Analysis of the influence of the transmission delay on the competition between TCP and TFRC

Armando Garcia Berumen, Michel Marot

Institut TELECOM; TELECOM & Management SudParis; CNRS-SAMOVAR
9 Rue Charles Fourier - 91011 Evry CEDEX- France
E-mail {armando.garcia_berumen, michel.marot}@it-sudparis.eu

ABSTRACT

In this paper we analyse the competition between TCP (Vegas and Reno) and TFRC, and particularly the effect of the transmission delay on the way these protocols share the bandwidth. This study is necessary when we want to design, in future systems, mechanisms to share the bandwidth in a parametrisable way, especially in satellite networks like DVB-S2/RCS allowing Bandwidth on Demand (BoD) assignment. Few performance studies have been published on TFRC and its competition with TCP. Our results show that the fairness is quite variable and is highly dependent on the transmission delay. This can be expected since TFRC has been designed to be less reactive than TCP in case of sudden change in the network conditions, but it is also surprising since TFRC is based on a model of TCP in order to be fair. Our study can be particularly useful when designing architectures with long transmission delays such as satellite systems.

Keywords: TCP, TFRC, Satellite networks, fairness.

1. INTRODUCTION

For a long time, TCP has been used to transport data across the Internet, and UDP has been the protocol dedicated for the transport of multimedia flows. While TCP allows facing congestion situations, UDP keeps the same communication rate all along the connection, generating more and more congestion. That is why several TCP friendly congestion control mechanisms for the transport of multimedia flows have been proposed around ten years ago. We are interested by one of them, TFRC [11], which is widely known and used.

Fairness is one of the most important criteria in quality of experience: if a multimedia flow manages well the situation of congestion but takes almost all the bandwidth, the overall feeling of the customer's quality of experience will be very bad. This work is motivated by the intention to propose later an architecture for triple-play services on different types of networks such as wireless and satellite systems [9], [10], where the delay is variable or long. To well understand the competition between both protocols is particularly interesting with wireless systems integrating multimedia services. The study is also relevant for the new satellite generation allowing Bandwidth on Demand (BoD) allocation, where cross-layer mechanisms can be proposed. Surprisingly, though there has been a lot of work on TCP, there have been few performance studies on TFRC. In [1], the authors studied on different timescales the fairness between TFRC and TCP in function of the number of flows and for a fixed delay. They found that both protocols share the bandwidth almost fairly. In [2], they found that by either decreasing or increasing rates of the TCP AIMD (additive increase and multiplicative decrease) mechanism can be adjusted to achieve fairness with TFRC. The authors of [3] noticed that TFRC takes more bandwidth compared to TCP when the number of flows increases. In [4], the same conclusion is obtained and this is attributed to the phase effect [5] which is probably an important factor, but not the only one because the authors still find unfairness between TCP and TFRC, though less, when replacing a Drop tail queue by a RED one.

In this work, we analysed the competition between two TCP versions, namely Reno [6] and Vegas [7] and TFRC for various transmission delays and connection durations. TCP Vegas was introduced in [7]. With respect to TCP Reno the main difference is the congestion control. Reno uses the loss of packets as a signal to inform that there is congestion in the network. TCP Vegas is proactive by detecting the congestion in order to reduce the throughput and trying to reduce the losses [8]. To detect network congestion, it uses the following equation:

$$diff = \left(\frac{W}{RTT_{\min}} - \frac{W}{RTT} \right) RTT_{\min} = W \frac{RTT - RTT_{\min}}{RTT} \qquad (1)$$

Where:

- W is the Congestion Window
- RTT is the actual Round-Trip Time
- RTT_{min} is the minimal RTT
- W/RTT_{min} is the expected rate
- W/RTT is the actual Rate

And it adjusts the window congestion as follows:

If $diff < \alpha$ Then $W := W+1$
Else if $diff > \beta$ Then $W := W-1$
Else $W := W$

α and β are threshold parameters with values of 1 and 3 respectively.

TFRC is considered in [11]. It was designed for applications with a fixed packet size and with the possibility of changing the sending rate in presence of congestion. It was developed to compete for bandwidth with other TCP flows. TFRC uses a throughput equation in its congestion control, this equation being a model of TCP Throughput. The interest for this kind of equation is that it is expected to be fair with TCP. This equation is a direct function of losses, round-trip time and the packet size and defines the allowed sending rate, which is also the expected TCP rate in the same conditions.

$$T = \frac{s}{R * \sqrt{2 * b * \frac{p}{3}} + (t\_RTO * (3 * \sqrt{3 * b * \frac{p}{8}} * p * (1 + 32 * p^2)))} \quad (2)$$

Where:

- T is the transmit rate in bytes/second;
- s is the packet size in bytes;
- R is the round trip time in seconds;
- p is the loss event rate (between 0 and 1.0);
- *t_RTO* is the TCP retransmission timeout value in seconds;
- b is the number of packets acknowledged by a single TCP acknowledgment.
- A simplification can be done if we consider *t_RTO =4*R.*

The paper is organised as follows. Section 2 describes the fluid approach for Reno, Vegas, TFRC and the integrated model used for the simulations. The simulations and numerical results are provided in Section 3. The conclusions and the future work are finally presented in Section 4.

