IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
ISSN: 2320 - 8791
www.ijreat.org

Novel Efficient And Scalable Packet Classification

Sherin Thankam Koshy[1] And Mrs K.Jaraline[2]

[1]Department of ECE
KCG College of Technology
Karappakam, Chennai- 600 097.

[2]Assistant Professor,
KCG College of Technology
Karappakam, Chennai- 600 697.

Abstract

The next generation routers is demanded to support a variety of network functionalities, such as firewall processing, quality of service (QoS) differentiation, virtual private networks, policy routing, traffic billing, and other value added services due to the booming development of the internet. This makes it necessary for the routers to classify the packets into different categories. Packet classification problem is traditionally considered in the fixed 5-tuple fields. Due to the rapid growth of the Internet traffic, and the rule set size, multi-field packet classification has become one of the fundamental challenges in designing high speed routers. This paper presents a decision tree based architecture for the next generation packet classification. By revisiting the traditional techniques for 5-tuple packet classification, we have proposed several optimization techniques for the decision tree based algorithm. Also, the designing of the mapping is done in such a way that the high throughput is sustained along with maximum utilization of memory.

Keywords- packet classification, Openflow, SRAM, tuple

1.INTRODUCTION

The development of the applications of Internet demands next-generation routers to support a variety of network functionalities, such as firewall processing, quality of service (QoS)differentiation, virtual private networks, policy routing, traffic billing and other value added services. In order to provide these services, the router needs to classify 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. Such a function is called multi-field packet classification.The process of categorizing packets into "flows" in an Internet router is called packet classification. In traditional network applications, packet classification problems usually consider the fixed 5-tuple fields: 32-bit source/destination IP addresses, 16-bit source/destination port numbers, and 8-bit transport layer protocol[1].Due to the rapid growth of the Internet traffic, as well as the rule set size, multi-field packet classification has become one of the fundamental challenges to designing high speed routers. It is impossible for existing software-based solutions to reach the throughput of current link rate. Most of the existing work in high-throughput packet classification is based on ternary content addressable memory (TCAM)[[2][3] or a variety of hashing schemes such as Bloom Filters.However, TCAMs are not scalable with respect to clock rate, power consumption, or circuit area, compared to SRAMs. Most of TCAM-based solutions also suffer from range expansion when converting ranges into prefixes. Hashing-based solutions such as Bloom Filters[4][5] have become popular due to their time performance and high memory efficiency[6]. However, hashing cannot provide deterministic performance due to potential collision and is inefficient in handling wildcard or prefix matching.Even though many multi-dimensional classification algorithms have been proposed, most of them are precluded from commercial equipments due to their high memory requirements. As an alternative, optimizing and mapping state-of-the-art packet classification algorithms onto SRAM-based parallel architectures such as field-programmable gate array (FPGA) is a better option. FPGA technology has become an attractive option for implementing real-time[7] network processing engines due to its ability to reconfigure and to offer abundant parallelism.State-of-the-art SRAM-based FPGA devices provide high clock rate, low power dissipation and large amounts of on-chip dual-port memory with configurable word width.

2.EXISTING SYSTEM

The TCAM memory array stores rules in decreasing order of priorities, and compares an input key against every element in the array in parallel. The *N*-bit bit-vector, *matched*, indicates which rules match and so the *N*-bit priority encoder indicates the address of the highest priority match. The address is used to index into a RAM to find the action associated with this prefix.

TCAMs are being increasingly deployed because of their simplicity and speed (the promise of single clock-cycle classification. However, some disadvantages to TCAMs are:

- A TCAM is less dense than a RAM, storing fewer bits in the same chip area. One bit in an SRAM typically requires 4-6 transistors, while one bit in a TCAM requires 11-15 transistors.
- TCAMs dissipate more power than RAM solutions because an address is compared against every TCAM element in parallel. At the time of writing, a 2 Mb TCAM chip running at 50 MHz dissipates about 7 watts. In comparison, an 8Mb SRAM running at 200 MHz dissipates approximately 2 watts.

