

# A Review on Effective Architecture of Packet Classification

Kranti M. Hande<sup>1</sup>, V. R. Wadhankar<sup>2</sup>

<sup>1</sup>Electronics Department, Agnihotri College of Engineering, Wardha, India

<sup>2</sup>Professor, Electronics Department, Agnihotri College of Engineering, Wardha, India

**Abstract:** *The process of making category of packets is called packet classification. All packets of the same category obey a pre-defined rule and are processed in a similar manner by the router. Packet classification is needed for various application. Due to increase of demand in throughput, increase in internet traffic, multi field packet classification becomes very difficult. In this paper we describe different types of algorithm and identification of parameters of algorithm suitable for different application. This paper includes the overview of the new challenges, existing solutions and future scope of packet classification.*

**Keywords:** bit vector, header field, Packet classification, router

## 1. Introduction

Packet classification enables routers to support access control, resource reservation, virtual private networks, and other services. Each packet header arriving at a router is compared against a set of rules. Each rule contains fields with value priority, and an action. The TCP/IP header field consist of source and destination IP addresses, source and destination port, and protocol identifier. Packet matches rule if it matches every field in that rule.

## 2. Basic Search Algorithms

### 2.1 Linear Search

In this algorithm rules are stored in order of decreasing priority. A packet is compared with each rule in sequential manner until a rule is found that matches all relevant fields.

It is simple. Memory requirement is less. It is highly dependent on rule set. If the number of rules increases time to classify packet increases.

### 2.2 Geometric Algorithms

#### 2.2.1 Grid of tries

It allocates a rule to only one trie node as in a hierarchical trie, and achieves query time by pre-computing and storing a switch pointer in some trie nodes. A switch pointer guides the search process and is labelled with 0 or 1. It decreases classification time complexity. It reduces memory requirement. It works good for two dimensional multiple field. It is highly dependent on ruleset.

### 2.3 Hardware based algorithm

#### 2.3.1 Ternary CAM

Rules in decreasing order of priorities are stored memory array and compares an header field with every element in the memory array parallely. The N-bit bit-vector indicates matching rules and so the bit priority encoder shows the address of the highest matching priority.

Due to parallel comparison speed is high. It has more power dissipation because it compare address with every element of TCAM in parallel manner. It stores less bits in same size of chip than RAM because one bit in TCAM requires more number of transistors as compared to transistors require for one bit in RAM. It is suitable for less number of rules, it is not suitable for large number of rules.

#### 2.3.2 Bitmap intersection

In this scheme set of rules matches a packet is the intersection of sets. For intersection of sets in hardware, each set is converted to N-bit bitmap with each bit corresponds to a rule. The set of matching rules is the set of rules whose corresponding bits are „1“ in the bitmap. The best matching rule is found from the sets which are intersected to give the set of matching rules from which the best matching rule is selected. In N bit wide bitmap in each d dimensions,  $O(N)$  ranges are present. Time required for packet classification is  $O(d \cdot T_{RL} + dN/w)$  where w is the width of memory and  $T_{RL}$  is time to do one range look up. Thus time complexity can be reduced by d. It is suitable for a less number of rules in multiple dimensions. It requires more memory and requires less classification time.

### 2.4 Heuristic algorithms

#### 2.4.1 Tuple Space Search

In tuple space search algorithm a classification query is decomposed into a number of exact match queries. The algorithm first maps each d dimensional rule into a d-tuple. Therefore the set of rules mapped to the same tuple are of a fixed length operations on each of the hash tables corresponding to all possible tuples in the classifier. If the number of tuples is small, the use of the tuple space search algorithm performs well for multiple dimensions.

#### 2.4.2 Recursive Flow Classification (RFC)

RFC algorithm is used for packet classification of multiple field it map the s bits of the packet header to a T bit action identifier. RFC attempts to perform the same mapping over several phases, the algorithm maps one set of values to a smaller set.

- 1) In the first phase, into multiple chunks are formed from fields of the packet header that are used to index into multiple memories in parallel. Result of lookup is small by selecting contents of memory.
- 2) In next phases, memories are indexed from the results of earlier phases memories are indexed.
- 3) In the final phase, the memory yields the action.
- 4) It is not efficient for larger ruleset, memory requirement and time complexity increases.