2. FLUID MODELS AND INTEGRATED APPROACH

We used a fluid approach, integrating the validated models given by Bonald in [12] and Barbera et al. in [13].

2.1 Model for Reno and Vegas

By using a fluid approximation (because their simplicity and speed), it is possible to characterise a TCP connection with the next scheme, Figure 1.

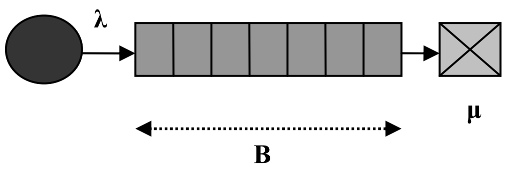

Figure 1. Scheme used to model TCP connections.

Let a TCP connection be and let define:

- μ : Throughput of the queue;
- λ : Throughput of the connection;
- τ : The latency for the connection;
- W: Window size;
- RTT: Round Trip Time;
- B: Buffer size;

According to [12]:

- The throughput for a connection can be calculated as follows:

$$\lambda(t) = \min\left(\mu\ , \frac{W(t)}{\tau}\right) \quad (3)$$

- The instantaneous value for the buffer at time t *Buffer (t)*,can be calculated as follows:

$$\mu = \frac{W(t)}{\frac{Buffer(t)}{\mu} + \tau} \quad (4)$$

- The possible values for RTT are given by:

$$RTT(t) = \max\left(\frac{W(t)}{\mu}, \tau\right) \quad (5)$$

- The evolution for the window is as follows:

For TCP Reno:

$$\frac{dW(t)}{dt} = \frac{1}{RTT(t)} \quad (6)$$

For TCP Vegas:

$$\frac{\mathrm{dW(t)}}{\mathrm{dt}} = \frac{-\frac{1}{2}[\mathrm{sgn}(diff(t) \times \tau - \alpha) + \mathrm{sgn}(diff(t) \times \tau - \beta)]}{RTT(t)} \quad (7)$$

$$W(t) = W_{\max} \Rightarrow W(t^+) = \gamma W(t) \quad (8)$$

2.2 Model for TFRC

In [13], a fluid model was proposed for TFRC. The equations to calculate the target rate are as follows for the case where there are several TFRC connections sharing a same buffer.

$$T^{(t)} = \begin{cases} \min\begin{pmatrix} T^{(t)} + s/RTT \\ \max(\min(T, 2T^{(R)}), s/t_{mbi}) \end{pmatrix} & if\ p > 0 \\ \max(\min(2T^{(t)}, 2T^{(R)}), s/RTT) & if\ \ p = 0 \end{cases} \quad (9)$$

Where:

- T is the TFRC throughput calculated using (2);
- *s/RTT* is the minimum sending rate during the slow-start;
- $T^{(R)}$ is the estimated throughput at which data was received at the receiver;

- t_{mbi} is the maximum inter-packet backoff interval (64 seconds);
- p is the loss event rate (p=0, determines the slow-start) ;
- $T^{(t)}$ is the target throughput;

The actual rate sent in the network is:

$$\frac{SQRTT(t)}{\sqrt{RTT(t)}}\times T(t)^{(t)} \qquad (10)$$

Where

- RTT is the round-trip time (in seconds) suffered by the packets at the time instant t;
- *SQRTT(t)* is the estimated square root long-term *RTT* (in seconds$^{1/2}$) at the time instant t;

This prevents TFRC to be too volatile which would be very bad for multimedia applications.

$$p^{(LOSS)}(t)=\begin{cases}0 & if\ q\ (t)\leq B\\ \dfrac{\max\left(0,\sum\lambda_k(t)-C\right)}{\sum\lambda_k(t)} & if\ q\ (t)=B\end{cases} \qquad (11)$$

Where:

- $\lambda_k(t)$ is the average arrival rate (in bytes/s) for a given connection;
- q (t) is the queue length at time t;
- B is the buffer size;
- C is the transmission capacity (bytes/s);

The evolution of the queue length q(t) can be calculated as follows:

$$\frac{dq(t)}{dt}=-C+(1-p^{(LOSS)}(t))\cdot\sum\lambda_k(t) \qquad (12)$$

With the condition $0\leq q\ \ (t)\leq B$

The output rate, μ (t) for the buffer is:

$$\mu\left(t+\frac{q(t)}{C}\right)=\frac{\lambda_k(t)\cdot(1-p^{(LOSS)}(t))}{1+\dfrac{1}{C}\dfrac{dq(t)}{dt}} \qquad (13)$$

The loss event rate p used in (2), is estimated via a set of equations. Because of lack of space, the reader is referred to [13], for details on how p is calculated.
Note: p is different from $p^{(LOSS)}$.

2.3 Integrated Fluid Model

We integrate both previous models sharing the same buffer in the next topology, Figure 2, and it is proposed to study the integrated approximation.

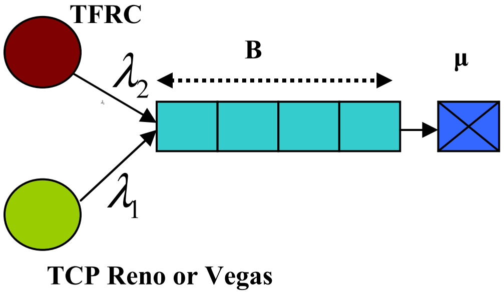

Figure 2. Topology used in the integrated model.

