

Research Article

PackeX: Low-Power High-Performance Packet Classifier Using Memory on FPGAs

Khalid Rehman ¹ and **Zahid Ullah** ²

¹Department of Electrical Engineering, CECOS University of IT and Emerging Sciences, Peshawar, Pakistan

²Department of Electrical and Computer Engineering, Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan

Correspondence should be addressed to Khalid Rehman; khalid@cecos.edu.pk

Received 23 January 2021; Accepted 24 May 2021; Published 7 June 2021

Academic Editor: Sungchang Lee

Copyright © 2021 Khalid Rehman and Zahid Ullah. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Networks are continuously growing, and the demand for fast communication is rapidly increasing. With the increase in network bandwidth requirement, efficient packet-classification techniques are required. To achieve the requirements of these future networks at component level, every module such as routers, switches, and gateways needs to be upgraded. Packet classification is one of the main characteristics of a stable network which differentiates the incoming flow into defined streams. Existing packet classifiers have lower throughput to cope with the higher demand of the network. In this work, we propose a novel high-speed packet classifier named as PackeX that enables the network to receive and forward the data packets in a simplest structure. A size of 128-rule 32-bit is successfully implemented on Xilinx Virtex-7 FPGA. Experimental findings show that our proposed packet classifier is versatile and dynamic compared to the current FPGA-based packet classifiers achieving a speed of 119 million packets per second (Mpps), while consuming 53% less power compared with the state-of-the-art architectures.

1. Introduction

Network devices in their early stages lack packet classification because of the limited application, but the use of Internet is now a multiservice ecosystem; a classifier is one of the main components of the whole networking system [1–3]. It enables the network to deploy service classification, i.e., security, quality of service (QoS), multimedia communications, and monitoring, and distinguish different network traffic from each other [4]. Some simple techniques are also applied to a router to decide if packets should be forwarded or dropped in order to prevent network infrastructure from being violated. Generally, rules are kept relatively static in conventional implementations, so fast classification can be accomplished with the algorithms running over the constructed classifier's well-designed data structure [5]. The primary objective of classifier design has been to perform high-speed packet processing in the past, such as content detection, load balancing, and packet filtering. The classifier can

be installed offline as the rule update is rare at that moment. These new applications can respond simultaneously to a wide range of requests from different users, so that the classifier must be updated regularly in order to satisfy different requirements. Standard network migration operations modify the topology of the network, and the procedure adjusts the classifier accordingly [5, 6]. This allows the sorting of packets to be carried out electronically with the help of updating the fast dynamic policy, which is undoubtedly a necessary prerequisite for current and future classifiers.

The introduction of software-defined networking (SDN) [7–9] provides immense potential to support innovative functionality and value-added functionality for network growth. This includes traffic engineering support [10], network function virtualization (NFV) [11, 12], and high-performance cloud computing [13, 14]. Field-programmable gate arrays (FPGAs) due to their immense level of hardware parallelism are becoming extremely famous for the implementation of high-speed networks [2, 15, 16].

Software-defined networks (SDN), traffic engineering, and network function virtualization (NFV) as the next-generation networks are providing the most flexible networks. FPGAs fit completely to its requirement of reconfigurability and flexibility [17, 18].

Figure 1 shows a generalized structure of a packet classifier that helps the network to forward the incoming packet into the corresponding node or hop based on its destination address. It uses content-addressable memory (CAM) to store the destination addresses (IP addresses in the packet) and random-access memory (RAM) to store the next node where the packet needs to be transferred. There are basically many types of packet classifiers but the one we cover in this paper is based on the destination address and the next-hop [19]. A next-hop packet classifier consists of RAM and CAM. RAM performs searches using a memory address and then returns the data from the address [20]. The CAM-based search does the opposite. A function calls the CAM by passing a key that consists of a data word structure, and the CAM search returns a memory address. CAM further differentiates itself from different types of memory that it can perform memory searches in a single clock cycle. CAM can be a binary CAM or a ternary CAM depending on the requirement of the application [21]. However, other classifiers use other techniques to classify the packet from the incoming stream.

2. Motivation

Existing packet classifiers on FPGAs have the performance bottlenecks and cannot achieve higher throughput as required by the high-speed networks [17]. Thus, a high-speed (high throughput) packet classifier is needed to achieve the demand of the next-generation networks while not compromising on power. The novelty of our proposed architecture, PackeX, is its minimalism without compromising the high throughput requirement of the network to classify the incoming traffic. The proposed architecture provides higher speed packet classification by not compromising on the throughput of the network and consumes less power as compared to the existing packet-classification architectures.

