International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 5, October 2021, pp. 3855~3863
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i5.pp3855-3863

High performance modified bit-vector based packet classification module on low-cost FPGA

Anita P.[1], Manju Devi[2]
[1]Visvesvaraya Technological University, Belagavi, India
[1,2]Department of Electrical and Computer Engineering, The Oxford College of Engineering (TOCE), Bangalore, India

Article Info

Article history:

Received Dec 22, 2020
Revised Feb 15, 2021
Accepted Mar 16, 2021

Keywords:

FPGA
Header extractor
Modified bit vector (MBV)
Packet classification
Packet generation module
Protocol
Range search

ABSTRACT

The packet classification plays a significant role in many network systems, which requires the incoming packets to be categorized into different flows and must take specific actions as per functional and application requirements. The network system speed is continuously increasing, so the demand for the packet classifier also increased. Also, the packet classifier's complexity is increased further due to multiple fields should match against a large number of rules. In this manuscript, an efficient and high performance modified bit-vector (MBV) based packet classification (PC) is designed and implemented on low-cost Artix-7 FPGA. The proposed MBV based PC employs pipelined architecture, which offers low latency and high throughput for PC. The MBV based PC utilizes <2% slices, operating at 493.102 MHz, and consumes 0.1 W total power on Artix-7 FPGA. The proposed PC considers only 4 clock cycles to classify the incoming packets and provides 74.95 Gbps throughput. The comparative results in terms of hardware utilization and performance efficiency of proposed work with existing similar PC approaches are analyzed with better constraints improvement.

Corresponding Author:

Anita P.
Visvesvaraya Technological University, Belagavi, India
Department of Electrical and Computer Engineering
The Oxford College of Engineering, Bangalore, India
Email: anita.p.research@gmail.com

1. INTRODUCTION

The network system speed is continuously increased, whereas the demand for network constraints parameters like traffic analysis, load balancing, security, firewall, and quality of service (QoS) also increases drastically. So the packet classification (PC) algorithms are the primary solutions to meet the above network requirements. The PC aims to match the header-fields of the incoming packets with the appropriate ruleset. In general, most of the network architecture uses fields for packet classification, namely, source address (SA), destination address (DA), source port (SP), destination port (DP), and protocol [1]. A few of the network parameters are kept in mind to design an efficient PC module, like fast updation, memory requirement, flexibility for real-time implementation, specifications, searching speed, and a number of fields. There are many PC approaches available in the past, like the Tuple space approach, geometric based approach, decision tree-based approach, and trie based approaches, divide and conquer approaches, and hardware-based approaches. The Tuple space approach includes tuple space pruning and search algorithms. The geometric based approach includes Fat inverted, segment tree area-based quadtree, and a grid of tries algorithms. The decision tree-based approach has hyper cuts and HiCuts algorithms for PC. The trie based approaches have set-pruning tries, hierarchical tries, and grid-of tries algorithms. Similarly, the divide and conquer approach has a recursive flow method, lucent and aggregated bit-vector algorithm, and cross producting method. Lastly, the hardware-based methods have bitmap intersection, and ternary content addressable memory (TCAM) approaches [2]-[4].

Most of the time, the PC module's use is for an extensive network, multi-field classification, and large rule sets, increasing the complexity. The TCAM is a hardware-based approach and provides high-speed time with less complexity and not consumes more energy while classifying the packets [5], [6]. The decision tree based on the binary search on level (BSOL) engages the replication control method to reduce the memory utilization of BSOL. The decimal tree-based BSOL updates dynamically improve the speed and reduce the memory than HiCuts approaches [7]. The non-partitioning based algorithms like exhaustive search-based approaches use more classification time and support less table search. Cross producting method supports more table search and consumes less classification time. The partition-based approaches like decision-tree based, tuple space, and hash-based approaches support moderate search tables and utilize average classification time [8].

The proposed modified bit vector (MBV) based packet classification (PC) module is designed and implemented on FPGA. The PC module offers scalable and memory-efficient features, which suits real-time network applications. Section 1 explains the review of existing Packet classification using different approaches. The proposed MBV based packet classification with detailed hardware architecture is explained in section 2. Section 3 elaborates on the results and discussion of proposed work, with different design constraints on FPGA and also a comparison with existing similar PC methods with improvements. Section 4 concludes the outcome of the proposed work with improvement.

