Relative Jitter Packet Scheduling for Differentiated Services

Thu Ngo Quynh (*) Holger Karl (**) Adam Wolisz(**) Klaus Rebensburg (*)

(*) Interdepartmental Research Center
for Networking and Multimedia Technology
PRZ / FSP-PV / TUBKOM

(**) Telecommunication Networks Group
Departement of Electrotechnic Enginnering
and Computer Science

Technical University of Berlin
Strasse des 17. Juni 136 10623 Berlin Germany
Email: {thu, klaus}@prz.tu-berlin.de, {wolisz,karl}@ee.tu-berlin.de

Abstract

There is a clear need for relatively simple and coarse methods for providing differentiated classes of service for Internet traffic to support various types of applications and specific business requirements. The differentiated services approach to providing quality of service in networks employs a small, well-defined set of building blocks from which a variety of aggregate behaviors may be built. In the Differentiated Service Architecture (DiffServ) exists *Relative Differentiated Services,* which seek to provide *per-hop, per-class relative services.* The *Relative Differentiated* S*ervices* approach can be further refined and classified to *Re)* for this *Relative Proportional Differentiated Service* Model, which will achieve proportional jitter between different classes.

Keywords: Differentiated Service, jitter, proportional scheduling.

1. Introduction:

Currently, the Internet can only provide a single service class - best effort - that requires no pre-specified quality of service (QoS) contracts and provides no minimum QoS guarantees for packet flows. To provide adequate service, some level of quantitative or qualitative determinism - IP services must be supplemented and that is what Quality of Service (QoS) protocols are designed to do. A number of QoS protocols have evolved to satisfy a variety of application needs: Intergrated Services (IntServ) and Differentiated Services (DiffServ).

The IntServ approach [1], [2] supports quantified services as minimum service rate or a maximum tolerable end-to-end delay or loss rate, but requires each router of the network to maintain per-flow state in order to process per-flow signalling messages on the control plane and to perform per-flow classification, scheduling, and buffer management on the data plane. This approach seems to be infeasible for core routers to perform all of the above actions efficiently when there are millions of flows traversing through the network simultanously.

The other approach, DiffServ [3], [4] is newer than the IntServ approach and proposes a coarser notion of quality of service, focusing primarily on aggregated flows in the core routers, and intends to differentiate between service classes rather than provide absolute per-flow QoS guarantees. In particular, access routers process packets on the basis of finer traffic granularity such as per-flow or per-organization and core routers do not maintain fine grained state, but process traffic based on a small number of Per Hop Behaviors (PHBs) [5] encoded in the packet header. Although other behaviors are possible, there are currently three standards ones: Assured Per Hop Behavior [6], Expedited Per Hop Behavior [7] and Virtual Wire Per Hop Behavior [8].

Since the DiffServ approach is still evolving, many of its aspects are not yet clear, but the DiffServ philosophy is to realize *Relative Differentiated Service* [9], which seeks to provide *per-hop per-class relative services,* wherein each router has *N* service classes, and *Class i is better (or at least no worse) than class (i-1) for 2 = i = N*, in terms of service metrics for that hop. For *relative differentiated service,* there are no absolute guarantees due to the lack of admission control or resource reservations. Consequently, the network cannot provide worst case bounds for any service metric. Instead, each router only guarantees that the service invariant is locally maintained, even though the absolute service might vary with network conditions.

Relative Differentiated Service can be further refined and quantified to *Relative Proportional Differentiated Service* [10] and *Relative Absolute Differentiated Service* [11]. According to the *Relative Proportional Differentiated Service* model, certain forwarding performance metrics are ratioed proportionally to the *class differentiated parameters* that the network operators choose. In the *Relative Absolute Differentiated Service* model, the network operator attempts to control the absolute metric spacing between classes.

There exist some proportioanl scheduling studies in order to achieve proportional bandwidth (WFQ and others), proportional delay (MDP [14], BPR [9], WTP [9]), proportional loss (LHB [10]). The problem of proportional jitter has not been studied so far.

However, providing proportional jitter will be required because jitter is important, too: for achieving an acceptable quality of sound and animated images, delay jitter limitation is required by both interactive and non-interactive applications involving digital continuous media. Delay jitter can be eliminated by buffering at the receiver [12]. However, the amount of buffer space and thus the delay due to buffering required at the receiver can be reduced if the network can provide some level of control of delay jitter. The reduction can be significant for high bandwidth communication. It therefore makes sense to ask whether the schemes for controling or bounding delays and loss rates can be extended to provide any level of delay jitter control and, if so, under what condition. As it turns out, the mechanism to control jitter reduces the amount of buffer space required not only in the receiver but also within the network [13].

On the other hand, we believe that most existing and emerging real-time applications are soft real time in nature, they are tolerant to occasional delay or jitter violations and hence do not require worst case bounds of delay or jitter [14].

Hence there is a need to create a scheduling algorithm for providing proportional jitter. Our goal is to show that it is possible to meet the requirement of *Relative Proportional Differentiated Service* Model in terms of delay jitter by using a simple new scheduling algorithm called RJPS (Relative Jitter Packet Scheduling).

The rest of the paper is organized as follows. In Section 2, we describe our algorithm at the router. In Section 3, we describe our simulation and evaluate the results in Section 4. Finally, Section 5 concludes the work and outlines further possible research on the direction.

2. Description of algorithm

2.1 The model for *Relative Proportional Differentiated Service ([10]):*

The proportional model states that certain class performance metrics should be proportional to the differentiated parameters that the network operator chooses. A generic description of the proportional differentation model follows. Suppose that $\bar{q}_i\ (t, t+t)$ is the performance measure for class *i* in the time interval $(t, t+t)$, where $t > 0$ is the *monitoring timescale*. If differentiation over short timescales is desired, the value of t should be relatively small. The proportional differentiation model imposes constraints of the following form for all pairs of classes and for all time intervals $(t, t+t)$ in which both $\bar{q}_i\ (t, t+t)$ and $\bar{q}_j\ (t, t+t)$ are defined:

$$\frac{\bar{q}_i(t, t+t)}{\bar{q}_j(t, t+t)} = \frac{c_i}{c_j}$$

where $c_1, c_2, \ldots, c_N$ are the generic *Quality Differentiation Parameters* (QDPs). The basic idea is that, even though the actual quality level of each class will vary with the class loads, the quality ratio between classes will remain fixed.

