

On the Accuracy of Analytical Models of TCP Throughput

Ibtissam El Khayat, Pierre Geurts, and Guy Leduc

Department of Electrical Engineering and Computer Science,

University of Liège

{elkhayat, geurts, leduc}@montefiore.ulg.ac.be

Abstract. Based on a large set of TCP sessions we first study the accuracy of two well-known analytical models (SQRT and PFTK) of the TCP average rate. This study shows that these models are far from being accurate on average. Actually, our simulations show that 70% of their predictions exceed the boundaries of TCP-Friendliness, thus questioning their use in the design of new TCP-Friendly transport protocols. Our study also shows that the inaccuracy of the PFTK model is largely due to its inability to make the distinction between the two packet loss detection methods used by TCP: triple duplicate acknowledgments or timeout expirations. We then use supervised learning techniques to infer models of the TCP rate. These models show important accuracy improvements when they take into account the two types of losses. This suggests that analytical model of TCP throughput should certainly benefit from the incorporation of the timeout loss rate.

1 Introduction and Motivation

TCP is a transport protocol widely used by applications like remote access (ssh, telnet), file transfer (ftp), and Peer-to-Peer. It occupies more than 90% of Internet resources [8]. The success of this protocol lies in the reliable transfer it offers. To avoid network collapse, TCP reacts to congestion by reducing its rate. This reduction depends on the way the loss, which is used as indication of congestion, is detected. If the loss is detected by duplicate acknowledgments (typically 3: RFC-2581), the congestion window is halved. Otherwise, the loss is detected by timeout and the sender reduces the size of its congestion window to one packet. Depending on the way the loss is detected, TCP enters a slow-start phase (in the case of timeout) or congestion avoidance phase (in the case of triple duplicates). The way the sender increases its congestion window is also phase dependent. More details can be found in RFC-1122, RFC-2581 and in [3].

The large usage of TCP makes it benefit from a special attention. Many studies have been done to understand its behaviour and the parameters it depends on. Several analytical models (e.g. [11], [14], [10]) have been developed for the throughput of long-term TCP connections and have helped understand the impact of certain parameters. However, these models have been obtained under different assumptions and all assume that the phase of fast recovery is negligible and that the source resumes the linear increase of its congestion window directly

after the reduction (as pointed by Altman et al. in [1]). More sophisticated models exist that try to alleviate some of these hypotheses. For example, [9] and [1] take into account the effect of the window size on the round-trip time and also the correlation between losses. The model proposed in [15] is more accurate and furthermore it takes into account the slow start, which makes it usable also for short sessions.

Another consequence of the success of TCP is that any new protocol deployed on the internet should be TCP-friendly [6] in order not to disturb 90% of the traffic. To reach this TCP-friendliness, some multicast and real time protocols (e.g. [16], [18], [7], [5]) have used the analytical models of the TCP throughput, to adapt their rate so as to obtain similar throughput as TCP in the same network conditions. The most popular models for these applications are the SQRT [11] and the PFTK [14] formula.

According to these two models, the TCP throughput is inversely proportional to the round-trip time (at least in the case of low loss rate). This statement has an important consequence: if two TCP sessions share the same bottleneck then the ratio between their throughput (in terms of packets) is equal to the inverse of the ratio between their round-trip times. A simple experiment can however show that this statement is not always true. To this end, we ran different scenarios (100) with a simple topology consisting of one bottleneck over which a certain number of TCP New-Reno sessions compete. They all have the same packet size and different round-trip times. In each simulation, the bandwidth of the bottleneck and the number of concurrent sessions are chosen randomly and the loss rate is low (under 2%). Over all the scenarios, we record for each pair (i, j) of TCP connections the ratio between their throughput $(\frac{B_i}{B_j})$ and the inverse ratio between the average of their round-trip times $(\frac{RTT_j}{RTT_i})$. All the pairs $(\frac{RTT_j}{RTT_i}, \frac{B_i}{B_j})$ are represented by a point in the scatter plot of Figure 1. According to the models, each point should be on the line $y = x$ since the two ratios should be equal. Figure 1 shows that even in the case of simple topologies, the scatter plot is not fitting the model. In some cases, the ratio between the throughput equals seven times the inverse ratio between the average round-trip times. This also means that TCP is not always fair towards other TCP sessions. In [2], the authors have observed this unfairness in short term sessions.

