IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Congestion Management In Wireless Networks: A Machine Learning Approach

[1]Mr P.RaviprakashAsst. Professor, Malla Reddy Engineering College for Women, Maisammaguda, Hyderabad,500100

[2]Dr. D.B.K. Kamesh, Professor, Malla Reddy Engineering College for Women, Maisammaguda, Hyderabad,500100

Abstract

In this research, we show how machine learning may be used to make wired and wireless TCP networks more resilient to congestion. Since TCP reacts the same way to losses caused by congestion as it does to losses caused by link faults, it is sub- optimal in hybrid wired/wireless networks. So, we propose simulating various network topologies and then using machine learning to create a loss classifier automatically. Several machine learning methods are compared, and decision tree boosting is found to be the most effective. In comparison to ad hoc classifiers from the networking literature, it performs better.

1 Introduction

Modern computer networks use a wide range of technologies (hybrid networks that combine wired, wireless, and satellite links) and must accommodate numerous tasks that have varying quality-of-service requirements (email, videoconferencing, streaming, etc.). The complex interplay of many uncontrollable variables (user actions, networkload, connection capacity, etc.) makes it hard to predict how these networks will perform. Therefore, data mining and machine learning are unquestionably the go-to methods for enhancing our knowledge of networks and assisting with the development of appropriate control strategies.

In this study, we suggest a method for improving TCP performance over hybrid wired/wireless networks by employing machine learning techniques. The limitations of the current TCP congestion control method are first covered, followed by a discussion of how we intend to use machine learning to address these issues. Next, we'll go over the data set built for this study in Section 4, followed by a discussion of the various machine learning approaches in Section 5. Section 6 details experiments performed using the revised protocol. Further

Refer to [9] and [5] for more information about the research project.

2. Connected and wireless TCP systems

The "Transmission Control Protocol" (or "TCP") is the most popular protocol used on the Internet. Success can be attributed to its consistent file transfer and ability to avoid congestion. Congestion occurs in a network when routers cannot handle the incoming data, leading to buffer overflows and the subsequent loss of data packets. Congestion can only be eased by reducing the number of users on the network. The Transmission Control Protocol (TCP) congestion protocol is predicated on the definition of a congestion window that regulates the pace at which packets are transferred from a sender to a receiver, as well as a mechanism for

dynamically adjusting the size of this window to account for changes in network conditions. The plan is to gradually raise the rate (by adding to it) when things are going well, and to drop it (by multiplying it) when anything goes wrong. TCP's congestion control techniques are described in greater depth in RFC-2581.
Because of this, the congestion management mechanism hypothesises that packet losses result from buffer overflows, and it does a decent job of preventing these losses in wired networks. Errors in the link, such as signal fading, are a more likely source of data loss in wireless connections. When experiencing congestion, TCP slows down consistently and cannot tell the difference between link errors and losses. When there is no congestion, this reduction is unwarranted, resulting in a lower than possible TCP throughput over a wireless channel. A number of reports, for instance [14], have pointed out the problematic nature of TCP over a wireless link.

3. The proposed machine learning approach

One easy way to improve TCP's throughput over wireless links is to stop it from slowing down in response to link errors. Consideration ofone of the end systems (either the sender or the receiver) with an algorithm that can determine the source of a packet loss using just information available at that end system. If the loss is identified as a congestion loss, the modified procedure will merely split its congestion window. If that's not the case, the congestion window will stay the same.Several heuristics and analytical derivations (such as in Veno [8] or Westwood [13]) have been offered as crude classification rules for losses. To

automatically construct a classification model for loss causes, we propose in this research to apply supervised machine learning techniques. Using a database derived from simulations of random network topologies, this model will take as inputs statistics generated on packets received by the sender or by the receiver.
Two specific restrictions should be considered while selecting a classifier to ensure the application's practical feasibility and usefulness. First, it is important to keep in mind that the processing time to create a prediction and the computer resource needed to save the model are not to be ignored when doing classifications in real time. Second, any new protocol needs to be TCP- Friendly ([11]) due to TCP's widespread use in modern networks. Sharing the network with TCP, a TCP-friendly protocol allows TCP to achieve throughput comparable to that achieved while competing with another TCP under the same conditions.In [5], we demonstrate that an upper bound on the misclassification error on congestion losses that guarantees TCP friendliness can be derived analytically. The formula for this limit is as follows (1) where ErrC is the likelihood of incorrectly labelling a congestion loss as a link error loss, RTT is the round-trip-time1, and p is the actual loss rate, i.e. the proportion of dropped packets regardless of the reason. Given that this bound is dynamic and depends on the current RTT and p values, we will dynamically adjust the classifier to guarantee TCP compatibility.Each observation xi, yi > of our learning sample will be an input/output pair, with xi representing some variables characterising the state of the network at the time of the loss and yi representing either C to denote a loss due to congestion or LE to denote a loss due to a link error.In order to create the database2, we ran simulations using a net-1the interval between the transmission of a packet and its acknowledgement by the recipient.You may get the database online at http://www.montefiore.ulg.ac.be/eurts/publications/ BD-Fifo.dat.gz.[12] A ns-2 work simulationplatform. Our loss observations have been generated using the following method: After a random generation of the network architecture, the network is simulated for some period of time with equally random generation of traffic. All losses that have occurred throughout this time period are recorded in the database at the conclusion of the simulation. We have collected 35,441 losses (22,426 of which are congestion-related) across over a thousand randomly generated network topologies for our investigation.
Only by collecting statistics on packet departure and arrival times can we anticipate a congestion at the end system (sender or receiver). To define our inputs, we have calculated a variety of statistics connecting the one-way delay to the inter-packet time of several packets in the vicinity of the loss. This yields a total of forty numerical input variables.