Be λ_1 and λ_2 the average arrivals rate for respectively the TCP Vegas (or Reno) and TFRC connections and substituting λ_1 and λ_2 in (11), we obtain:

$$p^{(LOSS)}(t)=\begin{cases}0 & if\ q\ \ (t)\leq B\\ \dfrac{\max(0,((\lambda_1+\lambda_2)-C))}{\lambda_1+\lambda_2} & if\ q\ \ (t)=B\end{cases} \qquad (14)$$

Given that only the TFRC connection is governed by the loss factor p, and equation (12) becomes:

$$\frac{dq(t)}{dt}=-C+(1-p^{(LOSS)}(t))\cdot\lambda_2+\lambda_1 \qquad (15)$$

With the condition $0\leq q\ \ (t)\leq B$

The modification for equation (13) is:

$$\mu\left(t+\frac{q(t)}{C}\right)=\frac{\lambda_2\cdot(1-p^{(LOSS)}(t))+\lambda_1}{1+\dfrac{1}{C}\dfrac{dq(t)}{dt}} \qquad (16)$$

With the previous equations (14-16), it is possible to analyse the throughput behaviour, when two connections (TFRC and Vegas or Reno) arrive at the same buffer and they share the same link. It consists in integrating a differential system. The evolution of *W(t)* is still governed by the equations (6), (7) and (8).

3. SIMULATIONS AND NUMERICAL RESULTS

We performed two sets of simulations. In the first part, only one TCP connection competing with one infinite TFRC connection is observed, in function of the time, for different transmission delays. It allows understanding the dynamic of the competition between both connections. In the second set of simulations, we observed what happens when, instead of having two infinite connections, there are two connections restarted after a random time. This allowed to take into account the slow start phase, the effect of which is important on the performance of long transmission delay networks. For all the simulations, the confidence interval was calculated with an error less than 1%.

3.1 One TCP and one TFRC infinite simultaneous connections

To investigate how TFRC coexists with Reno and Vegas when they are competing for the bandwidth with different delays values, we performed several simulations. This section presents the most important results. The parameters used for the simulations, are a capacity of 5 Mbit/s, a packet size of 1000 bytes, and a buffer size of 100 packets. We tested the model for different values of link delay: a small delay (10 ms), a long one (200 ms) and a very long one (540 ms).

Figure 3 shows the results using a delay of 10 ms. We can observe that most of the bandwidth is taken by the TFRC connection; it means that TFRC is more aggressive in this case than TCP. The occupation of the queue for the same link delay is shown on Figure 4. Related to the Figure 3, most of the packets on the buffer belong to the TFRC traffic.

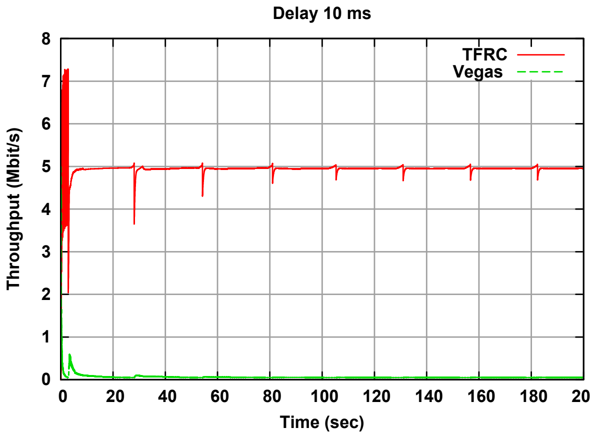

Figure 3. Throughput for a delay of 10 ms (TFRC-Vegas).

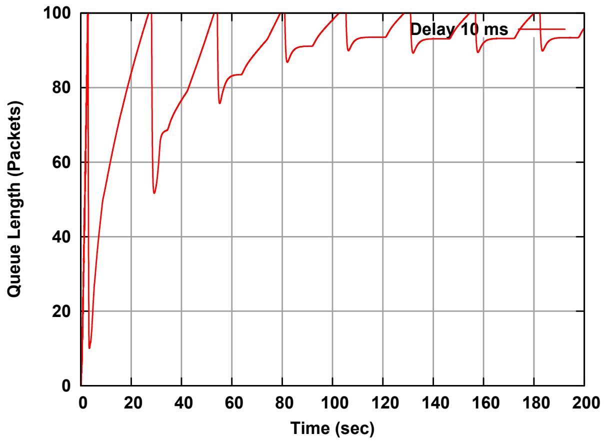

Figure 4. Queue occupation for a delay of 10 ms.

The results for a long delay (200ms) are shown on Figure 5 and 6.