3. Key Contributions

The key contributions of our proposed packet-classification architecture are as follows:

- (i) The proposed packet classifier, PackeX, is state-of-the-art architecture to classify the network packets compared to the existing architectures
- (ii) The proposed PackeX consumes half the power as compared to the state-of-the-art architecture
- (iii) PackeX process the incoming data packets at a rate of 119 Mega packets per second (Mpps) using the distributed RAM on target FPGA
- (iv) The proposed architecture is scalable and dynamically reconfigurable compared to the existing

state-of-the-art TCAM-based packet-classification architectures

The rest of the paper is organized in the following way: Section 4 addresses the related work. Section 5 explains the proposed PackeX classification system and the proposed architecture. The pipelining of the architecture proposed is defined in Section 6. The outcomes of the deployment and performance assessment of our proposed architecture are addressed in Section 7. Section 8 concludes the paper.

4. Related Work

Hardware-based packet classifiers can be divided into three main types: decision-tree, exhaustive search, and decomposition. Decision-tree classifiers have large hardware requirements and the time to process the defined rule set. Our proposed architecture has lower hardware resource requirement and is implementable on FPGAs, providing high performance compared to the state-of-the-art packet classifiers which is shown in Table 1.

Zhang and Zhou reported a scheme based on reducing the TCAM memory usage [22]. This code based on split function was proposed in this study. Firstly, the splitting of d-tuple rule set into d-tuple field takes place, followed by obtaining a unique field test for each dimension. Based on matching with the incoming packet, its storage in SRAM memory and indexing in TCAM using a concatenated field occurs. On the other side, our proposed packet classifier uses a novel and simple architecture to develop the packet addresses into a TCAM that is partitioned to store in the distributed memory of the target FPGA. This improves the throughput compared to the available state-of-the-art packet-classification algorithms.

Using a Multimatch Using Discrimination (MUD) approach, Lakshminarayanan et al. utilized the extra bits of TCAM entry for encoding it, followed by keeping the encoded value in the same TCAM entry [23]. Evidently, these multiple lookup cycles are one of the major disadvantages for high-speed networks. This fact has been explained in detail in the packet classification section of this article. Although the multiple lookup cycles consume a considerable amount of power, nevertheless, the individual power values of PackeX remain low owing to its simple structure [24].

Another packet classifier based on TCAM has been proposed in [25]. This packet classifier reduces the width of TCAM to 36 bits, which leads to reduce the space requirement for TCAM. Consequently, the storage of set rules takes place in the SRAM. However, due to field code formation and indexing mechanisms, this packet classifier requires large memory. In fact, the scattered and sparse SRAM array leads to wasting the majority of SRAM entries. PackeX uses the distributed RAM which is partitioned to reduce the RAM usage and provide the high-speed searching to achieve the high-speed classification of the incoming data packets.

Updating of the packet classifier is also an important aspect of the design, which sometimes becomes the bottleneck for dynamic networks. TCAM update modules are developed to speed up the updating of the searching module

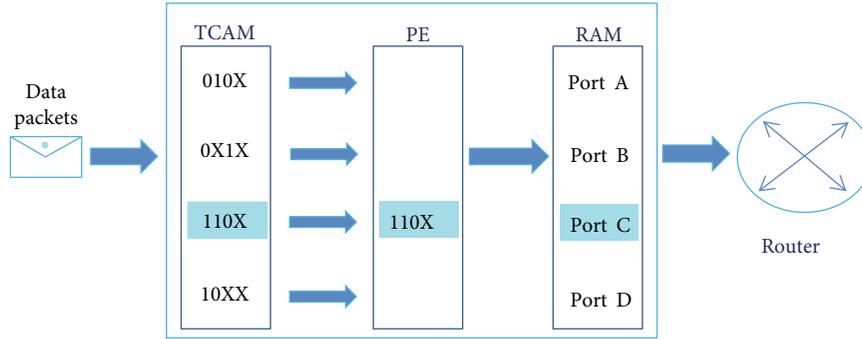


FIGURE 1: Generalized structure of packet classifier.

TABLE 1: Performance comparison of PackeX with the related work.

Packet classification engine	Platform	Memory/resource utilization	Power consumption (mW)	Throughput (Mpps)
Zhang et al.	FPGA	Maximum M/R utilization	1120	7 Mpps
Lakshminarayanan et al.	FPGA	Maximum M/R utilization	—	20 Mpps
Wang et al.	FPGA	Maximum M/R utilization	1200	—
Shen et al.	FPGA	Maximum M/R utilization	1900	—
Inayat et al.	FPGA	Average M/R utilization	33.72	20 Mpps
Ullah et al.	FPGA	Average M/R utilization	188	9 Mpps
Irfan et al.	FPGA	Average M/R utilization	—	33 Mpps
PackeX	FPGA	Minimum M/R utilization	17	119 Mpps

of packet classifier which is constantly improving and depends on the type of RAM used in FPGA [26, 27]. BRAM and distributed RAM have different update latencies based on their available depth in the target FPGA. We have chosen the distributed RAM which provides the minimum clock cycle update time which makes our packet classifier unique in its updating process. This makes PackeX dynamic as well as faster to support the modification of the network to mitigate the incoming traffic from time to time.