This report defines the different remote security dangers to wireless system and conventions at present accessible like wired equivalent privacy (WEP), Wi-Fi protected access(WPA) and Wi-Fi protected access2 (WPA2) . WPA2 is more security convention as compared to Wi-Fi protected access (WPA) it utilizes the Advanced Encryption standard (AES) encryption [14].

4. Machine learning techniques

We suggest a comparison of the following machinelearning methods in this investigation:
A forest of choices. The primary benefit of thisapproach is the speed with which we can compute a prediction, as pruned trees are typically relativelytiny. We've relied on the tried-and-true CARTalgorithm from [4] to construct our decision trees. Groups of decision trees. The goal of ensemblemethods is to enhance a learning process by firstlearning multiple models and then combining theirpredictions. We conducted an experiment in whichfour different ensemble approaches for decisiontrees—Bagging [2], Random forests [3], Boosting[7], and Extra-trees [10]—were pitted against oneanother. These procedures have all beenutilizedutilizing their standard parameter settings.
The use of simulated brains or ANNs. While this strategy typically yields more accurate models than decision trees, it is also significantly more time- and resource-intensive to implement. We have employed multilayer perceptron'soptimized with Levenberg-Marquardt [1] in our tests.
The k-Nearest Neighbors. The major problem of this approach is that it requires storing the whole learning sample and computing a prediction is quite computationally intensive.
As was mentioned previously, one crucial feature of the selected classifier is the ability to dynamically tune its error on congestion losses to guarantee TCP-friendliness. Despite their names, all of these techniques do more than just make a classification guess for each x value; they also provide an estimate of the conditional probability of each class (C or LE) given those inputs. Congestion losses are identified in the two-class scenario when the probability estimate P(C|x) is greater than 0.5.

The misclassification probability of each class can be modified, and our classifier retuned, by setting the threshold Pth to a value other than 0.5.
Using receiver operating characteristic (ROC) curves (for an introduction, see [6]) provides a logical technique to evaluate our models regardless of the value of Pth. For each setting of the threshold Pth, the true positive rate and false positive rate of a given class (between two classes) are plotted on a receiver operating characteristic (ROC) curve. We shall refer to the probability of incorrectly identifying a link error loss as the false positive rate, represented by ErrLE, and the genuine positive rate as 1 ErrC. Although ROC curves are two- dimensional, the area under the ROC curve (AUC) is a common one-dimensional summary of a ROC curve that is commonly used to compare classifiers; we will use this measure to rank our packet loss classifiers as well. The algorithms for calculating ROC curves and AUC are described in [6].
precision and processing speed. In terms of accuracy and efficiency, boosting is the best ensemble method. The results of two ad hoc categorization rules proposed in the networking literature in the context of two extensions of TCP for wireless networks, Veno [8] and Westwood [13], are also included in Table 1 for comparison. These rules provide outcomes that are drastically inferior to those produced by machine learning algorithms. This demonstrates the need for and interest in a machine learning approach to this application, both for the purpose of enhancing current classifiers and for the purpose of evaluating their accuracy. **The improved protocol has been simulated.**

5. Experiments

We have split the database into two parts, a learning set of 25,000 cases and a validation set of the remaining 10,441 examples, so that we can accurately quantify the error of each model. With each approach, a classification model is constructed using the training sample, and then its ROC curve, AUC, and error rate are assessed using the validation sample. Table 1 and Figure 1 show comparisons of the strategies using these criteria. In Table 1, we also detail how long it takes for each technique to correctly categorise the validation data. To implement ensemble methods, we construct T = 25 trees. All ensemble approaches have similar ROC curves, hence we just show the boosting findings in Figure 1. Using a leave-one- out cross-validation procedure, we found that the optimal value for k in the k nearest neighbors is 7. Several structures with ten to fifty neurons per layer were tested with various numbers of layers for the MLP. Results with two layers of 30 neurons are shown exclusively in Table 1.

