A Machine Learning Approach to Improve Congestion Control over Wireless Computer Networks

Pierre Geurts, Ibtissam El Khayat, Guy Leduc
University of Liège, Sart Tilman, B28
Liège 4000 - Belgium
{geurts, elkhayat, leduc}@montefiore.ulg.ac.be

Abstract

In this paper, we present the application of machine learning techniques to the improvement of the congestion control of TCP in wired/wireless networks. TCP is sub-optimal in hybrid wired/wireless networks because it reacts in the same way to losses due to congestion and losses due to link errors. We thus propose to use machine learning techniques to build automatically a loss classifier from a database obtained by simulations of random network topologies. Several machine learning algorithms are compared for this task and the best method for this application turns out to be decision tree boosting. It outperforms ad hoc classifiers proposed in the networking literature.

1 Introduction

Nowadays computer networks combine a large diversity of different technologies (hybrid networks that combine wired, wireless, and satellite links) and must respond to a large variety of tasks with very different requirements in terms of quality of service (e.g. e-mail, videoconference, streaming). Furthermore, the behavior of these networks depends on a large number of external factors (user behaviours, load of the networks, link capacities...) that are difficult to appraise. Hence, data mining and machine learning are certainly tools of choice to improve our understanding of networks and to help in the design of control solutions.

In this paper, we propose a specific application of machine learning techniques to the improvement of TCP over hybrid wired/wireless networks. We first discuss the drawback of TCP congestion control algorithm and how we propose to solve this problem with machine learning techniques. Section 4 then describes the database that has been generated for this study and Section 5 the different machine learning methods that have been compared. Experiments with the new protocol are described in Section 6. Further details about this study may be found in [9] and [5].

2 TCP over wired/wireless networks

TCP, for "Transmission Control Protocol", is the most widely used protocol in the Internet. Its success lies in its reliable file transfer and its capacity in avoiding congestion. A congestion arises in the network when routers are not able to process the data that arrive to them and this causes buffer overflows at these routers that result in the loss of some packets of data. The only solution to relax a congestion is to reduce the load of the network. TCP congestion protocol is based on the definition of a congestion window that controls the rate at which packets are sent by a sender to a receiver and a mechanism to adapt the value of this window size according to the state of the network. The idea is to increase the rate steadily (i.e. additively) when everything works fine and to reduce it more abruptly (i.e. multiplicatively) as soon as a loss occurs. Further details about congestion control mechanisms in TCP may be found in RFC-2581.

This congestion control mechanism thus makes the hypothesis that packet losses are due to buffer overflows and it works fairly well in the case of wired networks where this is indeed the case. However, in wireless links, losses caused by error in the link (for example by signal fading) are not so unlikely. TCP has no mechanism to distinguish loss caused by a link error from losses caused by a congestion and it reduces systematically its rate. This reduction is not justified when there is no congestion and the consequence is that the throughput of TCP over wireless link is lower than what it could be. Several studies have highlighted the bad behaviour of TCP over wireless link (e.g. [14]).

3 Proposed machine learning solution

A straightforward solution to increase the throughput of TCP over wireless links is to prevent it from reducing its rate when it faces a loss due to a link error. The approach

Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)
0-7695-2142-8/04 $ 20.00 IEEE

adopted in this paper is to endow one of the end systems (i.e the sender or the receiver) with an algorithm able to determine the cause of a packet loss only from information available at this end system. With such a classifier, the modified protocol will divide its congestion window only if the loss is classified as a congestion loss. Otherwise, it will maintain its congestion window constant.

Several rough classification rules for losses have been proposed in the literature based on heuristics or analytical derivations (e.g. in Veno [8] or Westwood [13]). In this paper, we propose to use supervised machine learning techniques to automatically derive a classification model for loss causes. This model will use as inputs statistics computed on packets received by the sender or by the receiver and it will be induced from a database obtained by simulations of random network topologies.

