ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Packet Classification Algorithm Using Boundary Cutting and Improving Search Performance

Nazma
Student,
Department of CSE,
Shadan Women's College of Engineering & Technology, Khairatabad, Hyderabad.

Syeda Nusrath Fatima
Associate Professor,
Department of CSE,
Shadan Women's College of Engineering & Technology, Khairatabad, Hyderabad.

Abstract-

Several efforts were made in the existing solutions to identify a successful packet classification solution. A variety of decision-tree-based packet classification algorithms were analysed in our works. Earlier decision tree algorithms chooses field as well as number of cuts based on a nearby optimized decision, which compromises search speed as well as memory requirement. Algorithms of decision tree make possible maximum priority match and multi-match packet classification.

New applications of networks have demanded a multi-match packet classification in which the entire matching results along with highest-priority matching rule have to be returned and it is essential to discover competent algorithms to resolve classification problems. A new resourceful packet classification algorithm was introduced in our work named as boundary cutting. Mainly conventional applications of packet classification necessitate the maximum priority matching.

The algorithm that was introduced has two principal advantages. Boundary cutting of projected algorithm is more effectual than that of earlier algorithms as it is based on rule boundaries to a certain extent than permanent intervals thus, amount of necessary memory is considerably reduced. Although boundary cutting loses indexing capability at internal nodes binary search recommend better-quality search performance.

Keywords:

Decision tree algorithms, Packet classification, Boundary cutting, Priority matching, Binary search.

INTRODUCTION:

Classification of packet is an important function providing value-added services in Internet routers. Multi match classification notion is fetching an important research item because of rising need for network protection, for instance network intrusion detection systems and worm detection. Usage of a high bandwidth and a tiny on-chip memory whereas rule database for packet classification resides in slower as well as superior capacity offchip memory by appropriate partitioning is enviable.

The amount of memory necessary to accumulate packet classification table must be considered. Performance metrics in support of packet classification algorithms mainly comprise processing speed, while packet classification has to be carried out in wire-speed for each incoming packet. Processing speed is assessed by means of number of off-chip memory accesses necessary to find out class of a packet since it is the slowest procedure in packet classification.

Our study analyzes a variety of decision-tree-based packet classification algorithms. Previous decision tree algorithms for instance HiCuts as well as Hyper Cuts select field as well as number of cuts based on a nearby optimized decision, which compromises search speed as well as memory prerequisite. T

his procedure requires a reasonable amount of preprocessing, which involves complex heuristics associated to each given rule set . If a decision tree is appropriately partitioned with the intention that the internal tree nodes are accumulated in an on-chip memory and a huge rule database is accumulated in an off-chip memory, decision tree algorithm can make available extremely high-speed search performance.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Decision tree algorithms obviously facilitate highest-priority match and multi-match packet classification. Innovative network applications have in recent times demanded a multi-match packet classification in which the entire matching results along with highest-priority matching rule have to be returned. It is essential to discover competent algorithms to resolve classification problems. In our work a novel system of packet classification on basis of boundary cutting was put forward which finds out space that each rule performs cutting consistent with space boundary.New proficient packet classification algorithm by means of boundary cutting is projected which finds out space that each rule performs cutting consistent with space boundary. Hence, cutting in projected algorithm is deterministic to a certain extent than involving difficult heuristics, and it is more effectual in providing enhanced search performance and more competent in memory requirement.

HiCuts and HyperCuts algorithms carry out cutting based on a fixed interval, and hence partitioning is unsuccessful in dropping the number of rules that belong to a subspace. In our work we put forward a deterministic cutting algorithm on basis of each rule boundary, named as boundary cutting (BC) algorithm. When the cutting of a prefix plane consistent with rule boundaries is carried out, starting and ending boundaries of each rule are used for cutting, however cutting by either is enough as decision tree algorithms usually search for a subspace in which an input packet belong and headers of specified input are evaluated for entire fields to rules belonging to subspace. The cuts at each internal node of boundary cutting decision tree do not contain permanent intervals. Consequently, at each internal node of tree, a binary search is necessary to find out proper edge to follow for a specified input.

OVERVIEW OF EARLIER DECISION TREE ALGORITHMS:

Packet classification can be formally defined as follows : Packet P matches rule Rk , for k=1.....N , if all the header fields Fd, for d=1....d, of the packet match the corresponding fields in Rk , where N is the number of rules and D is the number of fields. If a packet matches multiple rules, the rule with the highest priority is returned for a single-best-match packet classification problem and the list of matching rules is returned for the multimatch packet classification problem.

