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)

A scheduler for delay-based service differentiation among AF classes

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

IFIP Broadband Communications'99, Nov. 1999, Hong Kong.

RUN-PP-99-04
Public
November 1999

A SCHEDULER FOR DELAY-BASED SERVICE DIFFERENTIATION AMONG AF CLASSES

*Mudassir Tufail[1], Geoffroy Jennes[1], Guy Leduc[2]
University of Liege, Belgium.
[1] {mtufail,jennes}@run.montefiore.ulg.ac.be
[2]leduc@montefiore.ulg.ac.be

Abstract In the Differentiated Services (DS) framework, service differentiation is performed among the aggregates (collection of one or more microflows) rather than among the microflows (data stream pertaining to a single connection). We analyse three quality metrics namely bandwidth, loss and delay (that might be used for defining a service differentiation at a DS node) on two criteria 1) service differentiation should be **respected at all loads** and 2) service provision at aggregate level should scale down to microflow level **without being microflow aware**. We find that bandwidth requires microflow aware management, loss lacks in simplicity (though it satisfies the criterion # 2), and delay is the right candidate.

Ensuring better delays at an aggregate level also means ensuring better delays for all the included microflows, and additionally it is easier to define a scheduler that can adapt itself to the relative loads of the aggregates so that relative delays between aggregates are preserved at all loads. Our objective is to provide **relative quantification service** in DiffServ by a delay-based scheduler while satisfying both criteria.

Delay is also a meaningful QoS parameter for both interactive real-time applications and TCP applications, since the mean TCP throughput is roughly inversely proportional to the RTT.

We, therefore, develop a scheduler for Assured Forwarding (AF) PHB where service differentiation among aggregates is based on delays. We provide simulation results that prove that relative delays among aggregates are perfectly respected at all loads.

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

*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.

1. INTRODUCTION TO DIFFSERV

In Differentiated Services (DiffServ or DS) [1], service differentiation is performed at aggregate level rather than at microflow level. The motivation is to render the framework scalable. The service differentiation is ensured by employing appropriate packet discarding/forwarding mechanisms called Per Hop Behaviours (PHB) at core nodes along with traffic conditioning functions (metering, marking, shaping and discarding) at boundary nodes.

The DiffServ working group has defined three main classes: Expedited Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE). The EF can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end **quantitative** service through DS domains. The AF class is allocated a certain amount of forwarding resources (buffer and/or bandwidth) in each DS node. The level of forwarding assurance, for an AF class, however depends on 1) the allocated resources, 2) the current load of 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 [2]. Each AF subclass may have packets belonging to three drop precedences which eventually makes 12 levels of service differentiation under AF PHB group. The AF encompasses **qualitative** to **relative quantification** services [6]. In qualitative service, the forwarding assurance of the aggregates is not mutual-dependent, i.e. an aggregate may get forwarded with low loss whereas other with low delay [4]. In relative quantification, the service given to an aggregate is quantified relatively with respect to the service given to other aggregate(s). For example, an aggregate A would get x time better service than an aggregate B.

Motivation:. Despite of fact that the DiffServ proposition is simple and scalable, there are some important issues, notably:

- 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 service differentiation is preserved at all load?

This work aims at resolving the above mentioned two issues and focuses on AF PHB group meant for providing relative quantification service in DiffServ. Our paper is structured as follows: section 2 evaluates three metrics (bandwidth, loss and delay) for service differentiation with respective pros and cons and finally select one on which the rest of the paper is based, section 3 describes a formal description of the selected metric, section 4 develops a VirtualClock-based scheduling algorithm and finally section 5 presents the simulation results and their analysis.

2. DIFFERENT METRICS FOR SERVICE DIFFERENTIATION

There are three quality metrics which might be used for defining a service differentiation among AF classes at a DS node. These are: bandwidth, loss and delay. In the following sections, we study each of these metrics individually.

