Packet Classification Using Hierarchical Intelligent Cuttings

Sai Likitha Lagudu
COE14B033

Guide : Dr. Noor Mahammad

IIITDM Kancheepuram

April 26, 2018

Introduction

Packet classification - A fundamental processing pattern of modern networking devices.

- The process of categorizing packets by applying rules in classifier in an Internet router is called packet classification.
- Applied on headers of an incoming packet.

Packet classification objective

In packet classification , A rule R has F components, and the f th component of rule R,referred to as R[f], is a range match expression on the f th field of the packet header. A packet P is said to match a particular rule , if $\forall$f $\in$ [1,F], the f th field of the header of P satisfies the range expression R[f]. If a packet P matches multiple rules, the matching rule with the highest priority is returned.

Heuristic algorithm - HiCuts

- The HiCut algorithm firstly preprocesses the classifier (removing redundant rules) to build a decision tree data structure.
- When a packet arrives, the decision tree is traversed to find a leaf node, where the leaves are chosen when the search tree is built.
- The leaf node contains some rules, which are then searched linearly.

Terminology

- B(v): With each internal node v of a k-dimensional classifier, box B(v), is a k-tuple of ranges or intervals: ([l 1 :r 1], [l 2 :r 2],...).
- C(v): The cut C(v) is defined by a dimension d, cutting across the node v.
- np(C): the number of times that box B(v) is cut in dimension d i.e. cuts in the interval [l d :r d]. The cut thus divides B(v) into smaller boxes which are then associated with the children of v.
- R(v): Set of rules on node v.
- binth: Minimum number of rules that a leaf node should span.

HiCuts

- For a given rule-set, represent the entire rules at the root node v.
- Based on the length of the fields choose the Cut C such that it is divided into equal sized intervals each such interval represents the child of the cut node.
- Now find the set of rules spanning the child node.
- The process of cutting is performed at each level, and recursively on the child nodes of that level, until the number of rules in the box associated with each node follow below a threshold (which we will refer to as binth). A node with fewer than binth rules is not partitioned further and becomes a leaf of the tree.
- Now perform the linear search on the rules in the leafnode.

Example 1

Rules	Xrange	Yrange	Action
R1	0-31	0-255	A1
R2	0-255	128-131	A2
R3	64-71	128-255	A3
R4	67-67	0-127	A4
R5	64-71	0-15	A5
R6	128-191	4-131	A6
R7	192-192	0-255	A7