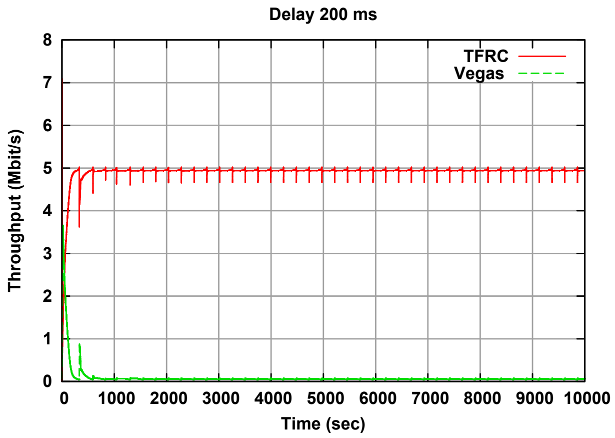

Figure 5. Throughput for a mean delay (200 ms).

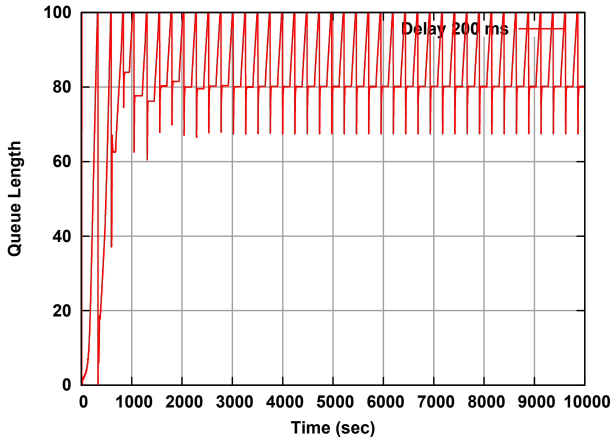

Figure 6. Queue occupation for mean delay (200 ms).

In order to simulate both connections over a satellite link, we consider this as a very long delay (540 ms). The results for the throughput using these values are shown on Figure 7. We note that at the beginning, in the slow-start phase, Vegas tries to take most of the bandwidth, but when the TFRC traffic increases, this causes a TCP Vegas diminution.

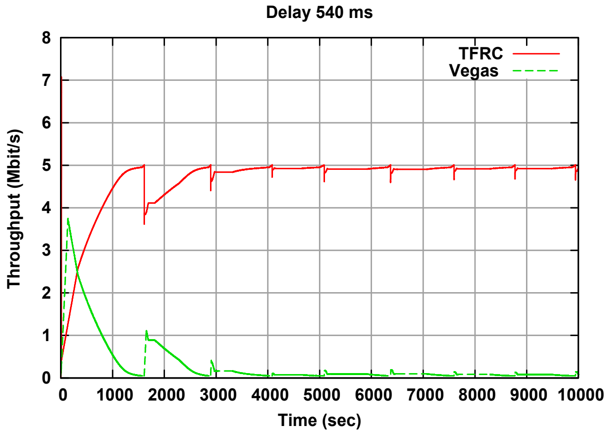

Figure 7. Throughput for a very long delay (540 ms).

The curve for the queue occupation is shown on Figure 8. The time to fill out the queue is also long. The diminution of the peaks of the TCP Vegas causes a diminution on the queue occupation. The window behaviour for TCP Vegas is presented on Figure 9. The value is always low, except at the beginning. This is due to TFRC which is in the slow-start phase.

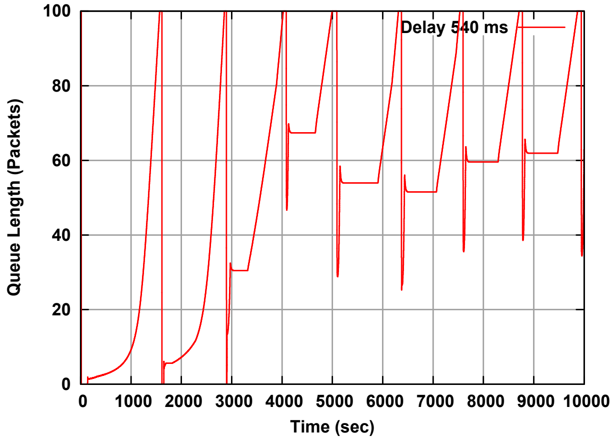

Figure 8. Queue occupation for a delay of 540 ms.

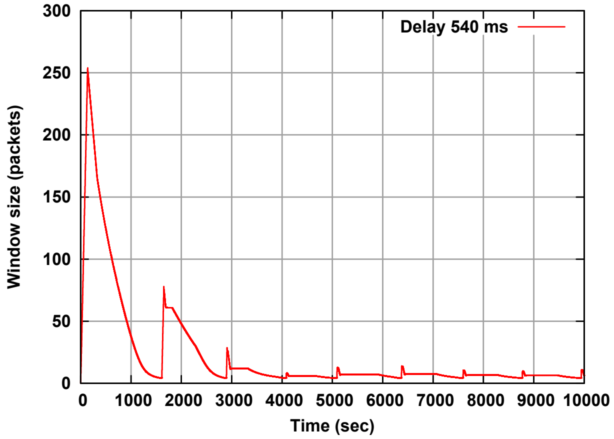

Figure 9. Window behaviour for TCP Vegas.

Figures 10 and 11, show the behaviour when TFRC and Reno, instead of Vegas compete for the bandwidth over a link delay of 10 ms. We can note that TFRC starves much of the bandwidth.

In conclusion for this section, it can be noticed that when there are simultaneously a TFRC connection and a TCP connection, (either Vegas or Reno) after certain time, TFRC starves always most of the bandwidth.