Lookup tables (LUTs) were used instead of memory blocks by Khatami and Ahmadi Partial reconfiguration in field-programmable gate array (FPGA) is used to shorten the time required to adjust the actions of an architecture. The pipeline system, on the other hand, results in unbalanced memory sharing, resulting in poor throughput and inefficient source allocation [28]. PackeX employs a very basic framework and a pipelining scheme, resulting in fast throughput and optimal use of source allocation.

Aceto et al. proposed a scheme of MIMETIC that can handle capitalized heterogeneity data traffic by using both inter- and intramodalities. The scheme outperforms the single modality-based techniques by supporting more challenging mobile traffic scenarios. MIMETIC uses three datasets to validate the performance improvement over the fusion classifiers, ML-based traffic classifiers, and single modality DL-based schemes [29]. The authors in [30] applied deep learning (DL) algorithm for packet classification on mobile traffic. The algorithm is centered on feature extraction, capable of running with encrypted and complex data traffic. It outperforms the ML state of art technique

and previous deep learning-based algorithms by improving F -measure. On the other hand, our proposed packet classifier employs a novel and easy design to convert the packet addresses into a TCAM partitioned for storage in the target FPGA's distributed memory.

The development of binary and ternary CAM also involves the packet classifier because it is the vital component in classifying the address from the stored address in a little time [31]. Thus, Bi-CAMs and TCAMs include the partitioning of the corresponding CAM, binary to ternary conversion of the storage cells, and pipelining of the internal signals to improve the throughput. The proposed packet classifier uses the idea of partitioning taken from HP-TCAM in [15] and pipelining taken from D-TCAM in [16] to develop a novel structure for packet classification in a fastest way. We thus attain a higher speed in terms of millions of packets per second, which is the best and fastest packet classifier to the best of our understanding, while consuming 40% to 53% less power compared to the state-of-the-art architectures.

5. Proposed Architecture

5.1. Terminology. Table 2 shows the notations that are used to describe the proposed packet-classification design (PackeX).

5.1.1. PackeX Internal Structure. PackeX internal structure is based on the distributed RAM available in modern FPGAs which implements the TCAM part of our design to store the network addresses. There are two major components in PackeX:

TABLE 2: Terminologies or notations.

Notations	Explanation
A_D	Destination address
W	Number of bits to each sub-TCAM. For 6-input, lookup tables (LUTs) w is 6
W	Number of bits in the destination address (A_D)
$D_T \times W_T$	Size of TCAM, where D_T represents depth of TCAM and W_T represents width of TCAM
$D_R \times W_R$	Size of RAM of RAM, where D_R represents depth of TCAM and W_R represents width of TCAM

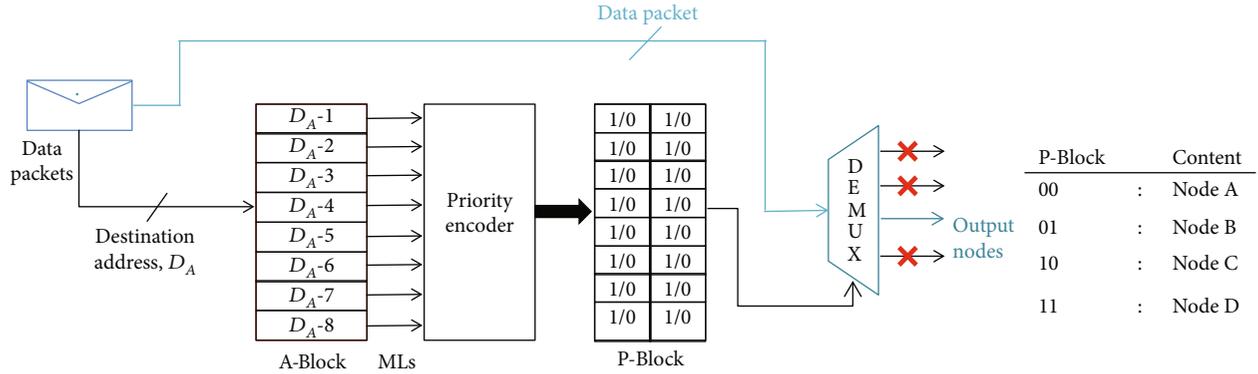


FIGURE 2: Structure of packet classifier, PackeX.

- (i) A-block: a TCAM that stores the addresses of the incoming packets. These addresses in this paper are considered to be destination addresses. However, it can also be a source of addresses by using PackeX as filters for data packets from which they come and determine whether or not to drop or forward the packet. This block is known as A-block (address block)
- (ii) P-block: a RAM that stores the information (pointers) for the corresponding nodes. It is known as P-block (pointers block)

Other blocks/components include demultiplexer, priority encoder or a simple encoder, ANDing modules, and connecting circuitry.