Rule sets are generally composed of five fields. The first two fields are related to source and destination prefixes and require prefix match operation. The next two fields are related to source and destination port ranges (or numbers), which require range match. The last field is related to protocol type and requires an exact match to determine the number of cuts for the chosen field.

The binth is a predetermined number of rules included in the leaf nodes of the decision tree for a linear search. For simplicity, considering a two-dimensional (2-D) plane composed of the first two prefix fields, a rule can be represented by an area. the areas that each rule covers in a prefix plane for a given 2-D example rule set. As shown, a rule with) lengths in and fields covers the area of , where is the maximum length of the field (W is 32 in IPv4).

EXISTING SYSTEM:

Our study analyzed various decision-tree-based packet classification algorithms. If a decision tree is properly partitioned so that the internal tree nodes are stored in an on-chip memory and a large rule database is stored in an off-chip memory, the decision tree algorithm can provide very high-speed search performance. Moreover, decision tree algorithms naturally enable both the highest-priority match and the multimatch packet classification. Earlier decision tree algorithms such as HiCuts and HyperCuts select the field and number of cuts based on a locally optimized decision, which compromises the search speed and the memory requirement. This process requires a fair amount of preprocessing, which involves complicated heuristics related to each given rule set.

DISADVANTAGES:

1.The computation required for the pre-processing consumes much memory and construction time, making it difficult.

2.Algorithms to be extended to large rule sets because of memory problems in building the decision trees.

3.The cutting is based on a fixed interval, which does not consider the actual space that each rule covers; hence it is ineffective.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

PROPOSED SYSTEM:

» In this paper, we propose a new efficient packet classification algorithm based on boundary cutting. Cutting in the proposed algorithm is based on the disjoint space covered by each rule.
» Hence, the packet classification table using the proposed algorithm is deterministically built and does not require the complicated heuristics used by earlier decision tree algorithms.

ADVANTAGES OF PROPOSED SYSTEM:

1.The boundary cutting of the proposed algorithm is more effective than that of earlier algorithms since it is based on rule boundaries rather than fixed intervals. Hence, the amount of required memory is significantly reduced.

2.Although BC loses the indexing ability at internal nodes, the binary search at internal nodes provides good search performance

IMPLEMENTATION:
Building a BC Decision Tree:

When the cutting of a prefix plane according to rule boundaries is performed, both the starting and the ending boundaries of each rule can be used for cutting, but cutting by either is sufficient since decision tree algorithms generally search for a subspace in which an input packet belongs and the headers of the given input are compared for entire fields to the rules belonging to the subspace (represented by a leaf node of the decision tree).

Searching in the Boundary Cutting:

The cuts at each internal node of the BC decision tree do not have fixed intervals. Hence, at each internal node of the tree, a binary search is required to determine the proper edge to follow for a given input.During the binary search, the pointer to the child node is remembered when the input matches the entry value or when the input is larger than the entry value.Consider an input packet with headers (000110, 111100, 19, 23, TCP), for example; since is used at the root node, a binary search using the header of the given input is performed.

The header 000110 is compared to the middle entry of the root node, which is 010000. Since the input is smaller, the search proceeds to the smaller half and compares the input to the entry 000100. Since the input is larger, the child pointer (the second edge) is remembered, and the search proceeds to a larger half. The input is compared to 001000, and it is found to be smaller, but there is no entry to proceed in a smaller half. Hence, the search follows the remembered pointer, the second edge. At the second level, by performing a binary search, the last edge is selected for the header 111100. The linear search,which is the same as that in the HiCuts or HyperCuts algorithm, is performed for rules stored in the leaf node.

Selective Boundary Cutting:

In this module we propose a refined structure for the BC algorithm. The decision tree algorithms including the BC algorithm use binth to determine whether a subspace should become an internal node or a leaf node. In other words, if the number of rules included in a subspace is more than binth, the subspace becomes an internal node; otherwise, it becomes a leaf node. In the BC algorithm, if a subspace becomes an internal node, every starting boundary of the rules included in the subspace is used for cutting. We propose a refined structure using the binth to select or unselect the boundary of a rule at an internal node. In other words, the refined structure activates a rule boundary only when the number of rules included in a partition exceeds the binth.

