ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2021.0345

DEEP PACKET INSPECTION USING HYBRID CLASSIFIER FOR UNKNOWN TRAFFIC FLOWS IN THE INTERNET

P. Shanmugaraja[1], K. Chokkanathan[2], K. Thangaraj[3] and P. Ilanchezhian[4]
[1,3,4]*Department of Information Technology, Sona College of Technology, India*
[2]*Department of Master of Computer Applications, Madanapalle Institute of Technology and Science, India*

Abstract

In this era of Internet every electronic device being manufactured comes with built-ins to connect itself with the Internet. Such electronic gadgets create a massive amount of traffic flows every second. Managing this immense Internet traffic has become an exhausting job. Classifying these massive flows of data and interpreting them has attained considerable importance in the sight of researchers and Internet Service Providers. Many methods have been presented by various researchers to address this classification issue. This paper presents a Hybrid Classifier algorithm, which is a combination of the well-known machine learning types; the supervised learning and unsupervised learning to solve this classification issue. Accurate and Efficient recognition of flows is the key to manage flows in real-time. This algorithm classifies the traffic flows into real time flows and non-real time flows. It uses the built decision support model for classifying the flows based on the target class. To further validate the classification, it applies the k-means model. A significant improvement in the classification accuracy has been obtained.

Keywords:

Classification Algorithms, Network Traffic, Decision Tree, QoS

1. INTRODUCTION

Network traffic or Data Traffic also known as data flows or network flows refers to the data travelling across a transmission medium at a point of time. Data traffic is measured, controlled, and classified for ensuring Quality of Service (QoS) to the network users [1] [2]. Network topologies are built based on the quantity of network traffic that is generated.

Data traffic are classified as [3]

- Elastic flows or Non real time flows – e.g., peer to peer, WWW, FTP file transfers, and electronic mail applications. Here delay of the network traffic can be tolerated
- Inelastic flows or sensitive traffic or Real-time traffic – e.g., VoIP, Multimedia applications, Video Conferencing, Webinar, Online Gaming, IP-TV and interactive applications – here the delay is not tolerated. Real time traffic demands a severe time period for transmitting packets. Data flows that reach the destination after the time period are useless. There is a strict demand on the network for timely delivery of the data flows. The delay of individual fragments greatly affects the performance of the applications which generates real-time traffic. These applications cannot tolerate delay. If a packet is marked as real-time packet, then the router which handles it gives priority for processing and forwarding it immediately.

The efficiency of the network depends not only on bandwidth, packet loss, jitter and delay, but it also depends on the user satisfaction. But the performance of the Real Time traffic is greatly affected by the delay of individual packets. They are not able to adapt to a wide range of packet delay and delay variance at the transmission over data networks. Non Real Time traffic flows are generated by applications like E-mail, Peer to Peer, etc. They are insensitive to delay.

Even though the packets are classified as real time or non-real time traffic flows if they are not scheduled properly then the classification is of no use. The job of the scheduler is to select a packet from the both the queues. There are number of scheduling mechanisms such as FIFO, Simple Priority queuing, generalize processor sharing, weighted round robin, deficit round robin, weighted fair queuing, least attained service etc.

These algorithms do not fairly and dynamically schedule the Internet traffic. The prediction of dynamic network conditions is not available in the above said scheduling disciplines. The proposed method schedules the Real time traffic flow as speedy as possible and also fairly treats the packets seated in the non-real time traffic queues.

Network administrators can decide the priority of the packets based on their requirements. For e.g., live tv packets can be given high priority with respect to others. User gratification is also considered as one of the criteria for evaluating the efficiency of the network. Organizations needs to prioritize their real time traffic also called as business-critical traffic [15] over other traffic flows.

Those organizations need a tool which can classify the data traffic as real time and non-real time flows based on certain criterions [4]. The proposed Hybrid Classifier is designed for classifying traffic flows in the Internet. The proposed algorithm buckets the traffic flows into real-time and non-real time flows comparatively better than other algorithms available. The accuracy of the classification is improved. The proposed Hybrid Classifier as shown in fig.1 is designed to classify the traffic flows based on their descriptive statistics, which includes protocol, duration of the flow, standard deviation and mean of inter-arrival time, packet length, packet size variance and packets in the flow etc. [5].