Figure 1: Classifier

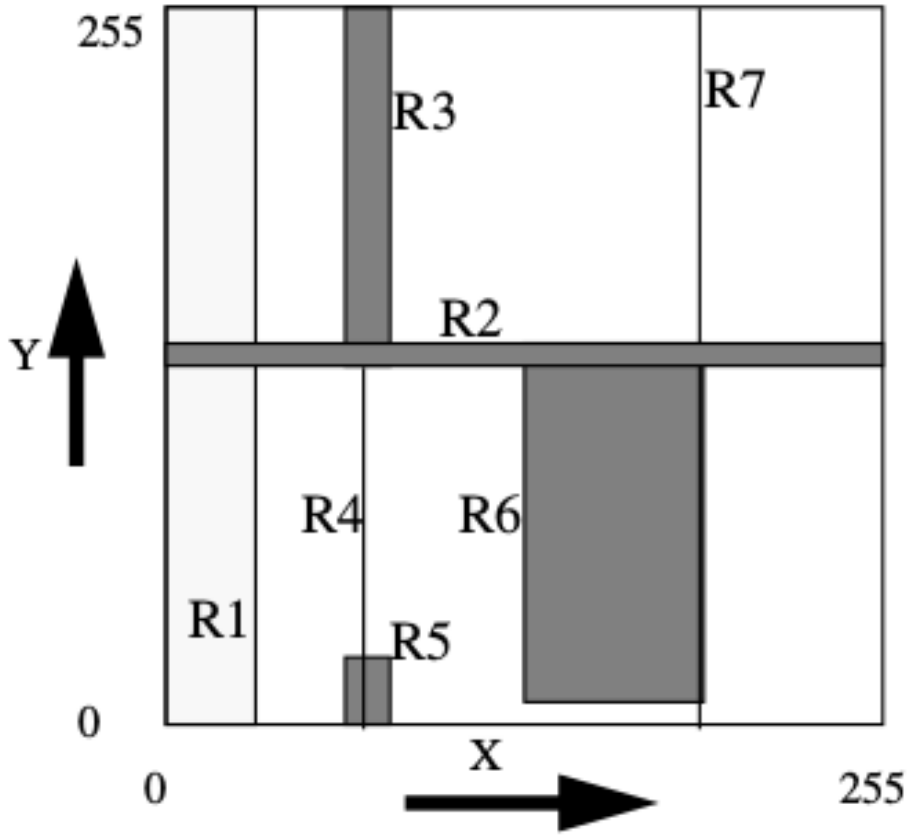


Figure 2: Geometric representation

Example

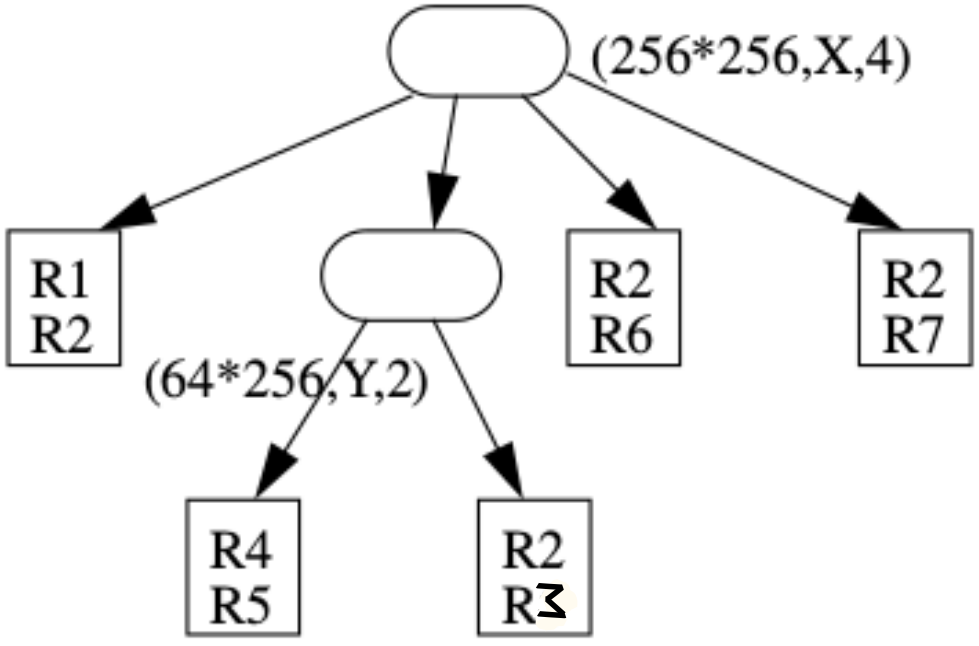


Figure 3: Decision tree

Example

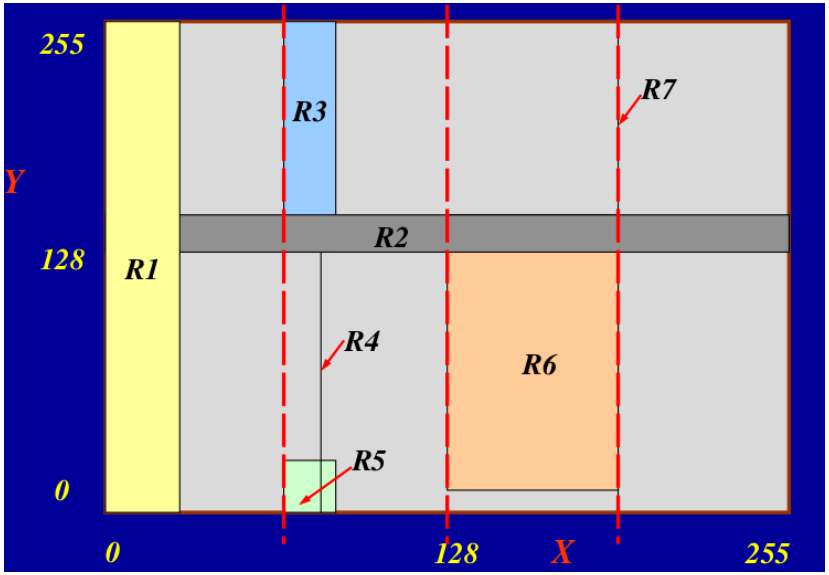
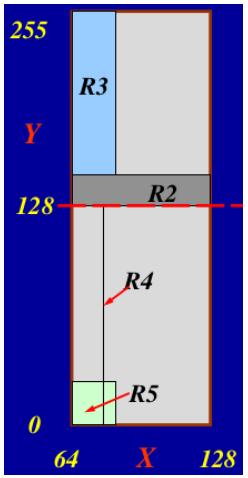


Figure 4: Cuttings



Methodology

HiCuts in higher dimensions

- Real life classifiers use five dimensions for identifying the flow. They are source-address, destination-address, source-port, destination-port and protocol.
- Using the original Hi- Cuts algorithm for five dimensions to minimize the number of rules, involves more pre-processing time, instead of that five dimensional classifier can be splitted.
- As the developed algorithm can be implemented in Parallel way we can divide into two 2-dimensional classifiers and a one 1-dimensional classifier

Table 1: Rule-Set

Rules	SIP/Mask	DIP/Mask	SP	DP	Protocol	Action
R1	150.160.28.96 /255.255.255.255	191.23.64.0/ 255.255.255.255	80	60	tcp	accept
R2	150.160.28.0/ 255.255.255.0	191.23.64.0/ 255.255.255.0	80	60	udp	deny
R3	164.142.36.20/ 255.255.0.0	148.165.222.3/ 255.255.0.0	100	60	udp	deny
R4	185.96.3.3/ 255.255.255.0	152.36.2.1/ 255.0.0.0	100	60	tcp	accept
R5	150.160.28.0/ 255.255.255.0	191.23.64.0/ 255.255.255.255	70	60	tcp	accept
R6	163.96.31.2/ 255.0.0.0	158.65.36.4/ 255.0.0.0	100	80	udp	accept
R7	148..78.63.4/ 255.255.255.255	123.68.96.123/ 255.255.0.0	80	60	tcp	deny
R8	185.26.36.8/ 255.255.0.0	186.67.23.5/ 255.0.0.0	80	60	udp	deny

