IMPROVED SELECTIVE BOUNDARY CUTTING ALGORITHM FOR IDENTIFYING PACKET LOSSES

B.HARITHA LAKSHMI [#1] , R.RAVI KANTH [#2]

[#1] M.Tech Scholar ,Department of Computer Science and Engineering, MVGR College of Engineering ,Chintalavalasa,Vizianagarm ,AP,India.

[#1] Assistant Professor ,Department of Computer Science and Engineering, MVGR College of Engineering ,Chintalavalasa,Vizianagarm ,AP,India

ABSTRACT

A computer network is defined as a telecommunications network which is mainly used to exchange between one computer with other computer either of limited distance or of far distances. In computer networks, all the computing device exchange their valuable data with each other using a new layer called as data link layer which are connected through either cable media or wireless media. Generally networks are classified into various types based on the usage and configuration. One among the best network is CSA (Client Server Architecture) network, in which all the clients try to connect or configure with server in order to get the resources from the server. In this type of network a client always try to send a request to the server at any time and in turn the server will receive the request from the client and try to send the response back to that client. During the communication between each and every nodes in the network the data is mainly divided into packets of equal sizes. If any node during the communication becomes inactive it will lead to a network cut between the consecutive nodes which may sometimes leads to edge cut. There are many decision tree based packet classification algorithms available in order to show the network performance of HiCuts, Hyper Cuts and EffiCuts.As the decision tree based algorithms are very efficient in identifying the cuts they are sometimes involve complicated heuristics for determining the fields and the number of cuts. So in this paper we have implemented a unique packet classification algorithm using the technique of finding boundary cut. By using this proposed algorithm, one can able to find out the cuts that occur during the data communication between nodes. In this paper we have implemented boundary cut detection algorithm based on TCP, UDP and FTP protocols, where the three protocols are mainly used to send data from sender to destination through router and they can able to identify the packet loss between the packets that travel all through three protocols. Once the data received at the receiver end, the receiver cab able to view the data and find how many packets was lost and what are the delay and throughput for that data. As an extension we can also find out the priority in which the packets are received by the receiver once the packets are received at its end. By conducting various experiments on our proposed system, we finally came to a conclusion that this proposed model is very accurate in identifying the packet loss at any end with an accurate delay and throughput.

Key Words: Wireless Media, Sensor networks, Hyper Cuts, EffiCuts, Client-Server Network

Issue 6, Vol. 6 (November 2016)
ISSN 2249-6149

I. INTRODUCTION

A wireless sensing element network could be an assortment of nodes organized into a network such every node having sensing and process capabilities. Every node has associate degree RF transceiver, sensor, and memory, battery-powered by battery. Today sensors are wide utilized in varied analysis fields since they'll monitor temperature and thus whether or not foretelling may be created easier. They're arbitrarily deployed in areas with sensors connected per the applications that they're getting used. Since they're being battery-powered up by batteries, energy consumption ought to be decreased so as to prolong the lifetime of sensing element nodes. in an exceedingly network, sensing element nodes communicate with one another in order that results are obtained as a part of their hand in glove combined work. Since every node must communicate with all the opposite nodes, wireless links are established between them. A cut is outlined because the failure of node. It will separate the network into disconnected methods incapable of communication with one another. Since they're arbitrarily deployed, loss of property may be quite black as they're going to result in the breakdown of entire network [1].