2.2 The model for *Relative Proportional Differentiated Service* for Jitter:

Based on the previous model, in the context of queueing delay jitter, we can write as follows:

$$\frac{\bar{j}_i(t, t+t)}{\bar{j}_j(t, t+t)} = \frac{\Delta_j}{\Delta_i} \qquad (1)$$

where parameters { Δ_i }are the *Jitter Differentiation Parameters* and $\bar{j}_i(t, t+t)$ is the average queueing delay jitter of class *i's* packets (class i is better than class j if $\Delta_i > \Delta_j$). In detail, we can write: the *Relative Proportional Differentiated Service* Model for jitter is characterized by the following *(N-1)* equations:

$$\begin{aligned} \bar{j}_1(t,t+t)\Delta_1 &= \bar{j}_2(t,t+t)\Delta_2 \\ \bar{j}_2(t,t+t)\Delta_2 &= \bar{j}_3(t,t+t)\Delta_3 \\ &\ldots\ldots \\ \bar{j}_{N-1}(t,t+t)\Delta_{N-1} &= \bar{j}_N(t,t+t)\Delta_N \end{aligned}$$

Leaving the problem of delay and jitter measure for further studies, we assume that jitter of one packet in a queue is the difference of queueing delay of this packet and the preceding packet in this class (this definition is based on the standards of IP performance metrics working group of IETF).

$$j^k = \left| d^k - d^{k-1} \right|$$

where d^k is the queueing delay of packet number k.

2.3 Description of forwarding mechanisms at the router

We will consider a *work-conserving* packet scheduler that serves N queues, one for each class. The choice of work-conserving forwarding mechanisms is important, because with a *non-work-conserving* scheduler it is possible to set the jitter spacing between classes to arbitrary levels. We believe that only *work-conserving* forwarding mechanisms will be used in practice, because of the competition for the best possible service between providers; this is mainly a non-technical issue however.

RJPS algorithm: Suppose that each router has a pre-specified number of jitter classes N. Each jitter class is served by a single first-in-first-out (FIFO) packet queue. Packets of a flow belonging to a jitter class i are queued in the corresponding queue in each router that the flow passes through. All flows with the same jitter class specification share the same FIFO queue at the router. The goal of our scheduling algorithm is to serve the packets such that the *short term average jitter* (calculated over a moving window) and *long term average jitter* (calculated from the beginning time of simulation) experienced by packets in a jitter class will satisfy equation (1) for all pairs of i and j.

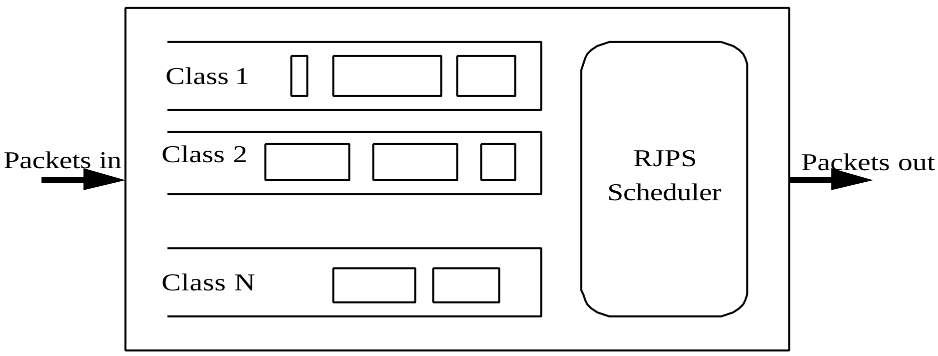


Figure 1. RJPS scheduler

Assuming that: in class i at *time* t, for each packet number k we know the arrival time (t_i^k), the starting time of transmission (T_i^k) and the transmission time TS_i^k.

- For all packets that have already been served, we call $j_i^*(t)$ the aggregate jitter experienced by all packets that have been served in the queue i at time t. This value is already determined, because all packets were served, and thus the queueing delays of these packets are already determined, too.

$$j_i^*(t) = \sum_k jitterofeachpacket = \sum_{k=1}^{s_i(t)} \left| d_i^k - d_i^{k-1} \right| = \sum_{k=1}^{s_i(t)} \left| (T_i^k - t_i^k) - (T_i^{k-1} - t_i^{k-1}) \right|$$

where d_i^k is the queueing delay of packet number k in class i, $s_i(t)$ is the number of packets served from jitter class i till time t.

- For all packets that are now queued in, we call $j_i^{\min}(t)$ minimum jitter for all packets that have already arrived. Assuming that no other packet will arrive for this class *i* in the future, this value can be calculated as:

$$j_i^{\min}(t) = \sum_{k=s_i(t)+1}^{s_i(t)+q_i(t)} \left| TS_i^k - (t_i^k - t_i^{k-1}) \right|$$

where TS_i^k is the transmission time of packet number *k* in the queue *i*, $q_i(t)$ is the number of packets that are now queued in this class. This formula is proved in Figure 2:

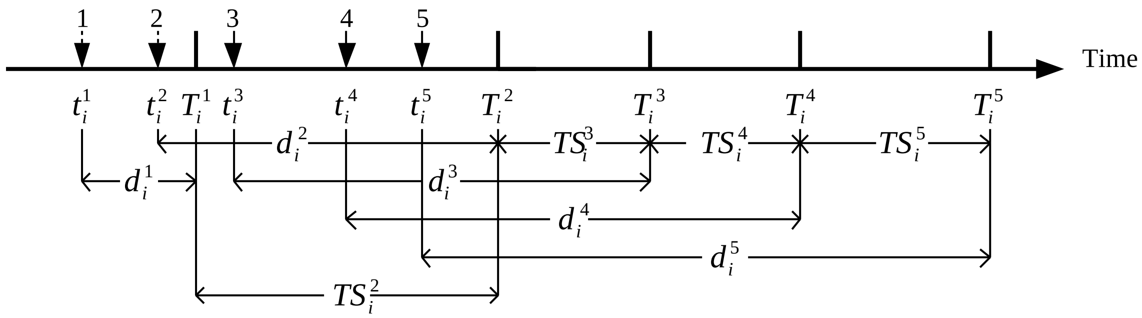


Figure 2. Packets in the class *i*

Following this example, the packets number 1 and 2 of this class have already been served and their queueing delays are determined. In this class there are still packets number 3, 4, 5 which should be scheduled (we have here $s_i(t) = 2$ and $q_i(t) = 3$). Recall that jitter of one packet in a queue is the difference of queueing delay of this packet and the previous packet in this class:

$$j_i^k = \left| d_i^k - d_i^{k-1} \right|$$

Because of a work-conserving scheduler, packets number 3, 4, 5 will achieve minimum jitter $j_i^{\min}(t)$ when all theses packets 3, 4, 5 are transmitted back-to-back in order to assure minimum assumed queueing delays. That means:

$$j_i^{\min}(t) = \sum_{k=3}^{5} \left| d_i^k - d_i^{k-1} \right|$$

if all packets 3, 4, 5 are transmitted back-to-back
In the Figure 2, we can rewrite:

$$\left| d_i^k - d_i^{k-1} \right| = \left| (T_i^k - t_i^k) - (T_i^{k-1} - t_i^{k-1}) \right| = \left| TS_i^k - (t_i^k - t_i^{k-1}) \right|$$

