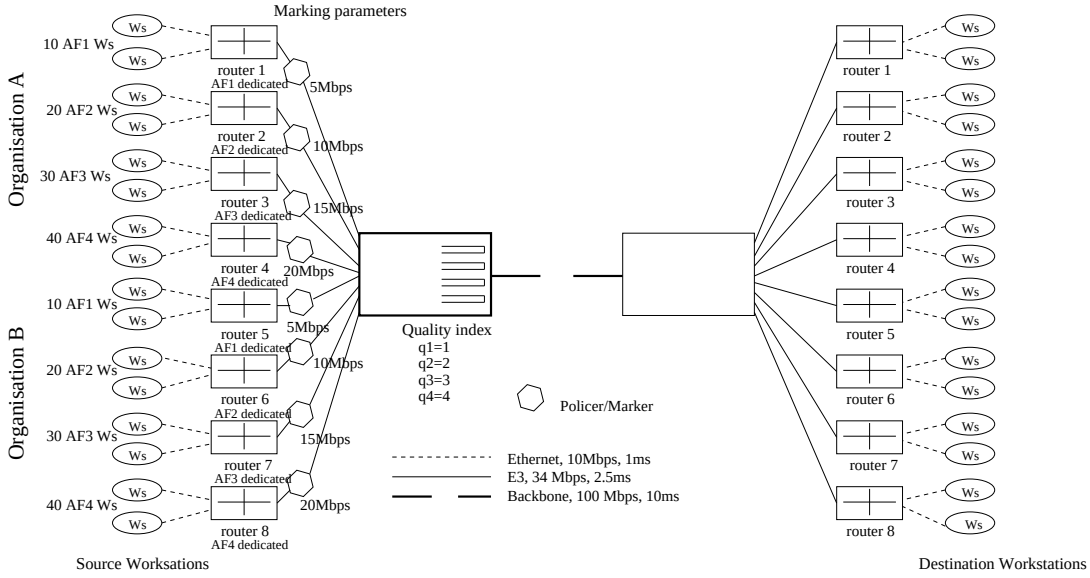

Figure 4. The 50/50 sharing simulation scenario.

F-GCRA markers at their respective SLAs. The important issue to study is how the sharing of resources between the organisations takes place so that SLAs are respected and fairness is maintained. We present in the following two simulations scenarios, and for each of them there are, per organisation, 10 workstations (i.e. TCP flows) for the AF1 class, 20 for the AF2 class, 30 for the AF3 class and 40 for the AF4 class.

1. **50/50 sharing simulation scenario:** In this scenario, the two organisations A & B have equal SLA values for respective AF classes, refer to figure 4. That is to say SLA for AF1 of organisation A is the same as that for AF1 of organisation B. Note that the backbone bandwidth is 100Mbps and same is the sum of all SLA values. In this configuration, the two organisations are expected to share equally the network resources (bandwidth and buffer). Table 4 presents the simulation results of the 50/50 sharing scenario. The two organisations attain nearly the same mean throughput per aggregate. The same is valid for goodput and PLR values. The results verify that the two organisations share equally the network resources and SLAs are respected.

Table 4. The simulation results for 50/50 sharing scenario.

		Class AF1	Class AF2	Class AF3	Class AF4
SLA: Bandwidth (Mbps)		**5**	**10**	**15**	**20**
Organisation A	Aggregate mean throughput (Mbps)	4.4	9.7	13.9	19.2
	Aggregate mean goodput (Mbps)	4.1	9.0	12.9	17.6
	Mean PLR (%) out-of-profile	4.6	5.3	5.9	6.3
Organisation B	Aggregate mean throughput (Mbps)	4.4	9.7	14.2	18.5
	Aggregate mean goodput (Mbps)	4.2	9.1	13.2	17.0
	Mean PLR (%) out-of-profile	4.8	5.4	5.9	6.3

2. **25/75 sharing simulation scenario:** We take the same scenario of figure 4 but change the SLA values in 25/75 proportion between two organisations A & B. The F-GCRA for AF1 of organisation A polices its traffic at 2.5Mbps whereas that for organisation B polices at 7.5Mbps, refer to figure 5. In this configuration, the two