This section discusses the existing packet classification (PC) approaches for different network application viewpoints. Qu and Prasanna [9] discuss the high-performance Packet classification module on FPGA, which offers dynamic updation, Clustering, striding, power gating, and dual-port memory mechanism in a single PC engine to improve the system performance. The 2-dimensional pipelined BV based PC is designed with a scalable architecture to overcome the existing PC challenges. The dynamic updates like modifying, insertion, and deletion in the ruleset improve the optimization of PC. The latency, energy, and throughput for different rules are analyzed with better results. Zhou *et al.* [10] present the decomposition based PC algorithm with multi-core processor implementation. The decomposition methods like linear search, range search, linear BV, and Range BV based approaches are incorporated in PC. Using these decomposition methods, the performance results are analyzed for latency and throughput for different processors like AMD and Intel concerning the number of rules. The work also analyzes the search and merges latency, cache performance, and threads per core. Linan *et al.* [11] describe the improved cutting-based PC method with multidimensional features. The Improved version of HyperCut work offers a better tradeoff between throughput and memory while creating the decision tree. The HyperCut algorithms describe the filter set, cutting methods, functional evaluation, decision tree building, and searching with updating methods to classify the packets.

Wang *et al.* [12] present the TCAM based PC to improve the packet forwarding rate constraints. This work provides a memory-efficient architecture in TCAM based PC by compressing the memory space utilized for the same data in the different ruleset. The global and block mask registers used in TCAM improves the packet forwarding rate. Meshram *et al.* [13] explain the field split BV (FSBV) algorithm for PC with modular architecture. This work uses the BV approach to classify the packets to improve the latency and throughput by providing the proper ruleset. The memory requirement for this work is less compared to the existing FSBV approach. Khan *et al.* [14] explain the PC module with high throughput, which uses simple XNOR gates for classifying the packets based on the ruleset. The work offers less memory utilization and low latency for the same rules used in the stride BV based PC approach. Mohan *et al.* [15] present the different Multi-match approaches like hyper-cut (HC) based, multi-match based discriminator (MMD), and B2PC along with pipelined and distributed hash table (PDHT) based Packet classification approaches. The hyper cut offers high speed and cutting freedom from the ruleset. Whereas MMD, which is the same as the TCAM approach, offers more processing speed and adopts a parallel search approach. The B2PC provides significant determination of rule set membership. The PDHT provides efficient and high throughput classification by not using TCAM.

Zheng *et al.* [16] present a total prefix-length (TPL) Cluster-based PC algorithm approach that decomposes the rules into different clusters with the same prefix length and area-based Quadtree approach with the highest priority. The work offers better space utilization, speed in search, dynamic updation to improve PC performance Yingchareonthawornchai *et al.* [17] present a fast and dynamic PC with a sorted–partitioning method. The ruleset sort ability function explains PC and field order comparison's usefulness to overcome PC's challenges. The multi-dimension interval Tree approach offers search time, deletion, and insertion for the sortable ruleset. The online and offline sort-ability partitioning for ruleset offers priority optimization and successful searches. Huang *et al.* [18] explain the geometric space partition, and hash-table

(GSPH) based hybrid PC algorithm. This work provides better classification speed with the same accuracy by using a hash table with a parallel structure and matches large packet sets for classification. The ruleset creates the subsets and generates the hash, followed by a decision tree structure with a packet strategy to classify the packets.

The packet classification usage is increased because the multiple fields have to be matched against many rule sets. The multiple fields matching is always challenging by maintaining the same performance, larger throughput, and efficient memory usage. It has been noticed from the existing works in the above section. Most of the classification approaches consume more chip area and latency and obtains low throughput on FPGA and ASIC platforms. The proposed PC module also provides low latency, and high throughput pipelined architecture with less resource utilization on-chip, which helps perform better classification processes for incoming packets from networks. The next section explains the proposed MBV based PC with detailed hardware architecture.