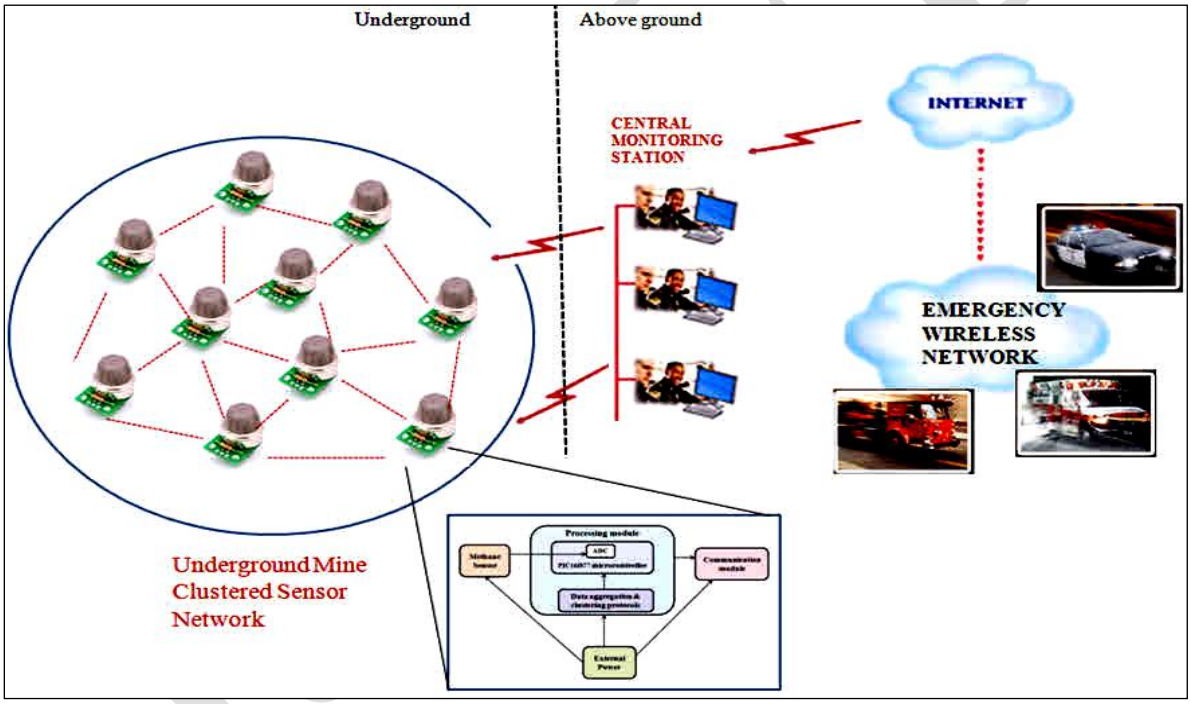

Figure. 1. Represents the Architecture of a Smart Wireless Sensor Network deployed both in Underground and Above Ground Regions

From figure 1, we can clearly find out the deployment architecture of a smart wireless sensor networks with a different forms of usage. As from the above figure we can clearly find out that many emergency networks are using wireless network as a source of communication. Hence all these networks are communicated with the help of a medium like internet from one region to other region. From the above figure we can clearly find out the difference in the deployment of the wireless sensor networks, where the underground network is almost clustered and arranged below the underground. If we look at the figure the above ground network is

connected we can able to find out that a wireless sensor networks are connected from a central monitoring station with the help of internet and it is widely used by a various services like fire,medical,police and emergency and so on.

As we all know that wireless sensor networks has many advantages in the current society ,but it still has some limitations in terms of deployment size and maintenance of that deployment takes a huge cost .So as the maintenance of the wsn takes a huge cost it is not widely spread and deployed in the primitive days. But now a day's almost all are using this wsn as an attractive medium for communication as due to its current low operational cost and reliability. Even in the current days there were some limitations that take place in the current wsn, in terms of node failure which is caused due to different problems and in different situations. Normally, node failure is expected to be quite common due to the typically limited energy budget of the nodes that are powered by small batteries [2],[3].If there was any failure happen in the network, it will leads to reduction of number of multi hop paths available in the network. Such failures cause a subset of nodes—that have not failed—to become disconnected from the rest, resulting in a"cut".