That means we can have: $j_i^{\min}(t) = \sum_{k=s_i(t)+1}^{s_i(t)+q_i(t)} \left| TS_i^k - (t_i^k - t_i^{k-1}) \right|$

We can evaluate the value of average jitter $\bar{j}_i(t)$ for all the packets in class *i* at time *t* as:

$$\bar{j}_i(t) \geq \frac{j_i^*(t) + j_i^{\min}(t)}{(s_i(t) + q_i(t))} \tag{2}$$

and $\bar{j}_i^{\min}(t) = \dfrac{j_i^*(t) + j_i^{\min}(t)}{s_i(t) + q_i(t)}$

In our scheduler, we set the priority of the Head of Line packet in class *i* at time *t* to:

$$p_i(t) = \bar{j}_i^{\min}(t)\Delta_i$$

Recall that the goal of our scheduler is to serve the packets such that the short term average jitter and long term average jitter experienced by packets in a jitter class will satisfy equation (1) for all pairs of *i* and *j*. A *simple heuristic* to achieve this equation is to serve the jitter class with the maximum value of $p_i(t)$ at any time *t*. In other words, the router selects the Head of Line packet of class *i* for which its priority is maximum among all backlogged classes.

Using this priority structure, after a time t, every classes' jitter will attain to the value :

$$\bar{j}_1\Delta_1 = \bar{j}_2\Delta_2 = \ldots = \bar{j}_N\Delta_N \tag{3}$$

That means the average jitter for each class is proportional to its weight, satisfying equation (1).

3 Simulations

Our simulation study (using the ns-2.1b6 Simulator [16]) shows that RJPS scheduler approximates the proportional jitter differentiation model of equation (1).

It is necessary to note that we should maintain a window of packets in order to address the inaccuracies caused by non-backlogged queues and accumulated history because when we do not reinitialize the variables of (2), as the number of packets that are served increases, current queue sizes start to have minimal impact on the service order. In our simulations, we use average jitter $j_i^*(t)$ taken over a moving packet window of size from 100 to 300 packets, thus making the forwarding behavior more responsive to current queue conditions.

The simulation model is as follows. RJPS scheduler uses packet sources of type on-off traffic. The topology used is shown in Figure 3. The links are 6Mps with a latency of 10ms. There is a total of 3 classes 0, 1, 2. Flow 1 from S1 to D1 (1.5Mbps) and Flow 2 from S2 to D2 (2Mps) belong to class 0, while flow 3 from S3 to D3 (0.5Mps) and Flow 4 from S4 to D4 (0.5Mps) belong to class 1 and flow 5 from S5 to D5 (2Mps) belongs to class 2. We run and collect our simulations in 100 seconds.

Our result shows that average jitter for each class at the router R4 which use RJPS algorithm is approximately proportional to theirs weights. The other routers use FIFO algorithm for scheduling packets.

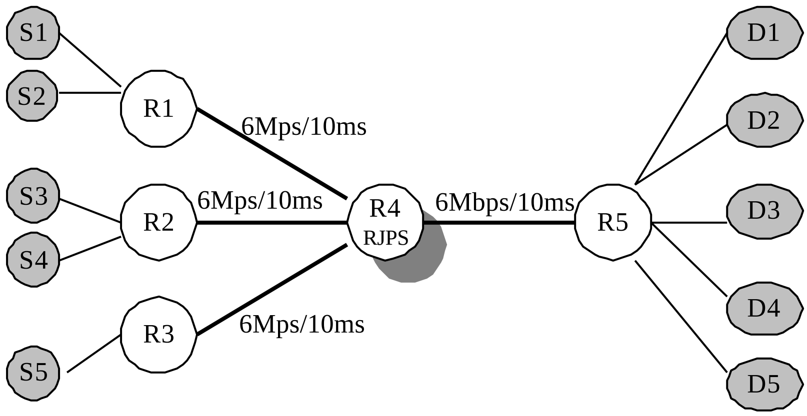


Figure 3. Network topology

The objective of this simulation study is to evaluate the behavior of RJPS scheduler in the context of the proportional jitter differentiation model. A critical issue is not only to investigate whether the RJPS scheduler can approximate the long term jitter (calculated from the beginning of the simulation) but also short term jitter (calculated over our moving window).

From (2), we have:

$$p_i(t) = \bar{j}_i^{\min}(t)\Delta_i = \frac{j_i^*(t) + j_i^{\min}(t)}{s_i(t) + q_i(t)}\Delta_i = \frac{j_i^*(t) + \sum_{k=s_i(t)+1}^{s_i(t)+q_i(t)} \left| TS_i^k - (t_i^k - t_i^{k-1}) \right|}{s_i(t) + q_i(t)}\Delta_i$$

It is easy to realise that when the size of packets varies, the transmission time TS_i^k varies widely and it makes the priority of HOL of packet in each class change, too. The differentiation of weights of classes will have an influence on the calculation of this priority. The other factors, such as link utilization will play an important role for the behavior of our scheduler, because under light link utilization, the delay of packets stay small, and no jitter differentiation is needed. Finally, the window's size should be examined because it is important for calculating $j_i^*(t)$. These factors will be discussed in the next section.

3.1 Behavior of RJPS with constant size of packets and heavy load

In this simulation, the jitter differentiation parameters of classes 0, 1, 2 are $\Delta_0 = 1, \Delta_1 = 3, \Delta_2 = 1.5$. The predefined ratio between class 0/class 1 and class 2/class 1 are 3 and 2, respectively. The window's size is set to 200 packets all packets have a size of 160 bytes. We intended to test the performance of RJPS in term of averge long term jitter and average short term jitter. The link utilization in this simulation is set to 100%.

a. Average long term jitter: The graph 4a shows that average long term jitter ratio for 3 classes achieves the pre-defined ratio 3:2. This ratio is achieved after a time of fluctuation of about some seconds (10s in our simulations).

b. Average short term jitter: Figure 4b shows that the short term jitter ratio fluctuates strongly and can reach up to 45, although the predefined ratio is only 3 and 2. We can say that our scheduler achieves poor quality with short term jitter. One reason is that we evaluate short term jitter over our window of size 200 packets only. When we extend this size of window, the accuracy will increase considerably.

c. Average delay: Clients will only be satisfied if they receiveboth better jitter and better delay. Hence, it is very important to examine the behavior of RJPS in term of delay because if our algorithm works well for proportional jitter, but a class with higher weight would receive higher delay, it is difficult to conclude that the class with higher weight is better than the class with lower weight.

