Tuning Delay Differentiation in IP Networks using Priority Queueing Models

Pedro Sousa, Paulo Carvalho, and Vasco Freitas

Universidade do Minho, Departamento de Informática,
4710-059 Braga, Portugal
email: {pns,paulo,vf}@uminho.pt

Abstract. This article evaluates the use of Priority Queueing models to achieve delay differentiation in IP networks operating under the Class of Services paradigm. Three models are considered: proportional model, additive model and a novel hybrid schema based on the upper time limit model. The characteristics, behaviour and viability of these models are analysed as regards traffic delay differentiation. The impact of each model on traffic aggregation and on individual flows is also evaluated. This study is complemented by the analysis of delay differentiation from an end-to-end perspective. An adaptive differentiation mechanism is also proposed and discussed.

1 Introduction

The new Internet applications require the rethinking of network protocols. To fulfill application requirements, new philosophies oriented to Quality of Service (QoS) provision are under research and development [1]. Two architectures have been distinguished in this domain: the Integrated Services [2] and the Differentiated Services [3]. Due to its simplicity and capacity of co-existing with the actual TCP/IP protocol stack, DiffServ architecture has been pointed out as a solution to provide a limited set of QoS profiles to users.

In this work, special attention is given to a particular QoS parameter - delay - and to scheduling mechanisms which are able to obtain delay differentiation. This delay differentiation can be extremely useful to integrate real-time applications, such as voice/video and other delay-sensitive applications [4]. Even when admission control mechanisms or reservation protocols (e.g. RSVP [5]) are not present, acceptable QoS can be obtained in the presence of an appropriate delay differentiation mechanism. Additional models can be useful to provide resources to the classes/flows (e.g. bandwidth [6, 7]) or in more relaxed models, to provide applications with adaptive and tolerant mechanisms [8–10].

This study examines the behaviour of three different delay differentiation models. The Proportional Model was considered as an efficient form to assure a proportional delay differentiation between traffic classes [11–13]. The Additive Model constitutes an alternative way to differentiate delays [12]. These two models are revisited and additional studies including flow granularity and end-to-end perspectives are evaluated. A more rigid schema called Upper Time Limit Model is also discussed and a novel hybrid differentiation mechanism is presented. In this context, this paper intends to pursue and complement the work presented in [11–13] and investigate an Upper Time Limit model. For this purpose a *Network Simulator* [14] based testbed was implemented in order to analyse the models responsiveness. The developed testbed was validated with mathematical results, including Priority Queueing Theory and Conservation Law confirmations. The three differentiation models are studied for short-time scales and for different load conditions. The goal is to understand the behaviour of each one according to the configuration parameters. Apart from the ability of each model to differentiate delay, jitter is measured for individual flows sharing a class. An adaptive differentiation mechanism is also proposed and discussed. Finally, the end-to-end relative differentiation between flows sharing a common set of differentiation nodes is investigated.

2 Proportional, Additive and Upper Time Models

Proportional, Additive and Upper Time Limit models belong to Priority Queueing (PQ) models [15]. In PQ models each queue is ruled by a priority function that varies over time (Time-Dependent Priorities). Different models can be implemented using Time Dependent Priorities. The nature of the priority function and its configuration parameters define the behaviour of the service assigned to each queue. The following subsections review the three models briefly. The study considers N distinct classes $C_{i(0 \leq i \leq N-1)}$ having C_0 the highest priority.

2.1 Proportional Model

Let $p_i(t)$ be the priority function associated with the queue i and U_i the corresponding differentiation parameter. In the proportional model this function is given by (1), with t_0 denoting the arrival time of packet to queue i and $U_0 > U_1 > ... > U_{N-1}$. The behaviour of (1) for two packets belonging to distinct classes is depicted in Fig. 1(a). As seen U_i represents the slope of the priority function. The expected behaviour of a scheduler operating under (1) is that, under heavy load conditions, the relation (2) is valid for all classes, i.e. $0 \leq i, j < N$, where $\bar{d}_i, \bar{d}_j$ are the mean queueing delays of the classes i and j. In other words, the higher the differentiation parameter is, the lower the delay in the class will be. Furthermore, the proportional relation expected in the delays results from the proportionality in the differentiation parameters.

$$p_i(t) = (t - t_0) * U_i \qquad (1) \qquad\qquad \frac{U_i}{U_j} \approx \frac{\bar{d}_j}{\bar{d}_i} \qquad (2)$$