In order to achieve the Quality of Service (QoS) for a network, it is very mandatory to determine the packets, which leads a major role in achieving security and QoS over the network [3]. A packet classifier should compare multiple header fields of the received packet against a set of predefined rules and return the identity of the highest-priority rule that matches the packet header. Many algorithms and architectures have been proposed over the years in an effort to identify an effective packet classification solution [4]–[6]. Use of a high bandwidth and a small on-chip memory while the rule database for packet classification resides in the slower and higher capacity off-chip memory by proper partitioning is desirable [7]. The main metrics that include for the packet classification are processing speed of each and every packet that to be carried out for a very wide region. This processing speed is mainly calculated by the number of off chip memory access regions that are required for determining the packet class.

In this paper, we mainly try to propose a novel efficient packet classification algorithm based on boundary cutting. Here cutting is nothing but the disjoint space that is available for each and every packet which is transmitted between the region. Here for each and every packet there are two types of protocols like TCP and UDP for transferring data from one system to other.So in this paper we are using a third protocol like FTP which can select a file and can be transferred from one node to other. Here in the proposed application the packet classification table is automatically built and don't require any of the complicated heuristics used by earlier decision tree algorithms [8]-[10]. The proposed algorithm has two main advantages over the primitive algorithms. First, the boundary cutting of the proposed algorithm is more effective than that of earlier algorithms since it is based on rule boundaries rather than fixed intervals. Hence, the amount of required memory is significantly reduced. Second, it is very accurately identified the number of white spaces that are available while transferring the packets over any of the three protocols that we use in the proposed application, this gives a better level of security for our current application. Here in our proposed application we have a facility to select the meta data from sender window like TCP takes the input as text entered by the sender and UDP takes a media files either like image or audio or video files and FTP takes any of the document files for sending from source to destination. All these three types of data combine we called them as meta

data ,as this will be transferred from sender to receiver.Initailly the sender will try to send the data ,which will be reached for the router who is available in the intermediate level. Once the router receivers the meta data information in the form of packets length, then it will automatically forward the data to the receiver window which is waiting for the packets. Once the receiver is received with all the meta data, here the receiver has the facility to find out the packet information like priority in which the data is received, the loss of packets in terms of empty spaces that are available for each and every packets during transmission, the delay and the throughput information of meta data.

II. BACKGROUND KNOWLEDGE

In this section we mainly discuss about the background work that was carried out in finding the boundary cut while sensing packets and we will also discuss about the different types of cuts that are available in the networks during data transmission.

MAIN MOTIVATION

The main motivation for developing this proposed application takes place from the origin of a sensor network designed or assumed as a undirected graph with G=(V,E), Where G represents the undirected graph with a set of vertices or nodes as 'V' and the distance or link between each and every set of vertices is represented as edges or 'E'.Here we try to take or assume two pair nodes like (u, v) such that nodes u and v can exchange messages between each other. Note that we assume that inter-node communication is symmetric in nature. Here nodes u, v are always said to be incident on both set {u, v}. The nodes which are having direct relation with a appropriate node like 'u' are known as neighbors of node 'u'. A cut is the failure of a set of nodes V from G results in G being divided into multiple connected components [6]. Recall that an undirected graph is said to be connected if there is a way to go from every node to every other node by traversing the edges, and that a component Gc of a graph G is a maximal connected sub graph of G. We are interested in devising a way to detect if a subset of the nodes has been disconnected from a distinguished node, which we call the source node, due to the occurrence of a cut.

Now we discuss about the various types of CUTS that are available within the network as follows, these cuts are generally happen when we try to transfer data from one terminal or node to other terminal which is available within the network.

HI-CUTS

This is the most prominent and primary cut that was studied and analyzed in the current paper. In HI-Cut each rule defines a five-dimensional hypercube in a five-dimensional space, and each packet header defines a point in the space. Normally the HiCuts algorithm [8] recursively cuts the space into subspaces using one dimension per step. Each subspace ends up with fewer overlapped rule hyper cubes that allow for a linear search. In the construction of a decision tree of the HiCuts algorithm, a large number of cuts consume more storage, and a small number of cuts cause slower search performance. It is challenging to balance the storage requirement and

the search speed. The HiCuts algorithm uses two parameters, a space factor (spfac) and a threshold (binth), in tuning the heuristics, which trade off the depth of the decision tree against the memory amount. The field in which a cut may be executed is chosen to minimize the maximum number of rules included in any subspace. The spfac, a function of space measure, is used to determine the number of cuts for the chosen field. The binth is a predetermined number of rules included in the leaf nodes of the decision tree for a linear search.