2.1 BANDWIDTH

If service differentiation at aggregate level is bandwidth-based then one needs to know the number of included microflows (for each aggregate) in order to determine the service differentiation at microflow level. For example, an aggregate getting 50 Mbps would deliver 5 Mbps per microflow if they are 10 whereas it would be 25 Mbps if there are just two microflows inside. Consequently, a microflow in an aggregate (supposed to give highest quality) may get a worse service (than a microflow in any other aggregate) if the aggregate contains a big number of microflows. This can be avoided by PHBs which are microflow aware. It's typically that kind of complexity that we would like to avoid in Differentiated Services deployment.

2.2 LOSS

The loss is often determined in terms of percentage of total data transmitted. Therefore, defining a certain loss ratio for an aggregate can easily be scaled down to all its microflows. Although the loss-based service differentiation does not require microflow aware PHB, it is rendered tedious when combined with packet precedence levels within an aggregate as explained below.

There are three packet drop precedences in an aggregate. The precedence of a packet defines how much it is prone to be discarded in case of congestion. The precedence level of a packet may be selected by the application or by an edge router. Introducing two levels of services differentiation (aggregates & precedences within an aggregate) based on a same metric (i.e. loss) needs to implement extra control and intelligent discard mechanisms. This is to manage all the thresholds (for aggregates & precedences) not only to respect the relative quality of services, at all loads, among aggregates, but also to ensure the relative quality of services among packets of different drop precedences within an aggregate.

2.3 DELAY

The 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. If a class X should have lesser delay (i.e. better service) than class Y then the

queue length of X should be kept smaller than that of Y. It can be done either by discarding packets at a higher rate or by serving the queue at higher rate. Discarding packets at higher rates, although keeps the delay shorter, does not offer a reliable service for loss-sensitive applications. On the other hand, servicing a class with a higher rate, so as to limit its queue length, offers a shorter delay as well as a better throughput to its applications. A delay-based service differentiation is thus required to have a self-regulation property, i.e. the service rate for an aggregate would then be calculated dynamically. Note that this dynamic calculation of the service rate does not need to be microflow aware and preserves the service differentiation at all loads. Self-regulation of an aggregate's service rate can be done by knowing just the current queue length of the aggregate and its relative quality index (refer to section 3).
Let us consider that the transport protocol is TCP as is the case with most of the Internet applications these days. The throughput of a TCP application depends on two factors, refer to relation 1 from [5]: RTT (delay[1]) and loss probability.

$$throughput = \frac{Constant}{RTT * \sqrt{loss\_probability}} \tag{1}$$

The self-regulation property of delay-based service differentiation ensures that an aggregate experiences packet loss, during congestion, in proportion to its current load. Consequently, the loss probability of all microflows of all aggregates tend to attain a same value at a DS node. Naturally, an application with lesser delay will end up getting a better goodput, an expected compensation for paying more.

2.4 CONCLUDING REMARKS

We presented three metrics for service differentiation among aggregates. Bandwidth is dropped as it requires the microflow aware PHB whereas the loss metric, when coupled with packet precedence level, is tedious to manage. The delay-based service differentiation, on the other hand, is easy to self-regulate and is microflow independent. We select the delay as a metric for service differentiation and present it formally in the next section.

3. FORMAL DESCRIPTION OF DELAY-BASED SERVICE DIFFERENTIATION

We consider a generic case of N backlogged classes. Let q represents the quality index associated with an aggregate i and service differentiation among aggregates follows the relation:

$$q_1 \leq q_2 \leq q_3 \cdots \leq q_N \tag{2}$$

The relation 2 declares the class N with the highest quality index and thus ought to be serviced better than all other aggregates. Let the quality index represents the delay-based (relative) service of an aggregate and d_i represents the delay for an aggregate i, then:

$$\frac{q_i}{q_j} = \frac{d_j}{d_i} \tag{3}$$

We introduce now a self-regulation property in the model. It means that the service rate r_i of an aggregate i is modified with respect to its current load, determined by its current buffer occupation, b_i. This self-regulation is *weighted* as it takes into account the aggregates quality index also:

$$\frac{r_i}{r_j} = \frac{b_i}{b_j} * \frac{q_i}{q_j} \tag{4}$$

In order to maintain the scheduling server work conserving,$\sum^N r_i = r$ where r is the speed of the scheduling server, we may rewrite the relation 4:

$$r_i = \frac{r b_i q_i}{\sum_{j=1}^{N} b_j q_j} \tag{5}$$

The above relation adjusts the service rate of an aggregate as its queue length changes but without violating the relative service differentiation among aggregates.