3.2 One TCP and one TFRC connections restarting randomly after a certain time

In order to observe the influence of the slow start convergence time on the fairness, the connections are restarted after a random exponentially distributed time. The experiments carried out were for the case where we increased in similar way both mean durations (from 10 to 240 sec), Figure 12, when the Vegas mean duration was fixed and the mean duration of the TFRC connection increased, Figure 13, and in the third case, where the TFRC mean duration was fixed and Vegas varied, Figure 14.

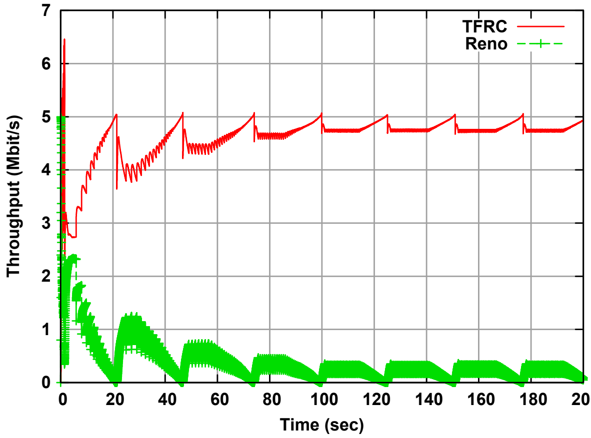

Figure 10. Throughput for a delay of 10 ms (TFRC-Reno).

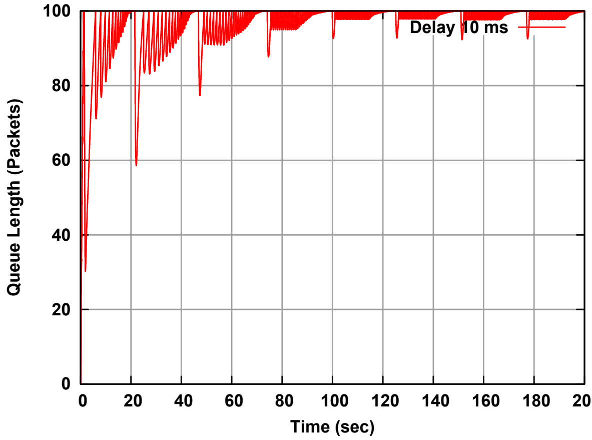

Figure 11. Queue occupation with a delay of 10 ms (TFRC-Reno).

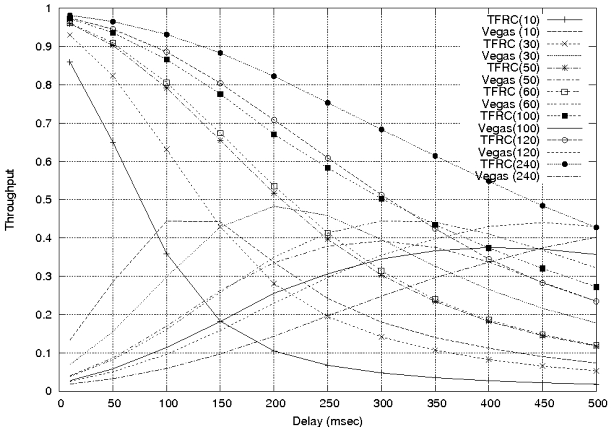

Figure 12. Throughput when the connection has the same exponential duration periods in seconds.

The mean throughput is very high for small delays for TFRC and decreases in function of the transmission delays; this is because TFRC has more difficulties to converge to the full bandwidth occupation. As a consequence, the TCP throughput increases, occupying the remaining bandwidth. As a consequence, it seems that it is TFRC which determines the proportion of bandwidth taken by each connection.

It can be noticed that, after a given threshold for the transmission delay, both throughputs decreases, TCP was lesser than TFRC but more reactive, but it experienced difficulties to converge towards the full bandwidth occupation in presence of long transmission delays.

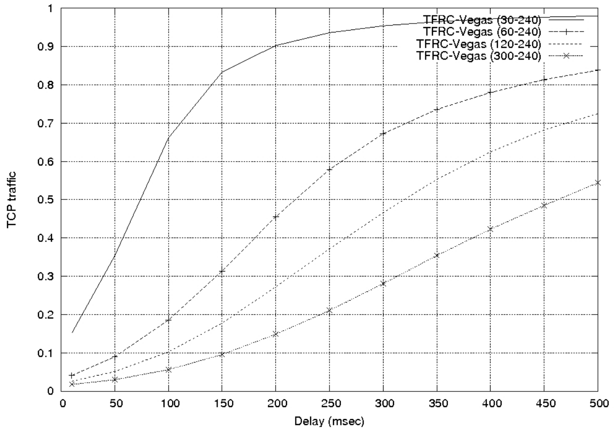

Figure 13. Ratio of the TCP throughput over the sum of TCP and TFRC throughputs when TFRC takes different values and TCP is fixed to 240 sec.

On Figures 13 and 14 are plotted the ratio of the mean throughput of the TCP connection over the sum of the mean throughput of the TCP connection plus the mean throughput of the TFRC connection for various mean connections durations and in function of the transmission delay. When the duration of the TCP connection increases, compared to the duration of the TFRC connection, its proportion of bandwidth increases, but it is not fair at all. One can clearly see that the fairness is a function of the transmission duration because of the difference of the convergence times of both connections towards the full bandwidth occupation.

