

Effective Architecture of Packet Classification on FPGA

Ms. Kranti M.Hande¹, Prof.V.R.Wadhankar², Prof.D.S.Dabhade³

Electronics Department, Agnihotri College of Engineering, Wardha, India^{1,2,3}

Abstract: For providing different quality services, router needs packet classification. Rules are frequently changing and due to multidimensional field it is difficult to maintain high speed and scalability in packet classification. In this work we use pipeline architecture with logic gates on FPGA. Due to use of logic gates and incorporate range search in architecture speed, memory efficiency, power efficiency of packet classification increases as compare to any previous method used for packet classification. This method is rule set independent.

Keywords: FPGA, packet classification, pipeline architecture, router, header field.

I. INTRODUCTION

The growth of the Internet and multiple dimensional field demands router to perform packet classification with high speed to support access control, traffic engineering, firewall processing, quality of service differentiation and other services. Classification of the packets into different categories based on a set of predefined rules, which specify the value ranges of the multiple fields in the packet header, is called multi-field packet classification.

The existing methods used different algorithms for packet classification, these algorithms require more memory, these methods use range to prefix conversion which require more memory because a single range converts to six different prefixes, therefore single rule is expanded into six rules causes more number of comparisons with header field. In our method we use XNOR gates, priority encoder and AND gate in architecture to identify the ruleset for packet header field. We use range matching, a range of upper and lower bound is provided by the rule and a match is found if the header field match within this range. Therefore it requires only two comparison for each rule i.e. it requires less number of comparison as compare to range to prefix conversion. Previous algorithms are not scalable i.e. there is no same performance for different rules. Now a days ruleset are changing frequently. Our method is ruleset independent therefore there is no problem for changing ruleset.

In our method bits of header field and bits of rules are applied to the input of XNOR gate, the output of the XNOR gate is the bits that indicates the rules which is matching for the input packet header. Use of XNOR gate eliminates use of lookup engine and field split which requires more memory. It also eliminates bit vector generation.

With the modular approach, multiple pipelines are required to support large-scale packet classification. Multiple rules from a classifier may match with an incoming packet, however, only the matching rule with highest priority is applied on the packet in the case of

packet classification. The priority is simply the order in which the rules appear in the classifier. In order to extract this priority encoder is used. In Integration of Range Search method the input packet header value is checked against both lower and upper bound of the rules, two values are loaded at each pipeline stage, per rule. In the range search method there is no necessity to store of entire bits in pipeline stages. The result of a sub-field is forwarded to the next stage and the comparison is continued till the last sub-field is reached. In our method FPGA based packet classification is used. It provides reconfigurability and compress the rules for on-chip which does not require external slow memory.

II. RELATED WORK

In basic search algorithm, Tuple Space Search algorithm has less throughput, high memory requirement and highly dependent on each ruleset. TCAM (Ternary Content Addressable memory) has high throughput but low memory efficiency, it requires more transistor to store same bits than RAM, It has low power efficiency and it is dependent on ruleset. In parallel bit vector algorithm each bit of bit vector represents rule of the entire ruleset, each bit of rule is compare with bit of header field independently than the other bits in the bit vector. This algorithm has high throughput but low memory efficiency.

In stride bit vector algorithm field is divided into sub-field. Sub field length of k bits is called as a stride. There is independent matching in between bits of the rule with corresponding k bits of an input packet header. Comparison is done in between k bit header field with the lower and upper bounds of the rules of stride. This method is called as incorporate range search. In range to prefix conversion rules are represented as a ternary string which needs more number of comparisons therefore needs more memory requirement. The result of the first stride is passed to the next stride comparison and they are bitwise AND ed to get entire classifier. Highest priority match is extracted by pipelined priority encoder. This algorithm has

moderate throughput moderate memory efficiency and it is independent on ruleset.

Distributed Cross-producing of Field Labels (DCFL), use independent search engines, and operate in parallel manner. Instead of using bit-vectors DCFL uses a network of efficient aggregation nodes, by employing BloomFilters and encoding intermediate search results. It is highly dependent on ruleset.

III. BLOCK DIAGRAM

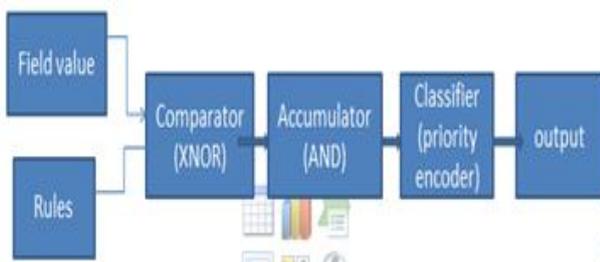


Fig.1 Block Diagram

For example rules and header field are considered of four bits. Header field bits and rules bits are applied to XNOR gate to get resultant bit vector. There is no need to split the field bits. Resultant bit vector is ANDed together to get highest priority match.