2. RELATED WORK

In the past, traffic classification techniques used well-known port numbers to identify the packets communicated on the Internet. This type of detection is the oldest methods which ease the analysis of data. This was easy and provided good results because many traditional applications used fixed port numbers assigned by or registered with the Internet Assigned Numbers Authority (IANA). After the birth of the Internet most of the applications used only one default port number. They communicated with the server using only those port numbers. John Postel detected, TCP and UDP packet headers and analyzed them by comparing port numbers with the official list of default port numbers assigned by IANA. For example, sending and receiving Email we use the Simple Mail Transfer Protocol (SMTP) on port 25 to send email and the Post Office Protocol version 3 (POP3) on port 110. In recent days Port-

based classification is ineffective because the latest applications do not communicate with standardized ports allocated to them.

3. HYBRID CLASSIFIER

The function of Hybrid Classifier is to differentiate flows into real-time and non-real-time flows. Here K-means clustering algorithm and C4.5 algorithm works together to increase the accuracy of classifying the traffic flows into real time and non-real time flows. The Fig.1 shows the architecture diagram of Hybrid classifier.

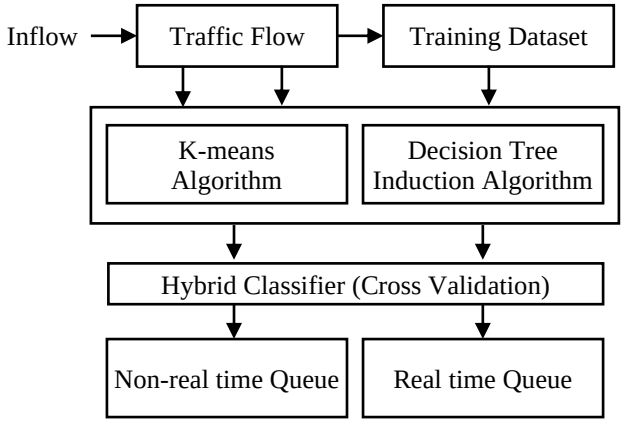

Fig.1. Hybrid Classifier Architecture

3.1 CLASSIFIER OUTPUT

The output from K-Means algorithm and C4.5 algorithm is further given as input to Hybrid Classifier out component for cross validation. If both the output of K-means and C4.5 labels a packet as real-time or non-real-time then it is positioned in the target class. If the outputs are different say for example if k-means algorithm classifies a packet to real-time and C4.5 classifies the same packet to non-real-time then the packet is positioned into the class pointed out by the classifier output.

3.2 REAL TIME AND NON-REAL TIME QUEUE

After classification the flows are placed in the appropriate queues i.e., either a real time queue or a non-real time queue.

3.3 DECISION TREE ALGORITHM

It is used to generate Decision tree from the training flows of the traffic data set.

3.3.1 Method:

Input: an attribute-valued dataset DS

1. Check for base cases
   a. All the samples in the list belong to the same class. When this happens, it simply creates a leaf node of the decision tree saying to choose that class.
   b. None of the features provide any information gain. In this case, C4.5 creates a decision node higher up the tree using the expected value of the class.
   c. Instance of previously unseen class encountered. Again, C4.5 creates a decision node higher up the tree using the expected value
2. For each flow vector attribute, *a*
   a. Find the normalized information gain ratio from splitting on *a*
3. Let *a_best* be the attribute with the highest normalized information gain
4. Create a decision *node* that splits on *a_best*
5. Recurse on the sub lists obtained by splitting on *a_best*, and add those nodes as children of *the node.*

The proposed method uses the built-in decision tree model to identify the traffic flow as soon as it is encountered and labels them. To further validate the results K-means algorithm is used and clusters the flows into the appropriate group.

3.4 K-MEANS ALGORITHM

The K-Means algorithm for is used for partitioning, where each cluster's center is represented by the mean value of the object in the cluster.

Input

1. *K*: the number of cluster class.
2. *D*: $\{f_1, f_2, f_3 \ldots f_n\}$ where f is the number of training flows in data set. Each flow (*f*) is represented as flow vector attributes $f_1=\{a_1, a_2 \ldots a_d\}$ where a_i contains flow based statistical parameter of the flow f_1.

Output

1. A set of K cluster with appropriate class label

Method

1. Arbitrarily choose *K* objects from *D* as the initial cluster centers;
2. Repeat
   a. Reassign each object to the cluster to which the object is the most similar, based on the mean value of the object in the cluster;
   b. Update the cluster mean, i.e., calculate the mean value of the object for each cluster;
3. Until no change;

The outcome of both the algorithms is further validated by the Hybrid classifier output using the following steps.