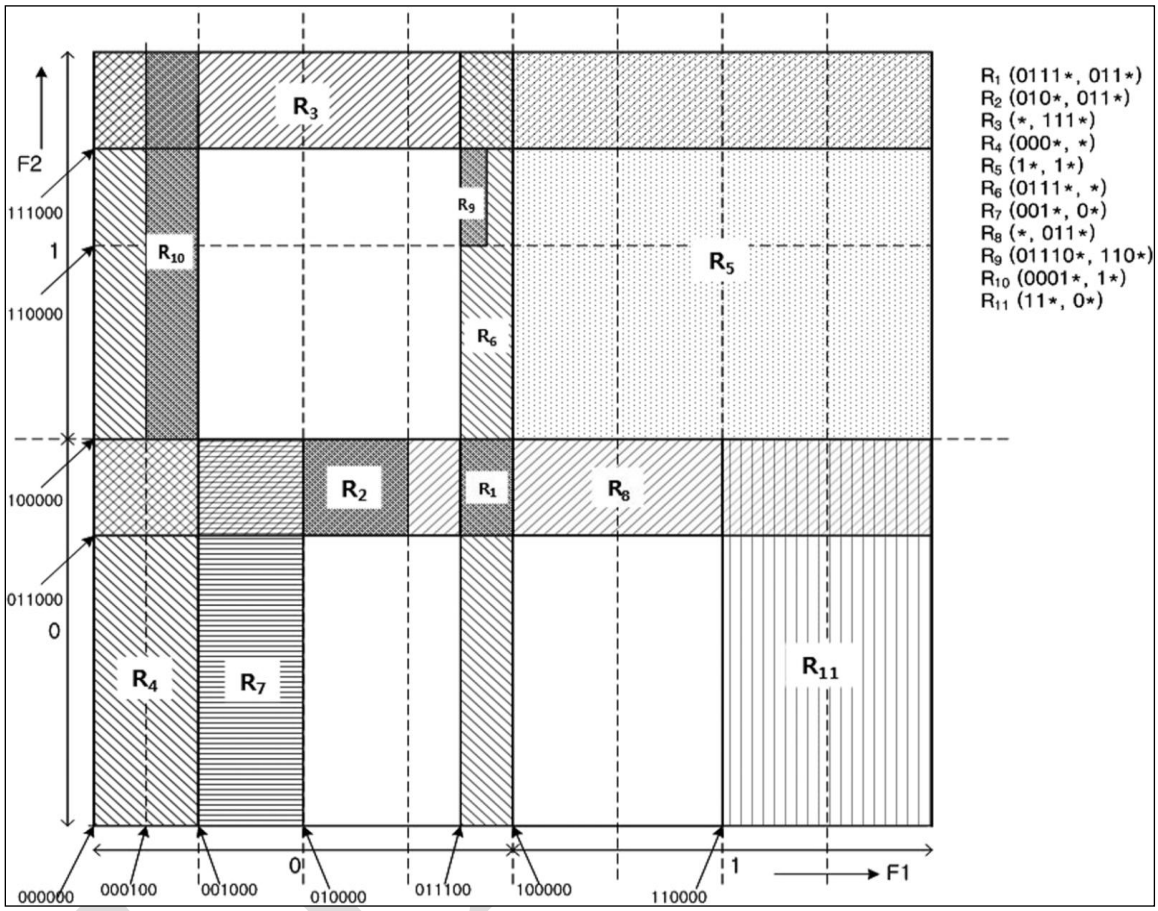

Figure. 2 Represents the Boundary Rules in a Prefix Plane

For example we can consider a two-dimensional (2-D) plane composed of the first two prefix fields, a rule can be represented by an area. Fig. 2 shows the areas that each rule covers in a prefix plane for a given 2-D example rule set. As shown, a rule with (i , j) lengths in F1 and F2 fields covers the area of $2^{W-i} \times 2^{W-j}$, where is the maximum length of the field (is 32 in IPv4).