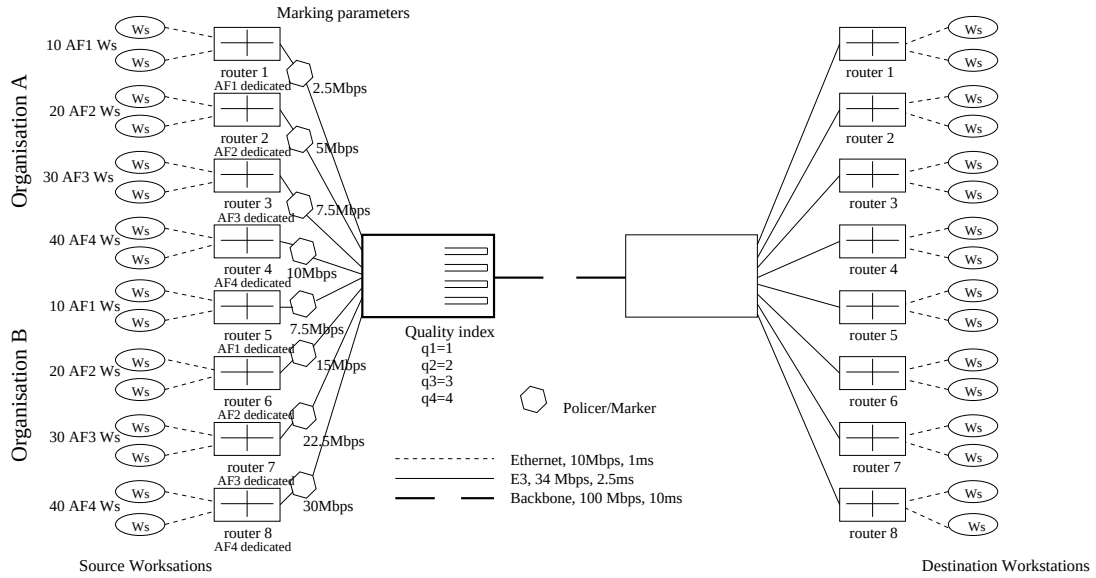

Figure 5. The 25/75 sharing simulation scenario.

organisations are expected to share the network resource in 1:3 proportion while preserving the SLAs. The table 5 presents the simulation results of 25/75 sharing scenario. We notice that aggregates of organisation A attain a mean throughput more than the ones specified by their respective SLAs whereas that aggregates of organisation B are unable to reach the SLA's bandwidths. It is mainly due to load (i.e. number of TCP connections per aggregate) to bandwidth (specified in SLA) ratio and partially due to the fairness explained later in this paragraph. The organisation B has lower load to bandwidth ratio and hence could not fully avail the SLA's bandwidths. As for the fairness, look at throughput values. The aggregate's values do not follow 1:3 proportion rather they stay close to 1:2 proportion. The same is true for goodput values. The marker may play an important role for recovering this fairness. We believe that using a simple marker like F-GCRA may not police properly the mean throughput of aggregate. The packets get marked, when the bursts are long, even if the mean throughput value is below the SLA. We can conclude that although the proposed implementation preserves the SLAs, the fairness is not totally respected. Another important observation is that aggregates of organisation A have slightly lesser end-to-end delay than those of organisation B. This is because, the organisation A has more packets marked than B, so suffers more packet-drop (see PLRs) under congestion. This means that there will be slightly fewer packets, of organisation A, being accepted when the queues are long, hence an improvement in end-to-end delays.

Table 5. The simulation results for 25/75 sharing scenario.

		Class AF1	Class AF2	Class AF3	Class AF4
Organisation A	**SLA: Bandwidth (Mbps)**	**2.5**	**5**	**7.5**	**10**
	Aggregate mean throughput (Mbps)	2.9	6.3	9.5	13.0
	Aggregate mean goodput (Mbps)	2.7	5.7	8.3	11.4
	Mean PLR (%) out-of-profile	6.9	7.8	8.4	8.4
Organisation B	**SLA: Bandwidth (Mbps)**	**7.5**	**15**	**22.5**	**30**
	Aggregate mean throughput (Mbps)	5.7	13.0	19.1	24.5
	Aggregate mean goodput (Mbps)	5.5	12.4	18.1	23.1
	Mean PLR (%) out-of-profile	3.6	4.1	4.7	5.0