4. EX-VC ALGORITHM

This section presents an Extended VirtualClock (Ex-VC) scheduling algorithm and emulates the model presented in section 3. The Ex-VC algorithm has an additional instruction of self-regulation compared to the traditional VC algorithm [7], 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[2]. 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 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 $flow_i$ and VS_i (Virtual Spacing). The $flow_i$ registers are initially set to zero and $VS_i = \frac{L_i^k}{r_i}$ 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:

```
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*/
    • $VS_i = \frac{L_i^k}{r_i}$
    • $stamp_i^k = max(v(t), flow_i) + VS_i$
    • $flow_i = stamp_i^k$
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: In [3], a similar delay-based approach for service differentiation is presented. It studies two schedulers, Backlog-Proportional Rate (BPR) and Waiting Time Priority (WTP). 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. Both schedulers require separate queues per aggregate (note that this constraint does not exist in the Ex-VC scheduler). The simulation results in [3; 9] show that the WTP is significantly better than the BPR. We envisage comparing the Ex-VC with the WTP in our future simulations.

5. SIMULATIONS

We simulate four AF classes: AF4, AF3, AF2 and AF1. These aggregates have relative quality indexes as: $q_4 = 4, q_3 = 3, q_2 = 2$ and $q_1 = 1$. We define a warm-up period during which the rate of packet arrival is higher than the packet service rate (i.e. the scheduler speed r). This makes the queue lengths grow. Once the warm-up period is over, the packet arrival rate becomes equal to packet service rate. Moreover, we define three scenarios of packets arrival: symmetrical, equal and asymmetrical (refer to figure 1).

- In the symmetrical packets arrival scenario, the queues are loaded proportionally to their delay guarantees. That is to say that aggregate AF4 will receive packets at rate half of that at which aggregate AF2 would receive the packets. Remember that aggregate AF4 experiences half the delay of AF2. This yields a buffer loading where Ex-VC algorithm self-regulates at a rather easy-going pace.

- In the equal packets arrival rate scenario, all aggregates receive packets at equal rates regardless of their quality indexes (i.e. delay guarantees).

- The third scenario, asymmetrical packets arrival, tests the Ex-VC algorithm in tending-to-worst buffer loading configuration and algorithm self-regulates at a hard-going pace. Here, queues are loaded inversely proportionally to their delay guarantees. In other words, the aggregate AF4 will receive packets at rate double of that at which aggregate AF2 would receive.

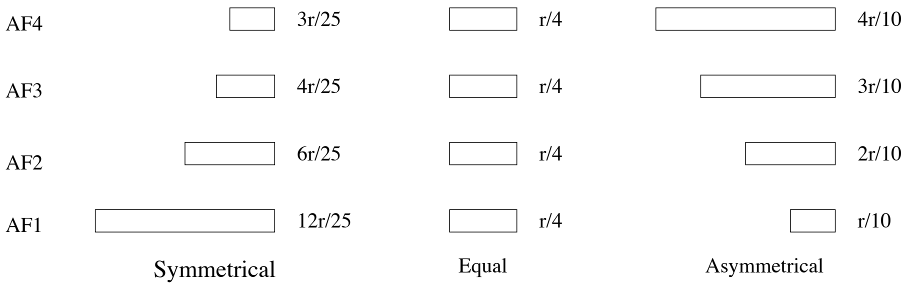