TCAMs[8] are appealing for relatively small classifiers, but will probably remain unsuitable in the near future for: (1) Large classifiers (256K-1M rules) used for microflow recognition at the edge of the network, (2) Large classifiers (128-256K rules) used at edge routers that manage thousands of subscribers (with a few rules per subscriber),(3) Extremely high speed (greater than 200Mpps) classification, and (4) Price-sensitive applications.

3.TUPLE PACKET CLASSIFICATION

Packet classification can be defined as the process of categorizing the packets into flows in an internet router and it is needed to access controls in firewall, policy based routing, traffic billing and provisioning of different quality of services. To classify which flow the packet belongs to is based on the content of packet header. All packets belonging to the same flow obey a pre-defined rule and are processed in a similar manner by the router.In general, packet classification on multiple fields is a difficult problem. The categorization function is performed by flow classifier, which maintains the set of rules. Packet classificationrequires comparing each packet against a database of rules and forwarding the packet according to the highest priority matching rule.Routers are now called upon to provide different qualities of service to different applications which means routers need new mechanisms such as admission control, resource reservation, per-flow queuing, and fair scheduling. All of these mechanisms require the router to distinguish packets belonging to different flows. The header of a packet contains instructions about the data carried by the packet. These instructions may include: Length of packet (some networks have fixed-length packets, while others rely on the header to contain this information), Synchronization (a few bits that help the packet match up to the network), Packet number (which packet this is in a sequence of packets), Protocol (on networks that carry multiple types of information, the protocol defines what type of packet is being transmitted: e-mail, Web page, streaming video), Destination address (where the packet is going), Originating address (where the packet came from) etc.A tuple is basically the field in the header of a packet. 5 tuple is a term used in computer networks to refer to a set of five different values that make up a Transmission Control Protocol/Internet Protocol (TCP/IP) connection.It can also be told as, tuple is a notion employed by network and system administrators in identifying the key requirementsto createan operational, secure andbidirectional network connection between two or more local and remote machines. The primary components of 5 tuple are the source and destination address. In this work we are considering 5-tuples for classifying the packets, which is shown in Table I.

Table I: Header fields supported in the design

Header field	*Notation*
Source IP address	SA
Destination IP address	DA
Protocol/Ethernet type	EtherType
Source port	SrcPort/SP
Destination port	DstPort/DP

Table II: Example of 5-tuple rule set

RULE	SA	DA	SP	DP	EtherType
R1	*	*	2-9	6-11	*
R2	1*	0*	3-8	1-4	10
R3	0*	0110*	9-12	10-13	11
R4	0*	11*	11-14	4-8	*

R5	011*	11*	1-4	9-15	10
R6	011*	11*	1-4	4-15	10
R7	110*	00*	0-15	5-6	11
R8	110*	0110*	0-15	5-6	*
R9	111*	0110*	0-15	7-9	11
R10	111*	00*	0-15	4-9	*

4.RULE SET PARTITIONING AND MATCHING

Rules can be defined as the individual entry for identifying packets. They can have one or more fields. We call a collection of rules a classifier. Each rule specifies a flow that a packet may belong to based on some criteria applied to the packet header.
The rules discussed here can be of two types

- *Simple rule* is the rule of which all the fields are specified as exact values.
- *Complex rule* is the rule containing wildcards or prefixes.

The 5-tuple header fields are matched against each rule. Each field of a rule can be specified as either an exact number or a wildcard. IP address fields can also be specified as a prefix.Each rule can have one or more fields and their associated values, a priority, and an action to be taken if matched. Different fields in a rule require different kinds of matches: prefix match for SA/DA, range match for SP/DP, and exact match for the protocol field. A packet is considered matching a rule only if it matches all the fields within that rule. A packet can match multiple rules, but only the rule with the highest priority is used to take action. Longest prefix matching for routing lookups is a special-case of one-dimensional packet classification. Concretely, each rule is generated as follows.
1) Each field is randomly set as a wildcard. When the field is not set as a wildcard, the following steps are executed.
2) For source/destination IP address fields, the prefix length is set randomly from between 1 and 32, and then the value is set randomly from its possible values.
3) For other fields, the value is set randomly from its possible value.

