TECHNICAL REPORT. A Method with Node Autonomy to Preserve QoS in Networks with Per-Route Admission. FULL RESULTS.

Alfredo P. Mateos-Papis (a).

(a) Division of Sciences of Communication and Design. Universidad Autónoma Metropolitana. Cuajimalpa. Mexico DF.
amateos@correo.cua.uam.mx

Abstract. This report presents extended results about a method called the Short Term Protection (STP) method. This method is applied to networks where flows are constrained to follow specific routes and where the admission process is distributed per-route. The bandwidth share of any route is protected in the short-term against the traffic increments of its intersecting routes. In the long term, in every congested link the bandwidth-share between routes that intersect in the link is proportional to the average relative measured demands of those routes. The nodes act autonomously without central administration. This method is expected to: 1- help network administrators to have confidence, within a time-window, about the amount of bandwidth they count with in every route; 2- allow a simple bandwidth management in the network, with scalable possibilities.

Keywords. . Quality of Service (QoS); Bandwidth Sharing; Distributed Traffic Control.

1. Introduction.

This report brings back the mature problem of bandwidth-sharing in QoS-aware networks with per-route distributed admission processes (as those proposed in [1] [2] where there is no bandwidth reservation considered for any route), presenting extended results of a method called STP [3] for bandwidth-sharing in these kinds of networks.

A network with per-route admission restricts every flow to run in a specific route, contributing, with this, for an easier evaluation of the traffic in the network, for the admission process.

A distributed admission processes is more scalable, compared with its global-admission counterpart, as it makes admission decisions only to a specific part of a network based on its partial view of that part; with the potential risk that an accomplished admission to one part of a network can inadvertently cause, in the other parts, an unacceptable deterioration of the QoS.

With the STP method in a network, each node acts autonomously, without a central admission authority considered. The nodes are aware neither of the existence of routes nor of the existence of flows. The method is intended to protect routes against bandwidth exhaustion cause from sudden increments in traffic in intersecting routes while allowing the network to have a simple distributed management scheme which adjusts in accordance to the bandwidth necessities of the routes. This method is expected to attain global behaviors in the network and be scalable, and it is projected to be helpful for the network's route-administrators to have confidence, at least within a time-window, about the amount of bandwidth they can count with.

Fig. 1 represents part of a network with three routes that join at node *x*, arriving to the node's input interfaces: *A*, *B* and *C*, respectively, and sharing the bandwidth of the node's output interface *D*. In the STI method a node where two or more routes gather is called an intersection node, as the case of node *x*. A node like this has, at every one of its output interfaces, one queue for every one of its input interfaces (considering that the queues in a node are made at its output interfaces [4]). Specifically, node *x* would have three queues at its output interface *D*, one for every input interface. In this report the routes that intersect at the output link of a node are called: *intersecting routes*.

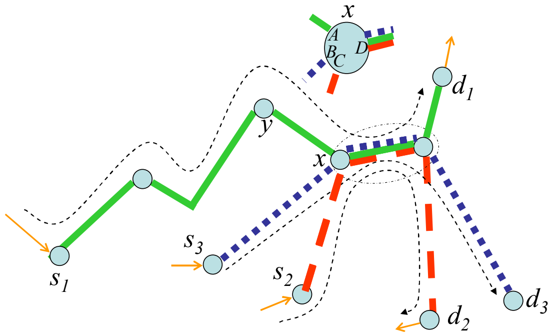

Fig. 1. Routes $s_1 \rightarrow d_1$ and $s_2 \rightarrow d_2$ and $s_3 \rightarrow d_3$ intersect at the output interface of node *x*. The intersection node *x* has three input interfaces, *A*, *B*, and *C*, and one output interface, *D*, in which the traffic of the intersecting routes converges.

Each queue of the output interface has a weight which may change, slowly, according to an updating algorithm which periodically compares the traffic coming from each input interface through the evaluation of the average length of each queue. Each queue at an output interface competes for bandwidth such that its weight decreases slowly if it has a relative smaller average length with regard to the other queues. For example, a route could lose up 20% of its weight in a 30-minute interval.

In this report the STP method uses a WFQ[1] scheduler [5], but it is not intended to be restricted to this kind of work-conserving scheduler.

The QoS parameter of interest in this report is the end-to-end delay in the routes, which includes the queuing delay in the intersection nodes. The term *delay-limit* is the maximum permissible delay that a packet can experiment while traversing a route.

Conditions, which are important to state about the STP method and the evaluation made to it in this report, are the following.

[11] Also called PGPS (Packet-by-Packet Generalized Processor Sharing).

The STP method is built to operate within one class of traffic in a network so this report considers traffic in a single service class. The impact of the STP method working in a class which makes bandwidth provisioning between classes, to optimize a global indicator in the network, is out of the scope of this report.

The evaluation of the STP method is made: 1- With simulations with generated traffic using a mixture of constant bit rate and Pareto sources; 2- Using small topologies; 3- Limited to a single direction in the routes. These simplified conditions are taken as a starting point for the evaluation of the method.

2. Background and Related Works.

The STP method is motivated by some characteristics of the nodes used by the DiffServ model [6] [7] and by the network conditions proposed in [1], [2], where the networks considered have predefined routes, the admission processes is distributed, per-route, as an option to *per-flow* admissions, with the admission decisions taken at the edge of the routes (the research goal of the report is the admission process –not the route-intersection problem).

There are methods for optimization of a global indicator in a network, which obtain the most effective bandwidth-share between classes, like that presented in [8], a bandwidth-sharing method described as proactive (not *reactive*) which uses a central bandwidth manager. Another bandwidth-sharing method between classes [9] adaptively adjusts the weights of a weighted-round-robin scheduler in every core node, where the nodes indeed operate autonomously and cooperate to obtain a QoS indicator.

In [10] a *Coordinated-Schedulers* method is presented to provide delay-bounds in a network which is not per-flow oriented. In it, the core nodes modify each arriving-packet's priority index depending on whether the packet was serviced late or early at the upstream node, as a consequence of cross-traffic. The method provides natural coordination between nodes but each node operates autonomously. This method can work in the context of a single class and it seems that it could have the benefit of protecting a route against intersecting routes. It is not in the scope of this report to study if the STP method could attain the benefits of the method presented by [10].

3. The STP Method.

An output interface of a node working with *N* queues, working the STP method, is considered. The weight of each queue should tend to be proportional to its relative average length with regard to the sum of the average lengths of all the queues. For the sake of simplicity, without losing generality, this explanation considers that there is one intersecting route associated to every queue.

When a queue increases its relative average length with regard to the other queues, the output interface releases more of its total bandwidth in favor of this queue, but slowly, in order to protect, temporarily, the routes associated with the queues that would lose weight.

The method starts at time t_0 where the queues are empty and the weighs have all the same value. Then, at the ending of every interval of duration τ, the method computes an indicator (represented with $I\Delta$ in (1)) for every queue of the output interface, to compare the current weight of the queue with its current relative length with regard to the lengths of the other queues. The interval τ should be sufficiently big as to be able to observe several arrivals and departures of packets (in this report $\tau = 1s$). With this indicator the method computes the new weights of the queues and substitutes those in operation. All this computation should take place in a time much smaller than τ. In (1), the computation of this indicator is made, for a queue i, at time $t_0 + (r+1)\tau$ where its current weight is represented with $\phi_i^{Act}\left(t_0 + r\tau\right)$ (which is valid from $t_0 + r\tau$ to $(t_0 + (r+1)\tau)^-$), and here r is an integer such that $r \geq 0$, and the average lengths of the queues are represented with Q_i.

$$I\Delta\left(\phi_i^{Act}\left(t_0 + r\tau\right)\right) = \frac{Q_i\left(t_0 + \left(r+1\right)\tau\right)}{\sum_{j=1}^{N} Q_j\left(t_0 + \left(r+1\right)\tau\right)} - \phi_i^{Act}\left(t_0 + r\tau\right) \qquad \textbf{(1)}$$

If, from equation (1), the indicator results negative for queue i then this queue should lose weight so that the result of its new weight, $\phi_i^{Act}\left(t_0 + \left(r+1\right)\tau\right)$, minus its weight in operation, $\phi_i^{Act}\left(t_0 + r\tau\right)$, reflects a negative increment (or a decrement). Equation (2) proposes a form of calculation for this negative increment, which is represented with $\Delta\left(\phi_i^{Act}\left(t_0 + r\tau\right)\right)$.

$$\Delta\left(\phi_i^{Act}\left(t_0 + r\tau\right)\right) = \ln\left(1-P\right)\frac{\tau}{T}\phi_i^{Act}\left(t_0 + r\tau\right) < 0 \qquad \textbf{(2)}$$

The two important parameters of (2) are P and T. Parameter P is called the *loss factor*, which is a positive constant much smaller than 1 (with values near 0.1). When P is very small, then $-\ln\left(1-P\right) \approx P$. The parameter T represents a *time span* and it might be on the order of 1200s. The ratio T / τ should be a big integer called K. As τ is much smaller than T the decrement of the queue from one interval to the next one should be very small.