In this paper, we propose to study the accuracy of the SQRT and PFTK models that are recalled in Section 2. The approach we propose for the validation is based on gathering an important number of TCP sessions obtained in different randomly generated topologies and scenarios, and comparing the throughput really obtained with the predicted one. The way we proceed and the results of the validation are given in Section 3. We show then, in section 4, that the main reason of the inaccuracy of the two models is the aggregation of the losses, which is also present in other TCP models [1, 9, 12, 13]. To the best of our knowledge, none of the previous work has made distinction between the two loss rates. In Section 5, we show that taking into account the timeout loss rate can greatly improve the accuracy of TCP throughput models obtained by supervised learning techniques. Finally, we conclude in Section 6.

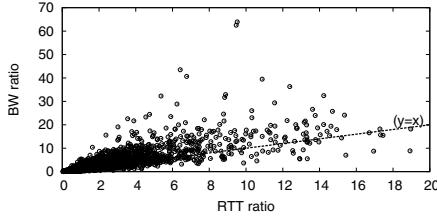


Fig. 1. The ratio of 2 session rates versus the inverse ratio of their RTTs

2 Analytical Models of TCP Throughput

In this section, we give a brief reminder of the PFTK and SQRT models which are often used by other protocols to offer TCP-Friendliness. Both have been developed for long-term TCP connections (i.e., for flows with a large amount of data to send, such as file transfers).

In 1997, the only formula modelling the throughput of TCP was the one developed by Mathis et al. in [11] which is:

$$B_{tcp} = \frac{C \cdot MSS}{RTT \sqrt{p}} \quad (1)$$

where $C \approx 1.22$, MSS is the maximum segment size, RTT the average round-trip time, and p the loss rate over the session. This model is often called the SQRT model. Only the congestion avoidance is taken into account in this model and all the losses are assumed to be detected by triple duplicates. This assumption implies that the loss rate is low and that there is no timeout expiration.

In 1998, Padhye et al. [14] have developed a more complex formula, taking into account losses detected by timeouts which are frequent in high loss rate environments. The details and the hypothesis made for this model can be found in [14]. The formula, called PFTK, is summarised as follows:

$$B_{tcp} \approx \min\left(\frac{rwnd}{RTT}, \frac{MSS}{RTT \sqrt{\frac{2bp}{3} + f(p)}}\right) \quad (2)$$

with $f(p) = T_0 \min(1, 3\sqrt{\frac{3bp}{8}})p(1 + 32p^2)$

where MSS , p and RTT are as described in the SQRT formula. T_0 is the initial value of the timeout, $rwnd$ the receiver window and b the number of packets acknowledged at once. This formula is said in [14] to be developed for Reno, but it is based on the hypothesis that all the packets sent after the loss and belonging to the same window are lost. This hypothesis means that the sender can only decrease its congestion window once by round-trip time, which is true only in the case of NewReno (see [1]).

The two formulas do not take the effect of slow-start phases into account, which makes them unusable for the prediction of the throughput of short TCP sessions. They also neglect the fast recovery phase as said previously. Other

assumptions have also been made for the modelling but we do not develop them in the paper. The reader can refer to the original papers and to [1] or [17] for more detailed discussions of these models.

3 Models Validation

In this section, we propose to validate the two models (SQRT and PFTK) by using a generic approach based on random simulations. More precisely, the quality of the models is measured by their ability at predicting the throughput of TCP in various topologies and scenarios. The way we generated these topologies and scenarios is described in Section 3.1. In Section 3.2, we give the criteria we use to measure the quality of model predictions. The results of the validation of the two formulas are discussed in Section 3.3.