4. PROPOSED HYBRID CLASSIFIER

1. Dataset '*DF*' which contains network flows is given as input to C4.5 algorithm
2. C4.5 classifies the network flows and those classified flows are given as input to the K-Means algorithm without a class label. The class information is stored in a data structure. Let output of C4.5 classifier as $DF_{c4.5} = \{f_1{:}c_1, f_2{:}c_1, f_3{:}c_2, \ldots, f_n{:}c_1\}$ where 'c_1' and 'c_2' are the labels of Real time and Non real time class.
3. The K-Means algorithm works with the network flows to find the cluster of each flow. It clusters each flows under real time or non-real time clusters. If a flow is not clustered in either of the clusters then it is considered as an outlier flow. Let output of K-Means algorithm as $DF_{K\text{-}Means}= \{f_1{:}c_1, f_2{:}c_2, f_3{:}O, \ldots, f_n{:}C_n\}$ where class label '*O*' is a outlier flow.

4. Each flow in $DF_{C4.5}$ and $DF_{K\text{-}Means}$ are further classified using the following strategy to find the final class label of the flow.
   a. If the algorithms point a flow to the same class label, then the flow is associated to the target class and the flow is queued in the appropriate output queue.
   b. If the algorithms point a flow to different class labels, then it is considered as ambiguity. This ambiguity can be resolved using majority voting. This voting can be done by matching the classes of neighborhood data objects in both K-means clustering and C4.5 decision tree model. The flow is associated with the class label that is selected in the majority voting and returned to the appropriate output queue.
   c. If the flow belongs to an outlier object, then it is labelled based on the C4.5 decision tree model and it is placed in the output queue.

5. ALGORITHM EFFECTIVENESS

Algorithm effectiveness is calculated by the classification accuracy of the model. Informally accuracy means whether our model predicted the data set correctly or not. It is calculated by using the Eq.(1)

$$\text{Accuracy} = \frac{Number\,of\ correct\ predictions}{Total\ number\,of\ \ predictions} \quad (1)$$

The effectiveness of the Hybrid classifier is compared with K-Means clustering and C4.5 evaluated using the following standard metrics [6] [7]

- Positive Detection Rate (PDR),
- False Positive Detection Rate (FPDR),
- Accuracy and
- F-Measure.

It is calculated using the Eq.(2)

$$F\text{-}measure = (2*Precision*Recall)/(Precision+Recall) \quad (2)$$

Precision is the ratio of correctly classified flows over all predicted flows in a class as shown in the Eq.(3).

$$Precision = TP/(TP+FP) \quad (3)$$

$$\text{Recall} = TP/(TP+FN) \quad (4)$$

where,

TP, FP and FN are the true positives, false positives and false negatives of the data set.

- *False Negatives (FN)*: Percentage of members of class '*f*' incorrectly classified as not belonging to class '*f*'.
- *False Positives (FP)*: Percentage of members of other classes incorrectly classified as belonging to class '*f*'.
- *True Positives (TP)*: Percentage of members of class '*f*' correctly classified as belonging to class '*f*' (equivalent to 100% - FN).
- *True Negatives (TN)*: Percentage of members of other classes correctly classified as not belonging to class '*f*' (equivalent to 100% - FP).

The Table.1 identifies the relation between False Negative, False Positive, True Positive and True Negative. A good traffic classifier aims to minimize the False Negatives and False Positives.

Table.1. Relationships between FN, FP, TP and TN

The flow '*f*' is classified as →	***F***	**!*f***
F	TP	FN
!*f*	FP	TN

6. TRACES AND COLLECTION OF DATA SET

Traces are collected from Sona college network (SONANET) and Thiyagarajar Polytechnic Network TPTNET. Data sets are captured using wire shark tool between the college network and the commercial Internet for about 16 weeks. Around 1000 computers with various flavors of operating systems are connected to Internet and data are captured. The systems are connected through 100BaseT links and/or 802.11b/g Access points. The data from the applications are clustered into one group. Streaming consist of both video and audio streaming. Unknown data flows don't have payloads in their packets.

The proposed hybrid algorithm does not classify the flows labelled unknown. The flow labelled under unknown have no payloads. The Fig.2 graphically represents the SONANET dataset and TPTNET data set overview. The applications identified are FTP, HTTP, IMAP, POP3, DNS, SOCKS, SMTP, SOAP, SSH, SSL, MYSQL, etc. E-mail, HTTP, streaming and database are the major flows in the campus network. The campus network is used by students, staff and faculties.