It can be shown that if queue i decreased its weight for the K consecutive intervals of size τ within the time-span T, then the ratio of its initial and ending weights, with regard to that time span, would be equal to $(1-P)$, that is: $1 - P = \phi_i^{Act}\left(t_0 + \left(l+1\right)T\right)$ / $\phi_i^{Act}\left(t_0 + lT\right)$, which is the same as: $\phi_i^{Act}\left(t_0 + \left(l+1\right)T\right)$ – $\phi_i^{Act}\left(t_0 + lT\right)$ = – $P\phi_i^{Act}\left(t_0 + lT\right)$. With this it can be stated that with this method the maximum weight deterioration for a queue in a time T is P%. This is demonstrated[2] using the fact, obtained from (1) and (2), that $\phi_i^{Act}\left(t_0+\left(m+K\right)\tau\right)$ = $\phi_i^{Act}\left(t_0+m\tau\right)\left[1-f\left(P\right)/\left(T/\tau\right)\right]^{T/\tau}$ where l and m represent integers greater than, or equal to 0.

The STP method also proposes a calculation of the weight of every queue that should not decrease its weight, as indicated by (1), after every interval of duration τ, in such a way that the sum of the weights of all queues be always equal to 1.

Following the method proposed in [11], the average length Q_i of queue i is computed as $Q_i=\left(1-w_q\right)Q_i+w_q q_i$ where w_q is the averaging parameter and q_i is the instantaneous queue-length. This equation acts as a low-pass filter. The smallest the value of w_q the smoother the output will be. The value for w_q is set to 0.002, to cope with the possibly burst behavior of the instantaneous queue-length.

4. Experimental Results

The first evaluations of the STP method are done with simulations using the topology of **Fig. 2**, a bounded network where the central node *c1* is the only one which can use the STP method. There are three edge nodes, *e1, e2* and *e3*. Outside the network boundaries there are three source nodes, *s1, s2, s3*, and three destination nodes, *d1*, *d2* and *d3*. Inside the network there are three routes: *Route1*, *Route2*, *Route3*. Links inside the network have a bandwidth of 3*Mb/s* for the operating class. The links connecting the network nodes with the outside nodes have a bandwidth of 100*Mb/s*, with 0.05ms of delay. *Route1* and *Route3* intersect at the output interface of *c1* (going to *e2*). *Route2* intersects with *Route1* at the exit interface of *e1* (going to *c1*). *Route1* suffers from two intersections which puts it in disadvantage against *Route3* which is not affected by *Route2*. In the experiments the situation is that *Route3* increases its traffic so that *Route1* is affected. The increase is made through the increase of sources of traffic in *Route3*.

When node *c1* uses the STP method, for the sake of simplicity of explanation, the first queue of *c1* is referred to as the queue for *Route1* and the second queue as the queue for *Route3*.

Because of the random nature of experiments each results exposed in this report is obtained from the mean value of the results of 40 to 60 experiments.

[2] See technical report *Rep-Tec-2-Equations-STP-Method.pdf* at http://ccd.cua.uam.mx/~amateos/.

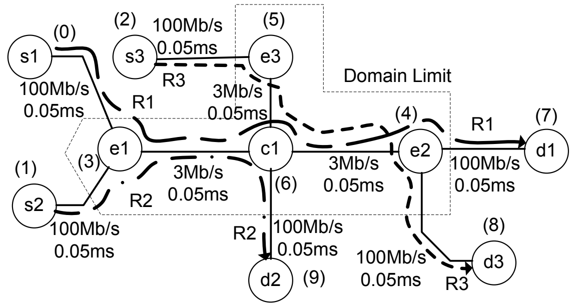

Fig. 2. The topology for the experiments has two routes: *Route1* traversing nodes *s1–e1–c1–e2–d1*, and *Route3* traversing nodes *s3–e3–c1–e2–d3*. The routes intersect at the output interface of node *c1* and separate at node *e2*. R1 stands for *Route1*, R2 and R3 have similar meanings.

4.1. Experimental Setting

The experiments were carried out with the ns-2 simulator[3]. The sources in *Route1* and *Route3* are *On/Off* Pareto sources with 250*ms* on/off duration, with 68 Kb/s during *On* periods and 95-byte packets [15][4] with Pareto shape parameter of 1.7 (infinite variance). The CBR sources have 1500-byte packets with 256Kb/s[5]. It should be clear that these rates are small compared with those actually used in the Internet, but they are used for the purpose of the evaluation of the STP method.

It is supposed that for a long time before the beginning of each experiment the routes have had the same traffic, then, at time $t = 0$ of the experiment *Route3* increases its number of sources (without knowing about the possible deterioration of the QoS inflicted on *Route1*).

Three different types of experiments are done, depending on the buffer used in the output interface of *c1*. In the first case, called *q1*, there is just one queue at that output interface (which handles the traffic of the two routes). The second case, called *q2*, uses the STP method, having two queues at the output interface of node *c1*, each one associated with *Route1* and *Route3*, respectively. The parameters employed for the *q2* case are: $T = 1200s$, $P = 0.25$. The time τ is 1*s*.

[3] The ns-2 simulator [12], version 2.33. Also, the simulations use the DS tools included in the simulator [13]. In the STP method these tools were modified to add the WFQ packet-scheduling operation, with the implementation proposed in [14], which uses the WFQ scheduler [5]. Additional modifications were made to incorporate the possibility to change the weights of the scheduler in a dynamic form, in accordance with the STP method (see technical report *Rep-Tec-3-WFQ-Changing-Weights.pdf* at http://ccd.cua.uam.mx/~amateos/).

[4] A voice packet can have 67 bytes which would make a 95-byte IP packet adding 8 bytes of UDP header and 20 bytes of IP header. A reasonable voice mean-rate of 3.0 Kbytes/s in one direction makes a traffic mean rate of 34Kb/s (3 Kbyte/s x 8 bits/byte x 95 / 67 = 34 Kb/s – considering the headers' overhead), making 68 Kb/s during the On period of On/Off sources.

[5] The information for the rate value is taken from [16].

The last case, called *q2f*, similarly, utilizes two queues at the output interface, but the weight of each queue is fixed throughout the duration of the experiment. For all the other queues (for the *normal* nodes) Droptail queues are used. The queue-lengths are big enough such that the total observed packet loss rate is always less than 2% in every experiment.

For the *q2* case and a *P* parameter value greater than 0, when the parameter *T* is close to 0 this case should work almost as the *q1* case does, and when the parameter *T* is very big this case should work almost as the *q2f* case does.

In every experiment, unless otherwise indicated, *Route1* and *Route2* have almost the same number of sources. *Route1* has 3 CBR sources + 17 Pareto sources, and *Route2* has 3 CBR sources + 16 Pareto sources. These numbers of sources are big enough to cause average packet-delays of 3ms at the exit interface of node *e1* without causing packet congestion losses.

Route3 initiates each experiment with 3 CBR sources + 8 Pareto sources. The initial traffic-share of *Route1* and *Route3*, at the exit of node *c1*, going to *e2*, is 0.5641 and 0.4359 for theses routes, respectively.

In the experiments it is supposed that this traffic has not changed for a long period of time, and then, at time *t* = 0, *Route3* changes its amount of traffic.

The STP method tends to give to every queue a weight which is proportional to its traffic proportion with regard to all the queues. As in these experiments *Route1* and *Route3* are injecting the traffic for the first and second queue of *c1* going to *e2*, respectively then these initial traffic-share values of *Route1* and *Route3*, reflected in the queues of *c1*, are taken as the initial values of the weights for these queues (as explained in the second scenario of the experiments presented in this report). Similarly, for the *q2f* case, these values are taken as the fixed weights for the queues.

Each row of Table 1 shows the number of sources of *Route1* and *Route3*, their rates and the bandwidth percentage share at the exit interface of node *c1* (going to *e2*) for a single experiment.

+PAR means the number of Pareto sources increased in *Route3* at the beginning of the experiment (*t* = 0). For example, in the second row the total number of Pareto sources of *Route3* is 8 + 1 = 9.