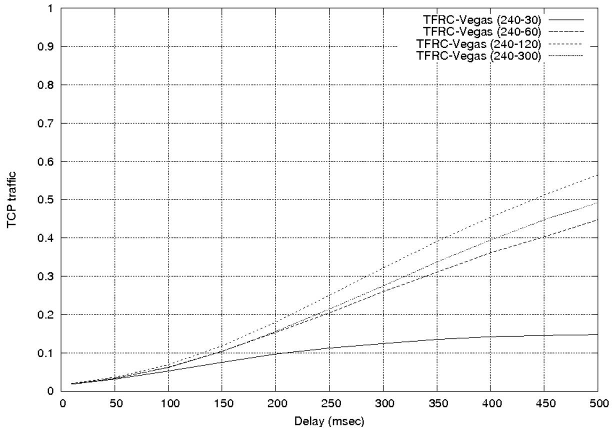

Figure 14. Ratio of the TCP throughput over the sum of TCP and TFRC throughputs when Vegas takes different values and TFRC is fixed to 240 sec.

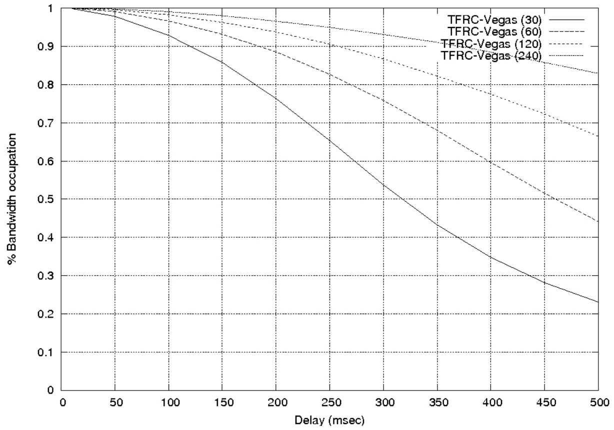

Figure 15. Bandwidth occupation for similar mean duration case (Vegas).

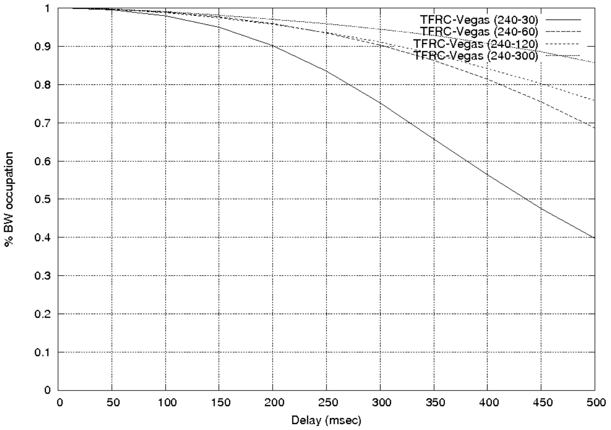

Figure 16. Bandwidth occupation when the mean duration of the TFRC connections is fixed to 240 sec.

Figures 15 to 17 show the bandwidth occupation for the three above mentioned cases. Both traffics were added to present the overall bandwidth occupation. The bandwidth occupation is mainly determined by the ability for both traffics to converge towards the full utilisation of the link, and so decreases in function of the delay.

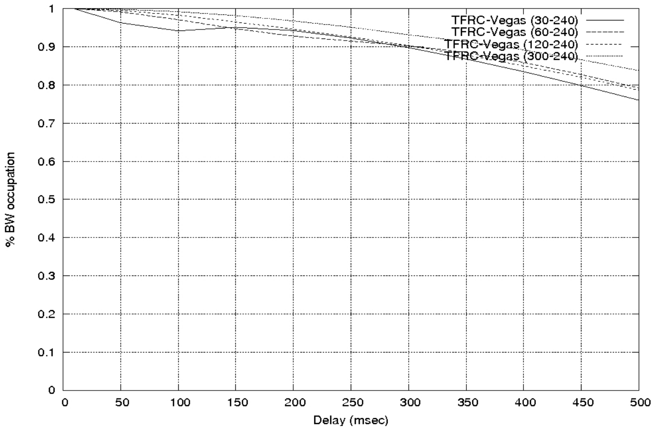

Figure 17. Bandwidth occupation when the mean duration of the Vegas connections is fixed.

To compare the performance of TFRC with another TCP version, we performed the same simulations when a connection starts and stops, but now with TCP Reno (Figures 18-23). For all the cases TFRC is more aggressive, but also affected by the long delays.

