Network Bandwidth Predictive analysis using Stacking

Bhushan Gehi
Student of IInd Year M.E Computer Science,
K.J. Somaiya College of Engineering,
Vidyavihar, Mumbai.
bhushan.gehi@gmail.com

Prof. Irfan Siddavatam
Department of Information Technology,
K.J. Somaiya College of Engineering,
Vidyavihar, Mumbai
irfansiddavatam@engg.somaiya.edu

Abstract – Study presented in this paper is based on analysis of network bandwidth usage and using it as baseline to predict and improvise future network bandwidth usage. Network bandwidth plays an important role in evaluating any application's performance running on a computer. Although it is not an application dependent parameter, however if affects overall performance of all the applications. Network bandwidth is finite and is limited by the laws of physics as well as technology limitations. Also, it is not free. For any organization, the cost of network bandwidth adds as a major and continual expense in overall infrastructure costs. Thus it becomes very important to continually monitor, analyze and improvise network bandwidth usage pattern. Historic data of network bandwidth usage is captured and used in this study. This historic data after some transformation defines the baseline, which is then absorbed by predictive analytics module. In this study we combine business intelligence along with bandwidth analytics. Business intelligence technique called as Stacking of predictive models is used to optimize the network usage patterns and thus improvise the overall usage of network bandwidth. An exhaustive research is also done while choosing the best combination of algorithms for stacking.

Keywords *- Stacking;Holtwinters; Neural network;*

*****

I. Introduction

This study uses Pcap(Packet Capture) dump of a network. This dump could be of any organization, institute, or any set of system whose usage pattern is to be analyzed. This dump forms the raw data which we will use to understand the historic network bandwidth usage pattern of the network.

This Pcap dump can be captured using any utilities like WireShark or any other custom packet capture utility. This dump should be ideally collected at least for a complete cycle for which analysis should be done. For example, if an organization has a weekly cycle defining the complete activities done on their network, they should take atleast a week's dump for this analysis. Taking dump of a complete cycle would ideally include all the parameters which might affect the study, and thus would give a fair analysis of network bandwidth pattern. However, for training the system, we would collect and use multiples of a data usage cycles. Number of cycles should not be restricted and should be captured as much as possible. This would result in improving the accuracy of prediction model.

Problem Statement: This study is focused on analyzing the current bandwidth utilization pattern and optimizes the overall usage pattern by predicting the bandwidth utilization for future events. A typical bandwidth usage pattern of a network would be as shown below in fig.1.

As can be seen in the graph, we would mostly have certain hours of the network wherein the bandwidth usage is high. On the contrary we might also have certain hours in which it is not used at all or least used.

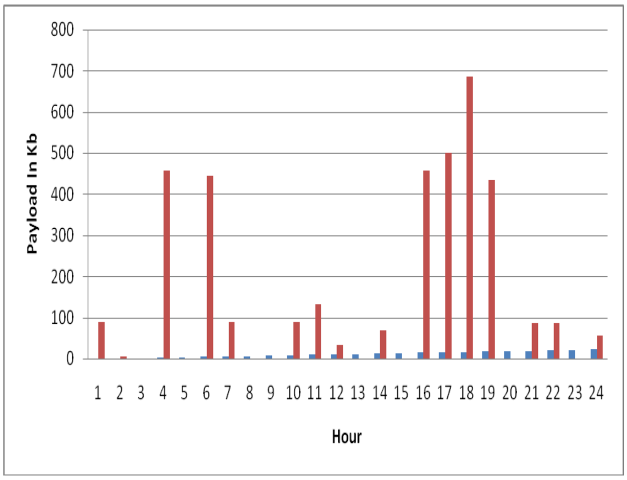

Fig. 1 Typical usage pattern of a network

In above usage pattern there are certain payloads which are periodic or have a certain cycle, like database refresh cycles, software update cycles, certain files being copied from remote servers, etc. However we also have random events of day to day work cycle which cause payloads to occur. So the problem statement of this case study is to predict the usage pattern of certain nodes of network so that we could optimize the overall usage of bandwidth. Output of this study can be certainly used to find the least used nodes or time duration in which bandwidth is least used. Such time events can then be used to schedule the periodic payload cycles; thus optimizing overall bandwidth usage pattern.