2.2 Additive Model

The additive model differentiates queues by an additive constant as expressed in (3), with $U_0 > U_1 > ... > U_{N-1}$. In this option, the priority difference between

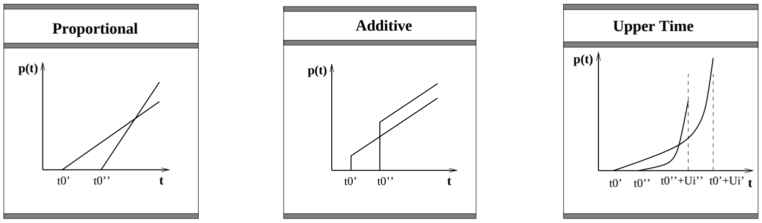


Fig. 1. (a) Proportional model (b) Additive model (c) Upper Time Limit model.

two packets remains constant over time, as depicted in Fig. 1(b). The interesting point in this model is to study the possibility of achieving additive differentiation in class delays, as expressed by (4). The equation (4) denotes that high priority classes may have a delay gain over low priority classes similar to the difference between the differentiation parameters. If this is true, it is an effective solution to spread class delays by a predefined value.

$$p_i(t) = (t - t_0) + U_i \quad (3) \qquad [\bar{d}_i - \bar{d}_j] \approx [U_j - U_i] \quad (i > j) \quad (4)$$

2.3 Upper Time Limit Model

The Upper Time Limit is a more rigid schema than additive and proportional models as it imposes a finite queueing delay. The idea is to define a boundary (reflected in U_i) for the packet queueing time (see (5)). In this model, the lower the boundary time is, the higher the priority function slope will be. At the limit ($(t - t_0) \geq U_i$) the server is *forced*[1] to dispatch the packet waiting service (see Fig. 1(c)). This model protects high priority classes, aiming that packets remain in queue for a maximum value U_i. In this model $U_0 < U_1 < ... < U_{N-1}$. Relations (2) and (4) for the Proportional and Additive models were obtained by the division and difference between the corresponding priority functions. The ratio (R) between priority functions of adjacent classes is defined[2] in (6) and represented in Fig. 2(a).

$$p_i(t) = \begin{cases} \frac{(t-t_0)}{U_i - t + t_0} \;\; if \;\; t < t_0 + U_i \\ \\ \infty \;\; if \;\; t \geq t_0 + U_i \end{cases} \quad (5) \quad R_{\frac{i}{i+1}} = \frac{p_i(t)}{p_{i+1}(t)} = \frac{\frac{t}{U_i - t}}{\frac{t}{U_{i+1} - t}} = \frac{U_{i+1} - t}{U_i - t} \quad (6)$$

The function evaluates roughly in a constant proportional mode (approximately with a value of $(\frac{U_{i+1}}{U_i})$ between the classes (white area), and as the time limit arrives, the function increases (grey area) tending then to infinity (black area). Our interest in this model is to use its capabilities to limit the queuing delay on the higher priority class. This class is oriented to extreme delay-sensitive

[1] Obviously when congestion occurs, or the load of high priority classes becomes very high, packets can be dropped or the waiting time limit exceeded.

[2] For simplicity simultaneous packet arrival times are assumed, i.e. t_0 is eliminated.

applications where a bound on delay is mandatory. Our objective is to establish such delay bounds and, simultaneously, achieve proportional differentiation between the other classes. This can be obtained by combining differentiation parameters conveniently. Fig. 2(b) shows an example where $Clas_1$ is *protected* by a realistic upper time limit, and $Class_2$ and $Class_3$ with *virtual* parameters (i.e. a queueing time limit much higher than the expected for the class and $U_2, U_3 \gg U_1$). Proportionality between $Class_2$ and $Class_3$ is obtained by configuring parameters as explained in Section 2.1, considering now that higher classes have lower differentiation parameters. As shown in Fig. 2(b) there is a server working region (slashed area) where hybrid differentiation is feasible.

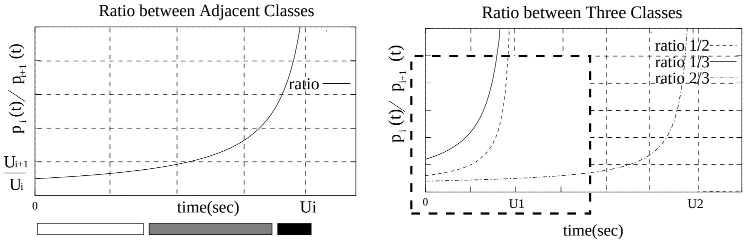


Fig. 2. (a) Ratio between priority functions (b) Expected server working region.