3.1 Topologies and Scenarios Used

To validate the SQRT and PFTK formulas, we use 7600 TCP New-Reno¹ sessions chosen randomly over thousands of topologies with different scenarios. Since the two formulas are developed for long-term throughput, we choose TCP sessions that last at least 400 seconds and for which the time needed to send all packets in a window is smaller than the RTT. The receiver window is chosen very large so as not to be the bottleneck. To create a topology and a scenario we proceed as follows: a network topology is generated randomly and then the network is simulated during a fixed amount of time, again by generating the traffic randomly. At the end of the simulation, we collect for all TCP New-Reno sessions that last at least 400 seconds, the loss ratio p computed over the whole session, the value of Maximum Segment Size (*MSS*), the value of the timeout T_0 , the average round-trip time *RTT*, the number of packets acknowledged at once b , and the TCP throughput obtained. This procedure is repeated until we have a sufficient number of sessions in the database.

To generate a random topology, we first select a random number of nodes (between 10 and 600) and then choose randomly the connections between these nodes. The bandwidth, the propagation delay, and the buffer size of the links were chosen randomly. The bandwidth is chosen between 56Kb/s and 100Mb/s while the propagation delay varies between 0.1ms and 500ms.

Concerning the traffic, the flows were chosen randomly among TCP and other types of traffic based on UDP and proposed by ns-2. The senders, the receivers, and the duration of each traffic were set randomly.

We are aware that this large set of random topologies and traffic conditions may also include many non realistic ones, but since the various analytical models of TCP are topology and traffic unaware, they are not supposed to give good results only in realistic scenarios. Moreover, as characterizing realistic scenarios

¹ We have used TCP NewReno, and not Reno, because, as stated earlier, both formulas are based on the assumption that the congestion window can only decrease once per RTT.

is beyond the state of the art, trying to restrict ourselves to a large set of so-called realistic scenarios may create the risk of being too restrictive and thus introduce a bias.

3.2 Evaluation Criteria

The quality of a model, or of an estimator, of which the goal is to predict a numerical output from some inputs, depends on how well it fits the data to predict. The closer the predicted value to the observed one, the more accurate is the model. Its accuracy, or its adjustment, can be measured by different statistics. The mean square error is often used as well as the coefficient of determination. The mean square error is equal to:

$$MSE = \frac{1}{N} \sum_{t \in \tau} (\hat{X}_t - X_t)^2, \quad (3)$$

where τ is the set of data to predict of size N , X_t is the value to estimate, and \hat{X}_t the value estimated by the model. The coefficient of determination is defined as:

$$R^2 = \frac{\sum_{t \in \tau} (\hat{X}_t - \bar{X})^2}{\sum_{t \in \tau} (X_t - \bar{X})^2} = 1 - \frac{\sum_{t \in \tau} (\hat{X}_t - X_t)^2}{\sum_{t \in \tau} (X_t - \bar{X})^2} \quad (4)$$

where \bar{X} is the average of X_t over τ . Note that $(\frac{\sum_{t \in \tau} (\hat{X}_t - X_t)^2}{\sum_{t \in \tau} (X_t - \bar{X})^2})$ is the ratio between the mean square error and the variance.

The closer the coefficient of determination to 0, the more the scatterplot is spread around the regression line, which means the less accurate the model is. And inversely, the lower the amount of spread points around the regression line, the more the coefficient of determination is close to 1 (a perfect fit).

Since PFTK and SQRT formulas are used to offer TCP-Friendliness, the ratio between the predicted value and the value to predict is another important measure of the model accuracy. This ratio should be close to one so that a protocol using one of the two formulas to determine its rate can provide TCP-Friendliness. This information is not carried in the MSE or in the coefficient of determination that are only sensible to absolute distances between the predicted and real values. The ratio can be very high whereas the distance between them can be very small (in comparison with the average distance between the predicted values and the values to predict in the set τ).