2. PROPOSED PACKET CLASSILIFCATION MODULE

The packet classification is partitioning the packets into various flows the group of packets that matches a predefined rule, which comes under one flow. The packet classification aims to identify the highest priority rule, which matches the incoming packets and performs specific actions with the particular ruleset. The packet classifier module performs a match of the input packets field against the database's rules. The rule set or classifier is made up of a particular set of rules, and these rules require the header fields of the input packets as a search engine. The overview of the proposed modified bit vector (MBV) based packet classification module is represented in Figure 1. The proposed module is designed based on the various functional requirements for classifying the incoming packets. The module mainly contains packet generation module (PGM), header extractor hodule (HEM), followed by the MBV packet classification module, range search module, network aggregator, and Priority encoder generation of classified output. The Various functional modules of the MBV based PC module are explained in the below subsections.

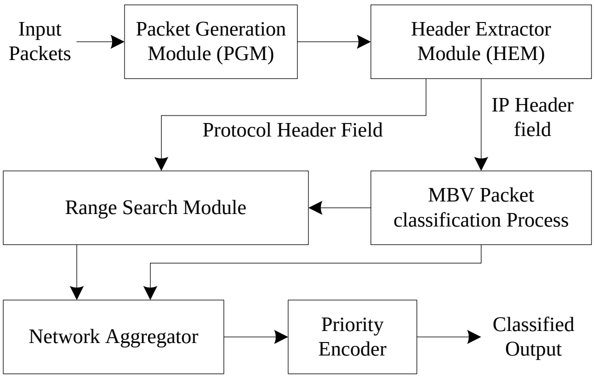

Figure 1. Overview of modified BV based packet classification module

2.1. Packet generation module (PGM)

The packet generation module (PGM) receives the incoming packets from the network system or the PC user and processes them further. The PGM initiates the first packet as the packet (SOP) packet to continue the PC process. Each packet contains 1 byte or 8-bits of data information. The PGM validates each incoming packet and processes further only the valid packets. The PGM also check any error packets appearing or not; if error packets are received, it will discard immediately and won't carry further.

2.2. Header extractor module (HEM)

The HEM is designed based on the internet protocol version (IPV-4) specification. The received packets are used to frame as different header formats for the PC purpose. The packet header fields are used in the classification process to match the packets against the rule set or not. The HEM is mainly used to extract the header field information from the received packets. The HEM is classified as the internet protocol (IP) header, ethernet header, and transmission control protocol (TCP) headers. Only IP and TCP headers are considered for the MBV Based PC. The IP headers are used for source and destination address generation. The TCP headers are used for the generation of source and destination ports.

2.3. Modified bit vector packet classification (MBV-PC) module

The MBV algorithm places a significant role in the packet classification process and is mainly used to match the source and destination IP address along with fields. The rule set is first converted into bit vectors (BV), and then matching processing needs to be done using header fields to finds the matching rule. The BV generation is generated first, and then the classification process continued further to find the highest priority match against the ruleset. The Pipelined Architecture of MBV based packet classification module is represented in Figure 2.

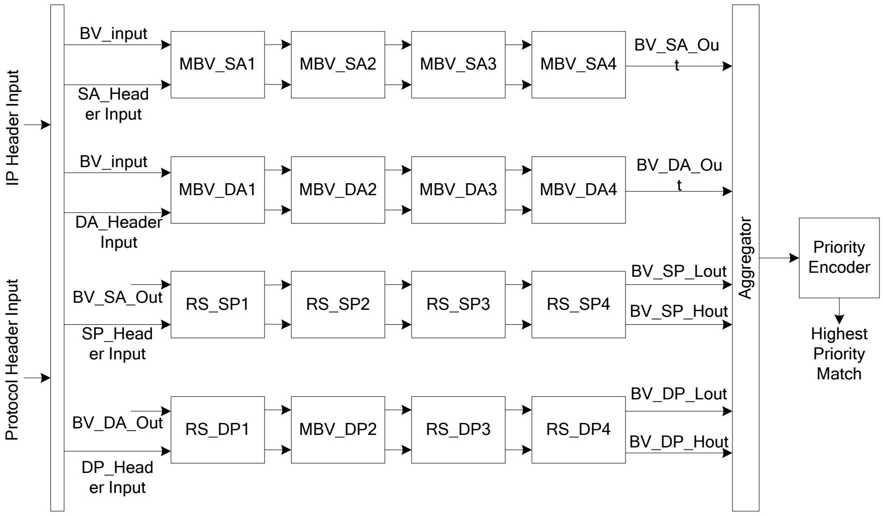

Figure 2. Pipelined architecture of MBV based packet classification module

The PC module has IP header, TCP header fields, BV fields as inputs, and the highest priority match as classified output. The IP header field is decomposed into source address (SA) and destination address (DA) as header inputs for the MBV process. Similarly, The TCP header field is decomposed into source port (SP) and destination port (DP) as header inputs for the range search process. The MBV based SA and DA are processed in a pipelined manner. In this design, 4-stages of the MBV based SA and DA and range search-based SP and DP processes are performed. The MBV process of single-stage for pipelined architecture is represented in Figure 3.