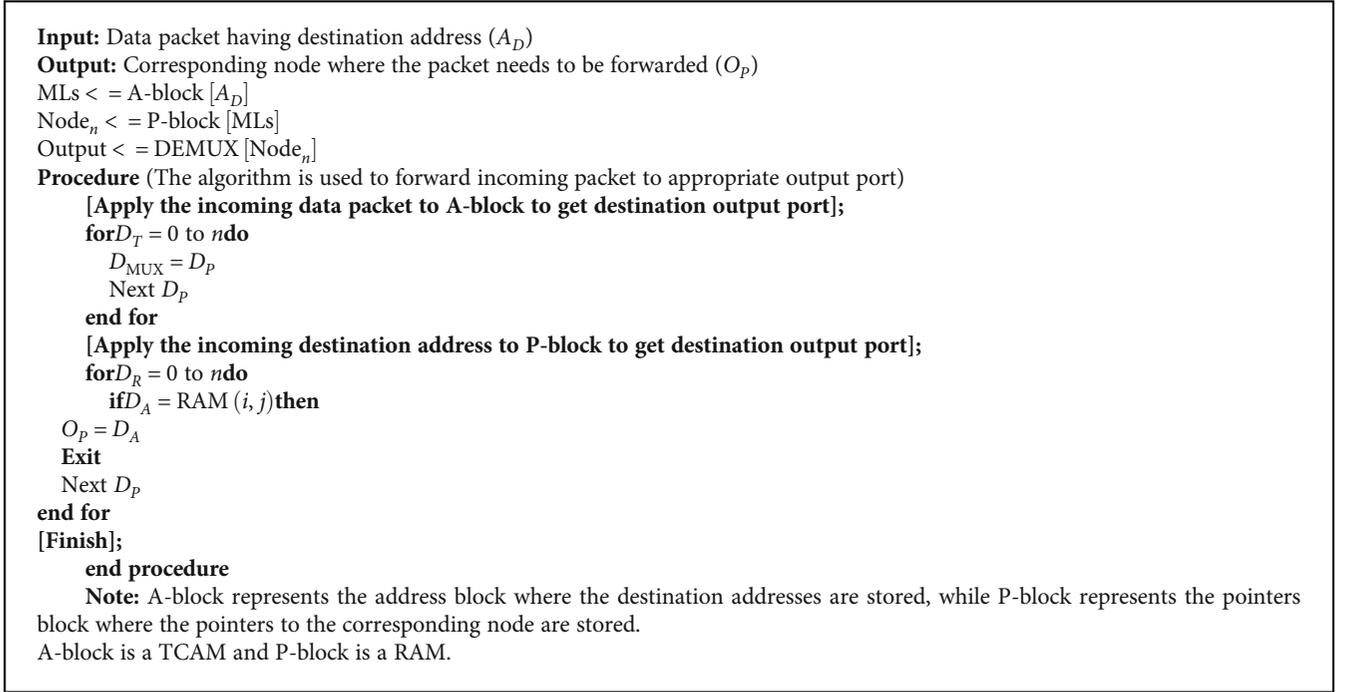
Figure 2 shows the internal structure of PackeX. The destination address (D_A) from the incoming packet is searched/compared with the stored addresses in A-block. One or more of the stored entries are matched with the input D_A and generates the corresponding Match-Lines (MLs). Priority encoder (PE) converts the Match-Lines into an address for P-block which provides the values of the corresponding node. For example, “00” represents node A, “01” represents node B, and so on. The packet is forwarded according to the node specification provided by P-block using a demultiplexer (DEMUX). A-block and P-block are the main blocks storing the address information and node information. The size of demultiplexer (DEMUX) depends on the number of nodes in the network. Here, the nodes are 4 which need a DEMUX of size 1 to 4.

5.2. *Classification Procedure.* The two basic blocks involved in the classification process are A-block and P-block. Algorithm 1 shows the procedure of the data flow. The destination addresses lead to the Match-Lines which find out the node, and the demultiplexer (DEMUX) activates only one channel out of many channels using the content stored in the P-block. The size of DEMUX is determined by the number of nodes that are supported by PackeX. For instance, 4-node PackeX require 1:4 DEMUX while the size of P-block depends on the number of destination addresses stored in the A-block.

5.3. *Partitioning of TCAM Module.* The TCAM part of PackeX has 6-bit sub-TCAM modules. Multiple sub-TCAM modules combine to form the complete TCAM component of the proposed packet classifier, as shown in Figure 3. The use of 6-bits is due to the built-in structure of distributed RAM inside modern FPGAs that are based on the 6-bit lookup tables (LUTs) also known as LUTRAMs. It holds the destination addresses of the packets that are received by PackeX for classification.

TCAM submodules of 6-bits each forms the TCAM module that stores the destination addresses of the incoming data packets to the packet classifier. The symbol W represents the total size of the stored addresses while the symbol w is 6 because of the 6-input LUTs available in the target FPGA.

