An Analysis on High-performance and Dynamically Updatable Packet Classification Engine on FPGA through Literature Survey

Mona Priya[1] & Prof. Ashish Raghuwanshi[2]

[1]M-Tech Research Scholar, [2]Research Guide Department of Electronics & Communication Engineering,
IES, Bhopal

Abstract: - Packet classification is a main requirement in routers to manage network security and traffic. In high speed networks packet classification in line rates has become a major challenge. The design mainly benefits from a parallel pipelined architecture implemented on field programmable gate arrays (FPGA) to achieve high speed packet processing Recently, the Internet has been used for real time applications such as video conferencing, HD video streaming over web and other voice over Internet Protocol (IP) processes that require certain delay guarantees. This state of the Internet leads internet service providers (ISP) to offer better service for real time applications. Hence, quality of service (QoS) levels in terms of delay and jitter are defined for different types of packets. In order to achieve predefined timing limitations, routers have to manage resources..

Index Terms—Packet classification, Field-Programmable Gate Array (FPGA), 2-dimensional pipeline, dynamic updates.

I. INTRODUCTION

Packet Classification

Packet scheduling and buffer management algorithms to decide which packet is served with priority, admission control algorithms to limit the load in a system to preserve enough resource for e services, and many other sophisticated functions and algorithms are analyzed for appropriate resource allocation.

In order to maintain resource allocation mechanisms, routers need to distinguish packets from each other and then make an action to satisfy the needs of QoS such as dropping or forwarding a packet accordingly. In other words, the traffic has to be classified into flows using predefined filters and then the corresponding action is taken. This process is called packet classification or packet filtering interchangeably. The routers that have the ability to classify packets into predefined flows based on the needs of the ISPs are called flow aware routers [7]. The flows, more commonly known as rules or filters are identified by the context of packets. The IP header comprising source address (SA), destinationaddress (DA), source port (SP), destination port (DP) and the protocol (PRTCL) fields of an incoming packet is compared with the rules in a rule set to identify its flow. In a simple rule, the addresses are represented as prefixes while port numbers are specified as ranges and protocol field is a fixed length number.

Packet classification problem in flow aware routers has been studied for a long time. However, it still attracts a great deal of attention from the researchers because of the fast growth of the Internet. The need for fast and efficient algorithms arises since the bandwidth of the network doubles in every nine months [21]. The growth of the network bandwidth requires providing admission control, traffic management, packet scheduling and network security for complex flows. Furthermore, the line rate of the Internet is increasing due to the new advancements in optical technology. Therefore, packet classification algorithms must be accommodated by suitable hardware to achieve packet processing inline rates. Due to all these reasons stared, packet classification is still a major problem that craves for novel solutions.

Performance Challenges

- *Speed*

Speed or throughput is the major performance metric that is used to evaluate packet classification algorithms. Packet classification requires fast processing to support the speed of the Internet which currently reached beyond 40Gbps [12]. In other words, an incoming IP datagram has to be processed in 8ns (assuming minimum sized 40 bytes IP packets). In order to achieve this goal, the hardware has to be as simple as possible since the complexity results in extra processing time [4].

- *Memory*

In addition to speed, memory requirement is another important metric for packet classification. The memory efficiency is defined as the storage requirement per rule [5]. The hardware resources are limited and should be used efficiently to serve larger rule sets. Additionally, larger memory needs extra power consumption and extra processing time. Therefore, memory efficient algorithms are required to achieve fast and power efficient results.

- *Scalability*

The size of rule sets in a router is increasing due to the growth of internet. Therefore, the proposed algorithms have to be scalable with respect to the number of rules in filter (rule) sets. In addition to scalability in the number of rules, scalability in the size and the number of header fields of a rule are also important for classification engines. The latest revision of the internet protocol IPv6 with 128 bits of address fields and next generation Open Flow switches with 12 header fields for each rule necessitate scalable solutions for efficient packet classification [17], [6].

- *Latency*

Latency is another major metric of a packet classification algorithm. The total time of packet processing in a router is called latency. Throughput and latency are different metrics and an algorithm might have a high throughput while having a high latency value. For example, pipelined algorithms achieve good results for the throughput. The processing time of one pipeline stage determines the throughput value. On the other hand, latency is the time from the packet enters the pipeline until it leaves. In other words, latency is equal to the number of pipeline stages multiplied by the processing time of a single pipeline stage. Deep pipeline in decision based algorithms increases the latency which eventually hasa negative effect on system synchronization of real time applications. Therefore, low latency is as significant as high throughput for delay guaranteed services.