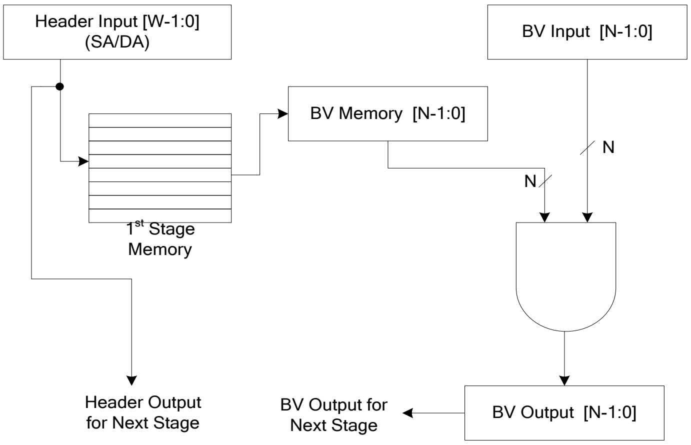

Figure 3. MBV process for pipelined architecture

The notations used in the MBV process are as follows: The 'W' is the width of the rule specified for the given field (in terms of bits), 'N' is the total number of rules used for the classification process in the ruleset, and 'k' is sub-field length (in terms of bits). The MBV algorithm working flow is represented as follows:

- Each of the header field 'W' bit rules is divided into sub-fields, and its length is 'k' bits. So the sub-field is known to stage or stride.
- The number of subfields' W/k' is considered, and each has a width of 'k' bits. Each 'k' bits of the rule will be matched with corresponding 'k' bits of header fields (SA/DA).
- The same process is continued to map the sub-fields as 2^k of 'N bit vectors.
- The corresponding rule is matched with the input header to get the match results. Each 'k' bits of the header field access the corresponding BV memory to generate the single 'N' bit vector. The same process is continued till to obtain the N bit vectors.
- Perform the bitwise AND operation of this BV memory and BV input fields to result in the matching output of the whole Ruleset of the particular Field.
- The Obtained BV output checks whether the corresponding rule set matches or not for the header field value.

The MBV algorithm process the independent matching for each sub-field with the corresponding header field. The 'W' bit rule is divided into the 'k' bits of a subfield. So the total number of sub-field is W/k. The W/k is equal to the number of searches for the whole PC, and it can be achieved by using W/k pipelined stages. In design, W=16, and k=4, so W/k=4 pipelined stages are considered.

2.4. Range search module

In the final stage, MBV matched outputs like SA, and DA outputs are input to the range search module as Bit vector inputs for the corresponding SP and DP, respectively. The packet classifier ruleset contains particular ranges for the corresponding fields, which request to match the ranges. So range search finds the ranges in the ruleset, which contains input header fields. The range search module is working the same as the MBV process in a pipeline manner. The pipeline stages are the same as W/k means 4. The range search module is performed in each stage for one sub-field. Range search module for each stage in pipelined architecture is represented in Figure 4. The range values of lower bound (LB) and upper bound (UB) are predefined against the header field. The header input field is considered as either SP or DP data. The Header field input is compared (≥) with LB values and generates the corresponding bit vector value, either '0' or '1'. Similarly, the same header field is compared (≤) with UB values and generates the corresponding bit vector value either '0' or '1'. The process continues until N-1 bit-vector values and generates the BV upper (BV_Hout) and lower bound (BV_Lout) outputs for one stage. The same range search process continued for different header fields and generated the corresponding BV lower and upper bound outputs. Finally, perform AND operation for all the stage BV lower and upper bound outputs to generate the final range search BV lower and upper bound outputs for SP and DP individually.