In this simulation, we evaluate average long term delay for each class, too. Figure 4c shows that delay of a class with higher weight is smaller than delay of a class with lower weight.

3.2 Behavior of RJPS scheduler with variation of packet's size:

In this simulation, the jitter differentiation parameters of classes 0, 1, 2 are $\Delta_0 = 2, \Delta_1 = 1, \Delta_2 = 1.5$. The predifined ratio between class 0/class 1 and class 2/class 1 are 0.5 and 0.667. Window size is 200 packets

The packet size plays an important role for the performance of our scheduler, as changing packet size will make the time of transmission of packets TS_i^k vary widely, and hence it will make the deviation of $j_i^{\min}(t) = \sum_k \left| TS_i^k - (t_i^k - t_i^{k-1}) \right|$ between different classes larger. Our proportional jitter ratio will be difficult to achieve.

Our traffic is based on the study of packet size in [17]. With UDP traffic, the size of packets varies around 157 bytes. When the size of packets varies, the ratio of short term jitter varies widely. The results in table 1 show the ratio of jitter when packet size varies from 72 to 256 bytes, 72 to 516 bytes and 72 to 1024 bytes:

	Long term jitter ratio						Short term jitter ratio					
	Class 0/Class 1 (predefined 0.5)			Class 2/Class 1 (predefined 0.66)			Class 0/Class 1 (predefined 0.5)			Class 2/Class1 (predefined 0.66)		
	average	max	min	average	max	min	average	max	min	average	max	min
72 to 256 bytes	0.4996	0.5021	0.498	0.6661	0.676	0.6579	0.5331	2.5249	0.245	0.65	5.4588	0.1879
72 to 516 bytes	0.4839	0.7731	0.4629	0.6669	1.3032	0.6543	0.7384	6.979	0.228	0.82	1026.4	0.3557
72 to 1024 ytes	0.4779	0.5013	0.4525	0.6652	0.706	0.6302	0.6975	23.769	0.1689	0.77	49.543	0.1432

Table 1. Performance of long term jitter ratio and short term jitter ratio with variable packet's size

Figure 5a and Figure 5b compare the average long term jitter ratio and short term jitter ratio between different classes. Results derived from these experiments showed that in most cases, the performance of long term jitter ratio of RJPS stays nearly constant. But when the packets with variable sizes come to our router, ratio of short term jitter fluctuates very strongly. The worst case is when packet size varies from 72 to 1024 bytes and the best case is, when packet size varies from 72 to 256 bytes. It is noteworthy that the short term jitter ratio of our scheduler depends strongly on the variation of packet size, for example, this ratio could fluctuate between 0.1432 and 49.5437 where the predefined ratio is only 0.66 when the packet size is between 72 and 1024 bytes.

3.3 Behavior of RJPS with variation of link utilization:

We will investigate the jitter ratio between different classes when the total traffic varies from moderate (60%) to heavy load (99%). In this simulation, the jitter differentiation parameters of classes 0, 1, 2 are $\Delta_0 = 2, \Delta_1 = 1, \Delta_2 = 1.5$. The predefined ratio between class 0 and 1 is 0.5, between class 2 and 1 is 0.6667. The result in Table 2 shows the performance of jitter ratio in this context. It is neccessary to note that our scheduler deviates remarkbly from the desired values at moderate loads, while the proportional jitter differentiation can be maintained more accurately in heavy-load situations. For example, with load of 60%, the average ratio of class 0/ class 1 is 0.6249 (predefined 0.5), while with load of 100%, this ratio is 0.5008. Figure 6a and 6b plot the variation of jitter ration with variation of link utilization.

Our scheduler works stable when there are enough packets in the queue. With light load, the packets should be scheduled immediately, the queueing delay stays small, jitter will stay small, too, and no jitter differentiaion is probably needed. That is why the proportional jitter model will work good only under heavy load condition.

	Long term jitter ratio						Short term jitter ratio					
	Class 0/Class 1 (predefined 0.5)			Class 2/Class 1 (predefined 0.66)			Class 0/Class 1 (predefined 0.5)			Class 2/Class1 (predefined 0.66)		
	average	max	min	average	max	min	average	max	min	average	max	min
60 %	0.6047	0.6058	0.5834	0.9094	1.034	0.6446	0.6249	1.1022	0.407	0.837	1.7735	0.449
70 %	0.5537	0.5608	0.5521	0.7386	0.7476	0.7277	0.5609	1.0633	0.286	0.744	1.3656	0.422
80 %	0.5039	0.5131	0.498	0.6725	0.6818	0.6709	0.5157	1.2803	0.3261	0.6898	1.679	0.4276
90 %	0.5009	0.51	0.499	0.668	0.67	0.667	0.5104	1.4082	0.3188	0.68	2.33	0.22
100%	0.5047	0.5067	0.5043	0.667	0.669	0.666	0.5008	1.4839	0.31	0.667	2.17	0.3689

Table 2. Performance of long term jitter ratio and short term jitter ratio under different load

3.4 Behavior of RJPS with variation of window's size:

	Long term jitter ratio						Short term jitter ratio					
	Class 0/ Class 1 (predefined 0.3334)			Class 2/ Class 1 (predefined 0.5)			Class 0/ Class 1 (predefined 0.3334)			Class 2/ Class 1 (predefined 0.5)		
	average	max	min	average	max	min	average	max	min	average	max	min
100 packets	0.33402	0.336	0.332	0.50075	0.501	0.4945	0.3385	0.92	0.038	0.503	5.25	0.053
200 packets	0.33302	0.335	0.332	0.49873	0.499	0.496	0.3246	0.6	0.058	0.491	0.86	0.14
300 packets	0.33289	0.334	0.332	0.50054	0.501	0. 49	0.328	0.53	0.109	0.4935	0.79	0.23

Table 3. Performance of long term jitter ratio and short term jitter ratio with different window's size

We will examine the performance of our scheduler when the window size varies. In this simulation, the jitter differentiation parameters of classes 0, 1, 2 are $\Delta_0 = 3, \Delta_1 = 1, \Delta_2 = 2$. The predefined ratio between class 0 and 1 are 0.333, between class 2 and 1 are 0.5, and the size of packet is 160 bytes.