II. HIERARCHICAL HYBRID SEARCH STRUCTURE FOR HIGH PERFORMANCE PACKET CLASSIFICATION

To achieve high throughput in packet classification, the hardware that supports the algorithm should be simple and easy to implement. For instance, trie algorithms that simply determine the next node by making one bit inspection can easily be realized with a simple hardware and higher thro ughputcan be achieved when compared to other designs. Although trie structures offer high throughput, they usually suffer from backtracking, which makes the search process difficult and complex.

The Hierarchical Trie (H-Trie) structured using SA and DA prefixes. First, all SA prefixes in the rule set are stored in a trie. Then, DA prefixes are connected to related SA nodes via DA tries. As a result, hierarchically connected SA and DA tries are built. When a packet comes, it traverses H-trie starting from SA trie. When a SA prefix is reached in the trie, the search continuous on DA trie. However, even if there is a match, the packet has to backtrack to SA trie and continue the search to find a prior match.

Table 1.1 shows a sample rule set for packet classification. In this table, the SA prefix of R4 and R5 (00*) is a descendant of SA prefix of R1, R2, R3, R8, R9 and R10 (0*). Moreover, these prefixes are both descendants of the SA prefix of R7 (*). When a packet whose SA prefix starting with “00” comes, search has to traverse all three sub-branches. Even if a match condition occurs somewhere, the search operation should continue by backtracking since there might be a higher priority match.

Table 1.1: Sample rule set

Rule	SA	DA	SP	DP	Protocol	Priority	Action
R1	0*	10*	80	*	TCP	1	Act0
R2	0*	01*	17	17	UDP	2	Act1
R3	0*	1*	44	*	TCP	2	Act2
R4	00*	1*	17	44	UDP	3	Act3
R5	00*	11*	*	100	TCP	4	Act4
R6	10*	1*	*	*	*	5	Act5
R7	*	00*	*	*	TCP	5	Act6
R8	0*	10*	*	100	TCP	6	Act7
R9	0*	1*	*	*	TCP	7	Act8
R10	0*	10*	17	17	UDP	7	Act9
R11	111*	000*	80	*	TCP	8	Act10

Fig.1.1 illustrates the backtracking of an incoming packet on H-Trie and clearly indicates all the paths that it follows in a proper search [5]. In this example, the incoming packet has SA value of “000” and DA value of “110”. Since the SA of this packet matches all the nodes on the path, the corresponding DA nodes must be visited.

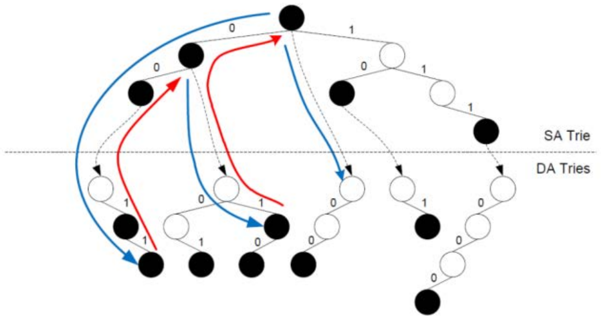

Fig. 1.1: Backtracking in H-trie structure.

II. LITERATURE SURVEY

Y. R. Qu and V. K. Prasanna, [1] High-performance and dynamically updatable hardware architectures for multi-field packet classification have regained much interest in the research community. For example, software defined networking requires 15 fields of the packets to be checked against a predefined rule set. Many algorithmic solutions for packet classification have been studied over the past decade. FPGA-based packet classification engines can achieve very high throughput; however, supporting dynamic updates is yet challenging. In this paper, authors present a two-dimensional pipelined architecture for packet classification on FPGA; this architecture achieves high throughput while supporting dynamic updates. In this architecture, modular Processing Elements (PEs) are arranged in a two-dimensional array. Each PE accesses its designated memory locally, and supports prefix match and exact match efficiently. The entire array is both horizontally and vertically pipelined. Authors exploit striding, clustering, dual-port memory, and gating techniques to further improve the performance of their architecture. The total memory is proportional to the rule set size. Their architecture sustains high clock rate even if authors scale up (1) the length of each packet header, or/and (2) the number of rules in the rule set. The performance of the entire architecture does not depend on rule set features such as the number of unique values in each field. The PEs are also self-reconfigurable; they support dynamic updates of the rule set during run-time with very little throughput degradation. Experimental results show that, for a 1 K 15-tuple rule set, a state-of-the-art FPGA can sustain a throughput of 650 Million Packets Per Second (MPPS) with 1 million updates/second. Compared to TCAM, their architecture demonstrates at least four-fold energy efficiency while achieving two-fold throughput.