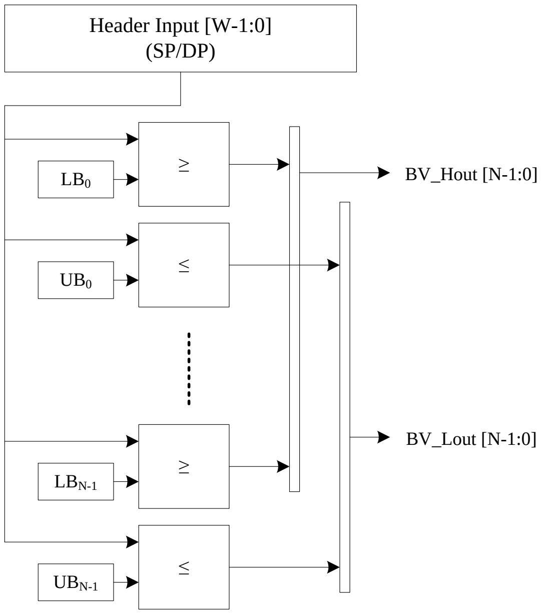

Figure 4. Range search module

2.5. Network aggregator and priority encoder

The network aggregator performs the AND operation for outcomes of BV source address output (BV_SA_out), BV destination address output (BV_DA_out), BV source port lower and upper bound outputs (BV_SP_Lout and BV_SP_Hout), and BV destination port lower and upper bound outputs (BV_DP_Lout and BV_DP_Hout). The network aggregator output is input to the priority encoder. The priority encoder collects the BV final outputs and extracts one highest priority matching rule. The priority encoder extracts the BV output bitwise and generates the highest priority output among all the matching rules as a classified output.

3. RESULTS AND DISCUSSION

This section elaborates on the proposed packet classification (PC) results using the modified bit vector (MBV) approach and its comparative analysis with existing approaches. The high-performance PC using the MBV approach is designed and synthesized on Artix-7 FPGA. The resource utilized on Artix-7 (XC7A100T) FPGA for PC using the MBV approach tabulated in Table 1. The proposed MBV_PC utilizes 3110 slices, 2167 LUT's, and 1886 LUT-FF pairs on Artix-7 FPGA as chip area resources. The MBV_PC operates at 493.102 MHz, with a combinational delay of 1.061 ns. The power consumption of the MBV_PC module utilizes 0.1W total power with the inclusion of the 0.017W dynamic power. The power consumption results are obtained using the Xilinx X-Power analyzer for clock frequency 100 MHz. The proposed MBV_PC Module takes only 4 clock cycles as latency to classify the incoming packets and obtains the 74.95 Gbps throughput as the module's speed. The comparative analysis of the MBV_PC module with the existing PC module's performance parameters like slices, LUT's frequency, and speed (Mpps) is tabulated in Table 2. (Mpps -millions of packets per second).

Table 1. Resource utilized for PC Module using MBV approach on Artix-7 FPGA

Resources	Modified BV_Packet Classifier
Chip Area	
Slice registers	3110
Slice LUT's	2167
LUT-FF pairs	1886
Time	
Minimum period (ns)	2.028
Max. frequency (MHz)	493.102
Combinational delay (ns)	1.061
Power	
Dynamic power (W)	0.017
Total power (W)	0.1
Latency and Throughput	
Latency (Clock cycles)	4
Throughput (Gbps)	74.95

Table 2. Performance analysis of proposed MBV_PC module with existing PC module's

Packet Classifier Designs	FPGA Family	Slices	LUTs	Frequency (MHz)	Speed (Mpps)
Decision tree PC [19]	Spartan-3E	NA	6442	100	NA
Ultra-scale PC [20]	Stratix-III	40070	16028	219	433
Large scale PC [21]	Virtex-5	10307	NA	125.4	250
Hi cuts [22]	Stratix-IV	15936	NA	150	100
TCAM-PC [23]	Kintex-7	NA	5200	213	426
Fast-PC [24]	Virtex-7	NA	7044	161.76	323.5
BC-based PC [25]	Artix-7	3636	2641	509.398	773.6
Proposed MBV_PC	Artix-7	3110	2167	493.102	986.2