3 Performance Evaluation

3.1 Experimental Framework

The differentiation mechanisms were implemented and tested in the *network simulator* (NS). Each mechanism determines the scheduler behaviour. The schedulers were implemented in C++ from *Queue Class* inheritance. Proprietary queues and monitors were also developed in order to collect results from the tests. Fig. 3(a) shows the implemented architecture. At Otcl level, the user selects the scheduler, defines the differentiation parameters of the queues/classes and provides classification information, i.e. $(packet_{flowid}, queue_{id})$ pairs. At the same level, the user indicates the state information granularity to be logged during scheduling. In the architecture core, the monitor module logs state information about flows/classes periodically for subsequently analysis.

3.2 General Comments

Simulation Tests: The models were tested in several simulation scenarios for different traffic patterns. The results presented here were obtained for the simulation scenario depicted in Fig. 3(b). $Class_A$ is used for on-off traffic (the duration of *on* and *off* periods follows a Pareto distribution with shape factor of

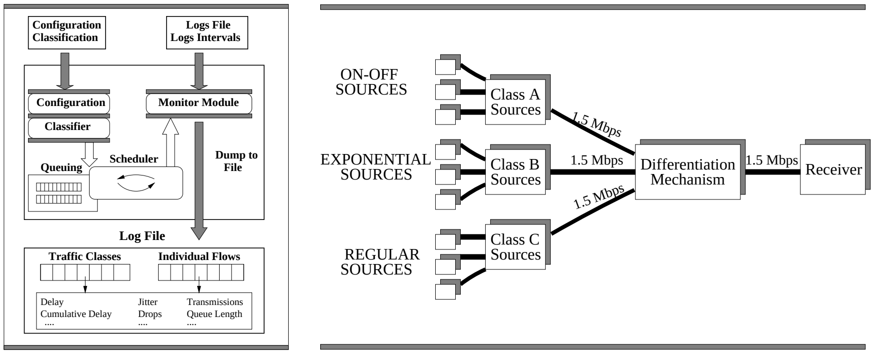


Fig. 3. (a) Developed Testbed (b) Simulation Scenario.

1.2). Additionally, $Class_B$ consists of Poisson traffic (exponential inter-arrival times), and $Class_C$ which includes regular traffic. A set of individual applications generates marked traffic for the corresponding class. The packet length is 500 *bytes*. In the delay differentiation examples presented in section 3.3, similar loads (≈33%) and queueing resources were also used for all classes. The server is connected to a 1.5 Mbps link[3]. The differentiation schemas were studied for heavy load conditions. The results are presented graphically where the *x axis* represents server packet transmission times, with a plot granularity of 80*ms*.

3.3 Delay Differentiation

This section intends to illustrate the differentiation characteristics for the three models. Fig. 4 to 5 show the delay differentiation behaviour for the models. For each model, the queueing delay is plotted by sampling interval (Fig. 4 to 5(a)(c)), and its average over the simulation period (Fig. 4 to 5(b)(d)).

Proportional Model: Fig. 4(a)(b) shows the performance of the proportional model for differentiation parameters $(U_A, U_B, U_C) = (4, 2, 1)$. The results show a proportionality between the class delays. In fact, $Class_C$ (low priority class) has a delay which is on average twice the obtained by $Class_B$, which in turn has a queueing delay around two times higher than $Class_A$. This behaviour is in accordance with equation (2) assuring that for heavy load, and for acceptable configuration parameters, the proportionality relations expressed by U_i parameters generate proportionality relations between class delays.

Additive Model: Fig. 4(c)(d) illustrates the obtained results for the Additive model when $(U_A, U_B, U_C) = (0.030,\ 0.010, 0.0)$. This means that under heavy load conditions $Class_A$ may have an advantage near 20*ms* over $Class_B$. Using the same reasoning $Class_B$ may have an advantage near 10*ms* over $Class_C$ and, transitively, $Class_A$ an advantage near 30*ms* over $Class_C$. In fact, the results presented in Fig. 4(c)(d) exhibits a behaviour which verifies equation (4). This approach provides additive class delay differentiation effectively.

[3] In which there is a queue with the architecture presented in Fig. 3(a).