–%R1 indicates, for a given row, the bandwidth percentage deterioration of *Route1* with regard to that which this route has in the first row. For example in the second row %R1 = 55.62 so that 1 –55.62/56.41 = 0.01405. The maximum traffic-share deterioration for *Route1* shown in the table is 13.55%.

It is important to notice that the queue-weight deterioration for a route with the STP method can never be bigger than the traffic-share deterioration of that route within a time-duration *T*.

CBR	CBR	PAR	PAR	+PAR	Kb/s	Kb/s	Kb/s	%	%	–%
R1	R3	R1	R3	R3	R1	R3	R3+R1	R3	R1	R1
3	3	17	8	0	1346	1040	2386	43.59	56.41	0.00
3	3	17	9	1	1346	1074	2420	44.38	55.62	1.41
3	3	17	10	2	1346	1108	2454	45.15	54.85	2.77
..	..	..	..							
3	3	17	16	8	1346	1312	2658	49.36	50.64	10.23
3	3	17	17	9	1346	1346	2692	50.00	50.00	11.37
3	3	17	18	10	1346	1380	2726	50.62	49.38	12.47
3	3	17	19	11	1346	1414	2760	51.23	48.77	13.55

Table 1. Number of sources of *Route1* and *Route3*, their rates and the bandwidth percentage share at the exit interface of node *c1* (going to *e2*) for every experiment.

4.2. Initial Results.

Fig. 3 presents the weight values for the *q2* case of the output interface in node *c1*, for four different values of *P*, after a period of time *T* = 1200*s* in a special group of experiments where the traffic of *Route3* is bigger than that of *Route1*[6].

In the case of *P* = 0.05 the weight of the queue attending *Route1* should be 0.564 * 0.95 = 0.5358, and the experiments gave weight value of 0.5291. For *P* = 0.15 this weight should be of 0.564 * 0.85 = 0.4794, and the experiments gave a weight value of 0.4661. For *P* = 0.25 the weight obtained should be 0.564 * 0.75 = 0.4230 and the experiments gave a value of 0.41. All these results are close to the projected ones.

[6] Route3 augments 30 Pareto sources at the beginning of experiment-time. Regarding **Table 1**, *Route1* has 40% of the traffic and *Route3* has 60% (only for this experiments the bandwidth of *c1*-*e2* is 4[Mb/s]).

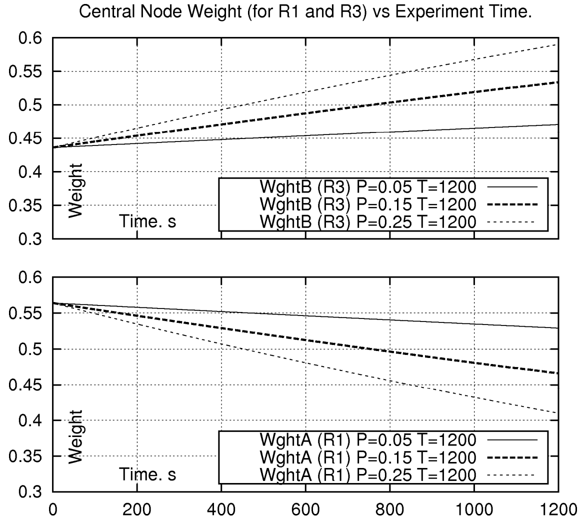

Fig. 3. Experimental results of weight-functions for the *q2* case with different values of *P* parameter. Label "Wght A (R1) *P*=0.15 *T*=1200" stands for weight of *Route1* for the *q2* case with parameters *P*=0.15 and *T*=1200.

4.3 Gain Results.

In order to assess the benefits of the STP method, this report uses a figure of merit which consists of an overall gain for each route, for each experiment. This gain rewards with one point for every packet that traverses its route within the delay-limit of the route, and penalizes with ten points otherwise.

The first scenario considers experiments of short duration, 120*s*, for cases *q1* and *q2f*, and a delay limit is 18*ms*. The reason to make this scenario is for observing the behavior of these two cases and to compare these behaviors with that of the *q2* case presented in the next scenario. **Fig. 4** shows the gains obtained for *Route1* and *Route3* (*Gain1* and *Gain3*) as a function of the number of sources added to *Route3*. Every point in the figure shows the gain obtained in each 120*s*-experiment. The upper abscissa axis shows the number of sources added to *Route3* in the experiment, and the lower abscissa axis shows the resulting rate in *Route3*. The gain in each experiment is normalized dividing it by the duration of the experiment in the simulation.

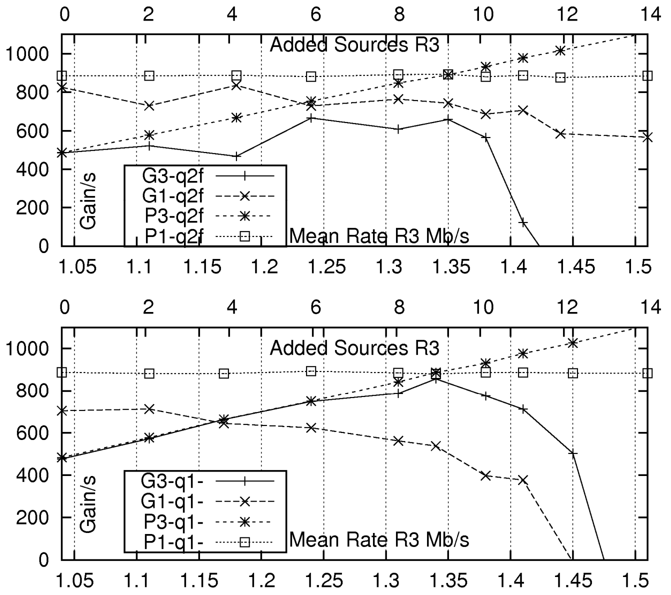

Fig. 4. Gains of routes against number of sources added in *Route3*, for cases *q1*, and *q2f* with a delay limits of 18*ms*. Label "G3-q1" stands for *Gain3* in the *q1* case, and label "P1-q2f" stands for stands for number of packets for Route1 for the *q2f* case. The meaning of the other labels is similar.

For the *q1* case *Gain3* has a maximum value at 9, decreasing afterwards. The admission process of *Route3* should avoid allowing an increase of more than 9 sources. *Gain1* deteriorates with the increase of sources in *Route3*. For the *q2f* case it is also clear that the admission process of *Route3* should neither allow more than 9 sources. From 0 to 9 sources, for the *q1* case, *Gain1* goes from 704 a 537 points, representing a loss of (704-537) / 704 = 27.7%. For the *q2f* case *Gain1* goes from 825 to 742 points, representing a loss of (825 – 742) / 825 = 10%. It is clear that the *q2f* case protects *Route1*. It is observed that the *q1* case is a good option when the available bandwidth is big compared to the traffic, in the routes[7]. Also, in the *q1* case, when the increase of traffic in *Route3* is big enough, for example from 1.25 to 1.45 Mb/s, the admission process of *Route3* should not allow that increase, giving an automatic protection to *Route1*, not requiring the operation of a *q2* case (STP method).

Fig. 5. shows the sum: *Gain1* + *Gain3* for each one of the cases *q1* and *q2f*. Each one of these sum-gains would be the overall gain obtained from *Route1* and *Route3* for each one of the cases. This figure compares the overall gain of case *q1* with that of case *q2f*. It can be seen that the gain-sums are approximately equal. The reason for this is that the schedulers for the *q2f* case (and *q2* case) is work conserving, that is, the scheduler serve at full transmission rate whenever there is data to be served, ideally

[7] The standard deviation obtained for the gain was big. For most cases of increased sources, it was bigger than the mean value * 0.5. This is because the gains obtained depend on values of Pareto traffic which can give big variations.

having a service, for the *q2f* case, as big as that of the *q1* case. This figure shows that the gain is not adequate to evaluate the protection benefit, neither the *q2f* case nor for the *q2* case, for *Route1*.

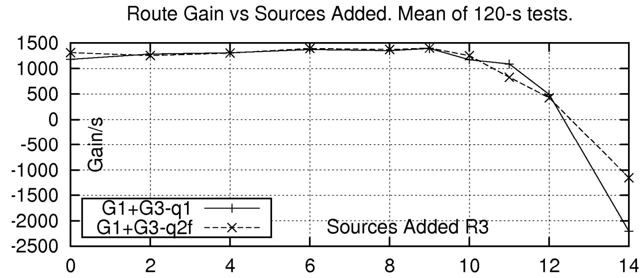

Fig. 5. *Gain1* + *Gain3* for each one of the cases *q1* and *q2f*. These two sum of gains are approximately equal.