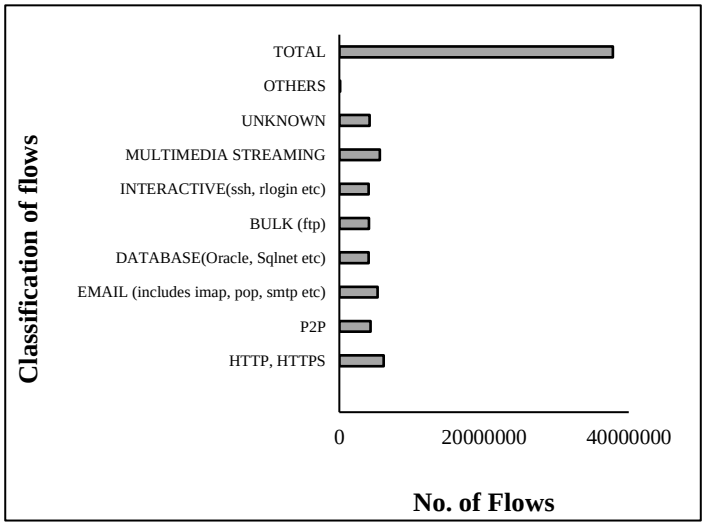

Fig.2. Dataset Overview

Algorithm: Hybrid Classifier Generate classified and cross validated Real Time and Non Real Time network flows

Input: Dataset *D* which is a set of flows $D = \{f_1, f_2, f_3,\ldots,f_n\}$. Each flow (*f*) is represented as flow vector attributes $f_1 = \{a_1, a_2,\ldots a_d\}$ where a_i contains flow based statistical parameter of the flow f_1.

Output: List of classified and cross validated Real time and Non real time network flows.

Stages:

1. Dataset 'D' which contains network flows is given as input to the built C4.5 classification model
2. The classified network flows using C4.5 is given as input to the K-Means algorithm without a class label, but the

retaining class information in some data structure. Let output of C4.5 classifier as $DF_{C4.5}$ = {f_1:c_1, f_2:c_2, f_3:O,…,f_n:C_n} where 'C_1' and 'C_2' are Real time and Non real time class label respectively.

3. The K-Means algorithm (discussed in section 4.2.1) acts on the network flows and determines the membership of each flow in either real time or non-real time cluster. The flow that fall outside the cluster regions are considered as outlier objects. Let output of K-Means algorithm as $DF_{K\text{-}Means}$= {f_1:c_1, f_2:c_2, f_3:O,…,f_n:C_n} where class label 'O' is a outlier data object.
4. Each flows in $D_{C4.5}$ and $D_{K\text{-}Means}$ is cross validated in the following way to find a final class label.
   a. If both flows have the same class label, then associate the common class label as a target class and return the flow to the output queue.
   b. If both flows have different class label, then it is a case of uncertainty, it can be solved by majority voting, which can be obtained by comparing the classes of neighborhood data object in K-Means clustering with their corresponding flow label of neighborhood data object in building C4.5 decision tree model. The class label that comes in majority voting is associated with a flow and returns it to the output queue
   c. If one of the flows is an outlier data object, then the flow is associated with the class label of same flow in C4.5 decision tree model and returns it to the output queue.

7. EVALUATING THE PROPOSED HYBRID APPROACH

The proposed work adapts the Hybrid Classifier approach to improve the overall classification accuracy of the system. The C4.5 algorithm is used in real world problems as it deals with numeric attribute and missing value. However, it is vulnerable to small variation in data that can lead to different decision trees and does not work well for the small training data set. Since the network traces data set tends to be skew in nature, the accuracy of classifier is getting affected.

On the other hand, K-Means algorithm addresses the problem of small variation in the data that arise in Decision Tree Induction C4.5 algorithm by unsupervised clustering. However, the accuracy of K-Means based classification is affected by the outlier data. Normally, outlier data objects will create an uncertain situation in the classification results. The outlier data problem associated with K-Means is easily addressed in Decision Tree Induction C4.5 classification algorithm by adapting a supervised approach. From the discussion, each method has its own limitation towards classifying data objects which has a variety of properties. There is a scope for improvement in accuracy if each classification method is implemented separately.