4. CONCLUSION

We presented a brief introduction to the DiffServ framework covering EF and AF PHBs. We then focused our attention on the AF PHB comprising the Ex-VC scheduler and RED/DI-RO packet accept/discard algorithm. The Ex-VC scheduler[2] is delay-based and adapts itself with the changing aggregate's load (self-regulation property). Moreover, it does not need to be microflow aware. As for packet accept/drop algorithm, we select : 1) RED when all packets have the same drop precedence or 2) DI-RO (Deterministic for In-RED for Out) algorithm when packets have different drop precedences. The latter accepts always the in-profile packets except if buffer overflows whereas it accepts probabilistically the out-of-profile packets. It is employed when SLAs are defined and there are markers at ingress of the DS domain marking the out-of-profile packets.

We then present simulation results of the above AF PHB. The simulations are described in a progressive manner where each configuration is tested to verify certain points. The first phase of simulations proved that service differentiation is respected at all loads and the quality at microflow level is independent of the load of the aggregates. Moreover, the PLR is the same for all the aggregates under a given configuration as packet accept/discard (RED) is implemented on head queue (i.e. global buffer). We then, in phase 2, introduce SLAs in configurations and consequently the addition of markers for traffic policing. RED is replaced by DI-RO. Here three configurations were tested and the results prove that the proposed implementation preserves SLA at all loads and aggregate's packet loss rates increase with their excess traffic beyond SLA. At the same time, fairness problem has been observed. We believe that using a simple marker like F-GCRA is one of the reasons for that. In our future simulations, we envisage replacing F-GCRA by a sophisticated marker (e.g. token bucket) and study the effect of having RED/DI-RO per aggregate. Additionally, we are currently doing a jitter-focused comparative study of Ex-VC with WTP (Waiting

Time Priority)[9] and are developing another WTP-like scheduling algorithm which, like Ex-VC but unlike WTP, does not oblige having separate queues per aggregate and still has a performance comparable to that of WTP.

REFERENCES

1. R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," *Internet RFC 1633* , 1994.
2. M. Tufail, G. Jennes, and G. Leduc, "A scheduler for delay-based service differentiation among af classes," *to appear in IFIP Broadband Communications'99* , Nov. 1999.
3. S. Blake, D. Black, M. Carlson, E. Davis, Z. Wang, and W. Weiss, "An architecture for differentiated services," *Internet RFC 2475* .
4. V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding phb," *Internet RFC 2598* , 1999.
5. J. Heinanen, T. Finland, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding phb group," *Internet RFC 2597* , 1999.
6. Y. Boram, J. Binder, S. Blake, M. Carlson, B. E. Carpenter, S. Keshave, E. Davies, B. Ohlman, D. Verma, Z. Wang, and W. Weiss, "A framework for differentiated services," *Internet draft: draft-ietf-diffserv-framework-02.txt* , Feb. 1999.
7. S. Manthorpe, "Stcp 3.2.6: Tcp/abr/atm network simulator," *http://lrcwww.epfl.ch/ manthorp/stcp/stcp.html* , 1997.
8. L. Zhang, "Virtualclock: A new traffic control algorithm for packet switching," *ACM Transactions on Computer Systems* **9(2)**, pp. 101–124, May 1991.
9. C. Dovrolis and D. Stiliadis, "Proportional differentiated services: Delay differentiation and packet scheduling," *to appear in ACM SIGCOMM-99, (http://www.cae.wisc.edu/ dovrolis/)* , 1999.
10. S. D. Cnodder and K. Pauwels, "Relative delay priorities in a differentiated services network architecture," *Alcatel Alsthom CRC (Antwerp, Belgium) deliverable* , 1999.
11. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking* , 1993.
12. V. Rosolen, O. Bonaventure, and G. Leduc, "A red discard strategy for atm networks and its performance evaluation with tcp/ip traffic," *to appear in Computer Communication Review, Vol. 29, No. 3* , July, 1999.
13. I. Yoem and A. L. N. Reddy, "Realizing throughput guarantees in a differentiated services network," *Proceedings of ICMCS (http://dropzone.tamu.edu/ ikjun/papers.html)* , June, 1999.