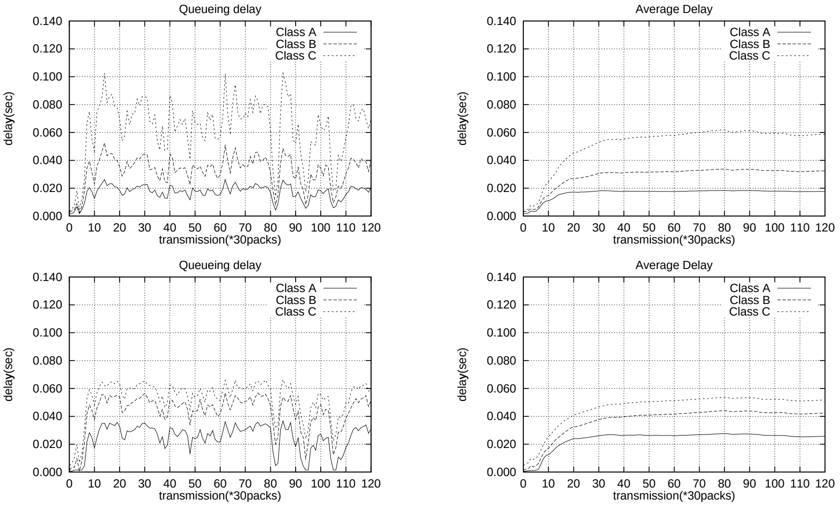


Fig. 4. (a) Queueing delay (b) Average delay for the Proportional Model, $(U_A, U_B, U_C) = (4, 2, 1)$ (c) Queueing delay (d) Average delay for the Additive Model, $(U_A, U_B, U_C) = (0.030, 0.010, 0.0)$.

Upper Time Limit Model: In this model two objectives were established: *(i)* obtain a queueing delay for $Class_A$ around $5ms$; and *(ii)* achieve proportional differentiation between $Class_B$ and $Class_C$. Fig. 5(a)(b) shows the results for this model using $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$. As explained in section 2.3 such configuration leads to a queueing delay below $5ms$ for $Class_A$ and a queueing delay for $Class_B$ two times lower than the obtained for $Class_C$. The results prove that this *hybrid* mechanism is feasible, and achieves differentiation successfully. As the results illustrate, as long as $Class_A$ queueing delays are confined to an upper-bound value ($5ms$), $Class_B$ and $Class_C$ queueing delays keep a proportional relation. Fig. 5(c)(d) represents a scenario where $Class_A$ has an upper limit of $10ms$ and $Class_C$ is assigned a queueing delay 10% higher than $Class_B$. This results in an approximation of $Class_B$ and $Class_C$ delays.

From the previous simulation examples one can argue that, under heavy load conditions, each model constitutes an effective differentiation mechanism and a good tuning scheme to provide network elements with delay differentiation capabilities. Additionally, the differentiation behaviour is achieved even in short-time scales ($80ms$ in the example). However, there are additional comments that must be made. For particular load conditions, all models can present some feasibility problems. This does not mean erroneous relative differentiation, but for particular scenarios (e.g. priority classes very loaded) the gap between queueing delays is lower than the one expressed by (2) and (4). The same occurs for the Upper Time Limit model if the traffic load on a high priority class impairs the delay limit imposed by (5). These occasional feasibility problems do not affect the essential conditions of relative differentiation, i.e. $\bar{d}_0 < \bar{d}_1 < \bar{d}_2 < ... < \bar{d}_n$.

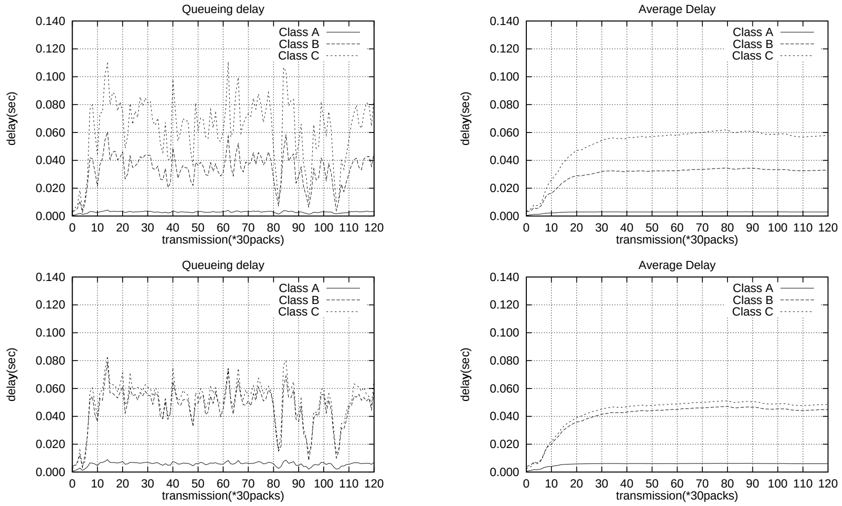


Fig. 5. (a) Queueing delay (b) Average delay for Upper Time Limit Model, $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$ (c) Queueing delay (d) Average delay for Upper Time Limit Model, $(U_A, U_B, U_C) = (0.010, 0.100, 0.110)$.