5.PACKET CLASSIFICATION ALGORITHM AND OPTIMIZATION

Next-generation packet classification can be viewed as a natural extension from traditional 5-tuple packet classification. Most of those algorithms fall into two categories:
Decomposition-based approaches
Decomposition-based algorithms[9], perform independent search on each field and finally combine the search results from all fields. Such algorithms are desirable for hardware implementation due to their parallel search on multiple fields. However, substantial storage is usually needed to merge the independent

Table III: Rule set used in the design

Rule	SA	DA	SrcPort	DstPort	EtherType
R1	*	*	2-9	6-11	*
R2	1*	0*	3-8	1-4	0x0812
R3	0*	0110*	9-12	10-13	0x0813
R4	011*	11*	1-4	9-15	0x0814
R5	0*	11*	11-14	4-8	*
R6	011*	11*	1-4	4-15	0x0812
R7	110*	00*	1-15	5-6	0x0815
R8	110*	0110*	0-15	5-6	*
R9	*	*	2-9	6-11	0x0812
R10	1*	0*	3-8	1-4	*
R11	011*	0110*	9-12	10-13	0x0813
R12	011*	11*	1-4	9-15	0x0816
R13	0*	11*	11-14	4-8	0x0814
R14	011*	11*	1-4	4-15	*
R15	110*	00*	1-15	5-6	0x0812

R16	110*	0110*	1-15	5-6	0x0815
R17	111*	0110*	0-15	7-9	*
R18	111*	00*	1-15	4-9	0x0812
R19	0*	0110*	9-12	10-13	*
R20	011*	110*	3-8	9-15	0x0816
R21	0110*	11*	11-14	4-8	0x0814
R22	011*	11*	1-4	4-15	0x0814
R23	110*	00*	4-15	5-6	0x0817
R24	1100*	0110*	1-15	5-6	0x0815

search results to obtain the final result. Decomposition-based algorithms have poor scalability, and work well only for small-scale rule sets.

Decision-tree-based approaches

Decision-tree-based algorithms, take the geometric view of the packet classification problem. Eachrule defines a hypercube in a *d*-dimensional space where isthe number of header fields considered for packet classification.Each packet defines a point in this *d*-dimensional space. Thedecision tree construction algorithm employs several heuristicsto cut the space recursively into smaller subspaces. Each sub-space ends up with fewer rules, which finally allows a low cost linear search to find the best matching rule. After the decision tree is built, the algorithm to classify a packet is simple.Based on the value of the packet header, the algorithm followsthe cutting sequence to locate the target subspace (i.e., a leafnode in the decision tree), and then performs a linear searchon the rules in this subspace.

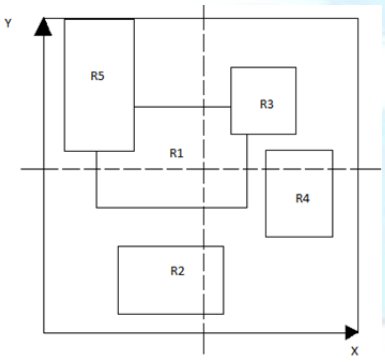

Fig1: X and Y axes correspond to SP and DP fields for R1–R5 in Table II.

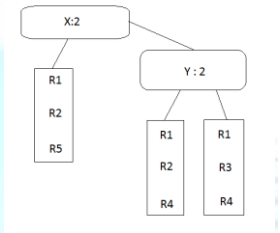

Fig.2: Hi-cut partitioning of Fig 1

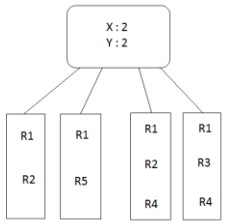

Fig.3: Hyper-cut partitioning of Fig 1

Decision-tree-based algorithms allow incremental rule updates and scale better than decomposition-based algorithms. The outstanding representatives of decision-tree-based packet classification algorithms are HiCuts and its enhanced version HyperCuts. At each node of the decision tree, the search space is cut based on the information from one or more fields in the rule.A rounded rectangle in Fig 2 and Fig 3 denotes an internal tree node, and a rectangle denotes a leaf node.HiCutsbuilds a decision tree using local optimization decisions at each node to choose the next dimension to cut, and how many cuts to make in the chosen dimension.The HyperCuts[10] algorithm, on the other hand, allows cutting on multiple fields per step, resulting in a fatter and shorter decision tree.

