TASHKENT UNIVERSITY OF
INFORMATION TECHNOLOGIES
NAMED AFTER MUHAMMAD AL-KHWARIZMI

MUHAMMAD AL-XORAZMIY NOMIDAGI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI

BULLETIN OF TUIT:

MANAGEMENT AND COMMUNICATION TECHNOLOGIES

ISSN:2181-1083 DOI:
Tashev Komil № 3s(31)2024

Analysis of hardware implementation of packet classification algorithms

Tashev Komil,
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi
Tashkent, Uzbekistan
k.akhmatovich@gmail.com

***Abstract*— This paper describes network packet classification algorithms for attack detection and prevention systems. Approaches to hardware implementation of network packet classification algorithms using decision trees, field splitting, and parallel processing are presented. The field splitting algorithm in the network context was initially studied in the Field- Split algorithm Bit Vector, the main purpose is to reduce memory consumption. To balance the bandwidth, and delay, which allows to improve the efficiency of power consumption and memory access, DPRAM is proposed for the FSBV algorithm, which will enable data to be read repeatedly. In the packet classification process, the source address, destination address, source port, and destination port are calculated by applying various operations for matching. To operate, to match the fields based on the FSBV algorithm, triggers, LUTs, and various logic elements will be applied.**

***Keywords*—** *packet classification, Intrusion detection system, matching packets.*

1. INTRODUCTION

Packet classification is the process of categorizing packets according to their header fields. This process is used in network devices such as routers, firewalls, intrusion detection and prevention systems (IDPS), and others to determine the context of packets and perform important actions.

Actions may include discarding unauthorized packets, copying, scheduling prioritizing, and encrypting protected packets [1].

Traffic forwarding may deal with different services for the same path, such as packet filtering, malicious attack prevention, accounting and billing, and traffic rate limiting [1, 2, 3].

2. RELATED WORKS

Nowadays, one of the important properties of packet classification algorithms is the processing speed, at low cost. The use of Ternary Content-Addressable Memory (TCAM) provides the simplest hardware solution for single-match packet classification. However, its high cost and high power consumption have led to extensive research into several alternative algorithmic solutions [4], including decision tree-based algorithms Hypercuts [5], Extended Grid of Tries with Path Compression (EGT-PC) [6, 7], Allotment Routing Table (ART) [8], Field Split Bit Vector (FSBV) [9] and others, which are discussed in more detail below.

The Hypercuts algorithm, the best-known single-match algorithm, has been evaluated using existing single-match C and VHDL implementations. Its advantage is that it requires little modification for use in multiple-match classification, other than modifying it to return all matches from a linear search of the leaf node rule list.

The preprocessing algorithm builds a decision tree based on the classifier structure. Cuts are made simultaneously in the selected dimensions, as well as at each level of the decision tree, and recursively on the child nodes of that level until the number of rules in each node is less than a predetermined value called the segment size. No cuts are made at a node with fewer elements than the segment size, and that node becomes a branch of the decision tree.

Each internal node in the decision tree that HyperCuts builds has the following three things associated with it:

B(v) A block representing a D-field of intervals;

NC. The number of sections made by the cuts and the corresponding array of NC pointers.

R(v) List of rules contained in the specified field.

However, the Hypercuts algorithm can be ruled out as a possible solution for several matches due to two serious drawbacks [5]:

• High degree of rule set overlap leads to memory explosion.

• Leaf nodes typically contain a large number of rules, resulting in lengthy linear search.

EGT-PC (Extended Grid of Tries with Path Compression) [6, 7] is based on a structure called a hierarchy grid.

A trie is essentially a binary search tree where each branch leaving a node is labeled with a 0 or 1. The prefix corresponding to a particular leaf node is the concatenation of all the bits encountered in the path from the root to that leaf node. A grid of tries is used for two-dimensional (i.e., two-field) matching. For each two-dimensional rule, nodes in the first-dimensional tree have a pointer to the root of the corresponding second-dimensional tree. Nodes in the second-dimensional tree contain a list of rules that match the node.

The grid of attempts is suitable for performing classification with multiple hits because the single-hit algorithm in its basic format finds all matches before performing a linear search to find the highest priority. However, EGT-PC jump pointers may, depending on the implementation, cause some matching rules to be missed. These algorithms aim to achieve a balance between

processing speed, memory efficiency, and implementation complexity. By leveraging advanced techniques, they address the limitations of TCAM-based solutions, such as scalability and energy efficiency. Among these, decision tree-based approaches have shown significant promise for adapting to varying packet classification requirements.