P. Gupta and N. McKeown, [2] The process of categorizing packets into “flows” in an Internet router is called packet classification. All packets belonging to the same flow obey a predefined rule and are processed in a similar manner by the router. For example, all packets with the same source and destination IP addresses may be defined to form a flow. Packet classification is needed for non-best-effort services, such as firewalls and quality of service; services that require the capability to distinguish and isolate traffic in different flows for suitable processing. In general, packet classification on multiple fields is a difficult problem. Hence, researchers have proposed a variety of algorithms which, broadly speaking, can be categorized as basic search algorithms, geometric algorithms, heuristic algorithms, or hardware-specific search algorithms. In this tutorial authors describe algorithms that are representative of each category, and discuss which type of algorithm might be suitable for different applications.

Y. R. Qu, S. Zhou and V. K. Prasanna, [3] Algorithms and FPGA based implementations for packet classification have been studied over the past decade. Algorithmic solutions have focused on high throughput; however, supporting dynamic updates has been challenging. In this paper, authors present a 2-dimensional pipelined architecture for packet classification on FPGA, which achieves high throughput while supporting dynamic updates. Fine grained processing elements are arranged in a 2-dimensional array; each processing element accesses its designated memory locally, resulting in a scalable architecture. The entire array is both horizontally and vertically pipelined. As a result, it supports high clock rate that does not deteriorate as the length of the packet header or the size of the rule set increases. The performance of the architecture does not depend on rule set features such as the number of unique values in each field. The architecture also efficiently supports range searches in individual fields. The total memory is proportional to the rule set size. Dynamic updates- modify, delete and insert operations for the rule set during run-time are also supported on the self-reconfigurable processing elements with very little impact on the sustained throughput. Experimental results show that, for a 1K 15-tuple rule set, a state-of-the-art FPGA can sustain 190Gbps throughput with 1million updates/second. To the best of their knowledge, authors are not aware of any packet classification approach that simultaneously supports both high throughput and dynamic updates of the rule set. Their architecture demonstrates 4× energy efficiency while achieving 2× throughput compared to TCAM.

F. Yu, R. H. Katz and T. V. Lakshman, [4] Today's packet classification systems are designed to provide the highest-priority matching result, such as the longest prefix match, even if a packet matches multiple classification rules. However, new network applications demanding multimatch classification - that is, requiring all matching results instead of only the highest-priority match - are emerging. Ternary content-addressable memory is becoming a common extension to network processors, and its capability and speed make it attractive for high-speed networks. The proposed TCAM-based scheme produces multimatch classification results with about 10 times fewer memory lookups than a pure software approach. In addition, their scheme for removing negation in rule sets saves up to 95 percent of the TCAM space used by a straightforward implementation.

F. Zane, Girija Narlikar and A. Basu, [5] Ternary content-addressable memories (TCAMs) are becoming very popular for designing high-throughput forwarding engines on routers: they are fast, cost-effective and simple to manage. However, a major drawback of TCAMs is their high power consumption. This paper presents architectures and algorithms for making TCAM-based routing tables more power efficient. The proposed architectures and algorithms are simple to implement, use commodity TCAMs, and provide worst-case power consumption guarantees (independent of routing table contents).

III. PROBLEM IDENTIFICATION

In this previous research work presented a 2-dimensional pipelined architecture for packet classification. The advantages of the proposed architecture include:

1) *Parameterized:* The architecture is highly parameterized; it can be optimized with respect tovarious performance metrics.

2) *Rule-set-independent:* The performance does not depend on any rule set features other than the rule set size.

3) *High-throughput:* All the PEs access their designated distRAM modules independently. The memory access is localized, resulting in shorter interconnections in each PE. This leads to high clock rate and high throughput on FPGA.

4) *Scalable with respect to rule set size:* The longest wire length is not significantly affected by the total number of rules; the architecture sustains high throughput for a large number of rules, assuming authors have sufficient hardware resources.

5) *Scalable with respect to input length:* The throughput is not adversely affected by the length of the packet header. Their architecture achieves good performance for both classic and Open Flow packet classification.

6) *Dynamically updatable:* The dynamic update is performed in a distributed manner on selfreconfigurable PEs; the update scheme has little impact on the sustained performance.

7) *Energy-efficient:* The proposed architecture demonstrates better energy efficiency. Compared to Stride BV, their approach sustains 2 × throughputs and supports fast dynamic updates with slightly more energy consumption.