The numbers of required sub-TCAM modules are determined from the required size of the addresses that need to be stored in the proposed packet classifier. The ideal addresses are always multiples of 6, i.e., 6, 12, 18, and so on. If the size of addresses/rules is 6 bits, the required sub-TCAM modules



ALGORITHM 1: The proposed packet-classification algorithm (PackeX).

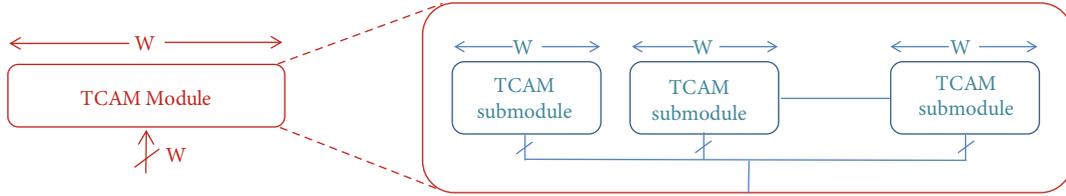


FIGURE 3: TCAM submodules.

are 1. If the size of addresses/rules is 12 bits, the required sub-TCAM modules are 2, and so on.

6. Pipelining

Performance of digital design degrades when the shortest path from input to output is larger as it reduces the speed of the whole circuit. Pipelined registers are introduced in stages to the digital design in order to reduce the path and improve the throughput. This is done recently by D-TCAM [16] which has shown improvement by incorporating flip-flops (FFs) to each of the distributed RAM that improved the throughput of TCAM on FPGA. We took the pipelining procedure from D-TCAM [16] to enhance the throughput of our system which is degraded by the ANDing circuitry, P-block, encoder, and demultiplexer. The proposed packet classifier achieves a high speed in classifying the incoming packets with 119 Mega packets per second (Mpps), while consuming 40 to 53% less power.

6.1. Example Design. The proposed packet classifier is scalable based on the available hardware resources on FPGA. To understand the basic architecture, we explain the design of 8 destination addresses and 4 nodes. The size of P-block

depends on the number of nodes and destination addresses, such that

$$W_R(\text{Width of RAM}) = \log_2(\text{Number of Nodes}),$$

$$D_r(\text{Depth of RAM}) = \text{Number of Destination Addresses.} \quad (1)$$

Thus, the size of RAM required for P-block is 8×2 , as shown in Figure 2. Priority encoder (PE) and demultiplexer (DEMUX) are 8:3 and 1:4, respectively.

6.2. Why an FPGA-Based Packet Classifier? Reconfigurability and high performance can easily put to work. As described in Section 4, packet classifiers are hardware and software-based. Hardware-based packet classifiers are faster but mostly fixed. FPGAs are reconfigurable according to the requirement of the system. Building a packet-classifier on FPGA gives us hardware-like performance (speed) and dynamic nature (reconfigurability) of the whole system. If the network grows in size, so is the packet classifier, because of the reconfigurable blocks available on FPGAs. Thus, our proposed architecture for packet classification, PackeX, has a simple structure

TABLE 3: Implementation results on Xilinx Virtex-7 FPGA for PackeX.

Approaches	No. of addresses/rules	Platform	No. of SRs	No. of LUTs	Throughput (Mpps)	Power (mW)
PackeX-I	64	Virtex-7	403	616	71	11
PackeX-II	128	Virtex-7	770	1095	119	17

Mpps: Mega packets per second; mW: milliWatt; SR: Slice Register; LUT: lookup table.

achieving improvement in important features, i.e., speed, power reduction, and reconfigurability.

7. Implementation

We have successfully implemented PackeX on Xilinx Virtex-7 FPGA using Xilinx Vivado 2018.2. The FPGA device xc7vx690tffg1157 is used with speed grade -2. Table 3 shows the implementation results for 64-rule 36-bit and 128-rule 32-bit packet classifier which classifies the incoming packets at a speed of 71 and 119 million packets per second (Mpps) and power consumption of 11 mW and 17 mW, respectively, compared to the state-of-the-art packet classifier.

Most importantly, the proposed design is purely a hardware-based design which eliminates the deficiencies of a software-based packet classifier. However, the proposed design is reconfigurable just like software-based design. Thus, PackeX combines the effective and useful properties of both software (i.e., reconfigurability) and hardware (i.e., speed). The experimental results show the feasibility and scalability of PackeX for future software-defined networks by its ability of adapting structure according to the need to the underlying application.

Table 4 shows the hardware utilization of the proposed packet classifier. The utilization of logical resources on the target FPGA is using under 1% of the available hardware except in case of the input/output (I/O) pins. The I/Os can be further reduced to optimize the performance of the system in future work. The lookup tables (LUTs) are 0.25%, LUTRAMs are 0.44%, and Slice Registers (SRs) are 0.09% of the available resources which show the scalability of our proposed architecture. Higher sizes of the packet classifier can be implemented on PackeX.