The decision tree PC [19] was implemented on Spartan -3E FPGA, which utilizes 6642 LUT's and operates at 100 MHz. Similarly, the ultra-scale PC [20] was implemented on Stratix-III FPGA, which utilizes 40070 slices, 16028 LUT's, operates at 219 MHz, works with the speed of 433 Mpps. The large-scale PC [21] was implemented on Virtex-5 FPGA, which utilizes 10307 Slices, operates at 125.4 MHz, and works with the speed of 250 Mpps. The HiCuts based PC [22] is implemented on Stratix-IV FPGA, which utilizes 15936 slices, operates at 150 MHz, and works with the speed of 100 Mpps. The TCAM based PC [23] is implemented on Kintex-7 FPGA, and it utilizes 5200 LUTs and operates with a speed of 426 Mpps. The fast PC [24] is implemented on Virtex-7 FPGA, utilizes 7044 LUTs, and operates 323.5 Mpps. The MBV based PC reduces the area overhead in terms of 14.46% for slices and 17.94% for LUT's than the BV based PC

[25]. The latency and throughput (Mpps) of the MBV based PC are improved around 20% and 21.6%, respectively, than the BV based PC [25]. The proposed MBV based PC utilizes fewer slices, operates at a suitable frequency, and works with better speed than the existing PC approaches. The efficiency parameters comparisons of the proposed MBV_PC module with existing PC modules are tabulated in Table 3.

Table 3. Efficiency parameters comparisons of proposed MBV_PC module with existing PC module's

Packet Classifier Designs	Ruleset dependencies	Latency	Throughput (Gbps)	Memory (Byte/rule)	Range-to -prefix
TCAM-PC [23]	High	NA	3.4	23.5	No
Fast-PC [24]	High	NA	51.76	38.9	No
Stride-BV [26]	No	31	111	52	No
Emulated TCAM [27]	Low	1	64	24	Yes
BV-TCAM [28]	High	11	75	154	No
DCFL [29]	High	5	19	90	No
Proposed MBV_PC	No	4	74.95	16	No

The TCAM based PC [23] is implemented on Kintex-7 FPGA, which works with a throughput of 3.4 Gbps and doesn't support a range to prefix feature, and considers 23.5 bytes/rule in memory. The fast PC [24] is implemented on Virtex-7 FPGA, gives a throughput of 51.76 Gbps and doesn't support a range to prefix feature, and considers 38.9 bytes/rule in memory. The stride BV based PC [26] is implemented on Virtex-7 FPGA, which works at 111 Gbps with a latency of 31 clock cycles. The stride BV considers 52 bytes/rule in memory, a rule set dependencies, and range-to prefix features are not supported. The emulated TCAM [27] is implemented on Stratix –IV, which utilizes only 1 clock cycle latency with a throughput of 64 Gbps. The emulated TCAM [27] supports a range of prefix features with the fewer rule set dependencies and considers 24 bytes/rule in memory. The BV TCAM [28] is implemented on Virtex-2 FPGA, which utilizes 11 clock cycle latency with a throughput of 75 Gbps. The BV TCAM [28] does not support a range to prefix feature with more rule set dependencies and considers 154 bytes/rule in memory. The DCFL [29] works at 19 Gbps throughput with a latency of 5 clock cycles. The DCFL [29] does not support the range to prefix and considers 90 bytes/rule in memory. The proposed MBV_PC module works at 74.95 Gbps with 4 clock cycles (Latency) and considers 16 bytes/rule in memory. The MBV_PC module does not range to prefix and rule set dependency features.

4. CONCLUSION

In this manuscript, an efficient MBV based Packet classification module is implemented on FPGA, which offers high speed, low latency, and better memory utilization. The PC mainly contains a packet generation module, header extractor module, MBV based source, destination address modules, range search based source, destination port unit, and priority encoder to classify the highest priority match output. The MBV based PC utilized 2% slices, 3% LUT's, works at 493.1 MHz and consumes 0.1 total power on Artix-7 FPGA. The MBV based PC uses only 4 clock cycles with a throughput of 74.95 and utilizes 16 bytes/rule in memory. The proposed MBV based PC utilizes less overhead in terms of 14.46% for slices, 2.91% for total power, 20% for latency, and 17.35% for throughput than our previous BV based PC. The proposed PC is also compared with different PC engines with better improvement in different design constraints.