7.1 ANALYSIS OF POSITIVE AND FALSE POSITIVE DETECTION RATE

The performance of the K-Means, Decision Tree (C4.5) and the proposed Hybrid Classifier is evaluated based on the positive/ false positive detection rate, accuracy and F-Measure. Two data sets SONANET and TPTNET are considered for performance evaluation. When Hybrid Classifier algorithm is executed on the SonaNet data, it provides positive detection rate of 98.8% and the false positive detection rate of about 6.5% as shown in the figure 3.0. The same algorithm provides a positive detection rate of 98.2% and a false positive rate of 7.32% on TPTNet data. It provides 98.0 positive detection rate and 6.5 false positive detection rate of the training data.

7.2 ANALYSIS OF ACCURACY AND F- MEASURE

Hybrid Classifier gives an accuracy of 95.5% and 87.2% F-measure of the training data set as shown in the Fig.3. For SonaNet data it results in 94.7% accuracy and 89.7% F-Measure. It gives an accuracy of 95.2% for TPTNet data and 90.3% of F-Measure. The results are graphically depicted in Fig.3.

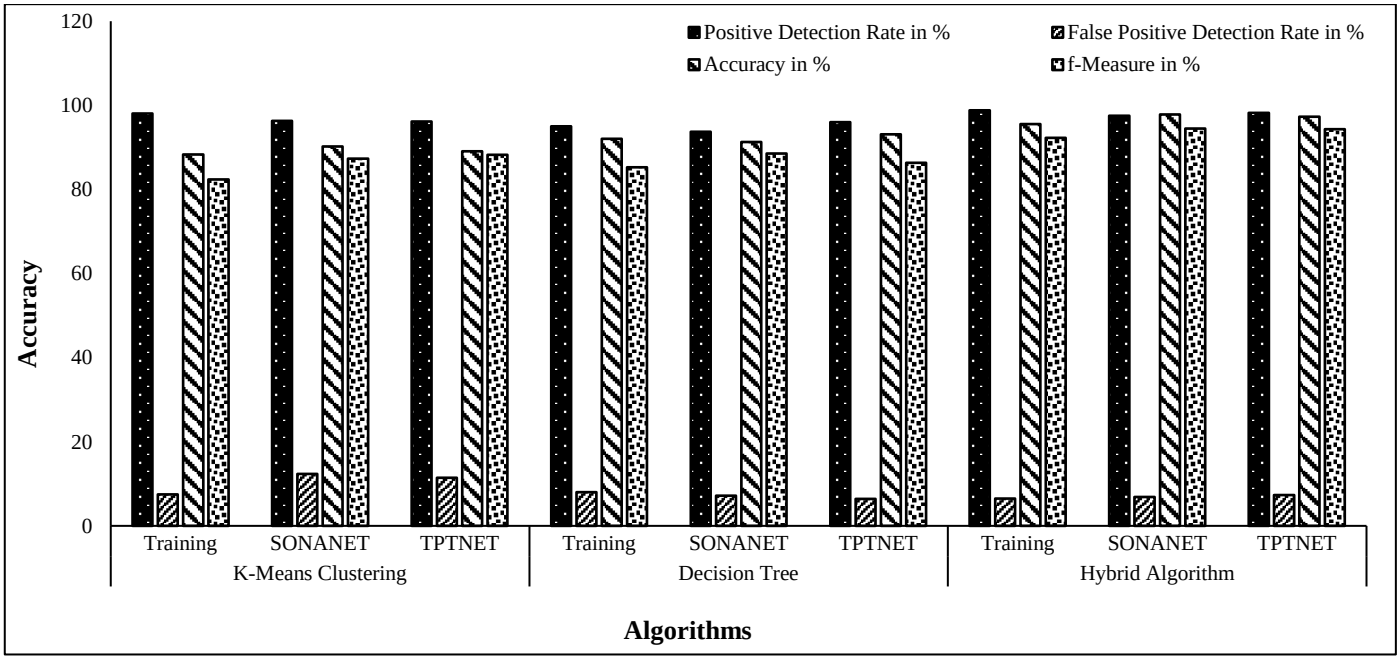

Fig.3. Accuracy of Proposed Hybrid Classifier

8. CONCLUSION

The results show that the hybrid classifier can achieve high accuracy and maximized the F-measure value when compared to K-Means and C4.5 algorithms. The following strategies make hybrid classifier better than the algorithms C4.5 and K-Means algorithms. The hybrid classifier has only two classes or cluster namely Real time and Non Real Time. A flow is categorized by both C4.5 and K-Means algorithms. If there is an uncertainty in the classification then the flows are cross validated by the proposed approach.