PackeX receives data from the server, classifies according to the stored addresses/rules, and forwards to the corresponding node. A node can be anything including but not limited to mobile phone, laptop, computer, and another server. Only one of the connected channels is activated to transfer the processed packet.

To implement the P-block, there are many options in modern FPGAs, i.e., single port distributed RAM, single port block RAM, and single port ROM. In our proposed design, the numbers of nodes are fixed to node 1, 2, $N - 1$, and N as shown in Figure 4.

We use single port ROM to store the content of P-block which provides the information about the type of node to which the packet needs to be forwarded. It can also be implemented using RAM whenever the system needs dynamic number of nodes in the network.

Table 5 presents the comparative analysis of various approaches in terms of metrics such as dynamic power consumption, latency, and FPGA resource utilization. A com-

TABLE 4: Implementation results of hardware utilization.

S.No.	Resources	Utilization	Available	Utilization (%)
1	LUT	1095	433200	0.25
2	LUTRAM	768	174200	0.44
3	SR	770	866400	0.09

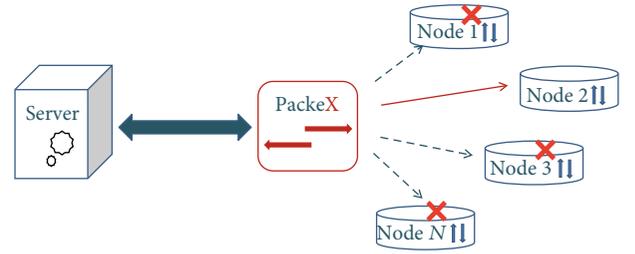


FIGURE 4: PackeX receives data from the server, classifies according to the stored addresses/rules, and forwards to the corresponding node.

parison of the proposed PackeX-TCAM architecture with the current state-of-the-art FPGA-based TCAMs in terms of power consumption output with the current state-of-the-art FPGA-based TCAMs [36] is given by equation (2).

$$\frac{\text{Normalized Power Consumption } [mW/D_R]}{\text{Performance}}. \quad (2)$$

Figure 5 presents the comparative analysis of logical resource utilization of existing state-of-the-art TCAM-based packet-classification architectures implemented on FPGA. As shown, compared with other TCAM-based classifiers, our proposed PackeX classifier is utilizing efficient logical resources.

In addition, Figure 6 represents the power ranking comparison of various TCAM-based architectures. It is envisaged from the figure that the power consumption for G-AETCAM [32], RE-TCAM [33], CLB-TCAM [34], and REST [35] is 36, 28, 22.23, and 161.18, respectively. Conversely, the power consumption for PackeX-I and PackeX-II is 11 and 17 mW, respectively.

Hence, the utilization and less number of resources for PackeX-I and PackeX-II approach lead to lower power consumption ranking without compromising on throughput.

8. Conclusions

Packet classification is a key operation needed in provisioning several important network services. One of the major challenges in designing the next-generation high-speed switches is to deliver high-speed low-power packet

TABLE 5: Power consumption of PackeX with RE-TCAM [32], G-AETCAM [33], CLB-TCAM [34], and REST [35].

Approaches	Platform	TCAM size ($D \times W$)	No. of SRs	No. of LUTs	Latency	Power (mW)
G-AETCAM [32]	Virtex-6	64×36	4672	3067	2	36
RE-TCAM [33]	Virtex-6	64×36	2337	2513	2	28
CLB-TCAM [34]	Virtex-6	64×36	4750	1472	1	22.23
REST [35]	Kintex-7	72×28	390	130	5	161.18
PackeX-I	Virtex-7	64×36	403	616	2	11
PackeX-II	Virtex-7	128×32	770	1095	2	17

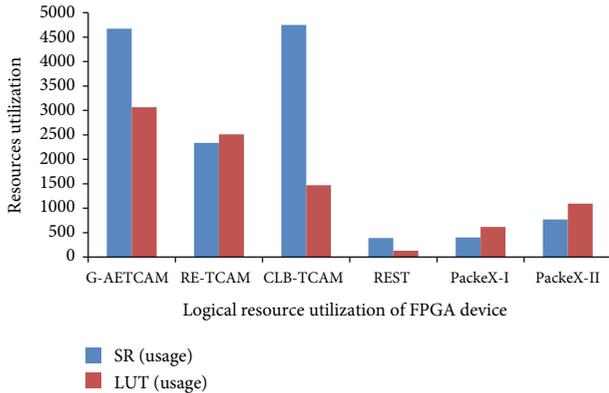


FIGURE 5: Comparison of resource utilization of FPGAs with PackeX.