3.4 Flow Granularity

This section studies the models behaviour at flow level. The aim is to examine how the delay-oriented QoS provided to each class is extended to the flow level. The knowledge of such flow characteristics can be useful since clever applications mechanisms (e.g. adaptive) can be deployed to improve their performance.

Per Flow Queueing Delay Consistency: The differentiation mechanisms should be fair for flows sharing a class. This means that for a generic time interval $[t_0, t]$, the queueing delay associated with each class should evenly affect flows belonging to that class. To verify this, the average queueing delay for two randomly selected flows of each class on each model is plotted in Fig. 6. As shown, different priority flows share the same delay relations of the classes. Additionally, flows in the same class have identical average queueing delays. This demonstrates the fairness of differentiation mechanisms even at flow level.

Delay Variation/Jitter: Jitter is an important measure from the applications' perspective. Many real time applications adapt themselves to network conditions. For example, applications involving isochronous media need to monitor delay and jitter in order to regulate the receiver's playout buffer [16]. If an adaptive perspective is assumed [9], applications can move across different traffic classes in order to achieve more suitable delay/jitter. Additionally to average queueing delays relations among classes/flows, it is important to find foreseeable relations for jitter. The distance between the max(+)/min(-) delay lines gives an estimative of the jitter experienced by the flow. The results show that jitter either is reduced or does not change significantly when the flow moves to high

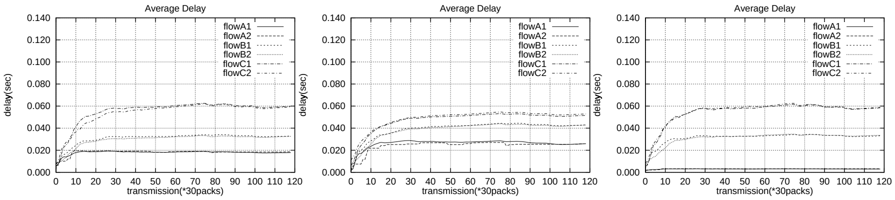


Fig. 6. Average queueing delay at Flow Level - (a) Proportional Model $(U_A, U_B, U_C) = (4, 2, 1)$, (b) Additive Model $(U_A, U_B, U_C) = (0.030, 0.010, 0.0)$, (c) Upper Time Limit Model $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$.

priority classes. Note that jitter issues are relevant for other than heavy load conditions (e.g. Fig. 7). In fact, it is possible that at t_0, when server is under heavy load conditions, $Class_i$ achieves a queueing delay $\bar{d_i}$, and subsequently at t_1, due to a load decrease on the server, the delay experienced by $Class_i$ may decrease sharply to a very low value, or even to zero. Therefore, jitter can assume a value $\bar{d_i} - 0 = \bar{d_i}$. Consequently, relations between jitter on each class are dependent on the relations between those classes' queueing delays.

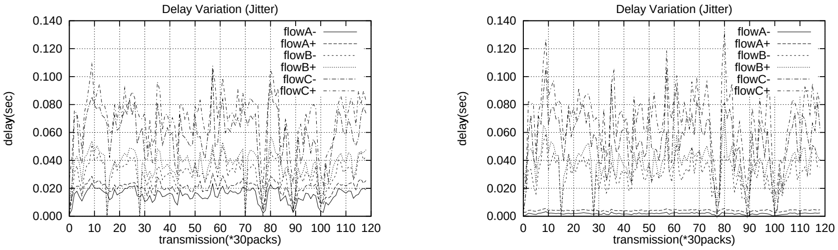


Fig. 7. Jitter at Flow Level, (a) Proportional $(U_A, U_B, U_C) = (4, 2, 1)$ (b) Upper Time $(U_A, U_B, U_C) = (0.005, 0.100, 0.200)$.