For this application to be practically feasible and useful, two specific constraints should be taken into account when choosing a classifier. First, because classifications are done in real-time, the computing times to make a prediction and the computer resource needed to store the model are not to be neglected. Second, since TCP is so popular in nowadays networks, any new protocol should be TCP-Friendly ([11]). A TCP-Friendly protocol is a protocol that, when sharing the network with TCP, allows this latter to have a throughput similar to the one it would get if it were in competition with another TCP in similar conditions.

In [5], we show that it is possible to derive analytically an upper bound on the misclassification error on congestion losses that ensures TCP-friendliness. This bound is of the form:

$$Err_C \leq f(RTT, p), \tag{1}$$

where Err_C is the probability of misclassifying a congestion loss as a link error loss, RTT is the round-trip-time[1], and p is the actual loss rate, i.e. the percentage of lost packets whatever the cause. Since this bound may change with time, our solution to ensure TCP-friendliness will be to dynamically adapt the classifier such that it always satisfies the constraint given the current RTT and p values.

4 Database generation

To solve our problem of loss classification, each observation $< x_i, y_i >$ of our learning sample will be an input/output pair where the inputs x_i are some variables that describe the state of the network at the occurrence of a loss and the (discrete) output y_i is either C to denote a loss due to a congestion or LE to denote a loss due to a link error.

The database[2] was generated by simulations with a network simulator called `ns-2`[12]. To generate our observations of losses, we have used the following procedure: 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, all losses that have occurred within this time interval are collected in the database. In our study, we have collected 35,441 losses (among which 22,426 are due to congestion) that correspond to more than one thousand different random network topologies.

At the end system (sender or receiver), the only information we can measure to predict a congestion is some statistics on the packet departure and arrival times. So, to define our inputs, we have computed different statistics relating the one-way delay and the inter-packet time of several packets surrounding the loss. All in all, this results in a set of 40 numerical input variables.

5 Machine learning techniques

In this study, we propose to compare several machine learning algorithms which are:

Decision trees. The main advantage of this method for our application is that pruned trees are usually quite small and the computation of a prediction is very fast. To build a decision tree, we have adopted the standard CART algorithm described in [4].

Decision tree ensembles. Ensemble methods are generic techniques that improve a learning algorithm by learning several models and then by aggregating their predictions. In our experiment, we have compared four ensemble methods that have been proposed for decision trees, namely Bagging [2], Random forests [3], Boosting [7], and Extra-trees [10]. All these methods have been used with their default parameter setting.

Artificial Neural Networks. This method usually gives more accurate models than decision trees but it is also much more demanding in terms of computing times and computer resources. In our experiments, we have used multilayer perceptrons with Levenberg-Marquardt optimization [1].

k-Nearest Neighbors. An important drawback of this method for this application is that the computation of a prediction is quite demanding and furthermore it requires to store the entire learning sample.

As discussed above, one important characteric of the chosen classifier is that it should be possible to adjust its error on congestion losses dynamically to ensure TCP-friendliness. Actually, all these methods not only provide a class prediction for each value of the inputs x but also provide an estimate of the conditional probability of each class, C or LE, given the inputs x. In the two class case, the default use of the model is to classify a loss as a congestion loss if the probability estimate $\hat{P}(C|x)$ is greater than 0.5.

[1] the time between the sending of a packet and the reception of its acknowledgment by the sender

[2] The database is available electronically at http://www.montefiore.ulg.ac.be/~geurts/publications/BD-Fifo.dat.gz.

Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)
0-7695-2142-8/04 $ 20.00 IEEE

However, by using a user defined threshold P_{th} different from 0.5, we can change the misclassification probability of each class and hence adjust our classifier to satisfy (1).

A natural way to evaluate our models independently of the value of P_{th} is to use receiver operating characterisic (ROC) curves (see [6] for an introduction). A ROC curve plots for every possible value of the threshold P_{th} the true positive rate versus the false positive rate of a given class (among two classes). In our case, the true positive rate is taken as $1 - Err_C$ and the false positive rate is the probability of misclassifying a link error loss, that will be denoted Err_{LE}. While ROC curves are two-dimensional, a common one-dimensional summary of a ROC curve to compare classifiers is the area under the ROC curve (AUC) which we will also use to rank our packet loss classifiers. ROC curves and AUC are computed according to the algorithms presented in [6].