Fig. 6 shows the gains and the delays of the routes for the *q1* and *q2f* cases. The delay-value is computed as the percentile of the 2% bigger values of the observed delays of the packets traversing their corresponding route. In other words, the delay of the route is the time exceeded by the delay of only the 2% most delayed packets. It can be seen that the delay of *Route1* for the *q2f* case increases very slightly as the traffic of *Route3* increases, still, the *q2f* case is not appropriate for networks where the traffic can change in the routes and where there is no reason to give a minimum fixed protection to a specific route.

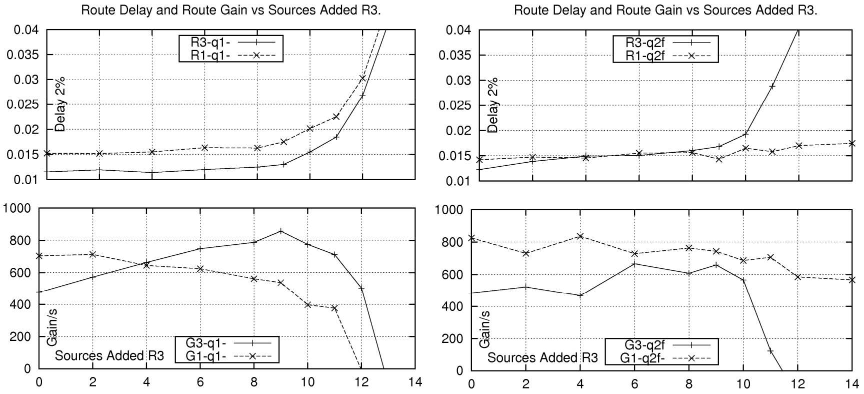

Fig. 6. Gains and the delays (in *s*) of *Route1* and *Route3*, for the *q1* and *q2f* cases.

Fig. 6 shows the gains and the delays of the routes for the *q2f* case. *G1* decays slowly as *Route1* is protected against the increase of traffic of *Route3*.

Fig. 7 shows the delay in *Route1* and *Route3* and the delays in node *c1* for cases *q1* and *q2f*. It can be observed how, for each case, the delay in node *c1* follows the delays of *Route1* and *Route3*, always below. In case *q1* the delays in node *c1* is the same for traffic coming from *Route1* and from *Route3*.

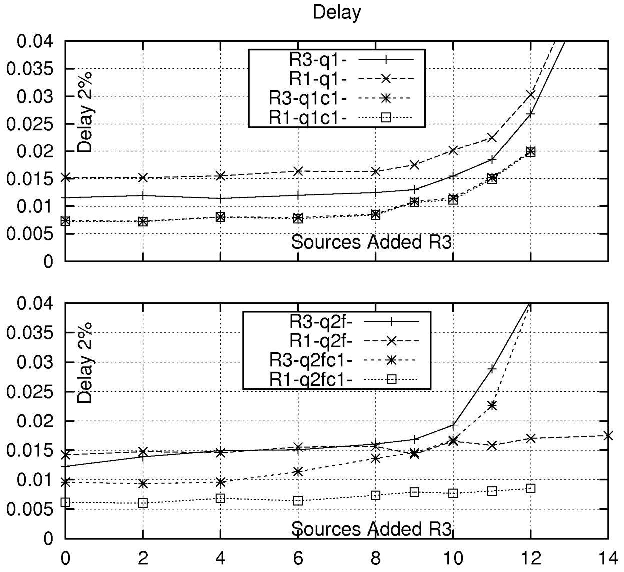

Fig. 7. Delays in *Route1* and *Route3* and the delays in node *c1* for cases *q1* and *q2f*.

The second scenario is one using *q2* case, with $P = 0.25$ and $T = 1200s$, with experiments of 1200*s* duration, where it is supposed that there is no change in traffic amount during those 1200*s*. It is also supposed that the traffic of the routes before time 0*s* has been the same for a long time, so that as the STP method tends to give the weight of each queue the same value of its to the traffic-share then the initial traffic-share values of *Route1* and *Route3*, reflected in the queues of *c1*, are taken as the initial values of the weights for these queues, that is 0.5641 and 0.4359 for *Route1* and *Route3*, respectively.

The gains are evaluated at the ending of subsequent intervals of 120*s* duration. In the first part of each experiment the *q2* case will protect *Route1* as case *q2f* does, and at the ending part of the experiments the *q2* case should tend to get the same weights as case *q1* does.

For this second scenario **Fig. 8** shows how the delay in the *q2* case augments with time in every experiment for *Route1* (and decreases for *Route3*).

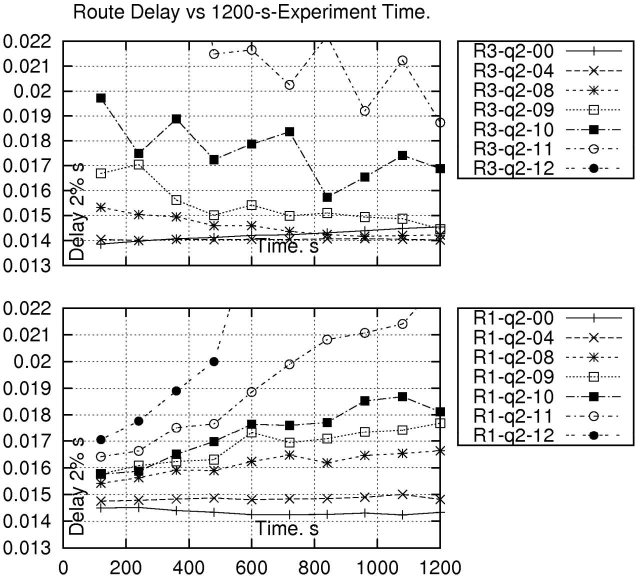

Fig. 8. Delays of routes for *q2* case, throughout the experiment time. Label "R1-q2-04" stands for delay of *Route3* for 4 sources added in *Route3*. The right side of the figure shows the gains of routes for the *q2* case. Label "G1-q2 120 s" stands for *Gain1* of *q2* case at 120*s* of experiment. The other labels have similar meanings.

Fig. 9 and **Fig. 10** show *Gain1* and *Gain3* for the *q2* case for experiments for various amounts of increased-Pareto sources in *Route3*, at the ending of each 120*s* segment time. From these figures it can be seen that *Gain1* has a decrease-tendency as *Route3* increases its traffic, and for increase-values bigger than 8 there is a clear reduction in gain as time runs by, in each experiment (as *Route1* has its gain-share reduced with time). From these figures it can be seen too that up to 8 increased-Pareto sources *Gain3* increases as *Route3* increases its traffic, but then, as *Route3* increases to 9 and more Pareto sources *Route3* begins saturating and *Gain3* decreases as *Route3* increases its traffic. It can also be seen that *Gain3* grows as time passes by, in each experiment (as *Route3* augments its gain-share with time).

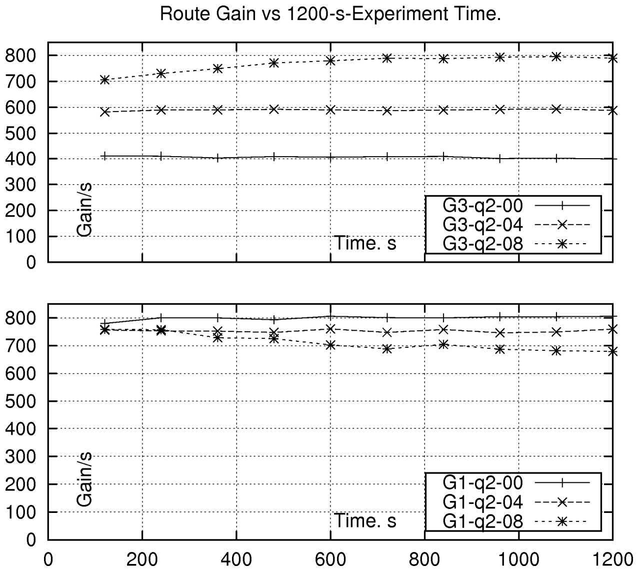

Fig. 9. *Gain1* and *Gain3* for the *q2* case for experiments for various amounts of increased-Pareto sources in *Route3*, at the ending of each 120*s* segment time. The label G1-q2-08 represents the *Gain1* for the *q2* case where *Route3* has had an increment of 8 Pareto sources at 0*s* of experiment-time. The meaning of the other labels is similar.