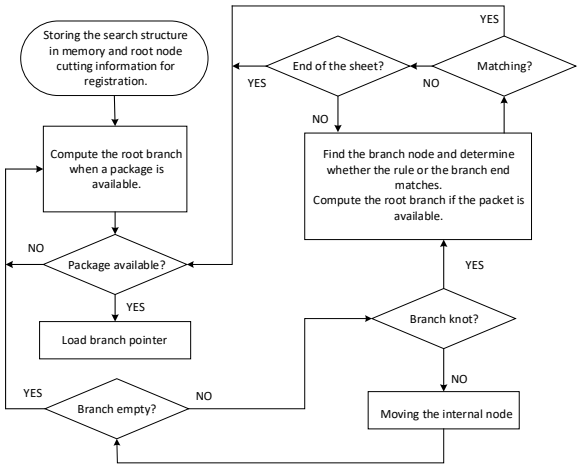

Fig. 2.13. Functional diagram of the packet classification mechanism

The architecture used to evaluate the EGT-PC algorithm for classifying multiple-match titles consisted of three blocks operating in parallel, as shown in Figure 2.15.

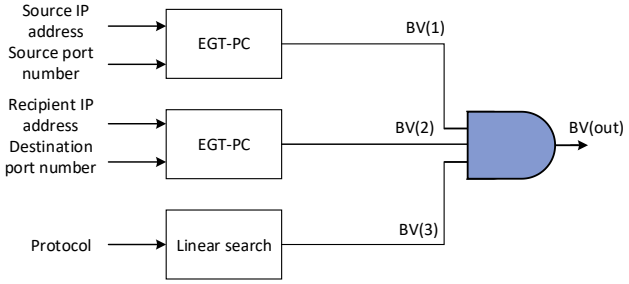

Fig. 2.15. EGT-PC multiple-match architecture

Two instances of the EGT-PC algorithm are used to classify a source address and port pair and a destination address and port pair, respectively. The IP address of each EGT-PC block is mapped to the 1st dimension tree , and the port number is mapped to the 2nd dimension tree . Note that all other address and port number pairs were tested, but those shown in Figure 2.15 were found to be the most efficient in terms of both speed and memory requirements. This is due to the relatively large number of unique port numbers compared to the number of unique IP addresses. Linear search is performed for a very small number of protocol types. Each block produces a bit vector with a length equal to the number of compressed rules.

The three bit vectors are then concatenated together to produce a total output bit vector.

ART Algorithm - This multi-bit tree-based routing table was invented by Donald Knuth [8]. Free software for single matchings implemented by Y. Hariguchi and D. Knuth was adapted to perform classification based on multiple matchings.

Large address lengths result in excessively large tables, so the address is usually broken into several short addresses called hops. In multi-level ART, each edge node contains a pointer to the table of the next level. This pointer will be NULL if there is no rule with a longer prefix than the one that matches the edge node. The address length of each level of the multi-level ART algorithm is chosen to minimize the number of tables created from a given set of rules.

ART extension for performing multiple matches. To perform a multiple-match search, each edge node can match multiple rules, and it must also include all rules that apply to its parent nodes.

The ART algorithm is used to perform matching separately for each IP address and port number field, and the matching rules are found by combining the output bit vectors using logical AND, as shown in Figure 2.17.

Research on packet classification with multiple hits has so far mainly focused on TCAM-based solutions since single-hit algorithms are considered unsuitable for classification with multiple hits due to the extensive intersections between rules in NIDS rule databases [10]. TCAM-based solutions require additional logic to return all hits since TCAMs typically return only one match with the highest priority [11]. Song and Lockwood [12] and Jiang and Prasanna [13] explored non-TCAM algorithmic approaches, but both papers propose architectures that include TCAM, albeit at a much smaller scale compared to fully TCAM-based solutions. Both approaches use bit vectors, where each bit corresponds to a rule index.