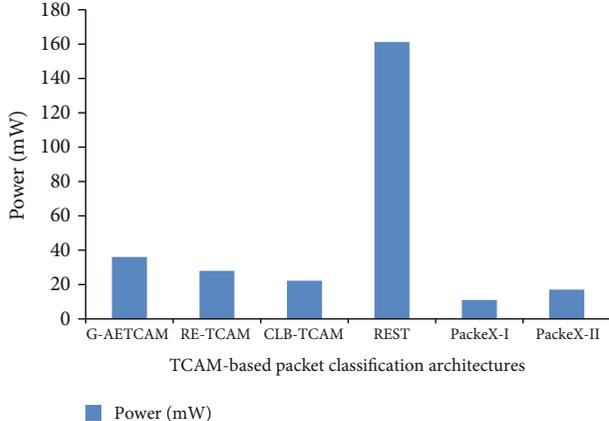


FIGURE 6: Power consumption ranking of various TCAM-based architectures.

classification. TCAM-based labeling of packets is a de facto standard for the high-performance processing of packets. However, due to inherent parallel structure, the high cost and high energy consumption are the major challenges of its efficient usage/implementation. TCAMs are the dominant industry standard used for multibit classifiers. However, as packet classification policies grow thorough and complex, a fundamental tradeoff arises between the TCAM space and the number of addresses for hierarchical policies.

This paper proposes a novel algorithm based on dynamic programming for solving important problems concerned with packet classification. PackeX presents a high throughput

in packet forwarding as compared to the existing algorithms with less power consumption. The proposed packet-classification design classifies the incoming packets with simplicity and provides high throughput and consumed less power compared to the existing algorithms. The PackeX design uses fewer hardware resources. It is reconfigurable when the network needs modification, making PackeX favorable to future defined networks. Our algorithms do not require any change to existing packet classification systems and can be easily deployed. As far as we know, this is the first work to study TCAM speed and memory optimization for packet classification.

In our current work, PackeX is deployed for fast mapping and updating algorithms to eliminate the complication of sequential steps in generating lookup tables (LUTs) and iterative procedure for calculating the required address of the packet. The future works may include deployment of submodules at such locations that can balance the energy consumption and speedup the process. This may be possible by using some statistical distribution for the deployment of modules with respect to the horizontal partitioning and virtual partitioning. In addition, the use of proper pipelining and partitioning can also be investigated in our proposed algorithm in the future.

Data Availability

The data that support the study's outcomes are all briefly introduced, and all information is included in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] H. T. Lin and P. C. Wang, "Fast TCAM-based multi-match packet classification using discriminators," *IEEE Transactions on Multi-Scale Computing Systems*, vol. 4, no. 4, pp. 686–697, 2018.
- [2] E. Norige, A. X. Liu, and E. Torng, "A ternary unification framework for optimizing TCAM-based packet classification systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 2, pp. 657–670, 2018.
- [3] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: a tutorial and