Objective

In this case study, there are two types of payload events that can occur. There are certain payloads which are periodic or have a certain cycle, like database refresh cycles, software update cycles, certain files being copied from remote servers, etc. However we also have random events of day to day work cycle which cause payloads to occur. Thus a single prediction algorithm like Linear regression or only Holt winters algorithm would not be able to predict future values accurately. Hence stacking of prediction models is done so as to consider both kinds of events. Algorihm like Neural network would be used along with Holt Winters algorithm. Holt Winters algorithm would help to find the periodic payload cycles and Neural network algorihm works accurately for random events. Thus combination of these two algorithms would result in better prediction in this case study.

Thus, this study is aimed to satisfy below objectives:

- ✓ Cater to the need of effective network bandwidth utilization
- ✓ Overcome the limitation of using only random event prediction models like Linear regression, Neural network, etc.
- ✓ Overcome the limitation of using only Holt winters prediction model
- ✓ Implementing the stacked prediction model combining prediction models across various classes of prediction models
- ✓ Generating graphical output for forecasted usage pattern

Motivation

Being a part of IT industry for few years and actually worked in an environment consisting of hundreds of workstations and servers in a network, has helped us understand the importance of network bandwidth. Although being in a high speed internet infrastructure, there would always be some applications or servers which are still struggling for bandwidth that they actually require to give the best throughput. On the contrary, the same network remains least used at some or the other instance during a work cycle. Thus this study focuses on understanding bandwidth usage and is motivated to come out with a pattern which a particular network follows. This pattern would than help to improvise and make the best use of the available network bandwidth.

II. Initial Analysis and Study of Prediction Models

In this study we are analyzing IP packets. A network packet is a formatted unit of data carried by a packet-switched network. When data is formatted into packets, the bandwidth of the communication medium can be better shared among users than if the network were circuit switched. A packet consists of two kinds of data: control information and user data (also known as payload). The control information provides data the network needs to deliver the user data, for example: source and destination network addresses, error detection codes, and sequencing information. Typically, control information is found in packet headers and trailers, with payload data in between. Each packet of dump consists of packets of various layers of OSI model. Packets of various layers are enclosed within one another. An IP-packet resides within an Ethernet-packet. A TCP-packet resides within an IP-packet. A HTTP-packet resides within a TCP-packet. Such a structure is represented in below diagram.

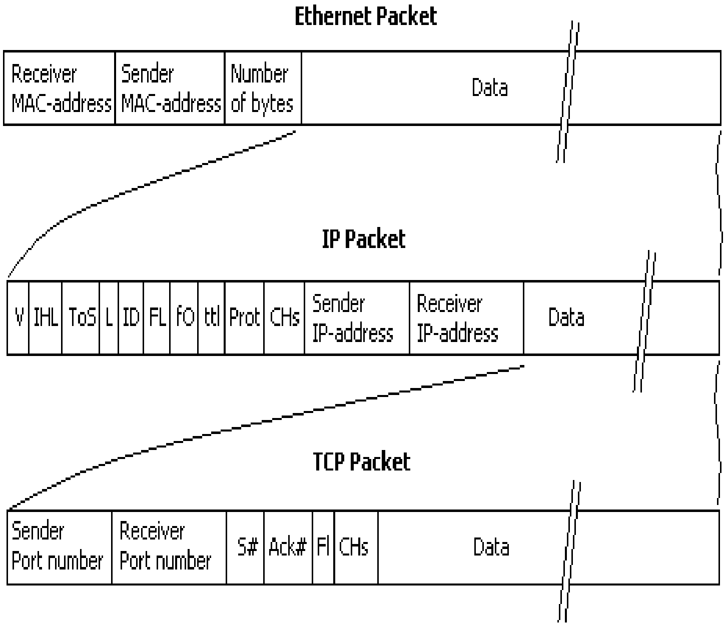

Fig. 2 IP packet structure

Prediction Models

Predictive modeling is the process by which a model is created or chosen to try to best predict the probability of an outcome. Predictive modeling encompasses a variety of techniques from statistics, modeling, machine learning, and data mining that analyze current and historical facts to make predictions about future, or otherwise unknown, events. Predictive models analyze past performance to assess how likely a customer is to exhibit a specific behavior in order to improve marketing effectiveness. This category also encompasses models that seek out subtle data patterns to answer questions about customer performance, such as fraud detection models.