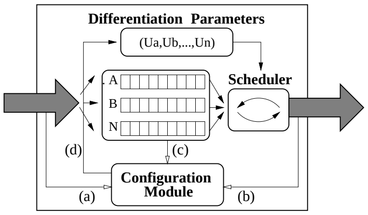
Playout Buffer dimensioning: The knowledge of jitter characteristics can be useful for playout buffer dimensioning if an adaptive behaviour of the application is assumed. When moving to low (high) priority classes, the application should protect itself by increasing (decreasing) the playout buffer. The ratio $\bar{d_i}/\bar{d_j}$ should guide the dimensioning degree adopted by the application. For the proportional model and using formula (2), a possible update strategy[4] is presented by (7), where $f_{i \Rightarrow j}$ is a flow moving from $Class_i$ to $Class_j$ and b_l the playout buffer length. Similar considerations can be made for the other models.

$$f_{i \Rightarrow j} : new(b_l) = old(b_l) * \frac{U_i}{U_j} \qquad (7)$$

[4] In this case it is assumed a single node between the sender and the receiver.

4 Adaptive behaviour of differentiation mechanisms

Network elements may assume adaptive behaviours (e.g. [17] presents a dynamic regulator mechanism for real-time traffic and [18] an adaptive packet marking scheme to achieve throughput differentiation). In our opinion, this principle can be applied at scheduler level in order to improve network resources usage (e.g. [19] suggests an Adaptive-Weighted Packet scheduler for premium service). The aim is to allow scheduling to react to certain operational conditions, modifying differentiation parameters *on-the-fly*. A possible adaptive scheduler architecture, which can be easily adopted in the three models, is proposed in Fig. 8(a).



```
$A_T \Leftarrow EstimateTotalArrivalLoad()$
$A_A \Leftarrow EstimateClassArrivalLoad(A)$
$Ratio \Leftarrow \frac{A_A}{A_T}$
$if(0 \leq Ratio \leq 0.4) \Rightarrow (U_A, U_B, U_C) = (16, 4, 1)$
$if(0.4 < Ratio \leq 0.6) \Rightarrow (U_A, U_B, U_C) = (4, 2, 1)$
$else \Rightarrow \{(U_A, U_B, U_C) = (2, 1.5, 1)$
```

Fig. 8. (a) Adaptive differentiation mechanism (b) Configuration module behaviour.

The *Configuration Module* has three distinct inputs: *(a)* input traffic information (e.g. aggregated load per class), *(b)* output traffic information (level of throughput share) and *(c)* QoS node state information (e.g. queueing delay, packet loss, jitter per class). Combining this information, the *Configuration Module* can modify the differentiation parameters *(d)* to achieve a certain objective. As a simple example, consider the *Configuration Module* has having the behaviour described in Fig. 8(b). When the thresholds are violated, new differentiation parameters are evaluated in order to deny excessive throughput allocation to high priority class. As shown in Fig. 9, the node becomes reactive to $Class_A$ load limits violation. As a consequence, new parameters are assigned to each class, resulting in a delay approximation between classes. Other variants of this mechanism for the three differentiation models will be matter of further research.

5 End-to-end relative delay differentiation

This section studies the proportional, additive and upper time limit models from an end-to-end perspective and establishes the corresponding upper bound limits for delay differentiation. The definition of a differentiation domain aims to achieve a foreseeable relative differentiation for flows[5] crossing a common set of

[5] As explained in subsection 3.4 the delay differentiation achieved for traffic aggregates is valid at flow level.

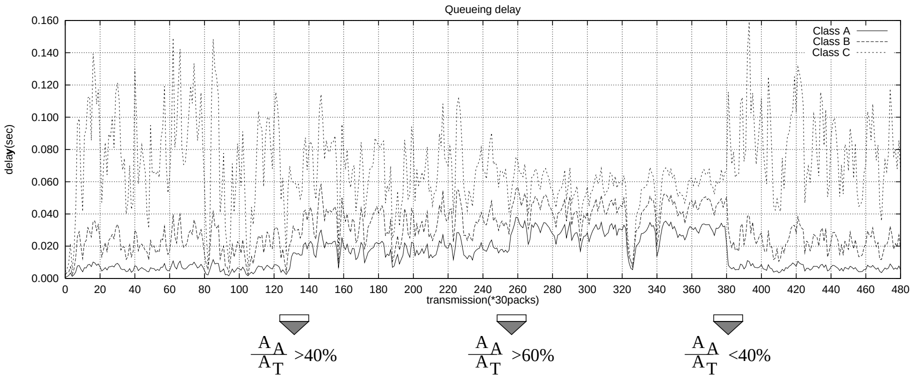


Fig. 9. Adaptive Proportional Model, (U_A, U_B, U_C)=(16,4,1)$\Longrightarrow$(4,2,1)$\Longrightarrow$(2,1.5,1).