Table 2: Packet Flow

Packets	SIP	DIP	SP	DP	Protocol
P1	150.60.28.96	191.23.64.0	80	60	tcp

Figure 5: Rule-Set

Using two 2-dimensional classifiers and a one 1-dimensional classifier

- From the given 5-dimensional classifier , the first two fields can be assigned to one 2- dimensional classifier and the next two fields can be assigned to the other 2-dimensional classifier, and the remaining one field to the 1-dimensional classifier.
- Using HiCuts the tree decision structure is constructed for both 2-dimensional classifiers. a simple linear search can be applied to match the remaining one field
- Set-A and Set-B represent rules from the 2-dimensional classifiers. Set-C represent rules from the 1-dimensional classifier
- The objective is to identify the common rules (represent a match on five fields) from these sets

Bit-Vector

- For each rule in set assign value one to the corresponding index of the vector and for the rules which are not present assign zero to it's corresponding indices.
- Set of rules matched for the source-address and destination-address are A={R1, R2, R5}. and the corresponding bit vector for this set is 11001000.
- Set of rules matched for the source-port and destination-port are B={R1, R2, R7, R8} and the corresponding bit vector for this set is 11000011.
- Set of rules matched for the protocol field are C={R1, R4, R5, R7} and the corresponding bit vector for this set is 10011010.