We thus need another criterion that takes this ratio $(\frac{\hat{X}_t}{X_t})$ into account. The maximum (resp. the minimum) of this ratio gives an idea of how much the prediction can overestimate (resp. underestimate) the true value. However, the average and standard deviation of the ratio should be analysed with caution. Indeed, underestimation affects the average and the standard deviations less than overestimation while both have the same importance for our application. Therefore, in addition to the ratio, we propose to use also the “*absolute ratio*” defined by $\max(\frac{\hat{X}_t}{X_t}, \frac{X_t}{\hat{X}_t})$, and which is sensitive in the same manner to both underestimation and overestimation.

Subsequently, we will use the coefficient of determination as well as the two ratios to evaluate TCP throughput estimators.

3.3 SQRT and PFTK Accuracy

For each session of the validation set (consisting of 7600 TCP sessions), we compute, based on the collected parameters, the throughput predicted by SQRT and PFTK. We then compute for the set of predicted data, the coefficient of determination, and several statistics (average, minimum, maximum and standard deviation) concerning the two ratios (\hat{X}_t/X_t and $\max(\hat{X}_t/X_t, X_t/\hat{X}_t)$).

We plot in Figure 2 the predicted throughput as a function of the real throughput (the one we try to predict) for the two models. Ideally, the scatter plot should fit the regression line ($y = x$), i.e. the predicted value should be equal to the value to predict. Figure 2 shows that both models are far from fitting the regression line. A great amount of points are in fact spread around the latter. Figures 3 and 4 show the ratio and the absolute ratio with respect to the real throughput. In the ideal case, both graphs should be reduced to the straight line $y = 1$. This is again far from being the case. Whatever the performance criterion, both models are inaccurate. Moreover, the range of the ratio in the case of SQRT

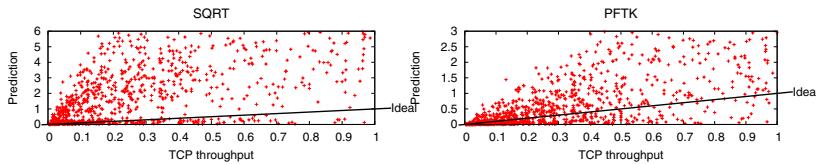


Fig. 2. The predicted throughput versus the real throughput

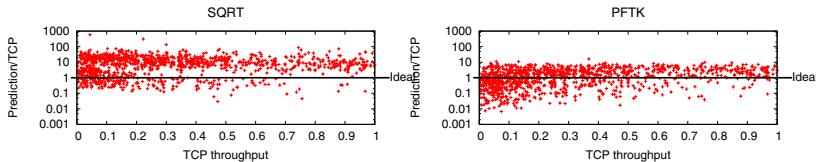


Fig. 3. The ratio between the predicted throughput and real throughput versus the real throughput

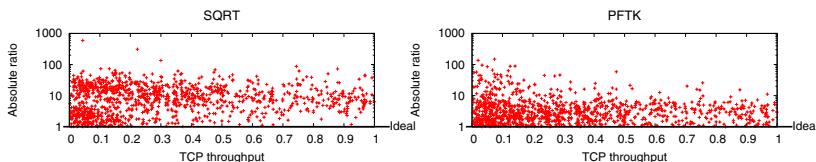


Fig. 4. The absolute ratio between the predicted throughput and real throughput versus the real throughput

Table 1. The coefficient of determination (R^2), the mean square error (MSE) and statistics of the ratio (R) and the absolute ratio (AR) of SQRT and PFTK models.

	R^2	$MSE 10^{-3}$	R				AR			
			avg	stdev	min	max	avg	stdev	min	max
Mathis	0.658	4.078	5.29	10.29	0.013	583.73	5.69	10.59	1	583.73
PFTK	0.814	2.211	2.2	1.19	0.006	16.11	3.15	5.81	1	152.59

is $[0.013, 583.73]$ while the range of PFTK is $[0.006, 16.11]$. That means that a protocol that uses the SQRT model to provide the fairness can get 583.73 times more than what a concurrent TCP would get, and 152.59 ($= 1/0.006$) times less when using PFTK. In both cases, the protocol would not be TCP-Friendly.