Rather than perform the rule set partitioning and the decision tree construction in two phases, they are combined efficiently in the algorithm. The outcome of the algorithm is multiple decision trees, which is called decision forest. The rule set is partitioned dynamically during the construction of each decision tree. At each node, the set of fields to cut and the number of cuts performed on each field should be figured out. The maximum number of cuts at each node is restricted to be 64. For the port fields, the precise cut points instead of the number of cuts are needed to be determined. Since more bits are needed to store the cut points than to store the number of cuts, the number of cuts on port fields is restricted to be at most 2.The algorithm being used here, differs from simplified HiCuts[11] and HyperCuts[12] in two aspects. First, when the port fields are selected to cut, the cut point which results in

the least rule duplication is chosen. Second, after the cutting method is determined, the rules whose duplication counts are the largest among all the rules covered by the current node is selected, and pushed into the internal rule list of the current node, until the internal rule list becomes full.

After rule set partitioning, the rule duplication due to wildcard fields will be reduced. However, the HiCuts/HyperCuts algorithm may still suffer from rule duplication due to its own inefficiency. As shown in Fig.2 and Fig.3 , rules R1, R2, and R4 are replicated into multiple child nodes, in both HiCuts and HyperCuts trees.While building the decision tree, rule duplication can be identified to come from two sources: 1) overlapping between different rules and 2) evenly cutting on all fields.

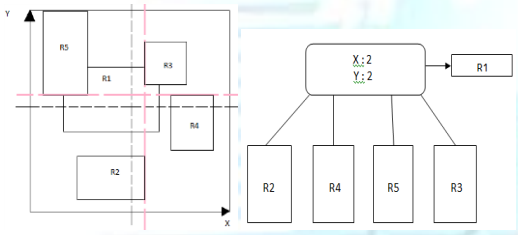

Fig 4:Reducing rule duplication in HyperCuts tree.

Accordingly, two optimization techniques are proposed, called rule overlap reduction and precise range cutting.

• Rule overlap reduction: The rules (e.g., R1 shown in Fig. 4) which will be replicated into child nodes are stored in a list attached to each internal node. These rule lists are called internal rule lists.

• Precise range cutting: Assuming both X and Y in Fig. 4 are port fields, the cutting points which result in the minimum number of rule duplication is selected, instead of deciding the number of cuts for this field.

6.ARCHITECHTURE

The components included in the design are mainly address counter, SRAM and packet classifier. The input stream of data is given to the components in a parallel fashion. Address counter can be defined as a counter which increments an initial memory address as a block of data is being transferred into the memory locations indicated by the counter. Static random-access memory (SRAM) is a type of semiconductor memory that uses bistable latching circuitry to store each bit. The term static differentiates it from dynamic RAM (DRAM) which must be periodically refreshed. SRAM exhibits data reminisce, but is still volatile in the conventional sense that data is eventually lost when the memory is not powered.A collection of rules can be called a classifier. When a stream of bits is given as input to the design, the classifier matches the rules stored in it to the incoming bits. If a matching rule is found, it gives a *valid* high or else a *valid* low. If the incoming bits matches with one or more rules, each header fields are further matched to the predefined rules, and the rule with the perfect matching is selected. Once the classification is done by the packet classifier, the outputs received are SA, DA, EthrType, SrcIP, DstIP, SrcPrt, and DstPrt. All these outputs help us to identify the packet. In this project, five packets can be identified since the rules to those packets are predefined.