Some of the widely used prediction models are:

Linear regression model: The linear regression model analyzes the relationship between the response or dependent variable and a set of independent or predictor variables. This relationship is expressed as an equation that predicts the response variable as a linear function of the parameters. These parameters are adjusted so that a measure of fit is optimized. Much of the effort in model fitting is focused on minimizing the size of the residual, as well as ensuring that it is randomly distributed with respect to the model predictions.

Time series models: Time series models are used for predicting or forecasting the future behavior of variables. These models account for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for. As a result standard regression techniques cannot be applied to time series data and methodology has been developed to decompose the trend, seasonal and cyclical component of the series. Modeling the dynamic path of a variable can improve forecasts since the predictable component of the series can be projected into the future.

Neural Network: Neural networks are nonlinear sophisticated modeling techniques that are able to model complex functions. They can be applied to problems of prediction, classification or control in a wide spectrum of fields such as finance, cognitive psychology/neuroscience, medicine, engineering, and physics. Neural networks are used when the exact nature of the relationship between inputs and output is not known. A key feature of neural networks is that they learn the relationship between inputs and output through training. There are three types of training in neural networks used by different networks, supervised and unsupervised training, reinforcement learning, with supervised being the most common one.

K-nearest neighbours: The nearest neighbour algorithm (KNN) belongs to the class of pattern recognition statistical methods. The method does not impose a priori any assumptions about the distribution from which the modeling sample is drawn. It involves a training set with both positive and negative values. A new sample is classified by calculating the distance to the nearest neighbouring training case. The sign of that point will determine the classification of the sample. In the k-nearest neighbour classifier, the k nearest points are considered and the sign of the majority is used to classify the sample

III. BUILDING THE MODEL

High level system structure of this model could be broadly divided in 3 components:

- Packet dump capturing
- Raw data transformation
- Stacked Prediction Module

Packet dump capturing

This is the first step of the system. We can use any readily available utilities to capture network dump of the network. One such utility is Wireshark or any other custom packet capture utility can be used. This dump should be ideally collected for a complete cycle for which analysis should be done. For example, if an organization has a weekly cycle defining the complete activities done on their network, they should take a week's dump for this analysis. Taking dump of a complete cycle would ideally include all the parameters which might affect the study, and thus would give a fair analysis of network bandwidth pattern. Considering a complete cycle for packet dump creation would ensure that all the components of current bandwidth usage cycle are captured. These components would include periodic updates of software's, weekly/monthly recurring download jobs, and all such components. In current implementation we are using Wireshark for creating network dump for our system.

Raw data transformation

This is an important step which involves transforming the raw data from packet dump into organized form which can then be used by the prediction module of the system. This organized data represents the current network bandwidth usage pattern. In this implementation we are using JnetPcap library for decoding pcap packets from the dump.

Few important attributes which need to be fetched from the packet structure are as follows:

- Source IP address
- Destinations IP address
- Payload
- Timestamp

Apart from the attributes which are directly available from the packet, there could be some parameters which can be called as derived attributes. Such attributes may not be readily available from the network packet; however such attributes can be derived from already available information. One such parameter useful in this study is weekday attribute, which can be derived from timestamp attribute.

This module of the system consists of various transformation steps like Data selection, Data Cleaning, and Data transformation. Here selection of the attributes which are readily available in the packet structure would be called as Data selection. Example of such attributes would be Source IP address, Destination IP address, etc. A network packet consists of lot of information in form of headers of various OSI layers as well as actual payload. However we might not need all this information in our study. Process of removing the unwanted information would be called as Data cleaning. For example, we also have information of source and destination Mac addresses in our network packets. However, this information is not useful for our study, and thus we clean out such data. Data cleaning is a very important step since it helps the system to focus only on useful information and eliminate the unwanted information. Data transformation would include derivation of derived attribute from already available information.