nodes in a given time period. This domain consists of M differentiation nodes, $0 \leq j \leq M-1$, traversed by individual flows. Let $\bar{d}_i^j$ be the average queueing delay of $Class_i$ at node j. If a flow crosses M servers $(0 \leq j \leq M-1)$ then the end-to-end average queueing delay $(\bar{d}_i^*)$ of $Class_i$ can be expressed by (8). For additive differentiation and under heavy load, (9) can be applied to a generic server, where U_i^j is the differentiation parameter of $Class_i$ in node j. Considering a flow crossing M independent nodes, (9) becomes (10), which combined with (8) results in (11).

$$\bar{d}_i^* = \sum_{j=0}^{M-1} \bar{d}_i^j \tag{8}$$

$$(\bar{d}_{i+1}^j - \bar{d}_i^j) \approx (U_i^j - U_{i+1}^j) \Rightarrow \bar{d}_{i+1}^j \approx (U_i^j - U_{i+1}^j) + \bar{d}_i^j \tag{9}$$

$$\sum_{j=0}^{M-1} \bar{d}_{i+1}^j \approx \sum_{j=0}^{M-1} (U_i^j - U_{i+1}^j) + \sum_{j=0}^{M-1} \bar{d}_i^j \tag{10}$$

$$\bar{d}_{i+1}^* \approx \left[\sum_{j=0}^{M-1} (U_i^j - U_{i+1}^j) + \bar{d}_i^* \right] \tag{11}$$

Equation (11) is obtained considering all servers in the flows path under heavy load conditions and for feasible configurations, otherwise the distance between class delays can become smaller than that. Equation (12) denotes this aspect and establishes an upper-bound for the end-to-end additive differentiation behaviour between two adjacent classes. A constant ϵ is introduced due possible inaccuracies of the models when the average delays are measured in very small time scales and, simultaneously, the server is under high class load oscillations.

$$(\bar{d}_{i+1}^* - \bar{d}_i^*) < \sum_{j=0}^{M-1} (U_i^j - U_{i+1}^j) + \epsilon \tag{12}$$

For the proportional model, (9) is now replaced by (13). Considering again a generic case of M servers under heavy load conditions, equation (13) becomes (14). The right term of equation (14) can be expanded as expressed by (15).

$$\frac{\bar{d}^j_{i+1}}{\bar{d}^j_i} \approx \frac{U^j_i}{U^j_{i+1}} \Rightarrow \bar{d}^j_{i+1} \approx \left(\frac{U^j_i}{U^j_{i+1}}\right) * \bar{d}^j_i \qquad (13)$$

$$\textstyle\sum_{j=0}^{M-1} \bar{d}^j_{i+1} \approx \sum_{j=0}^{M-1} \left[\left(\frac{U^j_i}{U^j_{i+1}}\right) * \bar{d}^j_i\right] \qquad (14)$$

$$\frac{U^0_i}{U^0_{i+1}} * \bar{d}^0_i + \frac{U^1_i}{U^1_{i+1}} * \bar{d}^1_i + ... + \frac{U^{M-1}_i}{U^{M-1}_{i+1}} * \bar{d}^{M-1}_i \quad (15)$$

Defining now X and Y as (16), the equation (15) can be bounded by (17). Using the same arguments of the additive models and considering equations (8) (14) and (17), equation (18) gives an upper bound limit for end-to-end proportional delay differentiation between two adjacent classes.

$$X = \min_{0 \le j \le M-1} \left(\frac{U^j_i}{U^j_{i+1}}\right) \quad Y = \max_{0 \le j \le M-1} \left(\frac{U^j_i}{U^j_{i+1}}\right) \quad (16)$$

$$X * \left(\textstyle\sum_{j=0}^{M-1} \bar{d}^j_i\right) \le (15) \le Y * \left(\textstyle\sum_{j=0}^{M-1} \bar{d}^j_i\right) \qquad (17)$$

$$\left(\frac{\bar{d}^*_{i+1}}{\bar{d}^*_i}\right) < \max_{0 \le j \le M-1} \left(\frac{U^j_i}{U^j_{i+1}}\right) + \epsilon \qquad (18)$$

For the upper time limit model, a high priority class under feasible load conditions is expected to obtain a maximum queueing delay equivalent to the sum of the differentiation parameters associated with $Class_0$ along the path (19). Within the hybrid model presented in section 2.3, low priority classes obtain an end-to-end relation also expressed by equation (18).

$$\bar{d}^*_0 = \textstyle\sum_{j=0}^{M-1} \bar{d}^j_0 < \sum_{j=0}^{M-1} U^j_0 + \epsilon \qquad (19)$$

We aim to corroborate equations (12), (18) and (19), along with ϵ significance, using simulation. The knowledge of this end-to-end differentiation behaviour is fundamental to provide applications with effective adaptation.