Figure 1 The rate of packet arrivals in three scenarios of buffer loadings

For each simulation type, we analyse two parameters: aggregate's service rates and aggregate's packets delay. All results are shown on the same respective scale. The simulation parameters are: warm-up period is 500 packet slots, simulation time is 10000 packet slots, buffer over-loading factor during warm-up is 2 and service scheduler speed r is 1 packet/time-slot. The first packet of AF1 arrives at $t = 0$, that of AF2 at $t = 1$, that of AF3 at $t = 2$ and that of AF4 at $t = 3$.

- Figure 2: We can notice that, by imposing symmetrical packet arrivals during the warm-up period, queue lengths reach an ideal relative load in order to offer the desired delay-based service differentiation. So it takes a relatively small time for the algorithm to stabilise itself once the arrival rates become equal (post warm-up phase). The aggregates experience relative packet delays in accordance to their respective quality indexes. The packets of the AF4 aggregate ($q_4 = 4$) have a delay four times shorter than those of aggregate AF1 ($q_1 = 1$).

- Figure 3: With equal rates of packet arrivals during the warm-up time, queues do not grow in required relative sizes and the algorithm self-regulates significantly (observe the crossing and fluctuating service rate curves) and takes longer than figure 2 to reach stabilisation. In post warm-up phase, packet arrivals are symmetrical and the algorithm works at easy-pace and ensures the relative service differentiation among aggregates.

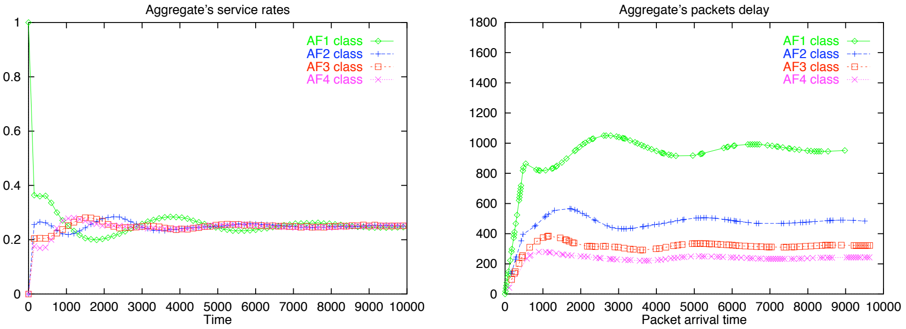

Figure 2 Symmetrical warm-up and equal post warm-up packet arrivals

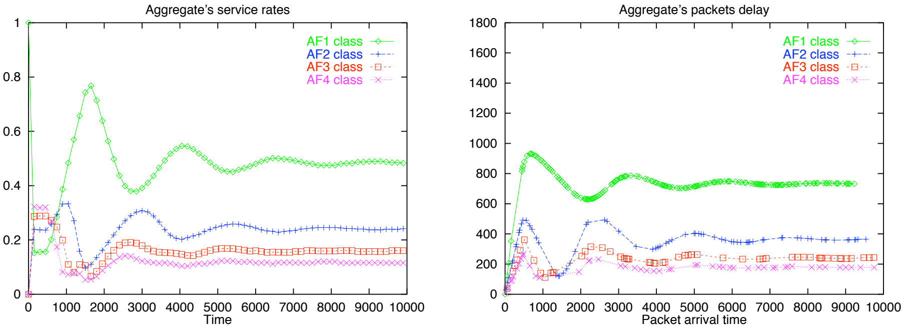

Figure 3 Equal warm-up and symmetrical post warm-up packet arrivals

- Figure 4: After an equal packet arrival warm-up period, the packet arrival rates are then changed to asymmetric rates. The algorithm self-regulates with changing queue lengths (due to changing packet arrival rates). We notice that aggregates suffer individually, though, more delays[3] (due to asymmetrical packet arrivals), relative delay differentiation is still respected.

- Figure 5: The asymmetric packet arrivals during the warm-up period have a great influence on the algorithm behaviour even though post warm-up packet arrival is symmetrical. This is because of the fact that queues are over-loaded (arrival rate is twice the service rate) during warm-up. Hard self-regulation is needed to respect the desired relative quality of service between aggregates and a longer period to reach stability.

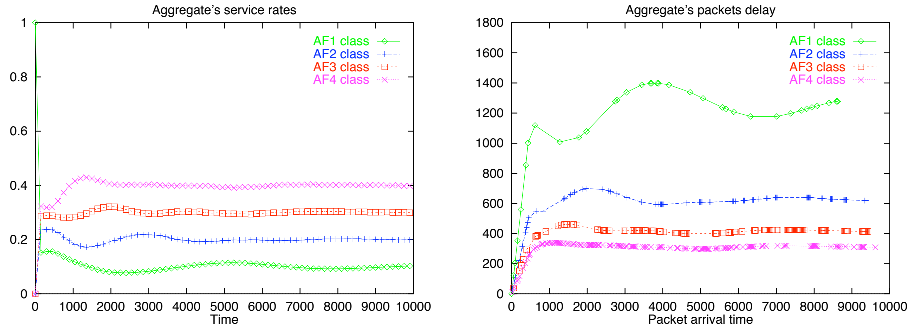

Figure 4 Equal warm-up and asymmetric post warm-up packet arrivals

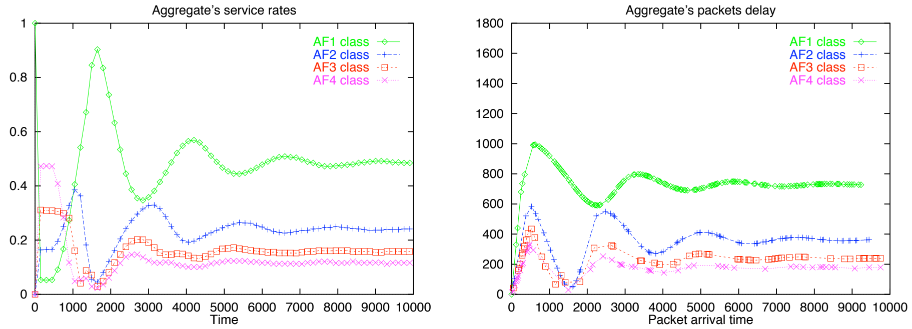

Figure 5 Asymmetric warm-up and symmetrical post warm-up packet arrivals

6. CONCLUSION