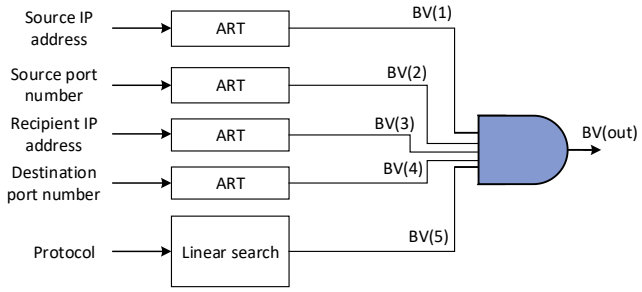

Fig. 2.17. Multiple-match architecture using ART

BV-TCAM (Bit Vector - TCAM) architecture by Song and Lockwood [12], as shown in Fig. 2.20, classifies each of the port numbers in parallel using a multi-bit tree to create two-bit vectors. The multi-bit tree used is based on a patent-pending tree bitmap [14]. A small TCAM implementation that can handle multiple matches, classifies the concatenation of source, destination, and protocol type to create a third bit vector. The reduction in the number of TCAM entries is achieved by matching the concatenation of IP addresses and protocol type in the rule set with a significantly shorter rule set containing only unique triplets. Consequently, the output bit vector the vector from the TCAM must be unpacked, i.e. mapped into a wider bit vector corresponding to the original set of rules.

Finally, the three-bit vectors are concatenated together to produce the final result.

Many tree-based algorithms use prefix expansion and leaf popping to improve performance. Prefix expansion [7] transforms a set of prefixes into an equivalent set with a shorter prefix length to allow multi-bit matching - as is also used in the ART algorithm.

Field-Split Parallel Bit Vector Architecture [9,16] The Field-Split Parallel Bit Vector (FSBV) splits the port number fields into individual bits, which are classified individually to produce a bit vector per bit. The two IP address fields and the

protocol type field are classified using two TCAMs and a CAM, respectively, similar to the BV-TCAM architecture. All the bit vectors are then concatenated together to produce the overall result, as shown in Figure 2.22.

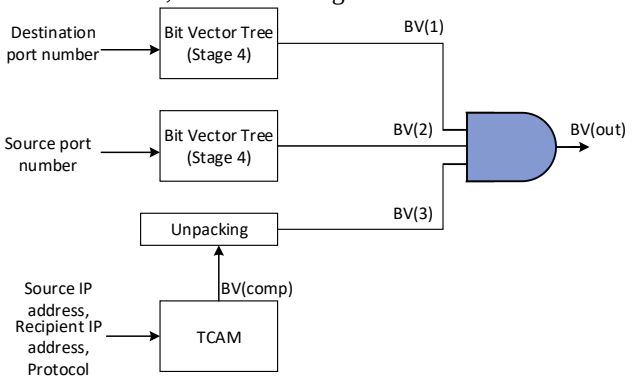

Fig. 2.20. General architecture of BV-TCAM

Table 2.6.

Example of a set of rules for a 4-bit field

4-bit field matching patterns	Rule
11 *0	R1
0101	R2
110 *	R3

Table 2.6 and Fig. 2.22 illustrate the FSBV scheme for a 4-bit field and a rule set consisting of three rules.

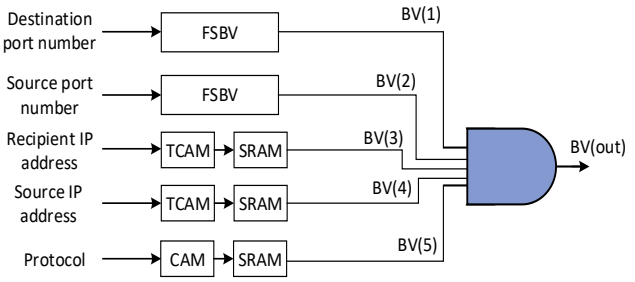

Fig. 2.22. FSBV diagram for the example rule set

This algorithm can be easily mapped to hardware architecture. Figure 2.21 shows the basic architecture of the FSBV algorithm for matching N rules, each of which can be represented as a triple string. According to the characteristics, the number of unique values in the SA/DA/ protocol fields is much smaller than the number of bits in these fields. Therefore, TCAM and CAM are used, which are effective in matching input with a small number of entries for these fields. Small TCAMs perform prefix matching in the SA/DA fields, while CAMs with 4 inputs perform exact matching in the protocol field.