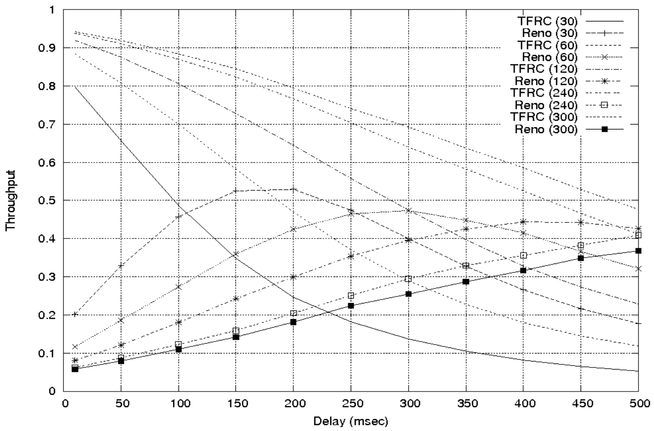

Figure 18. Throughput when the connection has the same exponential duration period in seconds (Reno).

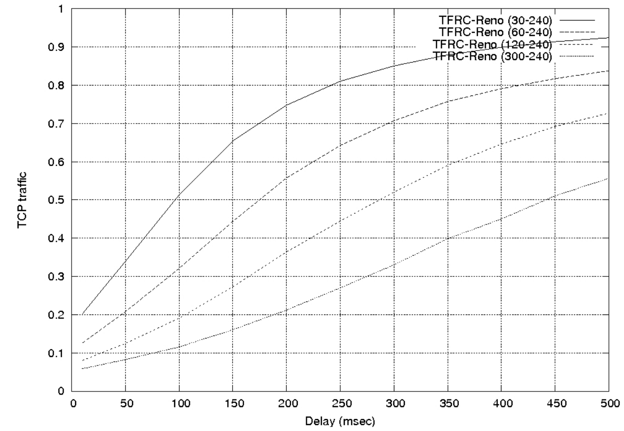

Figure 19. Ratio of the TCP throughput over the sum of TCP and TFRC throughputs when TFRC takes different values and TCP is fixed to 240 sec.

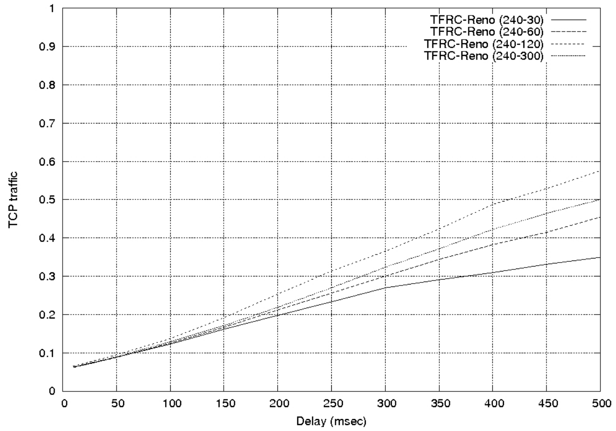

Figure 20. Ratio of the TCP throughput over the sum of TCP and TFRC throughputs when Reno takes different values and TFRC is fixed to 240 sec.

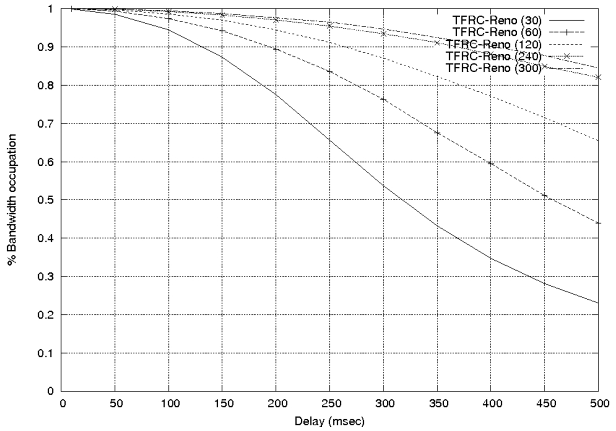

Figure 21. Bandwidth occupation for similar mean duration case (Reno).

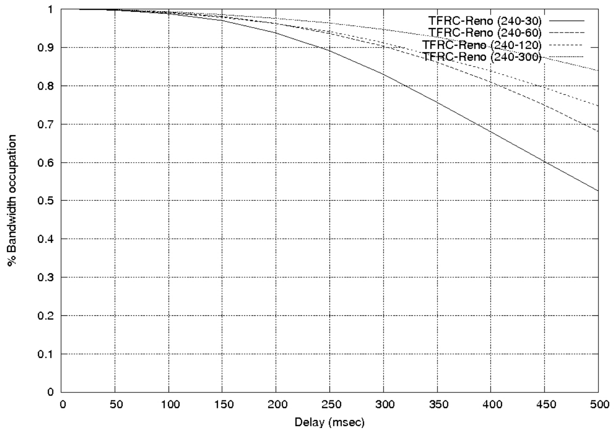

Figure 22. Bandwidth occupation when the mean duration of the TFRC connections is fixed (Reno varies).