6 Conclusions

This work assesses the use of PQ models to achieve foreseeable delay differentiation between traffic classes. In particular, three differentiation models are studied: proportional, additive and a hybrid one using an upper time limit model. A simulation testbed has been used and validated using theoretical models. All models show acceptable consistence in achieving the expected delay differentiation behaviour, i.e., both the proportional, additive and hybrid models distribute queueing delays among traffic classes. In our opinion, they are simple and useful to tune delay differentiation in IP networks. In particular, the hybrid differentiation mechanism shows consistence in limiting queueing delay on the highest priority class and, simultaneously, achieving proportional differentiation between the other classes. An additional study at flow aggregate level was carried out focusing on flow queueing delay consistence and jitter differentiation. In order to provide scheduling elements with adaptive behaviour, an adaptive differentiation architecture is proposed. Finally, the bounds for end-to-end delay differentiation between flows crossing a relative differentiation domain were determined for each model. As future work, the tuning process, which will be further consolidated, will include other performance aspects of differentiation domains using the differentiation delay strategies considered.

References

1. G. Armitage. *Quality of Service in IP Networks: Foundations for a Multi-Service Internet*, Macmillan Technical Publishing, Apr. 2000.
2. R. Braden *et al. Integrated Services in the Internet Architecture: an Overview.* RFC1633, Jul. 1994.
3. S. Blake *el al. An Architecture for Differentiated Services*, RFC2475, Dec. 1998.
4. M. Baldi. *End-to-End Delay Analisys of VideoConferencing over Packet-Switched Networks*, IEEE/ACM Trans. on Net., Vol. 8, N. 4, Aug. 2000.
5. R. Braden *et al. Resource Reservation Protocol (RSVP).* RFC2205, Sep. 1997.
6. I. Stoica and H. Zhang. *Providing Guaranteed Services without Per Flow Management*, In Proc. of SIGGCOMM'99, 1999.
7. Z. Zhang *et al. Decoupling QoS Control from Core Routers: A Novel Bandwidth Broker Architecture for Scalable Support of Guaranteed Services*, In Proc. of SIGCOMM'00, 2000.
8. P. Sousa and V. Freitas. *A framework for the development of tolerant real-time applications*, Computer Networks and ISDN Systems 30, 1531-1541, Dec. 1998.
9. T. Nandagopal, N. Venkitaraman. *Delay Differentiation and Adaptation in Core Stateless Networks*, Proc. of INFOCOM 2000, Tel Aviv, Israel - Volume 2.
10. I. Busse *et al. Dynamic QoS Control of Multimedia Applications based on RTP*, Computer Communications, Vol. 19, N. 1, pp. 49-58, Jan. 1996.
11. C. Dovrolis and P. Ramanathan. *A Case for Relative Differentiated Services and the Proportional Differentiation Model*, IEEE Network Magazine, 1999.
12. C. Dovrolis and D. Stiliadis. *Relative Differentiated Services in the Internet: Issues and Mechanisms*, In Proc. of ACM SIGMETRICS'99.
13. C. Dovrolis *et al. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling*, In Proc. of ACM SIGCOMM'99, 1999.
14. *ns* Documentation. http://www.isi.edu/nsnam/ns/ns-documentation.html
15. G. Bolch *et al. Queueing Networks and Markov Chains - Modeling and Performance Evaluation with Computer Science Applications*, John Wiley & Sons INC., 1998.
16. W. E. Naylor and L. Kleinrock. *Stream Traffic Communications in Packet Switched Networks: Destination Buffering Considerations*, IEEE Trans. on Communications, VOL. COM-30, No 12, 1982.
17. S. Introu and I. Stavrakakis. *A Dynamic Regulation and Scheduling Scheme for Real-Time Traffic Management*, IEEE/ACM Trans. on Net., Vol.8, N.1, Feb. 2000.
18. W. Feng *et al. Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet*, IEEE/ACM Trans. on Net.,Vol.7, N.5, Oct. 1999.
19. H. Wang *et al. Adaptive-Weighted Packet Scheduling for Premium Service* In Proc. of the IEEE Int. Conf. on Communications,Helsinki, Finland, Jun. 2001.