HYPERCUTS

While the HiCuts algorithm only considers one field at a time for selecting cut dimension, the HyperCuts algorithm [9] considers multiple fields at a time. For the same example set, the decision tree of the HyperCuts algorithm is shown in Fig. 3. The *spfac* and *binth* are set as 1.5 and 3, respectively. As shown at the root node, the F1 and F2 fields are used simultaneously for cutting. Note that each edge of the root node represents the bit combination of

Issue 6, Vol. 6 (November 2016)
ISSN 2249-6149

00, 10, 01, and 11, respectively, which is one bit in the first field followed by one bit in the second field. For the same input, the search follows the third edge of the root node and compares it to. The search then follows the first edge and compares to and at a leaf. Compared to the HiCuts algorithm, the decision tree of the HyperCuts algorithm generally has a smaller depth (not shown in the figures), as multiple fields are used at the same time in a single internal node.

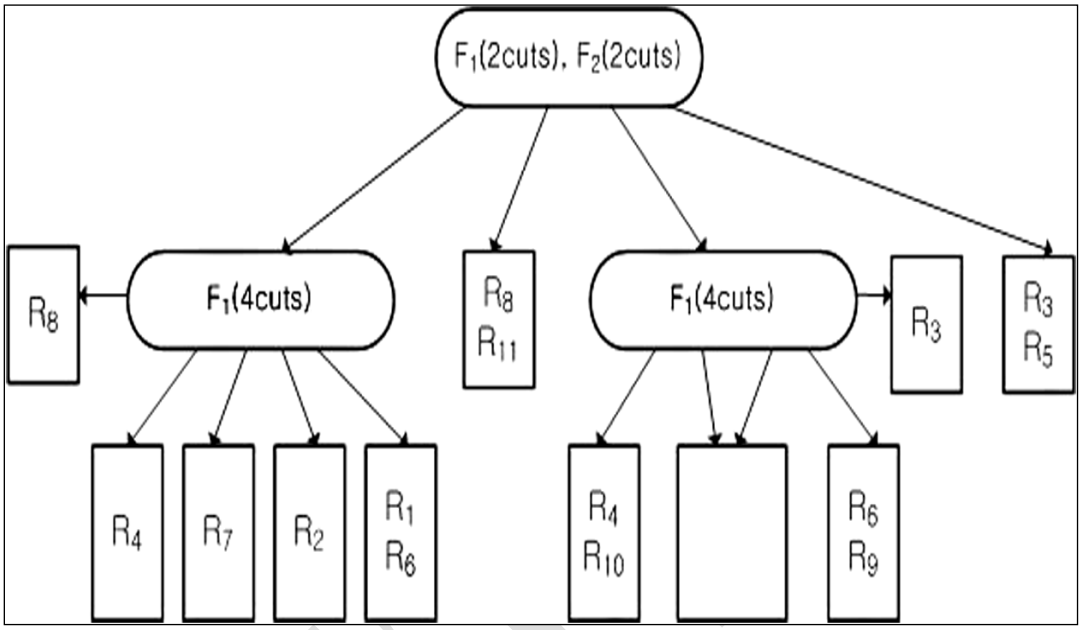

Figure. 3 Represents the Decision Tree Rules in a Prefix Plane

Recently, a new decision tree algorithm, EffiCuts, is proposed [11]. In solving the issues of previous decision tree algorithms, EffiCuts employs several new ideas such as tree separation and *equi-dense* cut. The tree separation is to separate small rules from large rules and makes multiple decision trees so that fine cutting for small rules does not cause the replication of large rules. Here, a small (large) rule means a rule covering a small (large) subspace. While HiCuts and HyperCuts employ equal-sized cuts as shown in Figs. 2 and 3, the *equi-dense* cut is to employ unequal-sized cuts based on rule density in each subspace in order to distribute rules evenly among the children.