6 Experiments

First, we evaluate classifiers and then we evaluate the new protocol with the best classifier only.

6.1 Comparison of loss classifiers

To make a reliable estimate of the error of each model, we have randomly divided the whole database into two parts: a learning set of 25,000 cases and a validation set containing the remaining 10,441 cases. A classification model is built with every method on the learning sample and its ROC curve, AUC, and error rate are evaluated on the validation sample. The methods are compared from these criteria in Table 1 and in Figure 1. We also give in Table 1 the time needed to classify the validation set with each method. For ensemble methods, we build $T = 25$ trees. As the ROC curves of all ensemble methods are very close, we only give the results obtained by boosting in Figure 1. The value of k for the k nearest neighbors was determined by leave-one-out cross-validation (the optimum is $k = 7$). For the MLP, we tried several structures of one and two layers with a number of neurons going from 10 to 50 neurons in each layer. Table 1 only shows the best result that was obtained with two layers of 30 neurons.

The results are quite good considering the diversity of the network topologies represented in the database. The decision tree has the lowest AUC but it is by far the fastest method to make a prediction. The k-NN has a better AUC than decision tree but its ROC curve is worst for small values of Err_C, which is the region of interest of our application. It is also the slowest method in terms of computing times. MLP improves upon both methods in terms of accuracy but it remains below ensemble methods in terms of accuracy and computing times. All ensemble methods are very close but boosting is superior along the two criteria.

Table 1. Comparison of different ML methods

Method	AUC	Error (%)	Time (msec)
DT	0.9424	8.92	150
Bagging	0.9796	6.65	650
Random forests	0.9823	6.48	600
Extra-trees	0.9813	6.91	940
Boosting	0.9840	6.34	570
MLP	0.9761	7.67	1680
k-NN	0.9541	10.16	316,870
Veno	0.7260	34.52	-
Westwood	0.6627	41.54	-

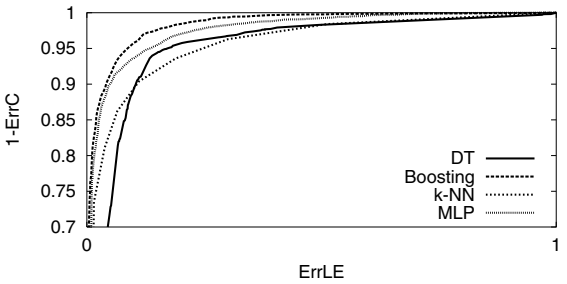

Figure 1. Comparison of the ROC curves of different ML methods

For comparison, we put also in Table 1 the result obtained on the validation set by two ad-hoc classification rules that have been proposed in the networking literature in the context of two extensions of TCP for wireless networks called Veno [8] and Westwood [13]. The results obtained by these rules are far below the results obtained by machine learning algorithms. This clearly shows the interest of a machine learning approach in the context of this application, both to improve existing classifiers but also to assess their validity.

6.2 Simulations of the modified protocol

We propose to use the boosting classifier in the following way to improve TCP. Each time a loss occurs, the minimal value of P_{th} such that (1) is satisfied is determined from the current (estimated) values of RTT and p. Then, the ensemble of trees together with this value of P_{th} are used to classify the loss. If the loss is predicted as caused by a congestion, TCP works as usual. Otherwise, it maintains its congestion window constant.

The main criteria to evaluate this protocol are bandwidth usage in the case of wireless links and TCP-friendliness in the case of wired networks. In this paper, we only describe two simple experiments showing the interest of the machine

Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)
0-7695-2142-8/04 $ 20.00 IEEE

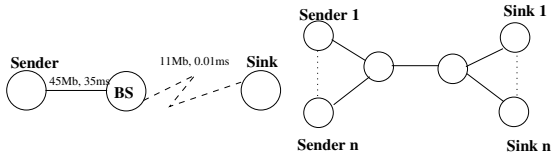

Figure 2. Two topologies used in simulations

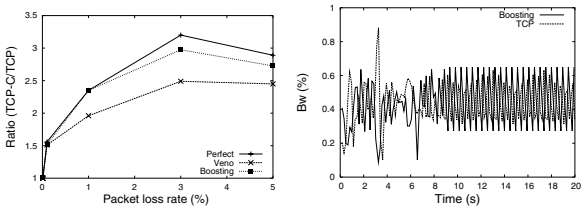

Figure 3. Top, the gain in wireless link, bottom, TCP-Friendliness

learning solution along these criteria. A more detailed validation of the protocol may be found in [5].