Bit-Vector

1	1	0	0	1	0	0	0
1	1	0	0	0	0	1	1
1	1	0	0	0	0	0	0

1	1	0	0	0	0	0	0
1	0	0	1	1	0	1	0
1	0	0	0	0	0	0	0

Figure 6: Bit vectors

Count-Increment

- Set of rules matched for the source-address and destination-address are A={R1, R2, R5}.At this stage the values updated are A[1] = 1 , A[2] = 1 ,A[5] = 1.
- Set of rules matched for the source-port and destination-port are B={R1, R2, R7, R8} , the values updated are A[1] = 2, A[2] = 2, A[7] = 1, A[8] = 1.
- Set of rules matched for the protocol field are C={R1, R4, R5, R7} , the values updated are A[1] = 3 , A[7] = 2, A[4]= 1, A[5] = 2.
- The value three is searched from the array.In this example R1 has it's corresponding array value as three. so the packet matches rule R1 and the action specified by this rule is performed.

HICUTS IMPLEMENTATION AND OUTPUTS

- Algorithm is implemented using C language. The algorithm includes the preprocessing of the classifier and for identifying the fields it need to traverse the decision tree.
- Complexity of the algorithm is height of the decision tree classifier. The serial code is implemented using C structures.

HICUTS IMPLEMENTATION AND OUTPUTS

- The processes which can be done parallely are found out. In preprocessing algorithm the leaf nodes spanning the rules can be searched parallely.
- For each leaf node a thread is assigned for finding the rules.
- On reaching the leaf node if the node spans more than one rule we use linear search which can be parallelised.
- The two 2d classifiers can be run parallel , each classifier is assigned one thread and internally for the classifier algorithm threads are assigned as described.