### 2.5 Distributed Cross-producing of Field labels (DCFL) algorithm

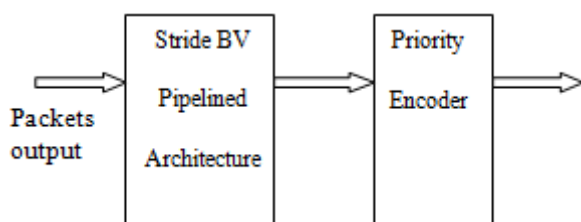
It is used for the multi-field searching and use independent search parallelly to find the matching conditions for each filter .DCFL uses a network of efficient aggregation nodes, employing bloom filter and encoding search result. This algorithm requires less memory. It is highly dependent on ruleset.

### 2.6 Decomposition-based algorithms

#### 2.6.1 Stride BV algorithm

In **Field Split Bit-Vector (FSBV)** rules and header field are converted in the form of bit-vectors. Each field bit vector was split into sub-fields. In this algorithm a sub-field length of k bits is considered as a stride. Each k bits of the rule perform independent matching on the corresponding k bits of an input packet header. k bit header produce a N bit-vector and header stride requested for packet classification is compared against the lower and upper bounds of the rules of stride instead of range to prefix conversion which needs more number of comparisons.

Thus it reduces memory requirement. The result of the first stride is forwarded to the next stride comparison and they are bitwise ANDed together in a pipelined fashion to arrive at the matching results for the entire classifier. With the help of pipelined priority encoder highest priority match is extracted.



**Figure 1:** Block Diagram

The memory requirement of stride bit vector is fixed for a classifier. Packet classification is independent of features of ruleset therefore packet classification is not affected due to changes in rules. Due to use of range search integration memory efficiency increases. Due to use of FPGA, on a single chip it can perform large-scale, high-speed, packet classification. Power requirement is very less due to less power dissipation. Throughput is high.

### 2.7 Decision Tree-Based Algorithm

In Decision-tree-based algorithms, for d number of header fields, each packet defines a point in this d-dimensional space. Each space is divided into smaller subspaces. Each

subspace allows a linear search to find best matching rule. From one or more fields in the rule , the search space is cut based on the information. The HyperCuts algorithm allows cutting on multiple fields per step, to form shorter decision tree. It allows updation of ruleset therefore it is more scalable as compare to other algorithm. Its dependency on ruleset is low.

#### 2.7.1 Parallel Bit-Vector (BV) algorithm

Rules and header field are converted in the form of bit-vectors. Independent search on each field is performed and then combine the search results from all fields. The lookup on each field returns a bit-vector and each bit of bit vector represents a rule. If the corresponding rule is matched on this field a bit is set and if the corresponding rule is not matched on this field a bit is reset. These bits are ANDed indicates the set of rules that matches a given packet. This algorithm provide a high throughput .Hardware requirement is less.

Memory efficiency is less.

## 3. Results and Findings

**Table 1:** Comparison of Algorithms

Algorithm	Throughput	Memory Requirement	Rulest Dependence
Linear search	LOW	LOW	HIGH
Ternary CAM	HIGH	HIGH	LOW
Grid of tries	HIGH	LOW	HIGH
Bitmap intersection	LOW	HIGH	HIGH
Recursive Flow Classification	LOW	HIGH	HIGH
Tuple Space Search	HIGH	HIGH	HIGH
Decision tree-based algorithm	HIGH	HIGH	LOW
Bit-Vector (BV) algorithm	HIGH	HIGH	HIGH
DCFL algorithm	LOW	LOW	HIGH
StrideBit Vector Algorithm	HIGH	LESS	NO

## 4. Proposed Method

After study of all algorithms it is found that Stride Bit vector Algorithm is the best algorithm among all algorithm because its throughput is high, it is bit vector based memory efficient packet classification algorithm suitable for hardware implementation. For any given ruleset it can use with high performance due to ruleset feature independence nature. Stride Bit Vector Algorithm is used on FPGA. It is proposed that using logical gates memory efficiency, throughput can further increase. With the help of logic gates further modification has scope to increase the performance.

## 5. 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 higher throughput, larger rule sets and more packet header fields. Growing Internet and media world efficient packet classification have a vast scope and opportunities in near future. This paper presented a comparison of different algorithm. After comparison it is

found that there is future scope in modification of algorithm and modification in architecture for packet classification.

## 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.