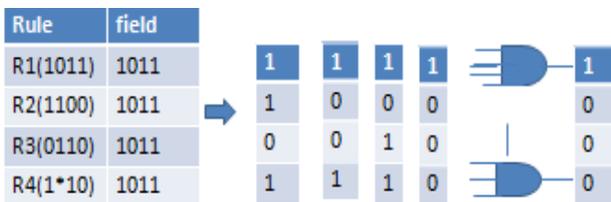


Fig.2.Example

In our example header bits 1011 is compared with all rules with the help of XNOR gate, resultant bits are ANDed together and applied to priority encoder which find out class four for header field 1011 because first bit set to one.

III. MODULAR PIPELINED ARCHITECTURE

In implementation of range search in hardware, the packet header value is checked against both lower and upper bounds of the rules for this two values need to be loaded at each pipeline stage, per rule. These values either can be stored locally in the pipeline stage or stored in stage memory. In the range search stages do not require storage of entire bits. In the architecture pipeline stages are used Each individual bit is independent on the other bits therefore partitioning is possible. Therefore there is no need to enter all N bits at each pipeline stage, only N/P bits are required to enter in pipeline stage, where P is number of partitions. Thus memory bandwidth requirement is decreased. In our method, we use XNOR gate for comparison of rules and header field. Rules and header field are the inputs of XNOR gate due to use of XNOR gate we eliminate the generation of bit vectors by field splitting which improves the memory efficiency and the output of XNOR gate is bitwise ANDed to get entire classifier Highest priority match is extracted by pipelined priority encoder.

This algorithm has high throughput, high memory efficiency and it is independent on ruleset. Pipelined priority encoder is used to overcome the problem of decreasing speed to find highest priority when the length of resultant bit vector increases. Pipelined priority encoder operates at high frequency. Bits of header field is used as address to stage memory.

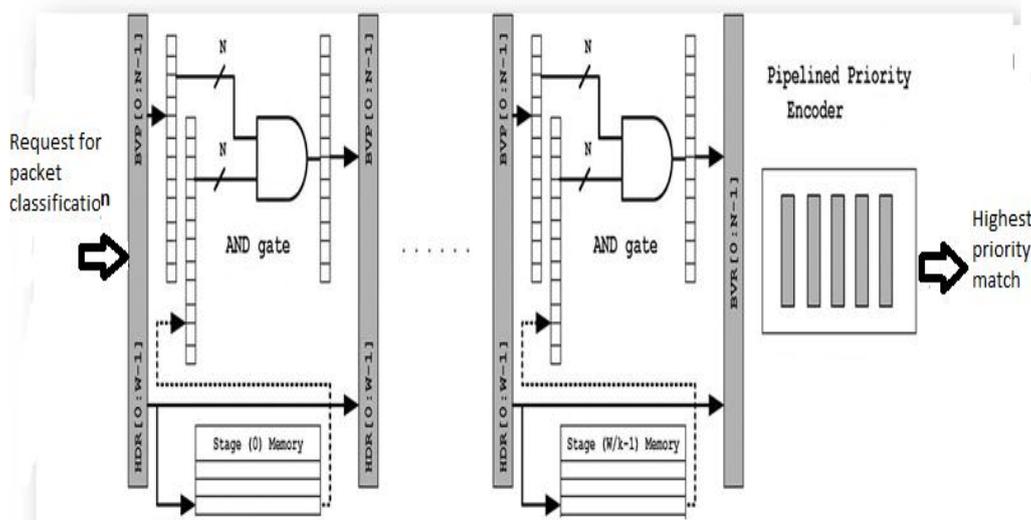


fig.2.Pipelined Architecture

The output bit vector of stage memory and the bit-vector generated from the previous stage are bitwise ANDed together and generate final bit vector of the current stage. Same construction is continued throughout the pipeline. and the Priority Encoder extracts the highest priority match from the resultant bit-vector Rules are arranged in order of decreasing priority. Priority encoder identifies first bit positions which is set to one and identify class in one cycle. But when the length resultant bits increases, time required to find highest priority matching increases. Therefore throughput decreases. To overcome this problem $\log^B N$ stages pipelined priority encoder is used which operate at high frequency, where N is number of resultant bits and B is number of partitions the resultant bits are split into in a given stage.