III. UNIQUE PACKET CLASSIFICATION ALGORITHM

In this section we will mainly discuss about the proposed unique packet classification algorithm and its design phase in detail in this section. we present the design of the unique packet classification algorithm as a tool which has three steps: 1) probing the path; 2) deriving ranks; and 3) partitioning packets with different forwarding priorities to different groups based on their ranks.

Now let us discuss about that in detail as follows:

Issue 6, Vol. 6 (November 2016)
ISSN 2249-6149

PROBING THE PATH

From the below figure 4, we can clearly find out our link probe method ,where we want to test the k packets of different types in our current application.Initailly the unique packet classification tool sends several bursts(n_b) from a specified source node to a destination node. The interval between bursts is represented as Δ . Each burst consists of n_r rounds, in which k packets, one for each packet type studied, are interleaved in random order. So, there is $n_{r \times k}$ back-to-back packets in each burst. There are three parameters Δ, n_b , n_r for the probe method. In order to achieve independence between bursts, i.e., to ensure the router's queuing busy period caused by one Δ burst does not interfere with the following one, should not be too small. On the other hand, in order not to experience large background traffic fluctuation duration the probe, we need to keep the whole probe duration within a relatively short period. In practice, is set to one to ten seconds to keep overall probe duration within several minutes.

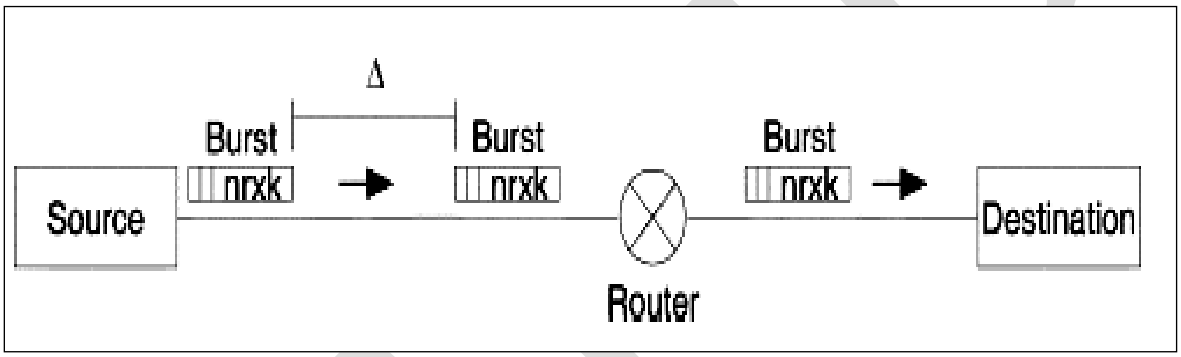

Figure. 4 Represents the Burst Consists of $n_r \times k$ packets.

The below algorithm or procedure 1 clearly justifies that ksplit (anrs) is used for splitting the packets which are send from source to the specified destination at the router node. Hence for splitting of packets we use K-Means as clustering algorithm and it will cluster the different packet types into the appropriate panels or spaces that are provided in the destination node.

Procedure 1 ksplit($anrs$)

1: $k_j \leftarrow \text{len}(anrs)$
2: **if** $k_j = 1 || (\max(anrs) - \min(anrs) \leq \theta_{1-\alpha,k_j,n_b,k})$ **then**
3: **return** $anrs$
4: **else**
5: $[anrs_1, anrs_2] = \text{kmeans}(anrs, 2)$
6: **return** $[\text{ksplit}(anrs_1), \text{ksplit}(anrs_2)]$
7: **end if**

As we are using three different protocols in our current application we need to use three types of bursts to give rankings for their individual data that are available in the meta data. This is clearly represented in figure 4, where there will be three bursts occur in a symmetric manner between source and destination through the router node.