The outcomes are respectable, considering the variety of network topologies included in the data set. Even though its AUC is poor, the decision tree is the quickest way to generate a forecast. Though its AUC is higher than that of the decision tree, the k-NN's ROC curve performs poorly for low values of ErrC, which is of particular interest in this application. The longest processing periods are also associated with this technique. In terms of accuracy, MLP is superior to both approaches, however it still lags ensemble methods.

Table1.ComparisonofdifferentMLmethods

Method	AUC	Error(%)	Time(msec)
DT	0.9424	8.92	150
Bagging	0.9796	6.65	650
Randomforests	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	-

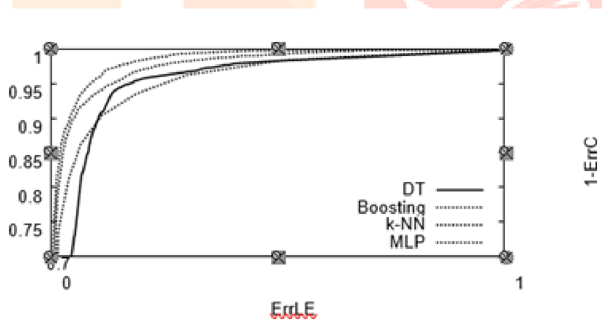

Figure1.ComparisonoftheROCcurvesofdiff erentMLmethods

Conclusions

In this work, we demonstrate how machine learning can be used to alleviate congestion issues. attheFNRS,Belgium. To enhance TCP, we propose the following application of the boosting classifier. After each loss, the current (estimated) values of RTT and p are used to calculate the smallest value of Pth that satisfies (1). The loss is then classified using an ensemble of trees and this Pth value. When congestion is suspected to be the root cause of a loss, TCP continues to function normally. Otherwise, the congestion window remains unchanged.

References

[1] C.Bishop.*NeuralNetworksforPatternRecognition*.Ox- ford:Oxford UniversityPress,1995.

[2] L.Breiman. Baggingpredictors. *MachineLearning*, 24(2):123–140,1996.

[3] L.Breiman.Randomforests.*Machinelearning*,45:5–32,2001.

[4] L. Breiman,J.Friedman,R.Olsen,andC.Stone.*Classifi-cationandRegressionTrees*.WadsworthInternational(Cal -ifornia),1984.

[5] I.ElKhayat,P.Geurts,andG.Leduc.Classificationof packetlosscausesinwired/wirelessnetworksbydecisiontree boosting.Technicalreport,UniversityofLiège,2004.

[6] T.Fawcett.Rocgraphs:Notesandpracticalconsiderations forresearchers.Technicalreport,HPLabs,2004.

[7] Y. Freund and R. E. Schapire.A decision-theoretic general-izationofon-linelearningandanapplicationtoboosting.In*Proceedings ofthesecondEuropeanConferenceonCompu-tationalLearningTheory*,pages23–27,1995.

[8] C.P.Fu andS.C.Liew.TCPveno:TCPenhancementfor transmission over wireless access networks.*IEEE (JSAC)JournalofSelectedAreasinCommunications*,Feb. 2003.

[9] P.Geurts, I.ElKhayat,andG.Leduc.A machinelearning approachtoimprovecongestioncontroloverwirelesscom- puternetworks.Technicalreport,UniversityofLiège,2004.

[10] P.Geurts,D.Ernst,andL.Wehenkel.Extremelyrandomized trees.Submitted,2004.

[11] M. Mathis, J. Semke, Mahdavi, and T. Ott. The macroscopicbehavioroftheTCPcongestionavoidancealg orithm.*ACMComputerCommunicationReview*,3:67– 82,July1997.

[12] S.McCanneandS.Floyd.*TheLBNLNetworkSimulator*. LawrenceBerkeleyLaboratory,1997.

[13] R. Wang, M. Valla, M. Sanadidi, B. Ng, and M. Gerla.Efficiency/friendliness tradeoffs in TCP westwood.In *InProceedingsofthe7thIEEESymposiumonComputersan dCommunications*, 2002.

[14] G.Xylomenos,G.Polyzos,P.Mahonen,andM.Saaranen. TCP performance issues over wireless links.*Communica-tionsMagazine*,*IEEE*,39(4):52–58,2001.

[15] P Nalajala, RV Krishnaiah, B Annapurna, B Godavarthi, security for wireless local area network with pre share key authentication using wi-fi protected access," International Journal of Civil Engineering and Technology (IJCIET), Vol 8, Issue 8, Pp: 841-851, 2017

4