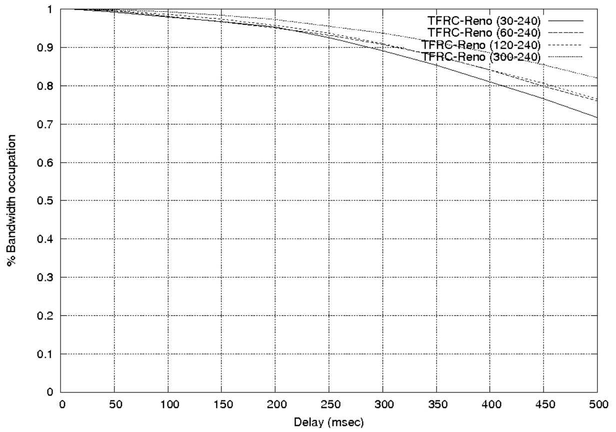

Figure 23. Bandwidth occupation when the mean duration of the Reno connections is fixed (240 sec)

Globally the results are similar for Reno and Vegas, except that Reno is more aggressive in case of long durations. It means that the steady state regime is more aggressive for Reno than for Vegas in competing with TFRC, which can be expected since TFRC is based on a model of TCP Reno.

4. CONCLUSIONS AND FUTURE WORK

Despite the existence of many fairness studies, few works had considered a comparison between TCP and TFRC. In this work we presented an analysis based on a fluid model to evaluate the competition between TCP and TFRC for TCP Reno and Vegas. Reno and Vegas were considered for TCP applications and TFRC for multimedia traffic. The results have shown that it is not possible to have fairness if Vegas and TFRC coexist. A similar situation appears when TFRC and Reno are present over the same link. When they compete for the bandwidth, TFRC is more aggressive for small delays. On the other hand, the performance of TFRC is more affected than TCP Reno and Vegas when they are used in long delay links. On one hand, since TFRC is based on a model of TCP, precisely in order to be fair with TCP, its behaviour could be expected to be fairer. On the other hand, since TFRC has been designed to be less reactive than TCP in order to avoid sudden changes in multimedia traffic, which would be very bad for the applications, it is not surprising to have TFRC more aggressive than TCP. The only solution to make TFRC fair would then to remove the inertia introduced in the behaviour of TFRC and to use buffering techniques jointly designed with adaptive coding techniques to avoid sudden changes in the experienced user quality.

A scenario where TFRC and TCP are mixed in a satellite link is possible when the user wants to pay less. Both traffics can be processed in a similar way as data, but using differentiated services.

The main importance of the study resides in showing the unfairness between the protocols, in which way it is, and how it varies in function of the transmission delay. Moreover the difference in the competition for bandwidth is greatly affected by the difference of aggressiveness of the slow start phases.

This kind of study can be used also to compare TFRC face to other recent versions of TCP such as CUBIC [14]. But, since these versions are more scalable than Vegas or Reno, the difference of fairness would probably still be observed. As future work, we will study the influence of the number of simultaneous connections observed together with the influence of the transmission delay. The authors are currently proposing a cross-layer mechanism to allow sharing the bandwidth in a parametrisable way between both TCP and TFRC protocols, in DVB-S2/RCS satellite systems. We also are interested in how to reduce the allocation delay in this kind of systems using cross-layer techniques.

5. REFERENCES

[1] Sally Floyd, Mark Handley and J. Widmer, "Equation-based congestion control for unicast applications", ACM SIGCOMM 2000, August 2000.

[2] Sally Floyd, Mark Handley, "A comparison of equation-based and AIMD congestion control", http://www.aciri.org/tfrc/, May 2000.

[3] M. N. Ahsan and Md. S. Rahman, "TCP friendly rate control performance with many flows", Journal of Electrical Engineering, Vol. EE 31, No. I & II, December 2004.

[4] J. Tian, S. Xiangzhi and W. Wenjun, "The effect of the inter-fairness of TCP and TFRC by the phase of TCP traffics", 0-7695-1381-6/ 2001 IEEE.

[5] Sally Floyd, Van Jacobson, "On traffic Effects in packet-switched gateways", Computer Communication Review V.21 N.2, April 1991.

[6] Kevin Fall and Sally Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP", Computer Communication Review, vol. 26, pp. 5--21, July 1996.

[7] L.S Brakmo, S. W. O'Malley, L.L Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", Proceedings of ACM SIGCOMM'94, 1994 pp.22-35.

[8] C. Samios and M. K Vernon, "Modelling the throughput of TCP Vegas", SIGMETRICS'03, San Diego California, June 2003.

[9] M. Kalama, M. Marot, A.G. Berumen, et al, "Cross-layer design for TCP applications and VoIP over satellite", Proc. 12th Ka-band and Broadband communications Conference, Napoli, Italy, September 2006.

[10] A. G. Berumen, M. Marot, "Cross-layer Algorithm for VoIP applications over satellite", Proc. 18th Annual IEEE international symposium on personal, indoor and mobile radio communications, PIMRC 2007, Athens Greece September 2007.

[11] M. Handley, S. Floyd, J. Padhye and J. Widmer, TCP Friendly Rate Control (TFRC): Protocol Specification", RFC 3448, Network Working Group, January 2003.

[12] T. Bonald, "Comparison of TCP Reno and TCP Vegas via fluid approximation", Rapport de Recherche N° 3563, INRIA, November 1998.

[13] M. Barbera, F. Licandro, A. Lombardio and G. Schembra, "A fluid-flow analytical model of networked multimedia TFRC Traffic Sources", IEEE Globecom St. Louis Missouri, November-December 2005.

[14] I. Rhee and L. Xu, "CUBIC a new TCP-Friendly high-speed TCP variant", In Proc. Workshop on Protocols for Fast Long Distance Networks, 2005, 2005.