HICUTS IMPLEMENTATION AND OUTPUTS

```
dakshaja@Lenovo:~/Desktop$ ./a.out
HICUT--------IP-----------------------------------------------------------------------------------------------------------------------

T = 4
Node 0, interval  X = 0 63 Y = 0 255
Node 0, Rule 0
Node 0, Rule 1
Node 1, interval  X = 64 127 Y = 0 255
Node 1, Rule 1
Node 1, Rule 2
Node 1, Rule 3
Node 1, Rule 4
Node 2, interval  X = 128 191 Y = 0 255
Node 2, Rule 1
Node 2, Rule 5
Node 3, interval  X = 192 255 Y = 0 255
Node 3, Rule 1
Node 3, Rule 6
============================================================

Node 0, interval  X = 0 255 Y = 0 63
Node 0, Rule 0
Node 0, Rule 3
Node 0, Rule 4
Node 0, Rule 5
Node 0, Rule 6
Node 1, interval  X = 0 255 Y = 64 127
Node 1, Rule 0
Node 1, Rule 3
Node 1, Rule 5
Node 1, Rule 6
Node 2, interval  X = 0 255 Y = 128 191
Node 2, Rule 0
Node 2, Rule 1
Node 2, Rule 2
Node 2, Rule 5
Node 2, Rule 6
Node 3, interval  X = 0 255 Y = 192 255
Node 3, Rule 0
Node 3, Rule 2
Node 3, Rule 6
============================================================


X CUT IS BETTER


>>>>>Splitting node: 1

T = 2
Node 0, interval  X = 64 95 Y = 0 255
Node 0, Rule 1
Node 0, Rule 2
Node 0, Rule 3
Node 0, Rule 4
Node 1, interval  X = 96 127 Y = 0 255
Node 1, Rule 1
============================================================

Node 0, interval  X = 64 127 Y = 0 127
Node 0, Rule 3
```

```
X CUT IS BETTER


>>>>>Splitting node: 1

r = 2
Node 0, interval  X = 64 95 Y = 0 255
Node 0, Rule 1
Node 0, Rule 2
Node 0, Rule 3
Node 0, Rule 4
Node 1, interval  X = 96 127 Y = 0 255
Node 1, Rule 1
============================================================

Node 0, interval  X = 64 127 Y = 0 127
Node 0, Rule 3
Node 0, Rule 4
Node 1, interval  X = 64 127 Y = 128 255
Node 1, Rule 1
Node 1, Rule 2
============================================================


Y CUT IS BETTER


HICUT--------PORT-------------------------------------------------------------------------------------------------------------------------------

r = 4
Node 0, interval  X = 0 63 Y = 0 255
Node 0, Rule 5
Node 0, Rule 6
Node 1, interval  X = 64 127 Y = 0 255
Node 1, Rule 2
Node 1, Rule 3
Node 1, Rule 4
Node 1, Rule 5
Node 2, interval  X = 128 191 Y = 0 255
Node 2, Rule 1
Node 2, Rule 5
Node 3, interval  X = 192 255 Y = 0 255
Node 3, Rule 0
Node 3, Rule 5
============================================================

Node 0, interval  X = 0 255 Y = 0 63
Node 0, Rule 0
Node 0, Rule 1
Node 0, Rule 2
Node 0, Rule 3
Node 0, Rule 6
Node 1, interval  X = 0 255 Y = 64 127
Node 1, Rule 0
Node 1, Rule 1
Node 1, Rule 3
Node 1, Rule 6
Node 2, interval  X = 0 255 Y = 128 191
Node 2, Rule 0
Node 2, Rule 1
Node 2, Rule 4
```

```
Node 0, Rule 0
Node 0, Rule 1
Node 0, Rule 2
Node 0, Rule 3
Node 0, Rule 6
Node 1, interval  X = 0 255 Y = 64 127
Node 1, Rule 0
Node 1, Rule 1
Node 1, Rule 3
Node 1, Rule 6
Node 2, interval  X = 0 255 Y = 128 191
Node 2, Rule 0
Node 2, Rule 1
Node 2, Rule 4
Node 2, Rule 5
Node 2, Rule 6
Node 3, interval  X = 0 255 Y = 192 255
Node 3, Rule 0
Node 3, Rule 4
Node 3, Rule 6
============================================================


X CUT IS BETTER


>>>>>Splitting node: 1

r = 2
Node 0, interval  X = 64 95 Y = 0 255
Node 0, Rule 2
Node 0, Rule 3
Node 0, Rule 4
Node 0, Rule 5
Node 1, interval  X = 96 127 Y = 0 255
Node 1, Rule 5
============================================================

Node 0, interval  X = 64 127 Y = 0 127
Node 0, Rule 2
Node 0, Rule 3
Node 1, interval  X = 64 127 Y = 128 255
Node 1, Rule 4
Node 1, Rule 5
============================================================


Y CUT IS BETTER


IP HICUT  MATCHED RULES   R3  R4

PORT HICUT  MATCHED RULES   R2  R3

PROTOCOL HICUT  MATCHED RULES   R1  R3  R4


IP ARRAY  0 0 0 1 1 0 0
Port ARRAY  0 0 1 1 0 0 0
PROTOCOL ARRAY  0 1 0 1 1 0 0
MATCHED ARRAY  0 0 0 1 0 0 0
MATCHED RULES FOR ALL FIELDS R3  dakshaja@Lenovo:~/Desktop$
```

Performance of HiCuts

HiCuts reduces the number of rules to be searched linearly, when the ruleset is extremely large it gives best performance. It balances both the processing time and memory requirement. Checking the constraints the best decision tree is constructed and used for processing the packet.

Conclusion

- The concept of packet classification have been explored in this project. The real time implementation of HiCuts is analyzed and observed that it takes more preprocessing time when it is implemented for 5-dimensional ruleset.
- To reduce preprocessing we used two 2-dimensional classifiers and one 1-dimensional classifier are considered.
- To combine the rules from these three classifiers bit-vector and count-increment are employed. The modules which can be parallelized in HICUTS are also identified.

Thank You