The choice of window size will have an important effect on the stability of our algorithm, because it will make the average jitter $j^*(t)$ vary. It is straightforward to see that the accuracy of our algorithm will increase with the size of window size choosen. But when the window size is high, the computation cost will grow rapidly, too In the previous simulation, we show the result of RJPS scheduler with the window of 200 packets. In this section we have evaluated the behavior of RJPS in the context of variable window size from 100 packets to 300 packets. The performance is shown in the Table 3.

As we see, the result of long term jitter ratio is similar, but the maximum of short time jitter ratio of class 2/class 1 when the window is 100 packets could increase to 5.247. With a window size of 200 packets this maximum value is only 0.8565 and with with a window size of 300 packets this value is 0.7840. Our result shows that performance of RJPS, especially short term jitter ratio, increases with the packet size while long term jitter ratio is not influenced by this window size. Figure 7a and 7b plot the variation of average long term jitter ratio and short term jitter ration between different classes.

3.5 Large Topology

In the previous sections, we have shown only performance of our scheduler with a small topology network, where there is only one router with RJPS. In this section, we present the result of long term jitter ratio and short term jitter ratio at different routers via a larger topology. This topology is shown in Figure 8

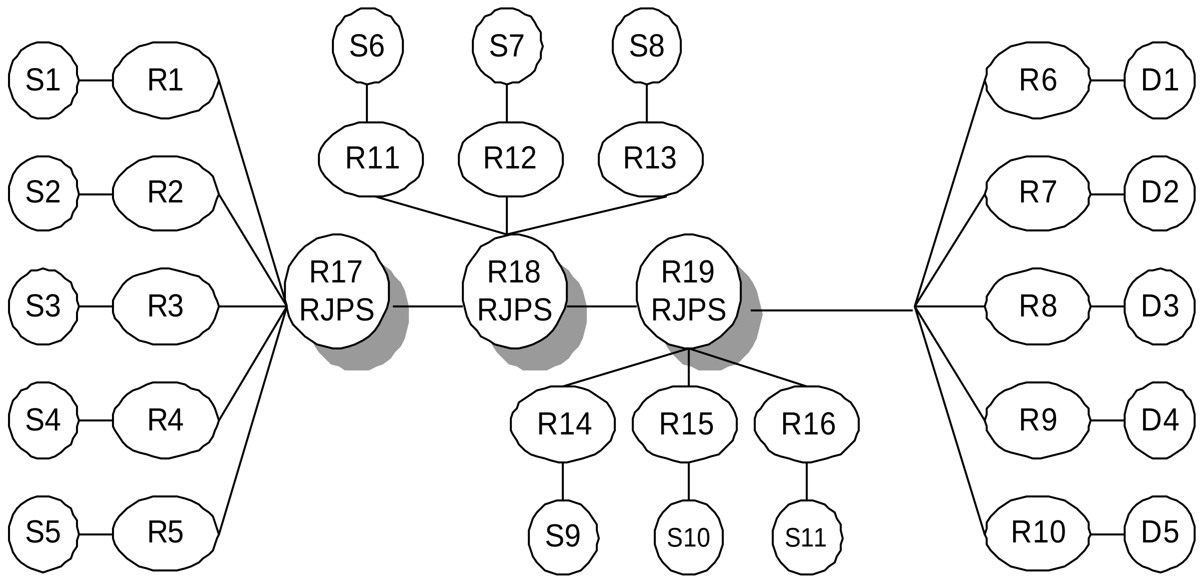


Figure 8: Network Topology

Our network is enlarged and RJPS is used in three routers: R17, R18 and R19 while the others are FIFO only. There are six more flows (S6 and S9 to D1 , S7 and S10 to D3, S8 and S11 to D5). The two first flows belong to class 0, the two next flows belong to class 2 and the two last flows belong to class 1. The weight of three classes are $\Delta_0 = 2, \Delta_1 = 1, \Delta_2 = 1.5$. All the links are 6Mbps with a latency of 10ms, but we are only interested in the jitter differentiation in the routers R17, R18, R19. The packets have a size of 160 bytes, and window size is 200. The total link utilization is set to 99%. The result in Figure 8a and 8b show that our scheduler achieves approximately the long term jiiter differentiation ratio via different routers.

4. Conclusion and future work

In conclusion, we have demonstrated that there is a need to create a scheduler providing differentiated jitter between different classes. We have presented a new scheduling algorithm which could provide proportional short term average jitter and long term average jitter between different classes. The result of our simulation shows that RJPS works well for long term proportional jitter, but the short term jitter depend strongly on the variation of the packet size, link utilization and on the choice of window size.

5. Acknowledgement

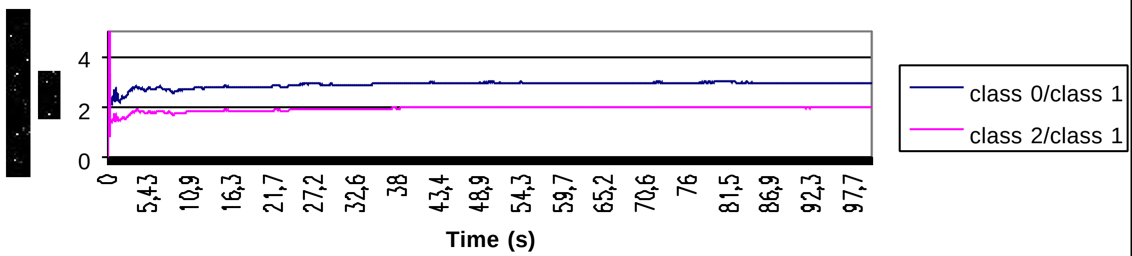
The authors would like to thank Nguyen Huu Thanh, Thomas Wolfram and Irina Piens for their useful help and comments.