Table 1 summarises the above figures numerically. The table shows the coefficient of determination and some statistics concerning the normal ratio and the absolute ratio of the two models. The SQRT model is less accurate than the PFTK one, which has already been shown in [14]. In other words, a protocol that will use the SQRT model will be less TCP-Friendly than one using the PFTK model. However, even if PFTK shows a better behaviour than SQRT, it is still not TCP-Friendly. The absolute ratio should be lower than 1.78 (as suggested in [7]) to provide the fairness towards TCP. Its average is 5.69 for SQRT against 3.15 for PFTK, which are both above 1.78. More precisely, over our validation set (7600 TCP sessions), 70% of PFTK predictions are not TCP-Friendly.

So, in conclusion, neither the SQRT model nor the PFTK model is accurate. This is due *in part*² to the fact that phases like slow-start and fast-recovery are not taken into account and that many hypotheses have been made to make the derivation of an analytical formula feasible[1, 17].

4 Analysis of Bad Predictions

In the previous section, we have shown that SQRT and PFTK are not always accurate and even more, the throughput they predict is often not TCP-Friendly. In this section, we propose to investigate the reasons for the bad predictions of the models. This study aims at characterising the conditions leading the formulas to under or overestimate the throughput. To this end, we propose to use an original approach based on the analysis of the validation test by the decision tree method (which provides “*interpretable*” models). With this method, we will build a classification model to discriminate the good and the bad predictions in function of different parameters gathered from the network. The analysis of the tree so obtained will then provide a characterisation of the conditions under which the models give bad predictions. Before going to the analysis in Section 4.2, we first explain how we classify the predictions into good and bad predictions.

² We will see latter that other causes exists.

4.1 Classification of Predictions

The model prediction will be considered as good if it preserves TCP-Friendliness. In other words, a prediction is considered as “good” if the ratio between itself and the true throughput it approximates belongs to $[1/K, K]$, where $K \geq 1$ is a factor that defines the bounds of fairness towards TCP. Out of this interval, the ratio can belong to $(0, 1/K)$ and in this case the prediction underestimates the throughput to predict, or the ratio belongs to (K, ∞) which is a case of overestimation. These three areas are represented graphically in Figure 5.

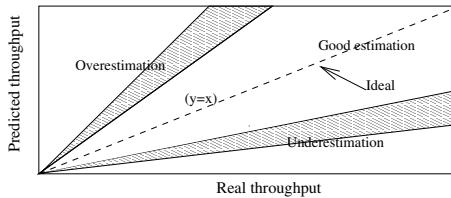


Fig. 5. The three areas defining the quality of the predictions of a model

To define bad predictions, we choose another parameter $K' > K$ and we consider as bad predictions the predictions that are such that the ratio belongs to $(0, 1/K') \cup (K', +\infty)$. This definition of bad predictions thus leaves a region of fuzziness between good and bad predictions which is represented in grey in Figure 5. Subsequently, by abuse of language, the word overestimation (resp. underestimation) will be used to denote the overestimation (resp. underestimation) area of the graph minus the gray area. Thus, the words underestimation and overestimation will be synonyms of bad prediction.

For our study, we use the commonly accepted value of $K = 1.78$ to define TCP-Friendliness and we consider that if the absolute ratio is higher than $K' = 3$, then the prediction is bad. The value of this threshold is purely subjective. However, a study had been done with a threshold equal to 10 and had led to similar results. The choice of the value three is thus not restrictive.