IV.SERIAL AND PARALLEL CLASSIFIER

In serial classifier one header field is compared with one rule at a time .It is called one rule classifier.Serial classifier has high latency ,packet classification latency increases linearly proportional to header length. For longer header length serial classifier is not useful, for application which demands low latency like multimedia, gaming parallel classifier is used.

In Parallel Classifier many of header field are compared with many rules at a time. Parallel classifier has low latency. In both serial and parallel classifier partition size of P,there will be N/P modular pipelines which find the matching and then priority encoder extracts highest priority match. Parallel classifier throughput is more as compare to serial classifier.

The memory requirement is according to number of rules present in packet classification. The memory require to store one rule is $O(1)$ space. Therefore memory requirement for N rules is $O(N)$.To change the classifier from serial to parallel or parallel to serial there is no change in memory requirement, modular pipelined architecture does not change requirement of memory. In FPGA on chip memory is fully utilized. There is no requirement of external memory.

V.COMPARISION WITH EXISTING METHODS

We compare several existing methods with this method of packet classification. By comparing with previous algorithm like Linear search, Grid of tries, Distributed Cross Field label, Recursive Flow classification Hierarchical Intelligent Cuttings, Field split bit vector, BV-TCAM, Tuple space search with method using XNOR gate, this method increases memory efficiency because there is no range to prefix conversion instead of incorporate range search is used which require less number of comparisons, again due to elimination of field split memory efficiency increases. This method is rule set independent it is scalable; This method is used for changing ruleset. Throughput is more. It is suitable for multidimensional field packet classification.

VI. CONCLUSION

Due to increase in demand of data, it is a great challenge to develop ruleset independent solutions for next-generation packet classification that support larger rule sets and more packet header field with high throughput.Our method presented high speed, scalable, high memory efficiency, low latency packet Classification on FPGA (Field Programmable Gate Array) which is better than any previous method.

REFERENCES

- [1] T. Ganegedara and V. Prasanna, "StrideBV: 400G+ Single Chip Packet Classification," in Proc. IEEE Conf. HPSR,2012, pp.1-6.
- [2] P. Gupta and N. McKeown, "Classifying Packets with Hierarchical Intelligent Cuttings," IEEE Micro, vol. 20, no. 1, pp. 34-41,Jan./Feb. 2000.
- [3] W. Jiang and V.K. Prasanna, "Field-Split Parallel Architecture for High Performance Multi-Match Packet Classification Using FPGAs," in Proc. 21st Annu. SPAA, 2009, pp. 188-196.
- [4] T.V. Lakshman and D. Stiliadis, "High-Speed Policy- Based Packet Forwarding Using Efficient Multi- Dimensional Range Matching," SIGCOMM Comput.Commun. Rev., vol. 28, no. 4,pp. 203-214, Oct. 1998.
- [5] Pankaj Gupta and Nick McKeown, "Packet Classification on Multiple Fields," Proc. Sigcomm, Computer Communication Review, vol. 29, no. 4, pp 147-60, September 1999, Harvard University.
- [6] V. Srinivasan, S. Suri, and G. Varghese, "Packet Classification using TupleSpace Search", Proceedings of ACM Sigcomm, pages 135-46, September 1999.
- [7] W. Eatherton, G. Varghese, and Z. Dittia," Tree Bitmap: Hardware/Software IP Lookups with Incremental Updates,"SIGCOMM Comput. Commun. Rev., vol. 34, no. 2, pp. 97-122, Apr. 2004
- [8] M. Faezipour and M. Nourani, "Wire-Speed TCAM-Based Architectures for Multimatch Packet Classification," IEEE Trans.Comput., vol. 58, no. 1, pp. 5-17, Jan. 2009.
- [9] C.R. Meiners, A.X. Liu, and E. Torng," Hardware Based Packet Classification for High Speed Internet Routers." Berlin,Germany: Springer-Verlag, 2010.
- [10] A. Sanny, T. Ganegedara, and V. Prasanna, "A Comparison of Ruleset Feature Independent Packet Classification Engines on FPGA," in Proc. IEEE IPDPS RAW, 2013, pp. 124-133
- [11] D. Taylor and J. Turner, "Scalable Packet Classification Using Distributed Crossproducing of Field Labels," in Proc. 24th annu. Joint IEEE INFOCOM, Mar. 2005, vol. 1, pp. 269-280.
- [12] C.A. Zerbini and J.M. Finochietto, "Performance Evaluation of Packet Classification on FPGA-Based TCAM Emulation Architectures," in Proc. IEEE GLOBECOM, 2012, pp. 2766-2771