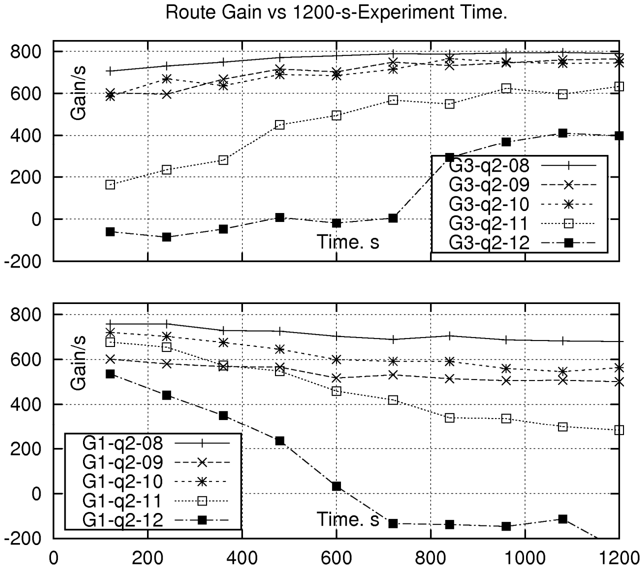

Fig. 10. *Gain1* and *Gain3* for the *q2* case for experiments for various amounts of increased-Pareto sources in *Route3*, at the ending of each 120*s* segment time. the meaning of the labels is similar to those of **Fig. 9**.

For this second scenario **Fig. 11** shows how the *Gain1* and *Gain3* of *q2* case tend to be equal to case *q1* at the ending of each experiment (at 1200*s*) and tend to be equal to *q2f* case at the beginning of each experiment (at 0*s*) (the gains are evaluated in 120*s* intervals –for example the gain of 1200*s* is evaluated from 1080 to 1200*s*). *This is a main result in this report*.

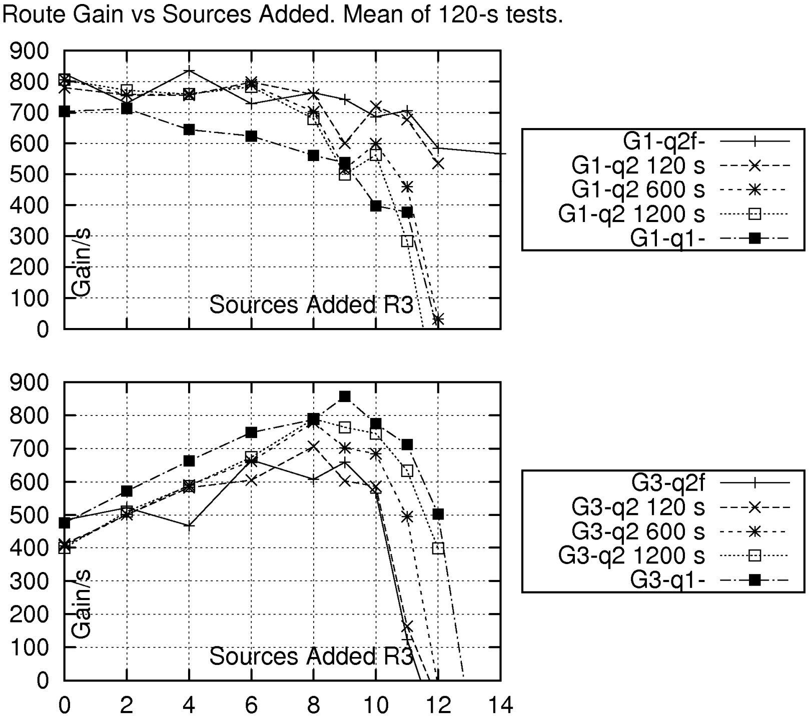

Fig. 11. The left side of the figure shows the delays of routes for *q2* case, throughout the experiment time. Label "R1-q2-04" stands for delay of *Route3* for 4 sources added in *Route3*. The right side of the figure shows the gains of routes for the *q2* case. Label "G1-q2 120 s" stands for *Gain1* of *q2* case at 120*s* of experiment. The other labels have similar meanings.

Fig. 12 shows the gains, the weights and the delays for the *q2* case for scenario 2 with 9 Pareto sources increase in *Route3* at 0*s* of experiment-time. It is observed how the weights of *Route1* and *Route3*, which are: 0.564 and 0.436 at time 0*s*, change and at time 1200*s* they have become roughly the same, that is, 0.5097 for *Route1* and 0.4903 for *Route3*. Indeed at 1200*s* *Route1* has not decreased its weight in 25% (only in 1–0.5097/0.564 which is 9.62%) as both Routes have the same amount of traffic, so it is a logical result that the weights of both routes become equal at 1200*s*. **Fig. 12** also shows the delay of the 2% most delayed packets. It is observed how *Route3* has more delay, compared to that of *Route1*, at 0*s*, and it has less delay at 1200*s*. **Fig. 12** also shows how the *Gain3* increases, as time passes by, with a corresponding decrease on behalf of *Gain1*.

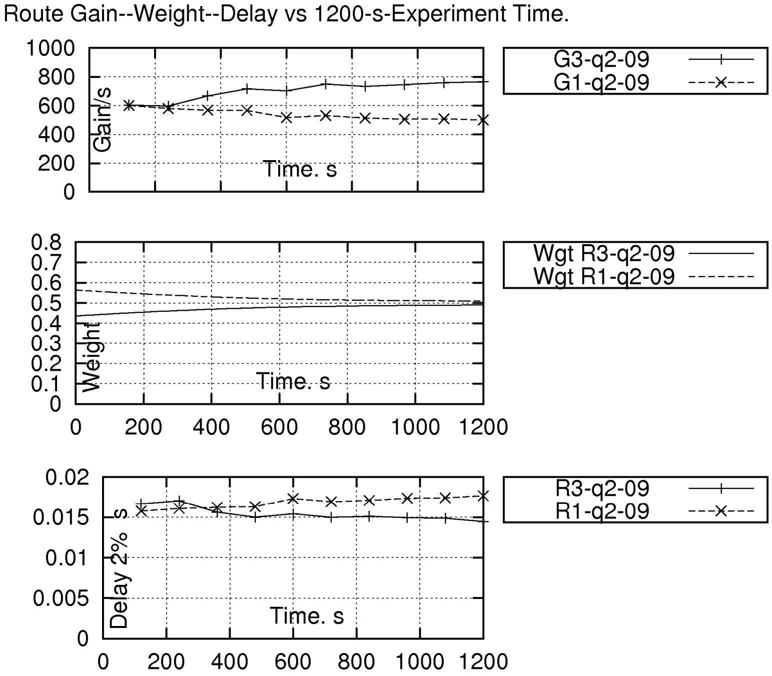

Fig. 12. Gains, weights and delays for the *q2* case for scenario 2 with 9 Pareto sources increase in *Route3* at 0*s* of experiment-time. Label Wgt R1-q2-09 stand for weight of *Route1* for the *q2* case where *Route3* has increased its number of sources in 9. The other labels have similar meanings.

For this second scenario, **Fig. 13** shows *Gain1* and *Gain3* for the *q2* case where *Route3* has an increment of 9 sources. At time 1200*s* of the experiment the weights of the queues for *Route1* and *Route3* should have changed to be both equal (having a value of 0.5 each). The traffic condition of this situation is shown in Table 1.

Taking these values of 0.5 as the initial weights for each route for case *q2*, other experiments simulating 120*s* are done with the same number of sources as the 1200*s*-experiments of the second scenario explained in the previous paragraph, with the exception that *Route1* has 14 Pareto sources instead of 17. These 120*s*-experiments can be taken as the continuation of the previous 1200*s*-experiments, that is, the start time would be at 1200*s* where *Route1* decreases the number of its Pareto sources from 17 to 14. The *Gain1*-result of these experiments is the point observed at 1320*s* with value 592.49 points, which is bigger than the *Gain1* value at 1200*s*, of 500.2 points. This can be viewed as an increase of *Gain1* as a result of a decrease of traffic in *Route1*, from 1200*s* to 1320*s*. The reason of this gain behavior is that *Route1* has had its weight decreased, from 0.564 to 0.500, in the time interval from 0s to 1200*s* and it had suffered from the extra delay imposed at the exit of node *e1* where it intersects with *Route2*. For this reason it is convenient for *Route1* to decrease its traffic at 1200*s*.

On the contrary, from 1200*s* to 1320*s* *Route3* takes advantage of the traffic decrease of *Route1*, considering the fact that the scheduler is energy-conserving, so that *Route3* augments it gain from 764.53 to 822.75 points, in the time interval 1200*s* to 1320*s*.