References:

[1] S. Shenker, and C. Patridge. Specification of Guarantted Quality of Service. RFC 2212, September 1997.
[2] S. Shenker, and J. Wroclawski. General Characterization Parameters for Intrgrated Service Network Elements. RFC 2215, September 1997.
[3] K. Nichols, V. Jacobson and L. Zhang, „ A Two-bit Differentiated Services Architecture for the Internet“, *draft-nichols-dsopdef-00.txt*, Internet Draft, November 1997.
[4] V. Jacobson, „An Architecture for Differentiated Services“, *Talk in the Intserv WG at the 39th IETF*, Munich, Germany, August 1997.
[5] S. Brim, B. Carpenter, F. Le Faucher. Per Hop Behavior Identification Codes. RFC 2836, May 2000.
[6] J. Heinanen, Telia Finland, F. Baker , W. Weiss, J. Wroclawski. Assured Forwarding PHB Group. RFC 2597. June 1999.
[7] V. Jacobson, K. Nichols, K. Poduri. An Expedited Forwarding PHB . RFC 2598. June 1999.
[8] Van Jacobson, Kathleen Nichols, Kedar Poduri. „The 'Virtual Wire' Behavior Aggregate “ *draft-ietf-diffserv-ba-vw-00.txt*. March, 2000.
[9] C. Dovrolis, D. Stiliadis and P. Ramanathan. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling. In Proceedings of the 1999 ACM SIGCOMM conference, Cambridge MA, September 1999.
[10] C. Dovrolis and Parmesh Ramanathan. A Case for Relative Differentiated Services and the Proportional Differentiation Model. In IEEE Network, 13(5):26-34, September 1999 (special issue on Integrated and Differentiated Services in the Internet).
[11] C. Dovrolis and Dimitrios Stiliadis. Relative Differentiated Services in the Internet: Issues and Mechanisms. In Proceedings of the 1999 ACM SIGMETRICS conference, Atlanta GA, May 1999.
[12] S. B. Moon, J. Kurose, D. Towsley. Packet Audio Playout Delay Adjustement: Performance Bounds and Algorithms. ACM/Springer Multimedia Systems, vol. 5, pp. 17-28, January 1998.
[13] D. Verma, H. Zhang, and D. Ferrari, ``Delay Jitter Control for Real-Time Communication in a Packet Switching Network", *Proceedings of TriComm'91*, Chapel Hill, NC, Mar. 1991.
[14] T. Nandagopal, Narayanan Venkitaraman, R. Sivakumar and V. Bharghavan. Delay Differentiation and Adaptation in Core Stateless networks. *IEEE Infocom 2000*, Tel Aviv, Israel. March 2000.
[15] L. Kleinrock. Queueing Systems, volume 2.
[16] UCB/LBNL/VINT Network Simulator-ns (version 2), http://www-mash.cs.berkeley.edu/ns/ns.html.
[17] S. McCreary, kc claffy. Trends in wide area IP traffic patterns. http://www.caida.org/ outreach/papers/AIX0005/