4.2 Decision Trees Analysis

As said in the previous section, there are two kinds of bad predictions: underestimation and overestimation. We have then 4 cases to analyse: PFTK and SQRT in both the underestimation and then overestimation case. Table 2 shows the

Table 2. Distribution of prediction types for the two models

Interval	PFTK	SQRT
$[0.001, 1/3)$ (under)	6.14%	3.43%
$[1/3, 3]$	33.42%	25.66%
$(3, 1000]$ (over)	60.44%	70.91%

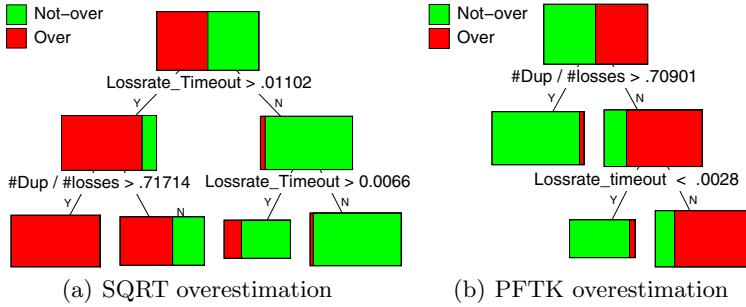


Fig. 6. The top of the decision trees classifying bad predictions. (lossrate_timeout is defined as the number of losses detected by expiration timeout (#timeout) divided by the number of packets transmitted).

distribution of the predictions of PFTK and SQRT into the different classes. The cases of underestimation are too rare for both PFTK and SQRT to obtain statistically meaningful conclusions from their analysis. So, we will drop these cases. It thus remains two situations: SQRT and PFTK in overestimation.

To analyse the reason for overestimation in both cases, we first classify each prediction into one of two classes: “*Over*” to denote the predictions that overestimate the TCP throughput and “*Not-Over*” to denote the other predictions. Then a decision tree is built to explain the classification using as inputs the PFTK parameters, the proportion of losses detected by triple duplicates, and the proportion of losses detected by timeout expirations. The top of each tree is represented in Figure 6 and is discussed below.

The tree of Figure 6(a) shows that if the proportion of losses due to timeout expiration exceeds a certain threshold then SQRT overestimates the throughput. Indeed, at each timeout loss no data is transferred, and this sender inactivity is not taken into account by the model. The model still considers that data are sent and the predicted throughput obtained is higher than TCP’s. This result is already known in the networking community.

The tree of Figure 6(b) is related to the overestimation of PFTK. It points out that if the proportion of losses due to triple duplicates is under a certain threshold, then the estimation exceeds the real throughput. When the number of losses detected by triple duplicates decreases, the number of losses detected by timeout increases. The loss rate p used in $E[A]$ (eq. (16) of [14]) becomes then higher than the value that should be used since it should include only losses due to triple duplicates. Thus, $E[A]$ is lower than what it should be and the predicted throughput, B (inversely proportional to $E[A]$), is then higher than the real one. In addition, when the number of timeouts increases, the number of slow-start phases not taken into account, and over which the throughput predicted is higher than the throughput to estimate, increases. The timeout loss rate affects also the prediction of PFTK.

In conclusion, a discrimination between the way the losses are detected seems to be required for a good prediction. To the best of our knowledge, no models of

TCP, even recent models such as [1] or [15], make this distinction. In the next section, we highlight the importance of incorporating the timeout loss rate in the context of models inferred by machine learning techniques.

5 Supervised Learning of TCP Throughput Models

In this section, we propose to use supervised learning methods to infer models to predict the TCP throughput. We choose these methods because, unlike analytical models, they do not make any assumption about the network and protocol. These methods automatically build a model of an input/output relationship solely from a database of observations of input/output pairs. As incorporating new inputs in these models is straightforward, we begin by inferring models of the TCP throughput using as inputs the same parameters as in PFTK formula³ and in a second step, we introduce the timeout loss rate in addition to these parameters. The comparison of these two classes of models will highlight the importance of distinguishing the two loss types.