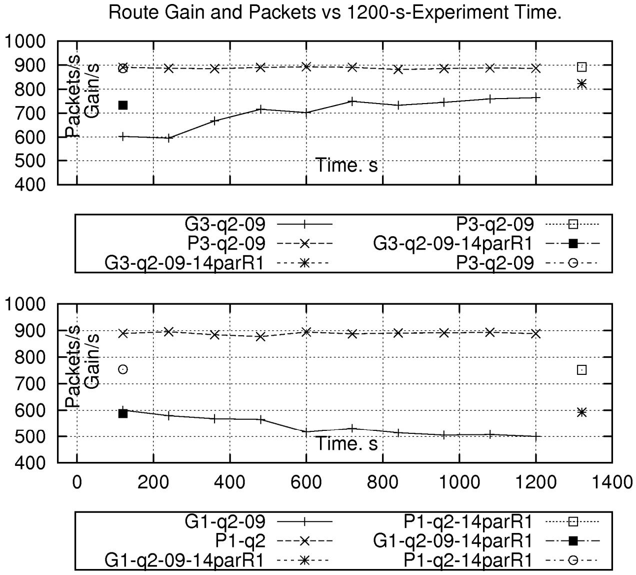

Fig. 13. *Gain1* and *Gain3* for the second scenario for *q2* case and in the special situation where *Route3* has an increment of 9 sources at 0*s* of the experiment-time. The weight of *Route1* decreases with the experiment time. The *Gain1* points show the behavior of this gain when *Route1* decreases its traffic at 0*s* or at 1200*s*. The label G1-q2-09-14parR1 stands for *Gain1* in the *q2* case where *Route1* has 14 Pareto sources and *Route3* has increase it number of sources in 9 at 0*s* (there are two points for this). The other labels have similar meanings.

Fig. 13 also shows a point of *Gain1*, for other set of 120*s* experiments for the case *q2* where the initial traffic-share of *Route1* and *Route3*, at the exit of node *c1* going to *e2* is the same of that of the experiments of scenario 2 of 1200*s* for case *q2*, that is: 0.5641 and 0.4359 for *Route1* and *Route3* respectively. Again, these experiments are done with the same number of sources as those of the 1200*s*-experiments, with the exception that *Route1* had 14 Pareto sources instead of 17. In this case at 120*s* *Gain1* is 588.28 points in lieu of the 600.76 points attained at 120*s* in the 1200*s* experiments where *Route1* had 17 sources. This result of smaller amount of gain with smaller amount of traffic obtained for *Route1* in the interval 0*s* to 120s is due to the fact that the weight of Route1 was 0.564, enough big to handle more traffic with a concomitant bigger gain. *Route3* again takes advantage of the decrease of traffic of *Route1* such that its value is 732.99 instead of the 602.91 value it had when *Route1* had 17 Pareto sources.

The following 120*s*-duration experiments are made for a third scenario for the *q2* case where parameter T = 120*s* (instead of 1200*s*) and the parameters P = 0.25 (the same as in the previous scenarios). For *Route1* and *Route3*, **Fig. 14** shows the delays for routes, **Fig. 15** and **Fig. 16** show the gains for routes, and **Fig. 17** show a combination of weight, gains and delays, for routes, for this scenario. These figures can be compared, respectively with **Fig. 8**, **Fig. 9**, **Fig. 10** and **Fig. 12** (those corresponding to the scenario where *T* = 1200*s*), where it can be observed that the behaviors of these delays, gains and weights are similar between the scenarios with *T* = 1200*s* and *T* = 120*s*, but the change with experiment-time is 10 times bigger (it is 10 times faster) for the results of the scenario with *T* = 120*s*.

This results could give an insight of how to select the parameter *T*, which could be on the order of the mean time between admission of flows, or the mean time of the life (duration) of the flows, in the network.

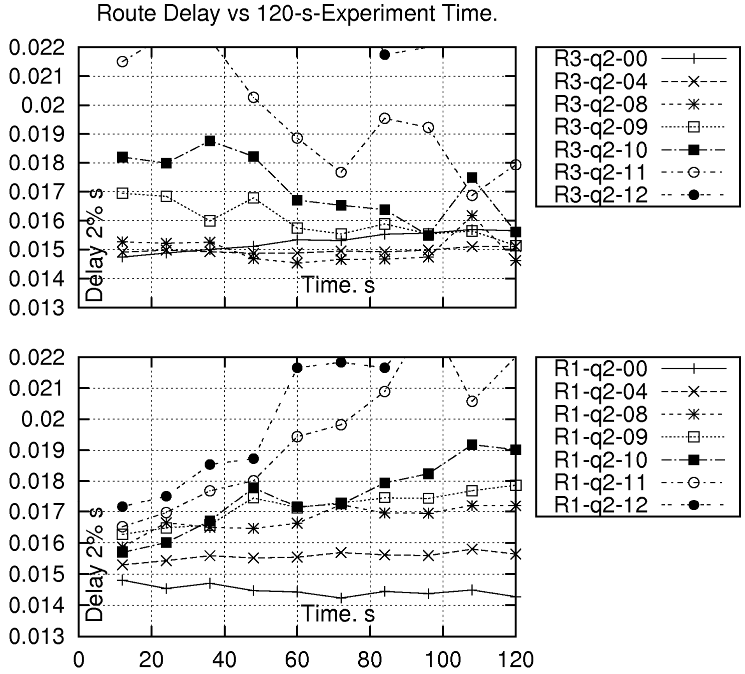

Fig. 14. Delay results for scenario 3 of *q2* case, which are similar to those of **Fig. 8**, but in this case the experiments have 120*s*-duration (instead of 1200*s*) and *T*=120*s*. The *P*-parameter value is still 0.25. Label R1-q2-04 stands for delay of *Route1* in the *q2* case where *Route3* has increased its number of Pareto sources in 4 at 0*s* of the experiment-time. The other labels have similar meaning.

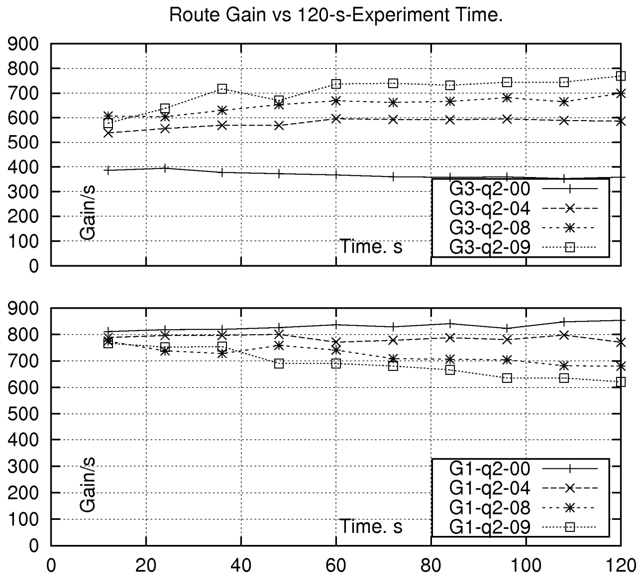

Fig. 15. Gain results for scenario 3 of *q2* case, which are similar to those of **Fig. 9**, but in this case the experiments have 120*s*-duration (instead of 1200*s*) and *T*=120*s*. The *P*-parameter value is still 0.25. Label G1-q2-04 stands for gain of *Route1* in the *q2* case where *Route3* has increased its number of Pareto sources in 4 at 0*s* of the experiment-time. The other labels have similar meaning.

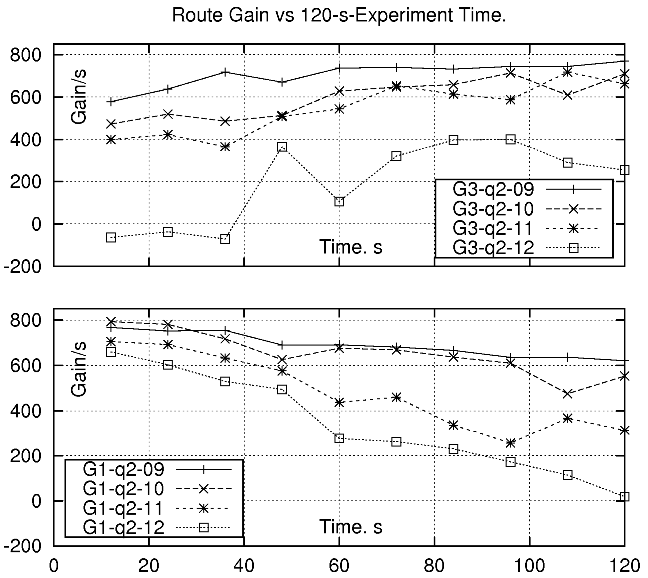

Fig. 16. Gain results for scenario 3 of *q2* case, which are similar to those of **Fig. 10**, but in this case the experiments have 120*s*-duration (instead of 1200*s*) and *T*=120*s*. The *P*-parameter value is still 0.25. Label G1-q2-10 stands for gain of *Route1* in the *q2* case where *Route3* has increased its number of Pareto sources in 10 at 0*s* of the experiment-time. The other labels have similar meaning.