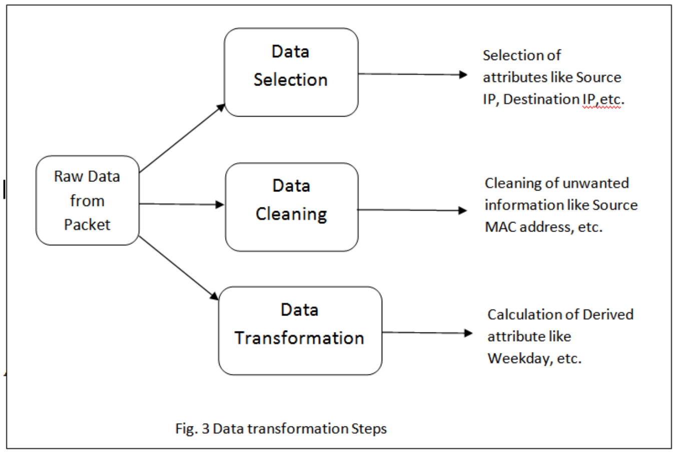
Data Selection
Selection of attributes like Source IP, Destination IP,etc.
Raw Data from Packet
Data Cleaning
Cleaning of unwanted information like Source MAC address, etc.
Data Transformation
Calculation of Derived attribute like Weekday, etc.

Fig. 3 Data transformation Steps

Apart from above transformation step, sometimes steps like grouping, summation, averaging might be required so as to organize the data in required format. In this study we use grouping and summation functions to group the data in required format which would later be absorbed by prediction module as an input.

Once our data is processed in our above flow we should have data in an organized format. Example of such an organized data is shown in below table. This organized data will now be the input for the prediction module.

Node	Date	Weekday	Hour	Payload (KB)
Node1	04/02/2014	Tuesday	00	2432
Node2	04/02/2014	Tuesday	02	424
Node3	06/02/2014	Thursday	03	5356
Node3	06/02/2014	Thursday	04	466
Node1	04/02/2014	Tuesday	02	79879
Node6	04/02/2014	Tuesday	09	977
Node1	06/02/2014	Thursday	20	677
Node1	06/02/2014	Thursday	19	5578
Node1	06/02/2014	Thursday	17	858
Node5	06/02/2014	Thursday	08	868

Table 1. Processed Data

Prediction Module

This is the last and the most important module of this system. Input to this module is the organized data from Data transformation step. One important factor in this module is to choose the right prediction model based on the current case study and available set of data. Based on input data from previous step, below algorithms are to be considered to build stacking based model:

- ✓ Neural Networks
- ✓ Holt Winters Algorithm

Stacking of prediction models

Stacking (sometimes called stacked generalization) involves training a learning algorithm to combine the predictions of several other learning algorithms. First, all of the other algorithms are trained using the available data, then a combiner algorithm is trained to make a final prediction using all the predictions of the other algorithms as additional inputs. Stacking typically yields performance better than any single one of the trained models. It has been successfully used on both supervised learning tasks and unsupervised learning.

Why Stacking is required: In this case study, there are two types of payload events that can occur. There are certain payloads which are periodic or have a certain cycle, like database refresh cycles, software update cycles, certain files being copied from remote servers, etc. However we also have random events of day to day work cycle which cause payloads to occur. Thus a single prediction algorithm like Linear regression would not be able to predict future values accurately. Hence stacking of prediction models is done so as to consider both kinds of events. Algorihms like Neural network would be used along with Holt Winters algorithm. Holt Winters algorithm would help to find the periodic payload cycles and Neural network algorihm works accurately for random events. Thus combination of these two algorithms would result in better prediction in this case study.

IV. Testing and Result Analysis

Accuracy of the model is largely dependent on the breadth of training set. For testing the model, we take initial data of 10 cycles and consisting of 2 nodes. Here a cycle means a week. Out of the total 10 cycles, we use 8 cycles of data to train the system and remaining 2 cycles are used to test the trained model.

Below shown is the graph plot of the output values predicted by HoltWinters algorithm in stage 1.