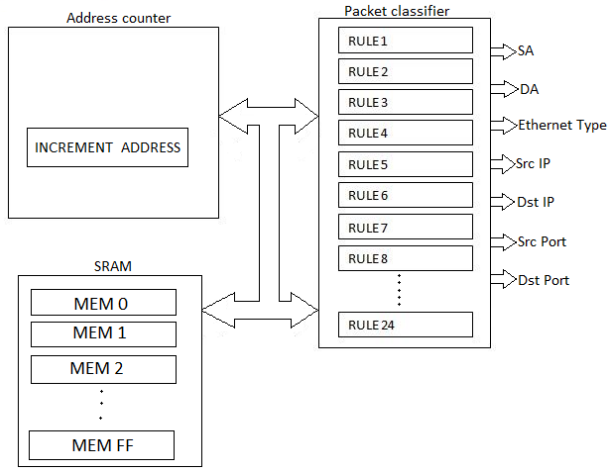

Fig 5: Architectural design

7.SIMULATION RESULTS

The packet input is given as a single bit which includes the tuple information. From the simulation results, it could be seen that each packet is identified and each tuple information is recognized from the packet input. After classification is done by the packet classifier, the outputs received are SA, DA, EthrType, SrcIP, DstIP, SrcPrt, and DstPrt. These outputs are helpful in identifying the packet.It also shows if the packet is a valid packet or not.

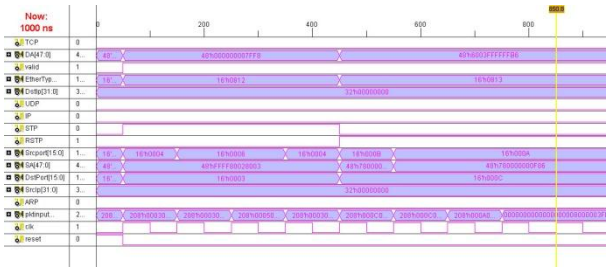

Fig 5: simulation result

8.CONCLUSION& FUTURE WORK

Hence the next – generation packet classification problems where 5-tuple packet header fields would be classified is done and several optimization techniques were proposed to reduce the memory requirement of the state –of- the – art decision-tree-based packet classification algorithm.The packets are classified according to the predefined rules based on the tuples, where the tuples are identified and presented in the output.The future work includes 12-tuple packet classification based on pre-defined rules and porting the design into real system and evaluating its performance under real-life scenarios such as dynamic rule updates.

REFERENCES

[1]Weirong Jiang and Viktor K. Prasanna, "Scalable Packet Classification on FPGA". IEEE transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 9, September 2012.

[2] K. Lakshminarayanan, A. Rangarajan, and S. Venkatachary, "Algorithms for advanced packet classification with ternary CAMs," in Proc.SIGCOMM, 2005, pp. 193–204

[3] H. Song and J. W. Lockwood, "Efficient packet classification for network intrusion detection using FPGA," in Proc. FPGA, 2005, pp. 238–245.

[4]P. Gupta and N. McKeown, "Algorithms for packet classification," IEEE Network, vol. 15, no. 2, pp. 24–32, 2001.

[5] S. Dharmapurikar, H. Song, J. S. Turner, and J. W. Lockwood, "Fast packet classification using bloom filters," in Proc. ANCS, 2006, pp.61–70.

[6] W. Jiang and V. K. Prasanna, "Large-scale wire-speed packet classification on FPGAs," in Proc. FPGA, 2009, pp. 219–228.

[7]M. E. Kounavis, A. Kumar, R. Yavatkar, and H. Vin, "Two stage packet classification using most specific filter matching and transport level sharing," Comput. Netw., vol. 51, no. 18, pp. 4951–4978, 2007. [8] P. Gupta and N. McKeown, "Classifying packets with hierarchical intelligent cuttings," IEEE Micro, vol. 20, no. 1, pp. 34–41, 2000.

[9] T. V. Lakshman and D. Stiliadis, "High-speed policy-based packet forwarding using efficient multi-dimensional range matching," in Proc. SIGCOMM, 1998, pp. 203–214.

[10]S. Singh, F. Baboescu, G. Varghese, and J. Wang, "Packet classification using multidimensional cutting," in Proc. SIGCOMM, 2003, pp.213–224.

[11]I. Papaefstathiou and V. Papaefstathiou, "Memory-efficient 5D packet classification at 40 Gbps," in Proc. INFOCOM, 2007, pp. 1370–1378

[12]A. Kennedy, X.Wang, Z. Liu, and B. Liu, "Low power architecture for high speed packet classification," in Proc. ANCS, 2008, pp. 131–140.