Appendix: Graphes

Figure 4a: Variation of long term jitter ratio with constant packet's size



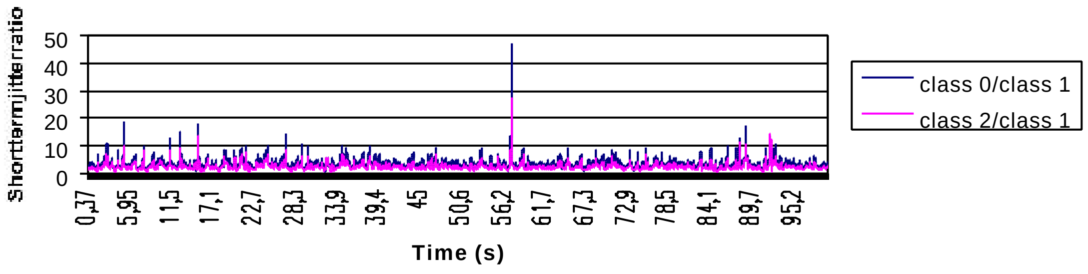


Figure 4b: Variation of short term jitter ratio with constant packet's size

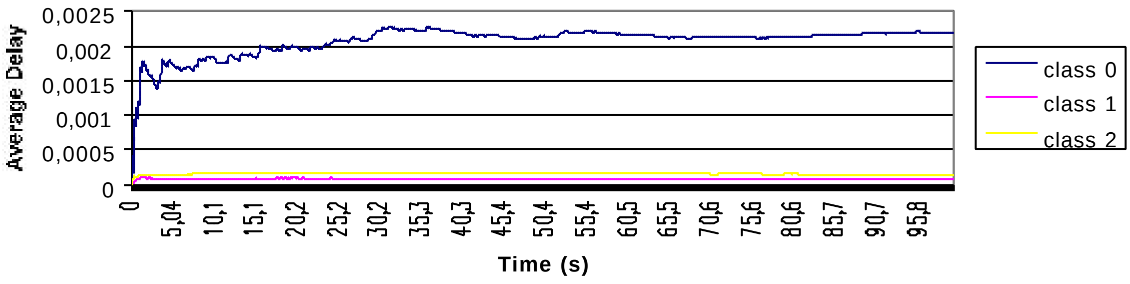


Figure 4c: Average delay of different classes

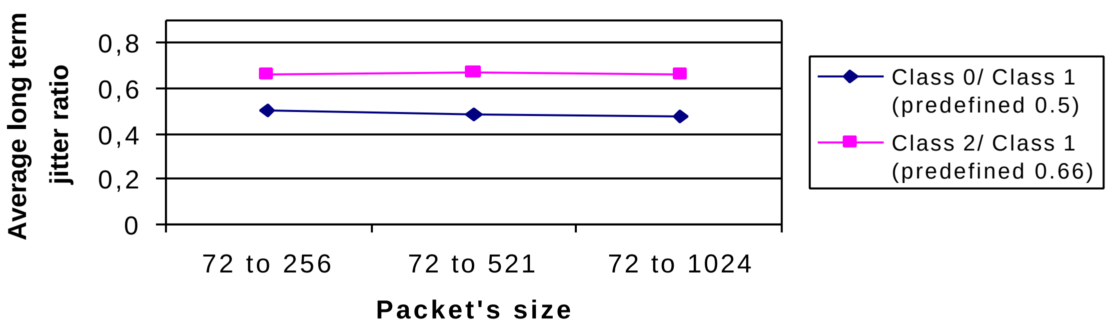


Figure 5a: Average long term jitter ratio with variable packet's size

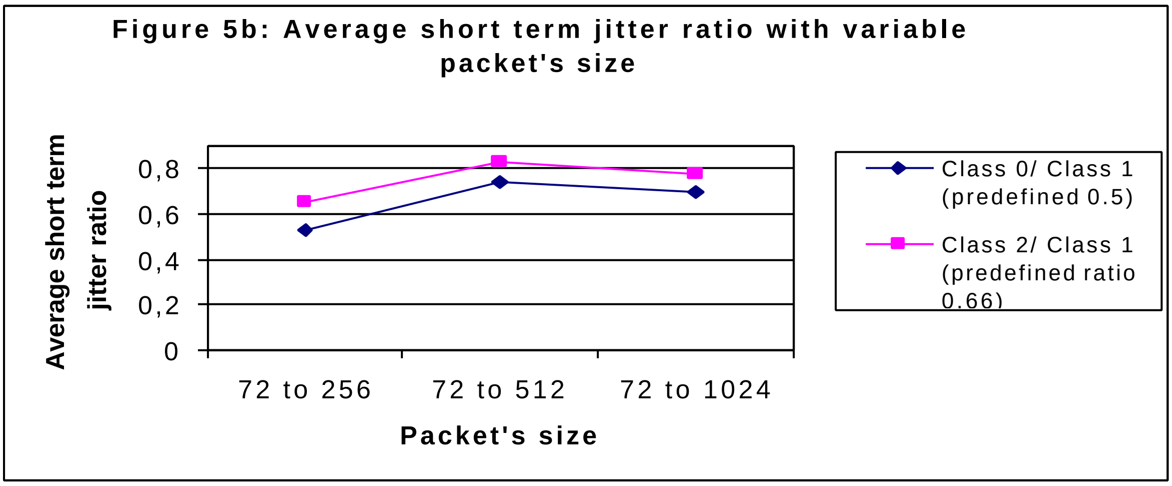


Figure 5b: Average short term jitter ratio with variable packet's size
Average short term jitter ratio
0,8
0,6
0,4
0,2
0
72 to 256
72 to 512
72 to 1024
Packet's size
Class 0/ Class 1 (predefined 0.5)
Class 2/ Class 1 (predefined ratio 0.66)

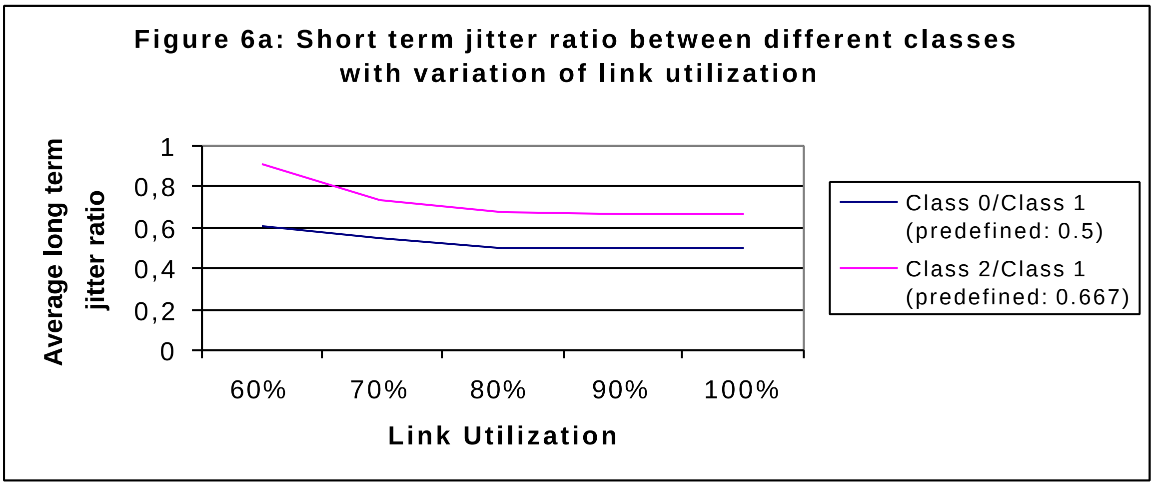


Figure 6a: Short term jitter ratio between different classes with variation of link utilization
Average long term jitter ratio
1
0,8
0,6
0,4
0,2
0
60%
70%
80%
90%
100%
Link Utilization
Class 0/Class 1 (predefined: 0.5)
Class 2/Class 1 (predefined: 0.667)

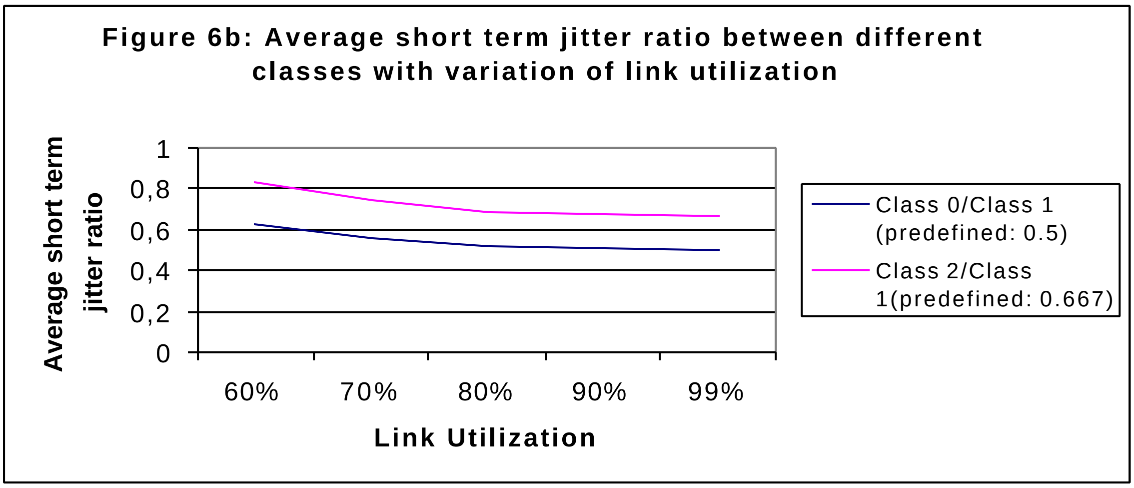


Figure 6b: Average short term jitter ratio between different classes with variation of link utilization
Average short term jitter ratio
1
0,8
0,6
0,4
0,2
0
60%
70%
80%
90%
99%
Link Utilization
Class 0/Class 1 (predefined: 0.5)
Class 2/Class 1(predefined: 0.667)