In the general architecture shown in Fig. 2.23, the FSBV scheme is used to map both port number fields, each of which consists of 16 bits. Thus, the hardware implementation requires 32 memories, each of depth 2, to store the rule bit vectors. Using FSBV to map port numbers is efficient because the number of unique source and destination port numbers present in the rule set is much smaller than the total number of rules. This allows for significant compression of the rule bit vectors. Jiang and Prasanna [13] chose TCAM and CAM to map IP addresses and protocol types, respectively, because they can efficiently map an extremely small number of unique values of each. The bit vectors for each are stored in SRAM and indexed by the TCAM/CAM output.

Table 2.7 summarizes and compares the results of the three algorithms considered with the performance estimates given by Song and Lockwood. [12] and Jiang and Prasanna [13] for BV-TCAM and FSBV systems. Based on these figures, the FSBV architecture uses relatively little memory, and its simplicity should lead to an efficient FPGA solution.

Table 2.7

Comparison of algorithms

No.	Meanings	Multiple Matching Algorithms					
		Hyper Cuts	EGT-PC	ART	BV-TCAM	FSBV	Stride B.V.
1.	Memory (bits per rule)	465693	120	142	74	17	104
2.	Cycles per package	109961	3824	658	13	0.5	428
3.	Energy consumption	6.85	0.55	0.15	0.01	0.01	0.01

BV-TCAM and FSBV schemes use only a minimal amount of TCAM, which accounts for a small fraction of the total energy consumption. As a result, they are much more energy efficient than schemes based solely on TCAM, and this is confirmed by the performance comparison results provided by Jiang and Prasanna [13] .

FSBV also performs significantly better than the EGT-PC and ART solutions. This is because FSBV leverages the characteristics of Snort's ruleset header sections by compressing source and destination IP addresses into very small TCAMs and separating port fields using FSBV's efficient technique.

A parallel approach to implementing FSBV. The field splitting algorithm in a network context was originally studied in the Field- Split algorithm Bit Vector (FSBV) [13]. In the FSBV algorithm, the main goal is to reduce memory consumption. Their study of 5-field packet classification rules presented in the rule set of many NIDPS [15] shows that the SA, DA, and PR fields of the rules contain very few unique values compared to the size of the rule set. To balance throughput, latency, which allows for increased energy efficiency, and memory access, DPRAM is proposed for the FSBV algorithm, allowing for multiple data reads. The block diagram of the proposed approach is shown in Fig. 2.24.

In this approach, the algorithm receives packets from both DPRAMs, with a packet scheduling module with 4 field-dividing packet modules. The peculiarity of this approach is that the four field-dividing packet modules transmit information for packet classification, which is associated with rules. All four packet classifiers work in parallel, creating a final classification result for balancing the throughput. The packet classification modules are associated with rules that are stored in lookup tables (LookUp Table).

The use of DPRAM allows multiple read or write operations to be performed simultaneously, compared to single-port RAM, which only aims to receive data at a time.

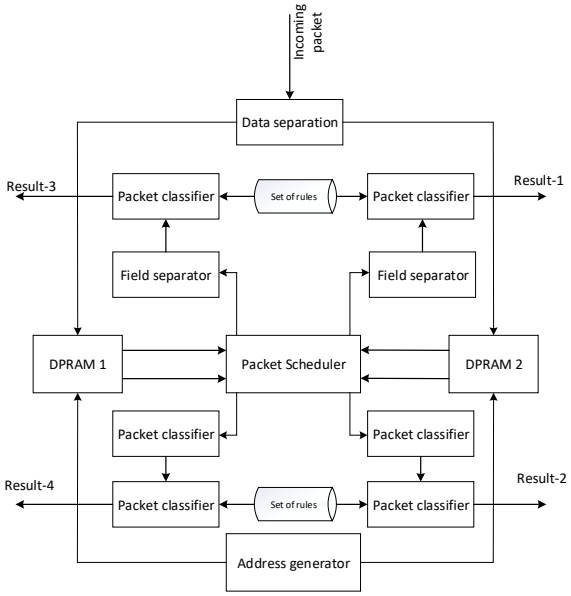

Fig. 2.24. Parallel multiple matching approach based on FSBV algorithm

An important module that is used to perform the packet classification process is the packet classifier block (Fig. 2.25). The source address, destination address, source port and destination port will undergo different matching processes. This block is composed of triggers, LUTs and various logical elements.