Data Structure:

There are two different ways of storing rules in decision tree algorithms. The first way separates a rule table from a decision tree. In this case, each rule is stored only once in the rule table, while each leaf node of a decision tree has pointers to the rule table for the rules included in the leaf. The number of rule pointers that each leaf must hold equals the binth. In searching for the best matching rule for a given packet or the list of all matching rules, after a leaf node in the decision tree is reached and the number of rules included in the leaf is identified, extra memory accesses are required to access the rule table. The other way involves storing rules within leaf nodes..

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In this case, search performance is better since extra access to the rule table is avoided, but extra memory overhead is caused due to rule replication. In our simulation in this paper, it is assumed that rules are stored in leaf nodes since the search performance is more important than the required memory.

RELATED WORK:

Packet classification is the core mechanism that enables many networking services on the Internet such as firewall packet filtering and traffic accounting. Using ternary content addressable memories (TCAMs) to perform high-speed packet classification has become the de facto standard in industry. TCAMs classify packets in constant time by comparing a packet with all classification rules of ternary encoding in parallel. Despite their high speed, TCAMs suffer from the well-known range expansion problem. As packet classification rules usually have fields specified as ranges, converting such rules to TCAM-compatible rules may result in an explosive increase in the number of rules. This is not a problem if TCAMs have large capacities. Unfortunately, TCAMs have very limited capacity, and more rules mean more power consumption and more heat generation for TCAMs.

Even worse, the number of rules in packet classifiers has been increasing rapidly with the growing number of services deployed on the Internet. In this paper, we consider the following problem: given a packet classifier, how can we generate another semantically equivalent packet classifier that requires the least number of TCAM entries? In this paper, we propose a systematic approach, the TCAM Razor, that is effective, efficient, and practical. TCAMs impose a high cost on cooling systems. TCAMs also cost about 30 * more per bit of storage than double-data-rate SRAMs. Moreover, for an -bit port range field, it may require 2(L-1) TCAM entries, making the exploration of algorithmic alternatives necessary.Packet classification can be formally defined as follows : Packet P matches rule Rk , for k=1.....N , if all the header fields Fd, for d=1....d, of the packet match the corresponding fields in Rk , where N is the number of rules and D is the number of fields. If a packet matches multiple rules, the rule with the highest priority is returned for a single-best-match packet classification problem and the list of matching rules is returned for the multimatch packet classification problem.

Rule sets are generally composed of five fields. The first two fields are related to source and destination prefixes and require prefix match operation. The next two fields are related to source and destination port ranges (or numbers), which require range match. The last field is related to protocol type and requires an exact match to determine the number of cuts for the chosen field. The binth is a predetermined number of rules included in the leaf nodes of the decision tree for a linear search.

HiCuts :

Each rule defines a five-dimensional hypercube in a five-dimensional space, and each packet header defines a point in the space. The HiCuts algorithm [8] recursively cuts the space into subspaces using one dimension per step. Each subspace ends up with fewer overlapped rule hypercubes that allow for a linear search. In the construction of a decision tree of the HiCuts algorithm, a large number of cuts consumes more storage, and a small number of cuts causes slower search performance. It is challenging to balance the storage requirement and the search speed. The HiCuts algorithm uses two parameters, a space factor (spfac) and a threshold (binth), in tuning the heuristics, which trade off the depth of the decision tree against the memory amount.

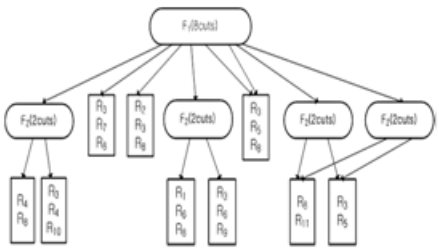

Fig:-HiCuts algorithm.

For simplicity, considering a two-dimensional (2-D) plane composed of the first two prefix fields, a rule can be represented by an area. the areas that each rule covers in a prefix plane for a given 2-D example rule set. As shown, a rule with (i,j) lengths F1 in F2 and fields covers the area of 2(w-i) * 2(w-j), where W is the maximum length of the field (W is 32 in IPv4).