This paper evaluates three possible metrics (bandwidth, loss and delay) for service differentiation among four AF classes. Each definition is individually analysed: bandwidth-based service differentiation suffers from scalability problem, loss-based definition lacks in simplicity, delay comes out as the rational quality metric to maintain a relative service differentiation among all aggregates independently of number of microflows in the aggregates. A formal model of delay-based service differentiation is described and is tested via simulations. The simulation results prove that the proposed model provides differentiated service to all aggregates in proportion to their quality indexes. Moreover, this differentiation is respected at all queue loads.

For a PHB to be defined completely, we need, in addition to a scheduling algorithm like Ex-VC, an algorithm which decides whether to accept or discard a packet, at it's arrival, depending upon the factors like congestion level and packet drop precedence. The two algorithms (packet accept/discard and scheduler), when implemented together at a node, helps ensuring the SLAs (Service Level Agreements). Currently we are working on packet accept/discard algorithms and on their simulations for TCP flows when coupled with the Ex-VC scheduler [8].

Notes

[1] The delay comprises queueing delay as well as transmission delay. The transmission delay, being insignificant (due to high speed links), does not contribute much in RTT. On the other hand queueing delay stays a non-negligible factor in RTT estimation.

[2] 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.

[3] Note that the average delay value (calculated on the whole buffer) per packet is same for all the configurations.

References

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

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

[3] 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/).

[4] P. Hurley, J. Y. Le Boudec, M. Hamdi, L. Blazevic, P. Thiran. "The Asymmetric Best-Effort Service". *SSC/1999/003,* (http://icawww.epfl.ch).

[5] M. Mathis, J. Smeke, J. Mahdavi and T. Ott. "The macroscopic behaviour of the TCP congestion avoidance algorithm". *Computer Communication Review,* July 3 1997.

[6] Y. Boram, J. Binder, S. Blake, M. Carlson, Brian E. Carpenter. "A Framework for Differentiated Services". *<draft-ietf-diffserv-framework-02.txt>,* Feb. 1999.

[7] L. Zhang. "VirtualClock: A new traffic control algorithm for packet switching". *ACM Transactions on Computer Systems, 9(2), May 1991.*

[8] M. Tufail, G. Jennes & G. Leduc. "Attaining per flow QoS with class-based differentiated services". To appear in *SPIE Symposium on Voice, Video and Data Communications, Conf: Broadband Network, Sept 1999.*

[9] S. De Cnodder and K. Pauwels. "Relative delay priorities in a differentiated services network architecture". *Alcatel Alsthom CRC (Antwerp) deliverable, May 1999.*