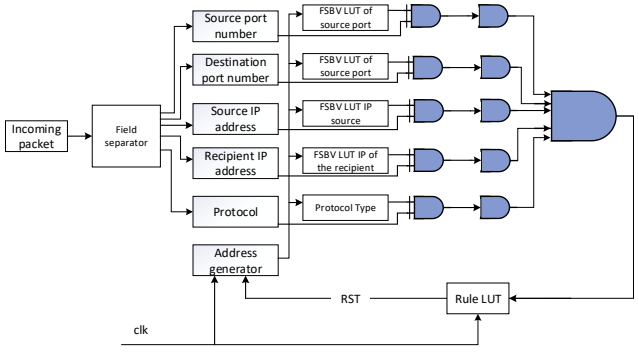

Fig. 2.25. Schematic diagram of the packet classification module

In the process of classifying packets, the source address, destination address, source port, and destination port are calculated by applying various matching operations. To perform the operation to match the fields based on the FSBV algorithm, flip-flops, LUTs, and various logical elements will be applied. It should be noted that the matching processes are performed independently for each source address, destination address, source port, and destination port. XNOR logical elements are used for the matching process of two packets or fields, as a result, the output signal is "1" when the two input signals are the same, otherwise, the result is "0" (The pseudocode of the algorithm is given in Appendix 1).

In this approach, a new mapping process is initiated by resetting the LUT address generator, which allows for real-time power savings.

A variety of queuing procedures can be used to schedule packets.

The procedure for using the packet scheduler is shown in Fig. 2.26. When scheduling, it is necessary to introduce a delay constraint, which is used to take into account the queuing discipline. This scheme uses the First Come, First Served (FCFS) mechanism - which is similar to FIFO (First Come, First Out), and is also easy to implement. The advantage of the FCFS queue is that it provides packets with a known delay D. In this case, when C is the connection speed, B is the maximum buffer size, the delay D can be calculated as follows:

$$D \leq \frac{B}{C}$$

Since the input stream represents all the traffic packets sent over the network, the classifier determines the class of packets and moves them to the appropriate queue. Priority queues must contain packets from different traffic classes, each traffic class has its own queue. The task of the packet scheduler is to select the queue to be serviced.

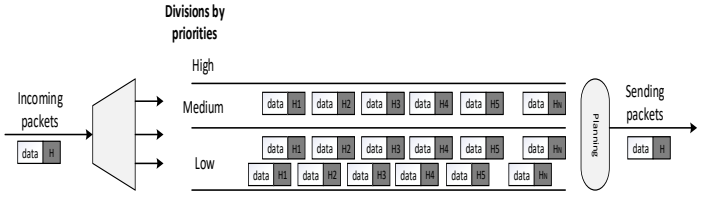

Figure 2.26 Using the Packet Scheduler to Schedule Packets

The classifier moves packets from the input stream into the appropriate queues. Each traffic classification has its own priority queue, denoted by q_i, where $I = 1, \dots, N$. Each queue has a limited capacity and can only store N_i packages. If q_i contains exactly N_i packets and the classifier finds the next packet from this traffic class in the input stream, then this packet will be rejected. All traffic classes have the same arrival distribution with a mean value λ_i For i^{-ro} traffic class. This means that in each time interval, a new packet can be observed in the queue q_i with probability λ_i. The average delay requirement is R_{ti} — the maximum permissible average delay per packet in a queue q_i and time interval t. This requirement is considered to be specific to queues $q_1, \dots, q_{N-1}$ and may vary over time. The queue with the greatest effort is the last queue q_N, which has no delay requirements and must be processed as quickly as possible.

The scheduler must select exactly one packet to be transmitted in each time slot. The scheduler must formally select one action (a_i) from the set of actions $A = \{a_1, a_2, \dots, a_n\}$. The server will receive one message from the queue q_i as a result of the action a_i.

CONCLUSION

For hardware implementation, one potential problem with the FSBV scheme is that it relies on previous IDS rule sets having a very small number of unique IP addresses. A number of single-match packet classification algorithms have been adapted to perform multiple matches, and their performance has been evaluated in terms of speed and power consumption. The memory efficiency of FSBV is due to the characteristics of recent IDS rule sets. The relatively small number of unique values of each header field in the rule set allows the bit vectors representing the rule set to be significantly compressed for each field. Its relative simplicity can be said to lend itself to implementation in hardware (e.g., FPGAs), where the bit vectors can be stored in block RAM to maximize performance.