DERIVING RANKS FOR THE VARIOUS PACKETS TYPES

For each and every burst that takes place, initially loss rate ranks are computed by first all the packets are sorted in an ascending order to their packet loss rates in that burst and then assigning ranks in order, i.e., the packet type with the largest loss rate has rank 1, the one with the second largest loss rate has rank 2 and etc. Also if we look at random of packet losses, the ranks of different packet types are always like random arrangements over the all bursts when the packet types are treated equally. If we go on other side, the ranks of certain packet types are always small when they are treated with low priority. The main advantage of using this ranks is we have a good research to bound the variance of loss ranks caused by the random effects whereas we do not have that bound for loss rates when the loss model is unknown.

PRINCIPLE OF PARTITIONING BASED ON RANKS

In this principle of partitioning based on ranks each and every packet burst can be treated as an observation. The problem of identifying whether there is any consistent difference among k ranks over the maximum 'n' observations is known as *problem of n rankings* [12].If we take an example of classic nonparametric solutions such as the Friedman test [13] we can find whether there is consistent difference, but they do not make partitions among packet types. Therefore in this paper we try to propose the use of Average Normalized Ranks (ANR) to group packet types when there is consistent difference. The ANR is the average of the ranks for a packet type over all bursts. Our statistical method is as follows:

Calculate ANR. Let $r_i^m = (1, 2, \ldots)$ denote the rank for packet type i in mth burst. The Normalized Rank NR_i^m is r_i^m/k. The range of NR_i^m is between $1/k$ and 1. The ANR_i for packet type i is

$$\mathrm{ANR}_i = \left(\sum_{m=1}^{n_b} NR_i^m \right) / n_b.$$

From the above ANR we can able to find the ranks for each and every packet type after a burst that takes place after each and every packet type. Now in the below section we will discuss about the modules that are used in the current application for development of proposed unique packet classification algorithm over network having boundary cuts and packet losses.Now let us discuss about that in detail in the following next section.

IV. IMPLEMENTATION MODULES

Implementation is the stage where theoretical design is converted into practical design. Here we divide the application into various modules and then decide the flow of the application in a modular fashion. The following application has totally 4 modules to implement this current unique packet classification algorithm over boundary cut with packet losses.

1. Preparing Meta Information Module
2. Dumping the Packets into the Network Module
3. Analysis of Network Policies Module
4. Report Generation Module

1. PREPARING META INFORMATON MODULE

In this first module in sender side it will generate sample packets from different packet types with certain counting. It will prepare a meta packet which consists of packet types and its counting with sending order of packets. This meta information is which will be send to the receiver first, before sending the sample packets it will be received by the receiver and store it feature reference. The receiver will accept the sample packets in an order which is specified in the meta information packets. It also contains the time gap (T) between packet types.

2. DUMPING THE PACKETS INTO THE NETWORK MODULE

It's the second phase of unique packet classification tool after preparing the meta data it has to generate the sample packets in all types. After generating the packets it has to send the types one by one in a particular time gap (T) which is mentioned in the meta info. Then all the packets will be dumped in to the network to the sender. To ensure that our tool will deduct the policies in high traffic fluctuation. Then the packets will be travels to the destination via the routers which can enforce the policies if it has then according to policy it will be reordered.

3. ANALYSIS OF NETWORK POLICIES MODULE

When the packets arrived in destination the analyzer in the destination will receive the packets in an order and keep track it. Then it will compare the received order of packets with meta info. Which is received first? Then it will check for packet numbers and types to find out the loss of packets. It will check the time gap (T) between the packet type to find out the delay and out of order. These information's are manipulated to generate the policies about the networks.

4. REPORT GENERATION MODULE

Based on the packets counting that received by the receiver after comparing with meta info. it will find out the loss. Based on the time gap (T) between packets types in the receiver side and sender side it calculates the delay. And based on the orders of packet type received by the receiver and send by the sender it finds out of order (OOO).Based on these metrics it will

generate a report and send it to sender about the network policies by unique packet classification tool.