The database used to infer models contains 18000 sessions generated by using the procedure described in Section 3.1. These models are then evaluated on the (independent) validation set of 7600 sessions used to evaluate the analytical models. In this paper, we present the results obtained by two machine learning algorithms: Multiple Additive Regression Trees (MART) and Multilayer Perceptrons (MLP). The interested reader can refer to [4] for more details about database generation and machine learning methods.

Table 3 compares the analytical models to the two machine learnt models, with and without the timeout loss rate (TLR). As in previous sections, we compute for each model the coefficient of determination, the mean square error as well as statistics related to the ratio and the absolute ratio.

Table 3. The coefficient of determination (R^2), the mean square error (MSE) and statistics of the ratio (R) and the absolute ratio (AR) of MART and MLP with and without the timeout loss rate (TLR).

		SQRT	PFTK	MART		MLP	
				Without TLR	with TLR	without TLR	with TLR
$MSE10^{-3}$		4.078	2.211	1.246	0.482	1.048	0.322
R^2		0.658	0.814	0.895	0.960	0.912	0.973
R	avg	5.29	2.2	1.23	1.11	1.62	1.12
	min	0.013	0.006	0.07	0.16	0.05	0.2
	max	583.73	16.11	25.91	6.49	18.9	5.2
	stdev	10.29	1.19	0.7	0.49	2.2	0.54
AR	avg	5.69	3.15	1.46	1.22	1.78	1.24
	min	1	1	1	1	1	1
	max	583.73	152.59	25.91	6.49	20	5.2
	stdev	10.59	5.81	0.83	0.55	2.14	0.58

³ These parameters are also used in [15].

We first notice that, on our validation set, supervised learning models are much more accurate than analytical models, whatever the set of parameters used. They reduce the mean square error by more than a factor two and they offer much better fairness towards TCP than the analytical model. Furthermore, the introduction of the timeout loss rate significantly improves their accuracy. The mean square error has been reduced by about a factor three for both methods and the coefficient of determination is much closer to one. In terms of fairness, the average absolute ratio is reduced from 1.46 to 1.22 in the case of MART and from 1.78 to 1.24 in the case of MLP. The reduction of the average is not as important as it is in the case of the maximum absolute ratio. Indeed, the maximum goes from 25.91 and 20 for MART and MLP respectively, down to 6.49 and 5.2. The positive impact of the introduction of the timeout loss rate is clear from this experiment and makes us believe that analytical models will also benefit from the introduction of this parameter.

Our machine-learnt models of TCP have been derived from a large set of random topologies and traffic conditions. The randomness of the learning set may be reduced by focusing more on so-called realistic topologies and traffic conditions, provided that good criteria are found to characterize them. Note however that doing so can only improve our learnt models, because the realistic cases, being a subset, will obviously drive the learning algorithms to more specialized models. A random learning set is thus actually a worst case for our approach. As our models are already quite better than existing analytical models in such a worst case, our approach and results are therefore already very encouraging.

6 Conclusions

In this paper we have studied the accuracy of SQRT and PFTK models. To this end, we have built a database with a high number of TCP sessions gathered in random scenarios and have compared the results predicted by the models with the observed throughputs. Neither SQRT nor PFTK is accurate and we have pointed out the reason of that. PFTK, which is the reference among the analytical models of TCP throughput uses the global loss rate p that accounts indifferently for losses detected by triple duplicates and losses detected by timeout expirations. This non discrimination affects the result: the throughput is overestimated. The application of machine learning algorithms allows us to highlight the importance of the distinction of the two types of losses, which can indeed greatly improve the quality of the models. Our analysis suggests also that future research aiming at the analytical modelling of the throughput of TCP should certainly take into account the timeout loss rate.