REFERENCES

[1] Madhi D., and Ramasamy K., 2007 , “Network routing algorithms, protocols, and architectures”, Morgan Kaufmann, USA, pp. 1-957.

[2] Gupta P., and McKeown N. b, 1999, “Packet classification using hierarchical intelligent cuttings,” In Proceeding of Hot Interconnects VII, pp. 34-41

[3] Meiners CR, Liu AX, and Torng E., 2010 , “Hardware Based Packet Classification for High Speed Internet Routers”, Springer, New York-USA, pp. 1-122

[4] Taylor, D. E. (2005). Survey and Taxonomy of Packet Classification Techniques. ACM Computing Surveys, 37(3), pp.238-275. DOI: http://dx.doi.org/10.1145/1108956.1108958

[5] Singh, S., Baboescu , F., Varghese, G. and Wang J. (2003). Packet Classification Using Multidimensional Cutting. Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03). pp.213-224. DOI: http://dx.doi.org/10.1145/863955.863980

[6] Baboescu , F., Singh, S. and Varghese, G. (2003). Packet Classification for Core Routers: Is there an alternative to CAMs? IEEE INFOCOM 2003. pp.53-63. DOI: http://dx.doi.org/10.1109/INFCOM.2003.1208658

[7] Srinivasan, V. and Varghese, G. (1998). Faster IP lookups using controlled prefix expansion. ACM SIGMETRICS Performance Evaluation Review. 26(1), pp.1-10. DOI: http://dx.doi.org/10.1145/277858.277863

[8] Hariguchi , Y. (2002). ART – Allotment Routing Table – A Fast Free Multibit Trie Based Routing Table; Webpage: www.hariguchi.org/art

[9] Jiang, W. and Prasanna, V. K. (2009). Field-Split Parallel Architecture for High Performance Multi-Match Packet Classification Using FPGAs. Proceedings of the 21st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '09). pp.188-196. DOI: http://dx.doi.org/10.1145/1583991.1584044

[10] Yu, F. and Katz, R. H. (2004). Efficient Multi-Match Packet Classification with TCAM. Proceedings 12th IEEE Symposium on Hot Interconnects (HOTI'04). pp.28-34. DOI: http://dx.doi.org/10.1109/CONECT.2004.1375197

[11] Yu, F., Lakshman, TV, Motoyama , MA and Katz, RH 2005. SSA: A Power and Memory Efficient Scheme to Multi-Match Packet Classification. Proceedings of the 2005 ACM Symposium on Architectures for Networking and Communications Systems (ANCS). pp.105-113. DOI: http://dx.doi.org/10.1145/1095890.1095905

[12] Song, H. and Lockwood, J. W. (2005a). Efficient Packet Classification for Network Intrusion Detection using FPGA. FPGA '05: Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays. pp.238-245. DOI: http://dx.doi.org/10.1145/1046192.1046223

[13] Jiang, W. and Prasanna, V. K. (2009). Field-Split Parallel Architecture for High Performance Multi-Match Packet Classification Using FPGAs. Proceedings of the 21st ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '09). pp.188-196. DOI: http://dx.doi.org/10.1145/1583991.1584044

[14] Eatherton , W. (1998). Hardware-based Internet Protocol Prefix Lookups. MS thesis, Electr. Eng. Dept., Washington Univ., St. Louis, MO, USA; http://www.arl.wustl.edu/~jst/studentTheses/wEatherton-1999.pdf

[15] Snort. Snort: Network intrusion prevention and detection system (ips /ids). http://www.snort.org/

[16] M. Karimov, K. Tashev and M. Yoriqulov , "Problems of increasing efficiency of NIDS by using implementing methods packet classifications on FPGA," 2019 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2019 , pp. 1-5, doi: 10.1109/ICISCT47635.2019.9011925.

[17] K. Tashev, I. Durdona and A. Mokhinabonu, "Comparative performance analysis the Aho-Corasick algorithm for developing a network detection system," 2022 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2022, pp. 1-7, doi: 10.1109/ICISCT55600.2022.10146815.

[18] K. Tashev and M. Karimov, "New Approach to developing efficient NIDPS," 2021 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2021, pp. 1-5, doi: 10.1109/ICISCT52966.2021.9670253.