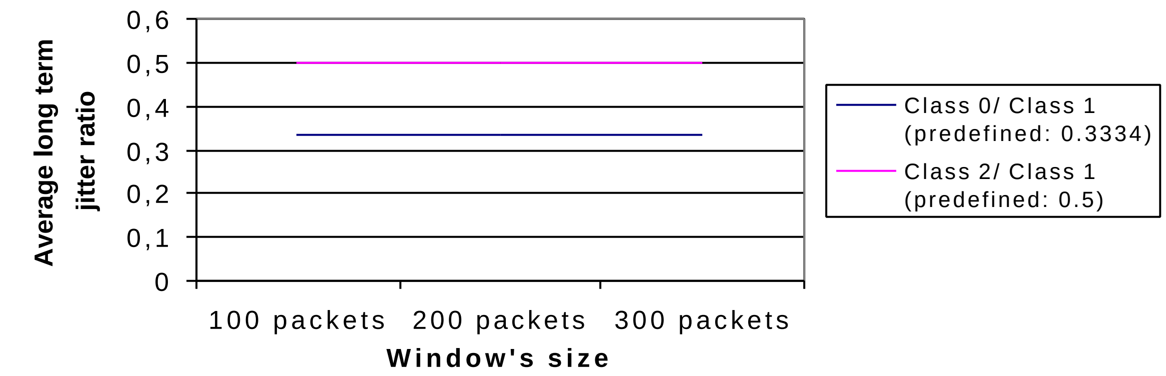


Figure 7a: Average long term jitter between different classes with variable window's size

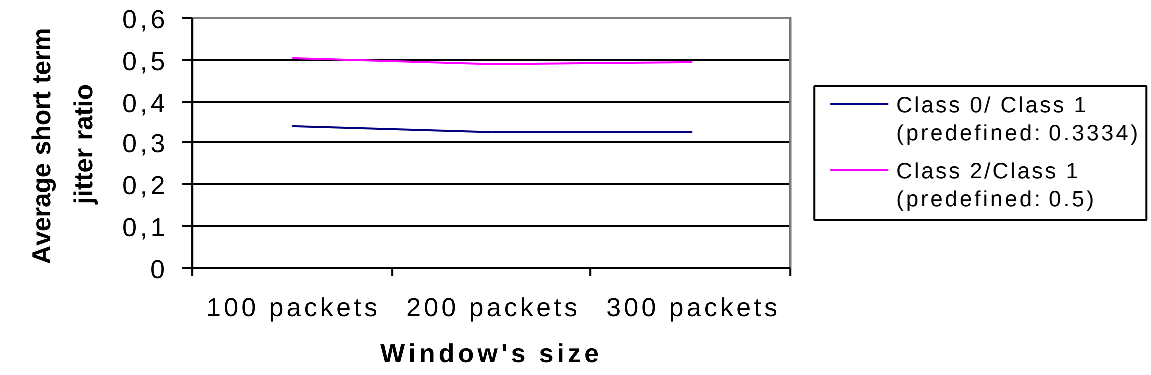


Figure 7b: Average short term jitter ration between different classes with variable window's size

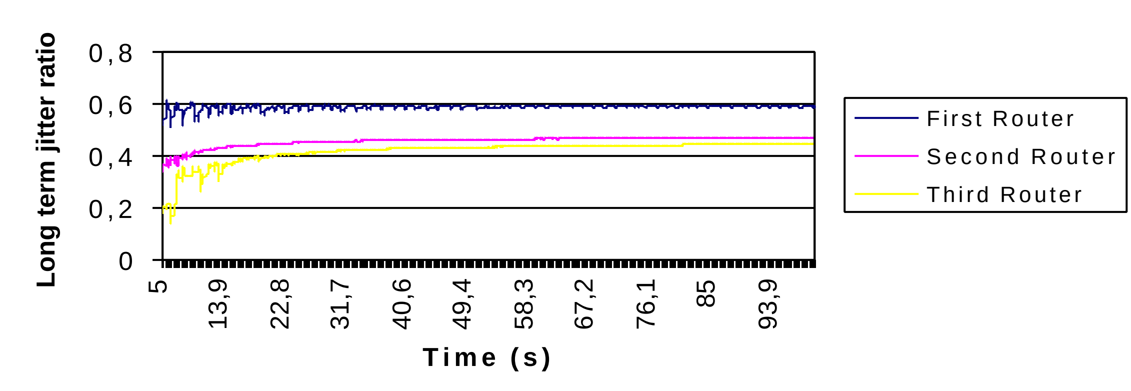


Figure 8a: Long term jitter ratio between class 0 and 1 with a large topology (predefined: 0.5)

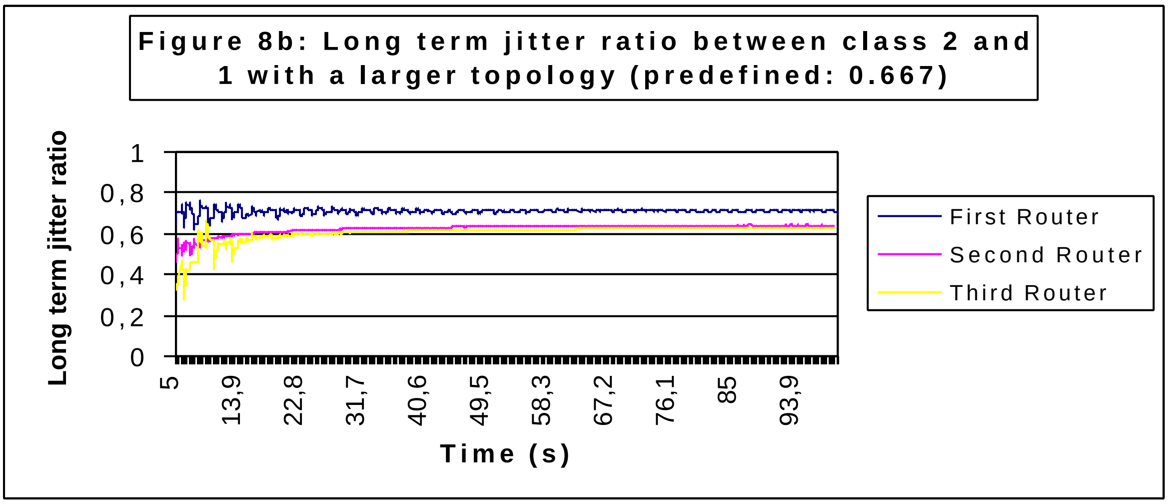


Figure 8b: Long term jitter ratio between class 2 and 1 with a larger topology (predefined: 0.667)
Long term jitter ratio
1
0,8
0,6
0,4
0,2
0
5
13,9
22,8
31,7
40,6
49,5
58,3
67,2
76,1
85
93,9
Time (s)
First Router
Second Router
Third Router