We have also proposed an alternative to analytical modelling based on supervised learning, which offers better results than the two tested models even without discriminating the losses. In the future, it would be very interesting to compare these models with more recent TCP models such as those of Altman et al. [1] and of Sikdar et al. [15]. However, the first one uses an infinite sum of terms which is difficult to compute in practice while the second one has been

developed for Reno and not NewReno. On a broader point of view, we would like to further exploit machine learning techniques in networking. These methods do not make any hypothesis and can take implicitly into account all TCP phases and long as well as short sessions, provided that they are represented in the database. The approach can also be easily extended to any version of TCP or any other protocol. Finally, the application of supervised learning techniques is automatic and needs much less time and effort than an analytical modelling, which may be of great importance in the rapidly evolving domain of networking.

Acknowledgements. This work has been partially supported by the Belgian Science Policy in the framework of the IAP programme (MOTION P5/11 project) and by the E-NEXT European Network of Excellence (NoE). P.G. is a Scientific Research Worker at FNRS, Belgium.

References

1. E. Altman, K. Avrachenkov, and C. Barakat. A Stochastic Model of TCP/IP with Stationary Random Losses. *IEEE/ACM Transactions on Networking*, 13(2):356–369, April 2005.
2. F. Baccelli and D. Hong. AIMD, Fairness and fractal scaling of TCP Traffic,. In *IEEE INFOCOM 2002*, volume 21, pages 229 – 238, June 2002.
3. C. Barakat. TCP/IP modeling and validation. *IEEE Network*, 15(3):38–47, 2001.
4. I. El Khayat, P. Geurts, and G. Leduc. Analysis and improvement of analytical models of TCP throughput by machine learning techniques. Technical report, University of Liège, 2005.
5. I. El Khayat, P. Geurts, and G. Leduc. Improving TCP in wireless networks with an adaptive machine-learnt classifier of packet loss causes. In *Proc. of the International Conference on Networking*, pages 549–560. Springer-Verlag, 2005.
6. S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans. Netw.*, 7(4):458–472, 1999.
7. S. Floyd, M. Handley, J. Padhye, and Jorg Widmer. Equation-based congestion control for unicast applications. In *SIGCOMM 2000*, pages 43–56, Stockholm, Sweden, August 2000.
8. C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S.C. Diot. Packet-level traffic measurements from the sprint ip backbone. *Network, IEEE*, 17(6):6– 16, Nov.-Dec. 2003.
9. M. Garetto, R.L. Cigno, M. Meo, and M. A. Marsan. Closed queueing network models of interacting long-lived TCP flows. *IEEE/ACM Trans. Netw.*, 12(2):300–311, 2004.
10. A. Kumar. Comparative performance analysis of versions of TCP in a local network with a lossy link. *IEEE/ACM Trans. Netw.*, 6(4):485–498, 1998.
11. M. Mathis, J. Semke, Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, 27(3):67–82, July 1997.
12. A. Misra and T. J. Ott. The window distribution of idealized TCP congestion avoidance with variable packet loss. In *INFOCOM (3)*, pages 1564–1572, 1999.
13. Vishal Misra, Wei-Bo Gong, and Donald F. Towsley. Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED. In *SIGCOMM*, pages 151–160, 2000.

14. J. Padhye, V. Firoiu, D. Towsley, and J. Krusoe. Modeling TCP Throughput: A Simple Model and its Empirical Validation. *Proceedings of the ACM SIGCOMM '98*, pages 303–314, 1998.
15. B. Sikdar, S. Kalyanaraman, and K. S. Vastola. Analytic models for the latency and steady-state throughput of tcp Tahoe, Reno, and SACK. *IEEE/ACM Trans. Netw.*, 11(6):959–971, 2003.
16. D. Sisalem and A. Wolisz. MLDA: A TCP-friendly congestion control framework for heterogeneous multicast environments. In *Eighth International Workshop on Quality of Service (IWQoS 2000)*, Pittsburgh, June 2000.
17. J. Widmer, R. Denda, and M. Mauve. A Survey on TCP-Friendly Congestion Control. *IEEE Network*, 15(3):28–37, 2001.
18. J. Widmer and M. Handley. Extending equation-based congestion control to multicast applications. In *Proceedings of SIGCOMM'01*, pages 275–285. ACM Press, 2001.