HyperCuts:

While the HiCuts algorithm only considers one field at a time for selecting cut dimension, the HyperCuts

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

algorithm considers multiple fields at a time. For the same example set, the decision tree of the HyperCut-salgorithm . The spfac and binth are set as 1.5 and 3, respectively. As shown at the root node, the and fields are used simultaneously for cutting. Note that each edge of the root node represents the F1 and F2 bit combination of 00, 10, 01, and 11, respectively, which is one bit in the first field followed by one bit in the second field.

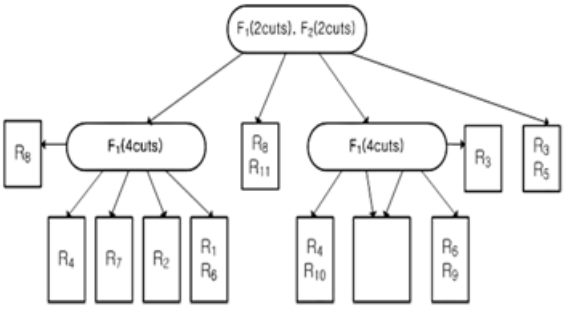

Fig:-hypercuts

Decision Tree Characteristics:

In the second step, we constructed decision trees of BC, SBC, HiCuts, and HyperCuts algorithms setting binth as the identi- fied lower bound value. Cutting is recursively performed and stopped if the number of rules included in a subspace is less than or equal to the lower bound binth. The performances of the HiCuts and HyperCuts algorithms depend on spfac as well.A decision tree is built without performing the optimization, and hence each leaf node has a number of rules less than or equal to binth. From the bottom of the tree, each subtree is examined to find common rules in every child, and those rules are pushed upward into the root of the subtree. This processing is repeated until the root of the decision tree is visited. In this case, the total number of rules located at internal nodes along a path from the root to a leaf node is guaranteed to be at most binth. Hence, the search performance is not degraded much by the optimization, but the rule replication becomes significant.

	Entry width	No. of bits	Field
Boundary Cutting	Node (3.625 bytes)	3	field selection
		$\lceil \log_2 N_e \rceil$	no. of entries
	Entry (7.75 bytes)	32	boundary value
		$\lceil \log_2 N_s \rceil$	child pointer
Selective Boundary Cutting	Node (3.5 bytes)	3	field selection
		$\lceil \log_2 N_e \rceil$	no. of entries
	Entry (7.625 bytes)	32	boundary value
		$\lceil \log_2 N_s \rceil$	child pointer
HiCuts	72.375 bytes	3	field selection
		9	number of cuts (512 cuts maximum)
		512	child map
		$\lceil \log_2 N_i \rceil$	pointer to the first internal child
		$\lceil \log_2 N_s \rceil$	pointer to the first leaf child
HyperCuts	72 bytes	5	field selection
		13	number of cuts in multiple fields (512 cuts maximum)
		512	child map
		$\lceil \log_2 N_i \rceil$	pointer to the first internal child
		$\lceil \log_2 N_s \rceil$	pointer to the first leaf child

Fig:-DATA STRUCTURES OF INTERNAL NODES IN EACH DECISION TREE

Rule	N	Src Prefix		Dst Prefix		Src Port		Dst Port		Protocol	
		No.	Rate(%)	No.	Rate(%)	No.	Rate(%)	No.	Rate(%)	No.	Rate(%)
ACL1K	958	3	0.31	1	0.10	958	100	283	29.54	68	7.10
ACL5K	4660	8	0.17	11	0.24	4660	100	1413	30.32	389	8.35
ACL10K	9735	19	0.20	30	0.31	9735	100	3008	30.90	786	8.07
ACL50K	48420	35	0.07	122	0.25	48420	100	9491	19.60	2446	5.05
ACL100K	95340	60	0.06	229	0.24	95340	100	16491	17.30	4232	4.44
IPC1K	988	27	2.73	12	1.21	835	84.51	522	52.83	330	33.40
IPC5K	4468	76	1.70	75	1.68	3615	80.91	2418	54.12	1353	30.28
IPC10K	8112	148	1.82	124	1.53	6373	78.56	4284	52.81	2188	26.97
IPC50K	49047	184	0.38	191	0.39	36344	74.10	23597	48.11	7447	15.18
IPC100K	94370	371	0.39	315	0.33	68292	72.37	43511	46.11	10587	11.22
FW1K	871	32	3.67	132	15.15	347	39.84	871	100	239	27.44
FW5K	3067	114	3.72	339	11.05	1167	38.05	3067	100	800	26.08
FW10K	4351	78	1.79	491	11.28	1527	35.10	4351	100	1073	24.66
FW50K	49163	1700	3.46	1139	2.32	36781	74.81	14764	30.03	700	1.42
FW100K	82473	2277	2.76	1564	1.90	61546	74.63	23502	28.50	1180	1.43