REFERENCES

[1] V. A. P. Kumar, V. Thiyagarajan and N. Ramasubramanian, "A Survey of Packet Classification Tools and Techniques," *2015 International Conference on Computing Communication Control and Automation*, Pune, India, 2015, pp. 103-107, doi: 10.1109/ICCUBEA.2015.26.

[2] B. Nagpal, N. Singh, N. Chauhan and R. Murari, "A survey and taxonomy of various packet classification algorithms," *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015, pp. 8-13, doi: 10.1109/ICACEA.2015.7164675.

[3] B. Nagpal, N. Singh, N. Chauhan and R. Murari, "A survey and taxonomy of various packet classification algorithms," *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015, pp. 8-13, doi: 10.1109/ICACEA.2015.7164675.

[4] Yie-Tarng Chen and Shin-Shian Lee, "An efficient packet classification algorithm for network processors," *IEEE International Conference on Communications*, 2003. ICC '03., Anchorage, AK, USA, vol. 3, 2003, pp. 1596-1600, doi: 10.1109/ICC.2003.1203871.

[5] M. Dixit, B. V. Barbadekar and A. B. Barbadekar, "Packet classification algorithms," *2009 IEEE International Symposium on Industrial Electronics*, Seoul, Korea (South), 2009, pp. 1407-1412, doi: 10.1109/ISIE.2009.5215939.

[6] S. Alinezhad and H. Saidi, "Efficient low power packet classification by grouping TCAM rules," *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*, Belgrade, Serbia, 2011, pp. 1531-1535, doi: 10.1109/TELFOR.2011.6143849.

[7] Y. Cheng and P. Wang, "Packet Classification Using Dynamically Generated Decision Trees," in *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 582-586, Feb. 2015, doi: 10.1109/TC.2013.227.

[8] W. Pak and Y. Choi, "High Performance and High Scalable Packet Classification Algorithm for Network Security Systems," in *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 37-49, 2017, doi: 10.1109/TDSC.2015.2443773.

[9] Y. R. Qu and V. K. Prasanna, "High-Performance and Dynamically Updatable Packet Classification Engine on FPGA," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 197-209, Jan. 2016, doi: 10.1109/TPDS.2015.2389239.

[10] Z. Shijie, Y. R. Qu, and V. K. Prasanna, "Multi-core implementation of decomposition-based packet classification algorithms," In *International Conference on Parallel Computing Technologies*, Springer, Berlin, Heidelberg, 2013, pp. 105-119, doi: 10.1007/978-3-642-39958-9_9.

[11] C. Linan, L. Zhaowen, M. Yan, H. Xiaohong and L. Chunqiang, "Multidimensional packet classification with improved cutting," *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, China, 2014, pp. 409-413, doi: 10.1109/ICNIDC.2014.7000335.

[12] C. Linan, L. Zhaowen, M. Yan, H. Xiaohong and L. Chunqiang, "Multidimensional packet classification with improved cutting," *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, China, 2014, pp. 409-413, doi: 10.1109/ICNIDC.2014.7000335.

[13] M. Meshram, S. Kakde, Y. Suryawanshi and Y. Deodhe, "FPGA implementation of modular architecture for packet classification using field split algorithm," *2015 International Conference on Communications and Signal Processing (ICCSP)*, Melmaruvathur, India, 2015, pp. 1098-1101, doi: 10.1109/ICCSP.2015.7322672.

[14] A. U. Khan, Y. Suryawanshi, M. Chawhan and S. Kakde, "Design and implementation of high performance architecture for packet classification," *2015 International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015, pp. 598-602, doi: 10.1109/ICACEA.2015.7164761.

[15] M. S. K. Mohan and J. S. Jayasudha, "Multimatch packet classification schemes," *2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS)*, Bangalore, India, 2016, pp. 1-4, doi: 10.1109/RAINS.2016.7764413.

[16] S. Zheng, X. Bi and J. Luo, "An Efficient Total Prefix Length-Based Clustering Packet Classification Algorithm," *2016 International Conference on Network and Information Systems for Computers (ICNISC)*, Wuhan, 2016, pp. 46-49, doi: 10.1109/ICNISC.2016.020.

[17] S. Yingchareonthawornchai, J. Daly, A. X. Liu and E. Torng, "A Sorted-Partitioning Approach to Fast and Scalable Dynamic Packet Classification," in *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1907-1920, Aug. 2018, doi: 10.1109/TNET.2018.2852710.