- survey,” *IEEE Journal of Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, 2006.
- [4] D. E. Taylor, “Survey and taxonomy of packet classification techniques,” *ACM Computing Surveys (CSUR)*, vol. 37, no. 3, pp. 238–275, 2005.
 - [5] T. Shen, G. Xie, X. Wang et al., “RVH: range-vector hash for fast online packet classification,” *arXiv preprint arXiv*, vol. 1909, 2019.
 - [6] S. Yingchareonthawornchai, J. Daly, A. X. Liu, and E. Torng, “A sorted partitioning approach to high-speed and fast-update OpenFlow classification,” in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pp. 1–10, IEEE, 2016.
 - [7] L. Molnár, G. Pongrácz, G. Enyedi et al., “Dataplane specialization for high-performance OpenFlow software switching,” in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 539–552, Florianopolis, Brazil, 2016.
 - [8] A. Sivaraman, A. Cheung, M. Budiú et al., “Packet transactions: high-level programming for line-rate switches,” in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 15–28, Florianopolis, Brazil, 2016.
 - [9] B. Li, K. Tan, L. Luo et al., “Clicknp: highly flexible and high performance network processing with reconfigurable hardware,” in *Proceedings of the 2016 ACM SIGCOMM Conference*, pp. 1–14, Florianopolis, Brazil, 2016.
 - [10] S. Jain, A. Kumar, S. Mandal et al., “B4,” *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.
 - [11] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, “Nfp: enabling network function parallelism in nfv,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pp. 43–56, Los Angeles, CA, USA, 2017.
 - [12] P. Zave, R. A. Ferreira, X. K. Zou, M. Morimoto, and J. Rexford, “Dynamic service chaining with dysco,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pp. 57–70, Los Angeles, CA, USA, 2017.
 - [13] D. Firestone, “VFP: a virtual switch platform for host {SDN} in the public cloud,” in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pp. 315–328, Boston, MA, USA, 2017.
 - [14] C. Lan, J. Sherry, R. A. Popa, S. Ratnasamy, and Z. Liu, “Embark: securely outsourcing middleboxes to the cloud,” in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pp. 255–273, Santa Clara, CA, USA, 2016.
 - [15] Z. Ullah, K. Ilgon, and S. Baeg, “Hybrid partitioned SRAM-based ternary content addressable memory,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 12, pp. 2969–2979, 2012.
 - [16] M. Irfan, Z. Ullah, and R. C. Cheung, “D-TCAM: a high-performance distributed ram based TCAM architecture on FPGAs,” *IEEE Access*, vol. 7, pp. 96060–96069, 2019.
 - [17] C. Li, T. Li, J. Li, D. Li, H. Yang, and B. Wang, “Memory optimization for bit-vector-based packet classification on FPGA,” *Electronics*, vol. 8, no. 10, p. 1159, 2019.
 - [18] A. Ullah, A. Zahir, N. A. Khan, W. Ahmad, A. Ramos, and P. Reviriego, “BPR-TCAM—block and partial reconfiguration based TCAM on Xilinx FPGAs,” *Electronics*, vol. 9, no. 2, p. 353, 2020.
 - [19] F. Baboescu, S. Singh, and G. Varghese, “Packet classification for core routers: is there an alternative to CAMs? IEEE INFOCOM 2003,” in *Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 1, pp. 53–63, IEEE, 2003.
 - [20] J. Van Lunteren and A. Engbersen, “Multi-field packet classification using ternary CAM,” *Electronics Letters*, vol. 38, no. 1, pp. 21–23, 2002.
 - [21] Y. D. Kim, H. S. Ahn, S. Kim, and D. K. Jeong, “A high-speed range-matching TCAM for storage-efficient packet classification,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, pp. 1221–1230, 2008.
 - [22] Z. Zhang and M. Zhou, *A code-based multi-match packet classification with TCAM*, In *Advances in Web and Network Technologies, and Information Management*, Springer, 2007.
 - [23] K. Lakshminarayanan, A. Rangarajan, and S. Venkatachary, “Algorithms for advanced packet classification with ternary CAMs,” *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 193–204, 2005.
 - [24] M. Abbasi, N. Mousavi, M. Rafiee, M. R. Khosravi, and V. G. Menon, “A CRC-based classifier micro-engine for efficient flow processing in SDN-based internet of things,” *Mobile Information Systems*, vol. 2020, 8 pages, 2020.
 - [25] R. Shen, X. Li, and H. Li, “A hybrid TCAM+ SRAM scheme for multi-match packet classification,” in *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pp. 685–690, IEEE, 2012.
 - [26] Z. Wang, H. Che, M. Kumar, and S. K. Das, “CoPTUA: consistent policy table update algorithm for TCAM without locking,” *IEEE Transactions on Computers*, vol. 53, pp. 1602–1614, 2004.
 - [27] T. Banerjee-Mishra and S. Sahní, “Consistent updates for packet classifiers,” *IEEE Transactions on Computers*, vol. 61, pp. 1284–1295, 2011.
 - [28] R. I. Khatami and M. Ahmadi, “High throughput multi pipeline packet classifier on FPGA,” in *InThe 17th CSI International Symposium on Computer Architecture & Digital Systems (CADSD 2013)*, pp. 137–138, IEEE, 2013.
 - [29] G. Aceto, D. Ciunzo, A. Montieri, and A. Pescapè, “MIMETIC: mobile encrypted traffic classification using multimodal deep learning,” *Computer Networks*, vol. 165, p. 106944, 2019.
 - [30] G. Aceto, D. Ciunzo, A. Montieri, and A. Pescapè, “Toward effective mobile encrypted traffic classification through deep learning,” *Neurocomputing*, vol. 409, pp. 306–315, 2020.
 - [31] I. Ullah, Z. Ullah, and J. A. Lee, “EE-TCAM: an energy-efficient SRAM-based TCAM on FPGA,” *Electronics*, vol. 7, no. 9, p. 186, 2018.
 - [32] M. Irfan and Z. Ullah, “G-AETCAM: gate-based area-efficient ternary content-addressable memory on FPGA,” *IEEE Access*, vol. 5, pp. 20785–20790, 2017.
 - [33] H. Mahmood, Z. Ullah, O. Mujahid, I. Ullah, and A. Hafeez, “Beyond the limits of typical strategies: resources efficient FPGA-based TCAM,” *IEEE Embedded Systems Letters*, vol. 11, pp. 89–92, 2018.
 - [34] I. Ullah, U. Afzaal, Z. Ullah, and J. A. Lee, “High-speed configuration strategy for configurable logic block-based TCAM architecture on FPGA,” in *2018 21st Euromicro Conference on Digital System Design (DSD)*, pp. 16–21, IEEE, 2018.

- [35] A. Ahmed, K. Park, and S. Baeg, "Resource-efficient SRAM-based ternary content addressable memory," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 1583–1587, 2016.
- [36] H. Nakahara, T. Sasao, H. Iwamoto, and M. Matsuura, "LUT cascades based on edge-valued multi-valued decision diagrams: application to packet classification," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 1, pp. 73–86, 2016.