Improvement on wireless links. To illustrate the gain of our protocol over wireless link, we simulated the simple hybrid network illustrated in the left part of Figure 2 with `ns-2`. The first link is wired and the second one that constitutes the bottleneck is wireless. We compare the ratio between the throughput obtained by the classifier and the one obtained by a standard TCP when we vary the packet loss rate from 0 to 5% over the wireless link. To have some points of comparison, we run also simulations with the classification rule proposed in Veno and, as our system is simulated, with a hypothetic perfect classification rule. The left graph of Figure 3 illustrates the ratio obtained by TCP enhanced with the three classifiers. The superiority of boosting over Veno and standard TCP is clear. It is also very close to the perfect model. Its gain with respect to TCP reaches about 300% when the loss rate is equal to 3%.

TCP-friendliness. To test the behaviour of our protocol when it competes with another TCP, we use the classical topology in the right part of Figure 2 with $n = 2$. The experiment consists in running a standard TCP in competition with our TCP plus the boosting classifier. The right part of Figure 3 shows the evolution of the throughput obtained by each traffic. This figure shows that the link is fairly shared between the two protocols. The same experiment was run with Veno's classification rules and in this case, the bandwidth used by Veno was five times higher than that of TCP.

7 Conclusions

In this paper, we have presented the application of machine learning techniques to the improvement of congestion control protocol in wireless networks. The application of the boosting classifier shows a significant improvement of bandwidth usage in wireless networks and no deterioration in traditional wired networks. The resulting classifier also compares very favorably to existing ad-hoc classification rules that have been proposed in the networking literature. This study thus shows the interest of the application of machine learning techniques to help designing new protocol in computer networks.

Acknowledgment

This work has been partially supported by the Belgian Science Policy in the framework of the IAP program (Motion P5/11 project). P. Geurts is a post-doctoral researcher at the FNRS, Belgium.

References

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[3] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[4] L. Breiman, J. Friedman, R. Olsen, and C. Stone. *Classification and Regression Trees*. Wadsworth International (California), 1984.

[5] I. El Khayat, P. Geurts, and G. Leduc. Classification of packet loss causes in wired/wireless networks by decision tree boosting. Technical report, University of Liège, 2004.

[6] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. Technical report, HP Labs, 2004.

[7] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the second European Conference on Computational Learning Theory*, pages 23–27, 1995.

[8] C. P. Fu and S. C. Liew. TCP veno: TCP enhancement for transmission over wireless access networks. *IEEE (JSAC) Journal of Selected Areas in Communications*, Feb. 2003.

[9] P. Geurts, I. El Khayat, and G. Leduc. A machine learning approach to improve congestion control over wireless computer networks. Technical report, University of Liège, 2004.

[10] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. Submitted, 2004.

[11] M. Mathis, J. Semke, Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *ACM Computer Communication Review*, 3:67–82, July 1997.

[12] S. McCanne and S. Floyd. *The LBNL Network Simulator.* Lawrence Berkeley Laboratory, 1997.

[13] R. Wang, M. Valla, M. Sanadidi, B. Ng, and M. Gerla. Efficiency/friendliness tradeoffs in TCP westwood. In *In Proceedings of the 7th IEEE Symposium on Computers and Communications*, 2002.

[14] G. Xylomenos, G. Polyzos, P. Mahonen, and M. Saaranen. TCP performance issues over wireless links. *Communications Magazine, IEEE*, 39(4):52–58, 2001.

Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)
0-7695-2142-8/04 $ 20.00 IEEE