Fig:-CHARACTERISTICS OF RULE SETS IN THE NUMBER AND RATE OF WILDCARDS

CONCLUSIONS:

In this paper, two approaches are given: Clustering and Boundary Cutting Algorithm. In testing phase, dataset is divided into clusters based on similarity. According to that clusters, classification is done to check whether that packet is normal or attack. Boundary cutting algorithm provides high accuracy than clustering as it perform cutting according to the space boundary.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

REFERENCES:

[1] H. J. Chao, "Next generation routers," Proc. IEEE, vol. 90, no. 9, pp.1518–1588, Sep. 2002.

[2] A. X. Liu, C.R.Meiners, andE.Torng, "TCAMrazor: A systematic approach towards minimizing packet classifiers in TCAMs," IEEE/ACM Trans. Netw., vol. 18, no. 2, pp. 490–500, Apr. 2010.

[3] C. R. Meiners, A. X. Liu, and E. Torng, "Topological transformation approaches to TCAM-based packet classification," IEEE/ACM Trans. Netw., vol. 19, no. 1, pp. 237–250, Feb. 2011.

[4] F. Yu and T. V. Lakshnam, "Efficient multimatch packet classification and lookup with TCAM," IEEE Micro, vol. 25, no. 1, pp. 50–59, Jan. –Feb. 2005.

[5] F. Yu, T. V. Lakshman, M. A. Motoyama, and R. H. Katz, "Efficient multimatch packet classification for network security applications," IEEE J. Sel. Areas Commun., vol. 24, no. 10, pp. 1805–1816, Oct. 2006.

[6] H. Yu and R. Mahapatra, "A memory-efficient hashing by multi-predicate bloom filters for packet classification," in Proc. IEEE INFOCOM, 2008, pp. 2467–2475.

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

[8] P. Gupta and N. Mckeown, "Classification using hierarchical intelligent cuttings," IEEE Micro, vol. 20, no. 1, pp. 34–41, Jan.–Feb. 2000.

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

[10] P. Gupta and N. Mckeown, "Algorithms for packet classification," IEEE Netw., vol. 15, no. 2, pp. 24–32, Mar.–Apr. 2001.

[11] B. Vamanan, G. Voskuilen, and T. N. Vijaykumar, "EffiCuts: Optimizing packet classification for memory and throughput," in Proc. ACM SIGCOMM, 2010, pp. 207–218.

[12] H. Song, M. Kodialam, F. Hao, and T. V. Lakshman, "Efficient trie braiding in scalable virtual routers," IEEE/ACM Trans. Netw., vol. 20, no. 5, pp. 1489–1500, Oct. 2012.

[13] J. Treurniet, "A network activity classification schema and its application to scan detection," IEEE/ACM Trans. Netw., vol. 19, no. 5, pp. 1396–1404, Oct. 2011.

[14] L. Choi, H. Kim, S. Ki, and M. H. Kim, "Scalable packet classification through rulebase partitioning using the maximum entropy hashing," IEEE/ACM Trans. Netw., vol. 17, no. 6, pp. 1926–1935, Dec. 2009.

[15] F. Baboescu and G. Varghese, "Scalable packet classification," IEEE/ACM Trans. Netw., vol. 13, no. 1, pp. 2–14, Feb. 2005.

Author's:

Ms.Nazma

has completed her btech -Information technology,from Sri raja rajeswari engineering college, karepalli from JNTUH University, Hyderabad. Presently, she is pursuing her Masters in Computer Science & Engineering from Shadan Women's College of Engineering and Technology, Khairatabad, Hyderabad, T.S, India.

Syeda Nusrath Fatima

has completed her B.tech - IT from osmania university and M.Tech - CSE from JNTUH,hyd.I have 4 yrs of teaching experience from vasavi college of engineering,ibrahimbagh, hyd. Currently working in Shadan college of engineering as an Asst.Prof.