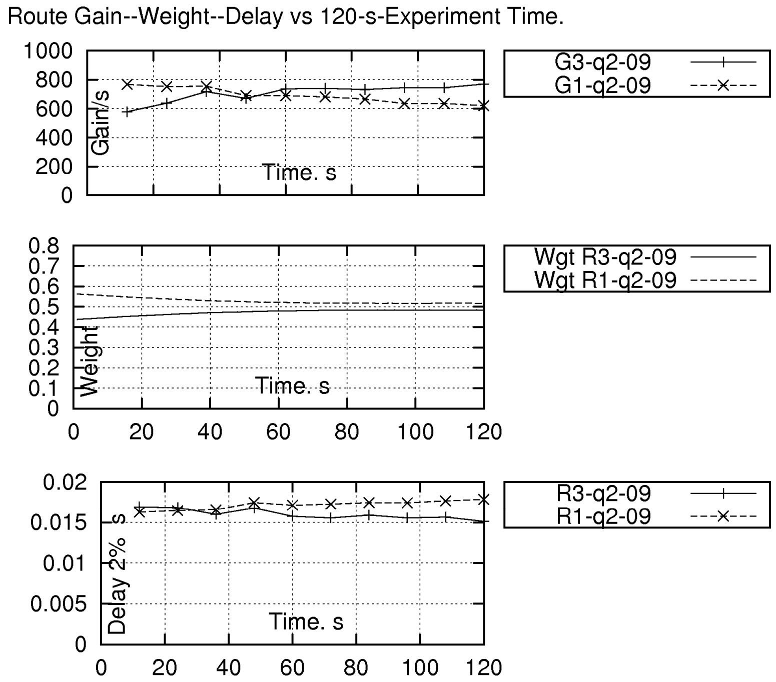

Fig. 17. Gains, weights and delay results for scenario 3 of *q2* case where Route1 increases its number of sources in 9 at 0*s* of experiment-time, which are similar to those of **Fig. 12**, but in this case the experiments have 120*s*-duration (instead of 1200*s*) and *T*=120*s*. The *P*-parameter value is still 0.25. Label G1-q2-09 stands for gain of *Route1* in the *q2* case where *Route3* has increased its number of Pareto sources in 9 at 0*s* of the experiment-time. The other labels have similar meaning.

The next scenarios use a topology where there are two nodes: *c0* and *c1* which are capable, each one, of using the STP method (for the *q2* case) , or of using two queues with fixed weights (for the *q2f* case), , although these scenarios do not use the *q2* case for the nodes but they use cases *q1* and *q2f*. The reason for this is to observe the limit-results to which a set of experiments could arrive to, if the *q2* case were applied at nodes *c0* and *c1*, as it has been explained. This topology is presented in **Fig. 18** which is very similar to that of **Fig. 2**.

In this new topology there are four edge nodes: *e1, e2, e3* and *e4* (instead of 3). Outside the network boundaries there are still three source nodes: *s1, s2, s3*, and still three destination nodes, *d1*, *d2* and *d3*. Inside the network there are still three routes: *Route1*, *Route2*, *Route3*. Node *c1* still acts as an exit node for *Route2*. Links inside the network have a bandwidth of 3*Mb/s*, for the operating Class. The links connecting bounded network with the outside nodes have a bandwidth of 100*Mb*/*s*, with 0.05ms of delay. *Route1* and *Route3* still intersect at the output interface of *c1* (going to *e2*), and *Route2* now intersects with *Route1* at the exit interface of the new core STP-node *c0* (going to *c1*). *Route1* suffers again from two intersections but in this case both intersections are done at the exit interface of STP-nodes. Again *Route1* has disadvantage against *Route3* which is not affected by *Route2*. In the following experiments the situation again is that *Route3* increases its traffic so that *Route1* is affected.

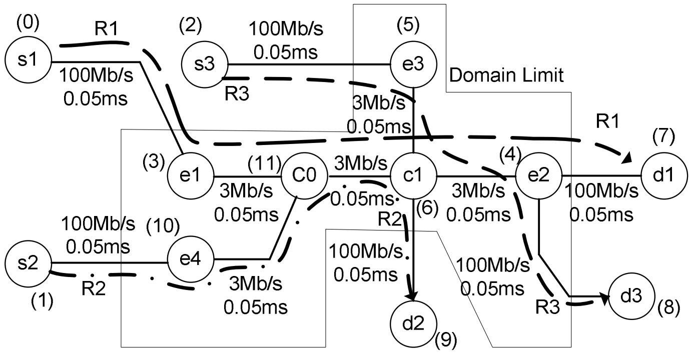

Fig. 18. Topology for next scenarios, where there are two nodes: *c1* and *c0*, capable of using the STP method. This topology is similar to that of **2** but in this case there is a new STP-central node: *c0*, and *Route1* suffers from two intersections at the exit-interface of two STP nodes. R1 stands for *Route1*, R2 and R3 have similar meanings.

It is supposed that the numbers of sources for the routes have been the same for long time before the initial time of the experiments. These numbers are:

Route1: 3 CBR + 17 Pareto
Route2: 3 CBR+ 8 Pareto
Route3: 3 CBR+ 8 Pareto

The traffic-share of *Route1* and *Route2* at the exit interfaces of *c0* is equal to 0.564 and 0.436, respectively. The traffic-share of *Route1* and *Route3* at the exit interface of *c1* is also equal to 0.564 and 0.436, respectively (see **Table 1**).

For the cases of two queues in *c0* and *c1*, the first queue has the traffic of *Route1*. The second queue of *c0* has the traffic of *Route2* and the second queue of *c1* has the traffic of *Route3*. As the STP method tends to assign the weights of the queues the same value as their corresponding traffic-share, then for the experiments with two queues the weights assigned, at t = 0, to *c0* and *c1* are: 0.564 for the first queue and 0.436 for the second queue.

The indicator obtained for the experiments is related with the delay of packets to traverse the routes. As in the previous scenarios, this delay-value is calculated as the 2% percentile of the 2% bigger values of the observed delays of the packets traversing their corresponding route.

Each one of these experiments simulates a time-duration of 120*s*. At time *t*=0*s* of the simulations the amount of sources at *Route2* or *Route3* can change.

The fourth scenario is then built with the new topology, for case *q1* in both nodes (nodes *c0* and *c1* use case *q1*). The sets of experiments are:

REFERIRSE EN CADA NUMERO A The experiment which uses...

1- Experiments which use the traffic indicated (the traffic that has been considered to exist long-time before the start-time of the experiments), from $t = 0$ to $t = 120s$. The result corresponding to the delay of *Route1* is 0.01650*s*.

2- Experiments which use the same traffic as the first set of experiments, with the exception that *Route2* initiates with 17 Pareto sources (*Route3* stays with 8 Pareto sources). The delay of *Route1* results in 0.01864*s* (an increase of 12.95% with regard to the delay in the first set of experiments).

3- Experiments which use the same traffic as the first set of experiments, with the exception that *Route3* which initiates with 17 Pareto sources (*Route2* stays with 8 Pareto sources). The delay of *Route1* results in 0.01864*s* (an increase of 12.97% with regard to the delay in the first set of experiments).

4- Experiments which use the same traffic as the first set of experiments, with the exception that each one of *Route2* and *Route3* initiate with 17 Pareto sources. The delay of *Route1* results in 0.2022 (an increase of 22.59% with regard to the delay in the first set of experiments).

The delay-values and the corresponding delay-values in percentage of these experiments are shown in the lower graphic and in the upper graphic, respectively, of **Fig. 19**.