In the future, authors plan to use this architecture vigorously for other network applications including traffic classification and heavy hitter detection for data center networks. Authors will also explore more techniques to improve the energy efficiency of this architecture.

IV.CONCLUSION

As information systems increase i n size and complexity the task of data acquisition and processing in and of such systems becomes ever more arduous and computationally expensive. To cop e with the sheer size of the tasks required new and faster techniques and systems are required to be developed; often app earing as either software packages or primarily hardware based systems such as the systems. This Literature review paper seeks to understand the literature survey of packetfilters implemented on FPGAs as high speed platforms and evaluates the suitability of FPGAs as platforms for packet filter implementation. It must be remembered that packet filters need to operate at line speeds or else they become detrimental to the health of any network they may be placed up on by introducing significant delays. This task is approached by a novice to the world of FPGAs and hardware description languages and begins with a review of other implementations of packet filters on FPGAs.

REFERENCES

[1] 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. 1 2016.

[2] P. Gupta and N. McKeown, "Algorithms for packet classification," in IEEE Network, vol. 15, no. 2, pp. 24-32, Mar/Apr 2001.

[3] Y. R. Qu, S. Zhou and V. K. Prasanna, "High-performance architecture for dynamically updatable packet classification on FPGA," Architectures for Networking and Communications Systems (ANCS), 2013 ACM/IEEE Symposium on, San Jose, CA, USA, 2013, pp. 125-136.

[4] F. Yu, R. H. Katz and T. V. Lakshman, "Efficient multimatch packet classification and lookup with TCAM," in IEEE Micro, vol. 25, no. 1, pp. 50-59, Jan.-Feb. 2005.

[5] F. Zane, GirijaNarlikar and A. Basu, "Coolcams: power-efficient TCAMs for forwarding engines," INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies, 2003, pp. 42-52 vol.1.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” SIGCOMM Comput. Commun.Rev., vol. 38, no. 2, pp. 69–74, 2008.

[7] W. Jiang and V. K. Prasanna, “Scalable Packet Classification on FPGA,” IEEE Trans. VLSI Syst., vol. 20, no. 9, pp. 1668–1680, 2012.

[8] J. Bispo, I. Sourdis, J. Cardoso, and S. Vassiliadis, “Regular Expression Matching for Reconfigurable Packet Inspection,” in Proc. of IEEE International Conference on Field Programmable Technology (FPT), 2006, pp. 119–12.

[9] Z. P. Ang, A. Kumar, and Y. Ha, “High Speed Video Processing Using Fine-Grained Processing on FPGA Platform,” in Proc. of IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM), 2013, pp. 85–88.

[10] T. Ganegedara and V. K. Prasanna, “StrideBV: Single Chip 400G+ Packet Classification,” in Proc. of IEEE International Conference on High Performance Switching and Routing (HPSR), 2012, pp. 1–6.

[11] I. Bonesana, M. Paolieri, and M. Santambrogio, “An Adaptable FPGA-based System for Regular Expression Matching,” in Proc. of Design, Automation and Test in Europe (DATE), 2008, pp. 1262–1267.

[12] A. Sudarsanam, R. Barnes, J. Carver, R. Kallam, and A. Dasu,“Dynamically Reconfigurable Systolic Array Accelerators: A Case Study with Extended Kalman Filter and Discrete Wavelet Transform Algorithms,” Computers Digital Techniques, IET, vol. 4, no. 2, pp. 126–142, 2010.

[13] R. Salvador, A. Otero, J. Mora, E. de la Torre, T. Riesgo, and L. Sekanina, “Self-Reconfigurable Evolvable Hardware System for Adaptive Image Processing,” IEEE Transactions on Computers, vol. 62, no. 8, pp. 1481–1493, 2013.

[14] L. Frigerio, K. Marks, and A. Krikelis, “Timed Coloured Petri Nets for Performance Evaluation of DSP Applications: The 3GPP LTE Case Study,” in VLSI-SoC: Design Methodologies for SoC and SiP. Springer Berlin Heidelberg, 2010, vol. 313, pp. 114–132.

[15] P. Gupta and N. McKeown, “Classifying Packets with Hierarchical Intelligent Cuttings,” IEEE Micro, vol. 20, no. 1, pp. 34–41, 2000.

[16] Y. R. Qu, S. Zhou, and V.Prasanna, “Scalable Many-Field Packet Classification on Multi-core Processors,” in International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), 2013, pp. 33–40.