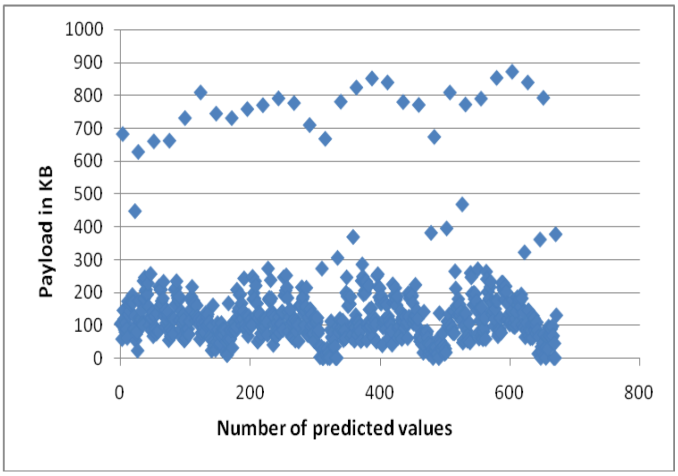

Fig.4 Predicted values using HoltWinters algorithm

When the same dataset is given to 2nd algorithm of stage 1 i.e. Neural network, output value graph produced is shown below:

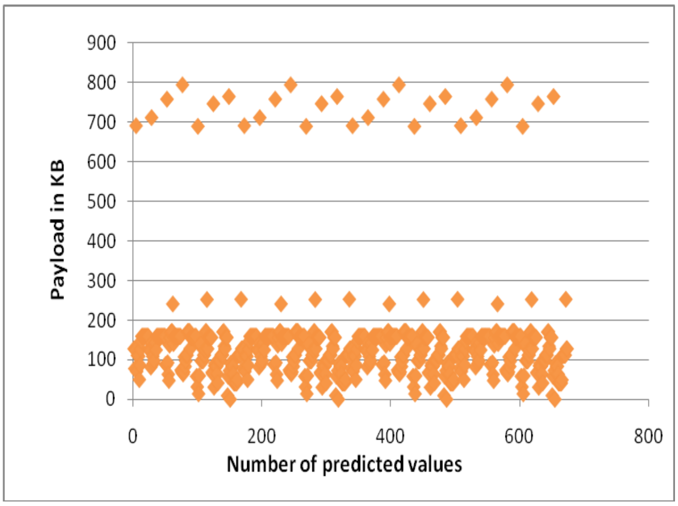

Fig.5 Predicted values using Neural network

Once we have the prediction values from both the algorithm, we pass this values to the decision making logic which helps us in choosing the final value amongst the two. The decision making logic used in this study comprises of calculation the RMSE (Root-mean-square deviation) for both the algorithms and then choosing the prediction values of the algorithm which has better RMSE value. Below shown are the RMSE values which were calculated during testing the algorihm:

Algorithm	RMSE
HoltWinters	82.96255
Neural Network	101.8338

Table 2. RMSE values

Based on the above value, the final output values of prediction module will be the values returned by HoltWinters algorithm. To verify the accuracy of the output values, prediction was done for few instances from training dataset and a table comparison was done. Few samples from comparison table are shown below:

Actual Values	Predicted Values
145	126.497873533164
180	168.039601433879
657	731.899519504767
12	112.348077059864
120	88.490383947514
170	114.221593274702
113	98.6074810028432
152	111.26938016779
154	93.1340197849753
78	103.60998850933
290	279.407836789637
249	254.372064707676
224	244.576073465425

Table 3. Actual Vs. Predicted Values

V. CONCLUSION AND FUTURE WORK

As seen from table 3, the predicted algorithm does a fair job in predicting values for most of the instances. However there might be few time instances where it could be improved. For such instances, there could be few more network parameters which should be captured. Such parameters could be studied in future and prediction algorithm could be updated to take care of such parameters.

VI. REFERENCES

[1] Keleher, P. ; Bhattacharjee, B. ; Sussman, A. Decentralized, accurate, and low-cost network bandwidth prediction. Dept. of Computer Science, Univ. of Maryland, College Park, MD, USA. INFOCOM, 2011 Proceedings IEEE

[2] Alaknantha Eswaradass, Xian-He Sun, Ming Wu. Network Bandwidth Predictor (NBP): A System for Online Network performance Forecasting. Department of Computer Science Illinois Institute of Technology Chicago, Illinois 60616, USA

[3] Introduction to time series and forecasting / Peter J. Brockwell and Richard A. Davis.—2nd edition ISBN 0-387-95351-5 SPIN 10850334

[4] Box G.E.P., Jenkins G.M., Reinsel G.C. 1994. Time Series Analysis: Forecasting and Control, 3rd edn. Prentice Hall, Englewood Cliffs, NJ

[5] David H. Wolpert: Stacked generalization. (1992) NeuralNetworks 5(2): 241-259