In the cross validation the uncertainty issue is solved by majority voting. If the flow is an outlier object then the flow is classified based on C4.5 decision tree model. With the help of this procedure positive detection rate is high and this work reduces the false positive detection rate. The proposed work also detects behavioral changes to the existing applications. The proposed approach is invulnerable to ephemeral changes in the network conditions. With the help of this procedure positive detection rate is high and this work reduces the false positive detection rate.

The proposed Hybrid Classifier reduces misclassification levels to a greater extent when compared to other classification methods. The flow labelled as 'unknown' by the well-known classification methods can be identified by the proposed Hybrid Classifier. All the results of above evaluation show that the proposed Hybrid Classifier that combines the C4.5 Decision tree induction and K-Means Clustering algorithm performs exceptionally well.

REFERENCES

[1] Changhua Sun, Lei Shi, Chengchen Hu and Bin Liu, "DRR-SFF: A Practical Scheduling Algorithm to Improve the Performance of Short Flows", *Proceedings of 3rd International Conference on Networking and Services*, pp. 1-13, 2007.

[2] K. Chokkanathan and S. Koteeswaran, "An Integrated Approach for Network Traffic Analysis using Unsupervised Clustering and Supervised Classification", *International Journal of Internet Technology and Secured Transactions*, Vol. 9, No. 4, pp. 1-12, 2019.

[3] S. Dhivya and P. Shanmugaraja, "A Study of Internet Traffic Classification using Machine Learning Algorithms", *International Research Journal of Engineering and Technology*, Vol. 2, pp. 154-157, 2015.

[4] C. Gu, S. Zhuang, Y. Sun and J. Yan, "Multi-Levels Traffic Classification Technique", *Proceedings of International Conference on Future Computer and Communication*, Vol. 1, pp. 448-452, 2010.

[5] L. Jeong Hwa and P. Jung Min, "Dynamic Scheduling for Usable Service in Network Robot", *Proceedings of International Conference on Internet and Data Structure*, pp. 252-259, 2013.

[6] C. Livadas R.Walsh, D. Lapsley and W.T. Strayer, "Using Machine Learning Techniques to Identify Botnet Traffic", *Proceedings of IEEE International Conference on Local Computer Networks*, pp. 967-974, 2006.

[7] T.T.T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning", *IEEE Communications Surveys and Tutorials*, Vol. 10, No. 4, pp. 56-76, 2008.

[8] Shang Ning, Tan Minggui, Wang Yaqing, Chui Zhongfa, Chui Yan and Yang Yong, "A Bp Neural Network Method for Short-term Traffic Flow", *Computer Applications and Software*, Vol. 23, No. 2, pp. 32-33, 2006.

[9] M.N. Kasirian and R.M. Yusuff, "An Integration of a Hybrid Modified TOPSIS with a PGP Model for the Supplier Selection with Interdependent Criteria", *International Journal on Production Research*, Vol. 51, No. 4, pp. 1037-1054, 2013.

[10] Hussein A. Mohammed, DAdnan Hussein Ali and Hawraa Jassim Mohammed, "The Effects of Different Queuing Algorithms within the Router on QoS VoIP and Communications", *International Journal of Computer Networks and Communications*, Vol. 5, No. 1, pp. 1-15, 2013.

[11] T. Karagiannis, A. Broido, M. Faloutsos and K.C. Claffy, "Transport Layer Identification of P2P Traffic", *Proceedings of International Conference on Internet Measurement*, pp. 121-134, 2005.

[12] S. Kumar, J. Turner and J. Williams, "Advanced Algorithms for Fast and Scalable Deep Packet Inspection", *Proceedings of ACM/IEEE Symposium on Architecture for Networking and Communications Systems*, pp. 81-92, 2006.

[13] S. Kumar, S. Dharmapurikar and F. Yu, "Algorithms to Accelerate Multiple Regular Expressions Matching for Deep Packet Inspection", *ACM SIGCOMM Computer Communication Review*, Vol. 36, No. 4, pp. 339-350, 2006.

[14] S. Kandula, S. Sengupta and R. Chaiken, "The Nature of Data Center Traffic: Measurements and Analysis", *Proceedings of ACM Conference on Internet Measurement*, pp. 202-208, 2009.

[15] P. Shanmugaraja, K. Chokkanathan, J. Anitha, A. Parveen Begam and N. Naveenkumar, "Dynamic Packet Scheduler for Queuing Real Time and Non-Real Time Internet Traffic", *International Journal of Recent Technology and Engineering*, Vol. 8, No. 3, pp. 1-12, 2019.