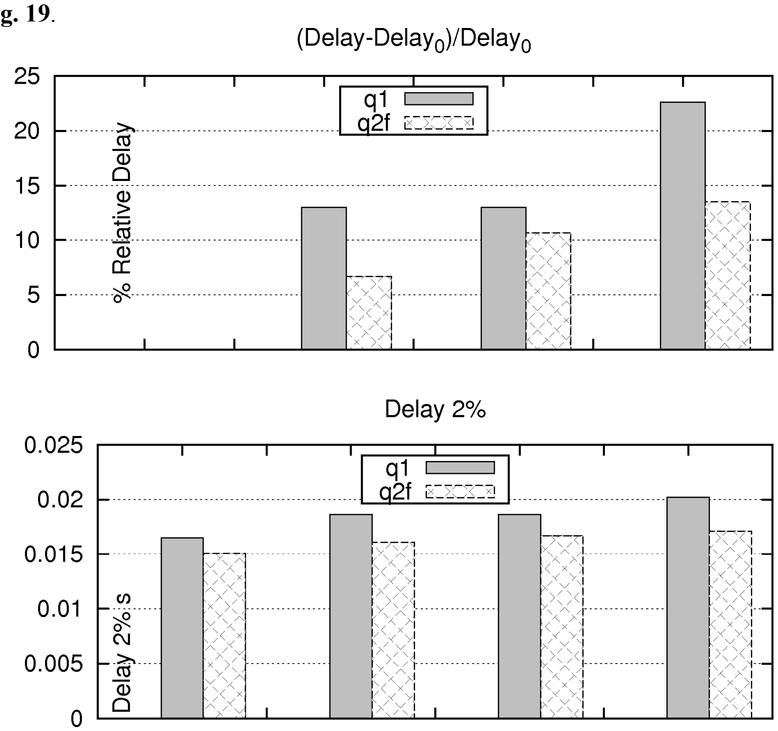

Fig. 19. Delay-values, and the corresponding percentage-values, for the four experiments in the fourth scenario (case *q1* for both nodes *c0* and *c1*), and for the four experiments in the fifth scenario (case *q2f* for both nodes *c0* and *c1*).

The fifth scenario is built with the new topology, for case *q2f* in both nodes (nodes *c0* and *c1* use case *q1*). The sets of experiments are:

1- Experiments which use the traffic indicated (the traffic that has been considered to exist long-time before the start-time of the experiments). The result corresponding to the delay of *Route1* is 0.01507*s*.

2- Experiments which use the same traffic as the first set of experiments, with the exception that *Route2* initiates with 17 Pareto sources (*Route3* stays with 8 Pareto sources). The delay of *Route1* results in 0.01608*s* (an increase of 6.65% with regard to the delay in the first set of experiments).

3- Experiments which use the same traffic as the first set of experiments, with the exception that *Route3* which initiates with 17 Pareto sources (*Route2* stays with 8 Pareto sources). The delay of *Route1* results in 0.01668*s* (an increase of 10.64% with regard to the delay in the first set of experiments).

4- Experiments which use the same traffic as the first set of experiments, with the exception that each one of *Route2* and *Route3* initiate with 17 Pareto sources. The delay of *Route1* results in 0.01711 (an increase of 13.53% with regard to the delay in the first set of experiments).

From Fig 9 it is observed that *Route1* obtains more protection from the use of the *q2f* case at nodes *c0* and *c1* (against the use of case *q1* at those nodes) when the increase of traffic is done in the two routes which intersect with *Route1*, compared with the situation where the increase is done in just any one of those two routes. The q2 case applied at both nodes, *c0* and *c1*, should give as much protection as the case *q2f* immediately after an increase of traffic in the intersecting routes.

6.0 Disadvantage of STP Method (case *q2*).

An example of a possible disadvantage of the *q2* case is, if at a certain moment *Route3* had a big weight and suddenly it decreased its traffic allowing for *Route1* to increase its traffic, and then, before *Route3* began to lose, substantially, its weight share, it increased again its traffic, then *Route1* could suffer from this increase, even more than what it would do in a similar situation with a *q1* case. This disadvantage could be accentuated if the situation is presented in more than one intersecting route of *Route1*.

7.0 Conclusions and Future Work.

The objectives of the proposed method are two-fold. 1- To protect in the short term, a route against traffic increments in intersecting routes, and 2- To dynamically assign, in the long term, a bandwidth share proportional to the average measured demands of the routes traversing congested links. The proposed *Short Term Protection*

(STP) method should help the network administrators to have confidence, at least within a time-window, about the amount of bandwidth they count with.

The method uses scheduling algorithm that allows the use of dynamic weights. The scheduler is work-conserving, so this method does not impose a decrease of the transmission capacity in the network.

The computations to obtain the new weights are separated by relatively long intervals of time (1 second), thus the algorithm should not cause noticeable performance-degradation in actual routers.

The experiments show that the STP method is not better than cases using single-queue configurations when there is enough bandwidth to serve the traffic requirements. This is obvious since in underutilized networks there is no need to protect flows against each other.

Bandwidth reservation, as happens with the fixed weight settings, is still an appreciated form of service which can be a source of important economic-compensations. However, its overall performance depends heavily on the proper weight assignments, which have to be setup by a central authority. On the contrary, the STP method is a simple method that dynamically adapts to the changing network conditions and works independently in every node without the need of a central administration.

The set of experiments presented in this report provide a first insight about the feasibility and performance of the STP method.

The gain figure-of-merit for specific routes helps to observe the protection of a route against the traffic increase in its intersecting routes for the *q2* and *q2f* cases, but the sum of gains of the routes (the global gain) is not appropriate to evaluate this protection.

The route protection of the STP method can be bigger if applied in more than one node in the route's path, although the possible disadvantages of the method in this situation are still to be analyzed.

An analysis about how the STP parameters affect its performance is another important line of research. The loss factor (*P*), the convergence interval (*T*) and the update period (τ) affect the method performance and are dependent on the traffic conditions.

References.

1 Centikaya, C., Knightly, E.: Egress Admission Control. In: IEEE INFOCOM 2000, vol. 3, pp. 1471-1480. (2000).

2 Cetinkaya, C., Kanodia, V., Knightly, E.: Scalable Services via Egress Admission Control. IEEE Transactions on Multimedia: Special Issue on Multimedia over IP, vol. 3(1), pp. 69-81. (2001).

3 Mateos-Papis, A.: A Bandwidth Sharing Method under Node Autonomy and Short-Term Protection for QoS. Accepted in: The 2011 Iberoamerican Conference on Electronics Engineering and Computer Science (CIIECC 2011). Universidad Panamericana. Aguascalientes. Mexico. (2011), in press.

4 Kumar, V., Lakshman, T., Stiliadis, D.: Beyond Best Effort: Router Architectures for the Differentiated Services of Tomorrow's Internet. IEEE Communications Magazine, vol. 36(5), pp. 152-164. (1998).

5 Parekh, A., Gallager, R.: A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single Node Case. IEEE / ACM Transactions on Networking, vol. 1(3), pp. 346-357. (1993).

6 Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W.: RFC 2475. An Architecture for Differentiated Services. The Internet Society (1998).

7 Nichols, K., Blake, S., Baker, F., Black, D.: RFC 2474. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. The Internet Society (1998).

8 Hui, T., Tham, C.: Adaptive Provisioning of Differentiated Services Networks Based on Reinforcement Learning. IEEE Transactions on Systems, Man, and Cybernetics, vol. 33(4), pp. 492-501. (2003).

9 Wang, H., Shen, C., Shin, K.: Adaptive-Weighted Packet Scheduling for Premium Service. In: IEEE International Conference on Communications'2001, pp. 1846-1850. (2001).

10 Li, C., Knightly, E.: Schedulability Criterion and Performance Analysis of Coordinated Schedulers. IEEE/ACM Transactions on Networking, vol. 13(2), pp. 276 – 287. ISSN: 1063-6692. (2005).

11 Floyd, S., Jacobson, V.: Random Early Detection Gateways for Congestions Avoidance. IEEE/ACM Transactions on Networking, vol. 1(4), pp. 397-413. (1993).

12 **ns-2** network simulator, http://nsnam.isi.edu/nsnam/index.php/Main_Page, available Jun 2010.

13 Pieda, P., Ethridge, J., Baines, M., Shallwani, F.: A Network Simulator Differentiated services Implementation. Open IP, Nortel Networks, http://www-sop.inria.fr/members/Eitan.Altman/COURS-NS/DOC/DSnortel.pdf, available Jun 2010.

14 Mrkaic, A. (tutor: U. Fiedler, supervisor: Prof. Dr. B. Plattner). Porting a WFQ Scheduler into ns-2's Diffserv Environment. Computer Engineering and Networks Laboratory (Insitut für Technische Informatik und Kommunikationsnetze). Swiss Federal Institute of Technology. Eidgenössische Technische Hochschule Zürich (2001).

15 Salman, S., Schulzrinne, H.: An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. Department of Computer Science. Columbia University. 2004, http://www1.cs.columbia.edu/~salman/publications/skype1_4.pdf, available Jan.2011.

16 Skype, https://support.skype.com/en/faq/FA1417/How-much-bandwidth-does-Skype-need;jsessionid=5A4C03A96FCF4A2AE9CE5D72D15093CD?frompage=search&q=how+much+bandwidth+for+a+call&fromSearchFirstPage=false, available Jan. 2011.