V. CONCLUSION

In this paper we finally proposed a unique packet classification algorithm using the technique of finding boundary cut. By using this proposed algorithm, one can able to find out the cuts that occur during the data communication between nodes. In this paper we have implemented boundary cut detection algorithm based on TCP, UDP and FTP protocols, where the three protocols are mainly used to send data from sender to destination through router and they can able to identify the packet loss between the packets that travel all through three protocols. Once the data received at the receiver end, the receiver cab able to view the data and find how many packets was lost and what are the delay and throughput for that data. As an extension we can also find out the priority in which the packets are received by the receiver once the packets are received at its end. By conducting various experiments on our proposed system, we finally came to a conclusion that this proposed model is very accurate in identifying the packet loss at any end with an accurate delay and throughput.

VI. REFERENCES

[1] H. J. Chao, "Next generation routers," *Proc. IEEE*, vol. 90, no. 9, pp.1518–1588, Sep. 2002.

[2] A. X. Liu, C.R.Meiners, andE.Torng, "TCAMrazor: A systematic approach towards minimizing packet classifiers in TCAMs," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 490–500, Apr. 2010.

[3] C. R. Meiners, A. X. Liu, and E. Torng, "Topological transformation approaches to TCAM-based packet classification," *IEEE/ACM Trans.Netw.*, vol. 19, no. 1, pp. 237–250, Feb. 2011.

[4] F. Yu and T. V. Lakshnam, "Efficient multimatch packet classification and lookup with TCAM," *IEEE Micro*, vol. 25, no. 1, pp. 50–59, Jan.–Feb. 2005.

[5] F. Yu, T. V. Lakshman, M. A. Motoyama, and R. H. Katz, "Efficient multimatch packet classification for network security applications,"*IEEE J. Sel. Areas Commun.*, vol. 24, no. 10, pp. 1805–1816, Oct. 2006.

[6] H. Yu and R. Mahapatra, "A memory-efficient hashing by multi-predicate bloom filters for packet classification," in *Proc. IEEE INFOCOM*,2008, pp. 2467–2475.

[7] H. Song and J. W. Lockwood, "Efficient packet classification for network intrusion detection using FPGA," in *Proc. ACM SIGDA FPGA*,2005, pp. 238–245.

[8] F. Baboescu, S. Singh, and G. Varghese, "Packet classification for core routers: Is there an alternative to CAMs?," in *Proc. IEEE INFOCOM*,2003, pp. 53–63.

[9] S. Dharmapurikar, H. Song, J. Turner, and J. Lockwood, "Fast packet classification using Bloom filters," in *Proc. ACM/IEEE ANCS*, 2006,pp. 61–70.

[10] P. Wang, C. Chan, C. Lee, and H. Chang, "Scalable packet classification for enabling internet differentiated services," *IEEE Trans. Multimedia*,vol. 8, no. 6, pp. 1239–1249, Dec. 2006.

[11] A. Coates, A. Hero, III, R. Nowak, and B. Yu, "Internet tomography,"*IEEE Signal Process. Mag.*, vol. 19, no. 3, pp. 47–65, May 2002.

[12] J. D. Gibbons, *Nonparametric Statistical Inference*. New York: Marcel Dekker, 1985.

[13] G. E. Noether, *Introduction to Statistics: A Nonparametric Approach*. New York: Houghton Mifflin, 1976.

VII. ABOUT THE AUTHORS

B.HARITHA LAKSHMI is currently pursuing her 2 years M.Tech in Department of Computer Science and Engineering at MVGR College of Engineering Vizianagaram, AP, India.Her area of interest includes Computer Networks.

R.RAVI KANTH is currently working as an Assistant Professor in Department of Computer Science and Engineering at MVGR College of Engineering Vizianagaram, AP, India. He has more than 5 years of teaching experience in engineering colleges.His research interest includes Data Mining.