[18] J. Huang, Y. Lu and K. Guo, "A Hybrid Packet Classification Algorithm Based on Hash Table and Geometric Space Partition," *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, Hangzhou, China, 2019, pp. 587-592, doi: 10.1109/DSC.2019.00095.

[19] F. Saqib, A. Dutta, J. Plusquellic, P. Ortiz and M. S. Pattichis, "Pipelined Decision Tree Classification Accelerator Implementation in FPGA (DT-CAIF)," in *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 280-285, Jan. 2015, doi: 10.1109/TC.2013.204.

[20] A. Kennedy and X. Wang, "Ultra-High Throughput Low-Power Packet Classification," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 286-299, Feb. 2014, doi: 10.1109/TVLSI.2013.2241798.

[21] W. Jiang and V. K. Prasanna, "Large-scale wire-speed packet classification on FPGAs," in *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays*, California, USA, 2009, pp. 219-228, doi: 10.1145/1508128.1508162.

[22] Z. Tao, W. Yonggang, Z. Lijun, and Y. Yang, "High throughput architecture for packet classification using FPGA," In *Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*, pp. 62-63, 2009.

[23] W. Yu, S. Sivakumar and D. Pao, "Pseudo-TCAM: SRAM-Based Architecture for Packet Classification in One Memory Access," in *IEEE Networking Letters*, vol. 1, no. 2, pp. 89-92, Jun. 2019, doi: 10.1109/LNET.2019.2897934.

[24] C. Yeim-Kuan and Han-Chen Chen, "Fast packet classification using recursive endpoint-cutting and bucket compression on FPGA," *The Computer Journal*, vol. 62, no. 2, pp. 198-214, 2019, doi: 10.1093/comjnl/bxy052.

[25] Anita P. and M. Devi, "Design of a Low Latency and High Throughput Packet Classification Module on FPGA Platform," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, pp. 1468-1474, 2020.

[26] T. Ganegedara, W. Jiang and V. K. Prasanna, "A Scalable and Modular Architecture for High-Performance Packet Classification," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1135-1144, May 2014, doi: 10.1109/TPDS.2013.261.

[27] C. A. Zerbini and J. M. Finochietto, "Performance evaluation of packet classification on FPGA-based TCAM emulation architectures," *2012 IEEE Global Communications Conference (GLOBECOM)*, Anaheim, CA, USA, 2012, pp. 2766-2771, doi: 10.1109/GLOCOM.2012.6503535.

[28] S. Haoyu and J. W. Lockwood, "Efficient packet classification for network intrusion detection using FPGA," in *Proceedings of the 2005 ACM/SIGDA 13th international symposium on Field-programmable gate arrays*, California, USA, 2005, pp. 238-245, doi: 10.1145/1046192.1046223.

[29] D. E. Taylor and J. S. Turner, "Scalable packet classification using distributed crossproducing of field labels," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, vol. 1, 2005, pp. 269-280, doi: 10.1109/INFCOM.2005.1497898.

BIOGRAPHIES OF AUTHORS

Anita P. is a research scholar in the department of ECE at The Oxford College of Engineering Bangalore. She had worked as an assistant professor at CMRIT, Bangalore. She obtained her B.E (Electronics and Communication Engineering) degree in 2002 from (GVIT) Bangalore University and M.Tech degree in VLSI design in the embedded system from CMRIT. She has nine years of academic teaching experience and worked for both the NBA and NAAC. She published 5 international journal papers. Her areas of interest are VLSI design, Analog, and Digital Electronics.

Manju Devi is working as a Professor and Head in the ECE's department at The Oxford College of Engineering Bangalore. She has worked as Vice-Principal and professor at BTLIT, Bangalore. She obtained her B.E (Electronics and Communication Engineering) degree in 1996 from Anna University, M.Tech degree in Applied Electronics from BMSCE, and PhD from Visvesvaraya Technological University (VTU), Karnataka. She has almost twenty-two years of academic teaching experience and worked for both the NBA and NAAC. She has almost 85 publications in international conferences and journals. She is guiding eight students from Visvesvaraya Technological University (VTU), Karnataka. Her areas of interest are VLSI design, analog and mixed-mode VLSI design, and digital electronics.