YIDDISH ISSN NO: 0364-4308
Vol 13 Issue 02,April 2023

Study of an Enhanced Cross-Product Method for Packet Classification

Mr Yendluri Ramaiah [1], V ANJAIAH [2]
Assistant professor[1,2]
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
P.B.R.VISVODAYA INSTITUTE OF TECHNOLOGY & SCIENCE
S.P.S.R NELLORE DIST, A.P , INDIA , KAVALI-524201

Abstract

A network forensics machine is required to classify IP packets, and the results of this classification define how the packets are handled in a gigabit network intrusion forensics system. In this study, a novel method for classifying packets is presented, one that makes use of cross-products. Before the addresses are cross-translated, they are separated into several categories. In this research, the large cross-product table was compressed using the no-overlapping partition strategy, which involves merging repeated rows or columns. The effectiveness of our method has been tested in simulated environments.

Keywords:

separation into non-overlapping groups by classifying packets, calculating the cross product, etc.

Introduction

The term "packet classification" refers to the process of categorizing data packets based on its contents and additional information (often referring to the IP packet and transmission layer head)[1].It does this by looking for matching rules that may be used to tell packets apart. The packet forwarding engine uses the categorization results to take the necessary steps to satisfy the users' demands, such as rejecting unlawful grouping, dispatching processing, or utilizing the results as the foundation for routing. The network intrusion forensics system on a gigabit network relies heavily on the collecting, processing, and forwarding of packets. The large data sets produced by computers used to gather evidence highlight the importance of network forensics.A computer that sorts IP packets and then filters them out according to their categorization. Relevant IP packets are extracted from data flows and sent to a forensics storage system for further examination.

Similar Research

work the parallel decomposition-integration method is an example of a class of techniques for classifying packets. When searching in many dimensions, results from several searches are conducted in parallel and then combined to get a best match. Using the rule mapping bitmap, algorithms like BV[2] and ABV[3] search for a match in all dimensions at once, and the match rules are derived from the findings in each dimension. Bitmap size is directly proportional to rule set size. The RFC[4] strategy is a multi-step, simultaneous decomposition-integration process. Each successive level of deconstruction and search uses the previous level's output as input. Improved search

effectiveness is made possible via pipeline technology. However, the algorithm's speed is limited by factors like the number of stages and how to choose the results from the previous stage, and its storage efficiency is not high. The Cross-Product method [5] initially considers all potential prefixes across all dimensions concurrently in order to identify suitable matching criteria. The cross-product method has various advantages over other search algorithms. In the K-dimensional example, it only takes K linear searches and 1 table lookup. The procedure is O(KN) time and O(KN) space intensive. (NK).The amount of space required to accommodate extensive criteria for classification might quickly become prohibitive. The prior cross product method used IP packet address presentation differences to classify data. The addresses are first divided into subsets, from which a cross-product is derived. Due to the overlapping nature of these classes inside a two-dimensional address space, a simple one-dimensional approach to locating a specific location is not achievable. In this analysis, the space is separated into discrete areas. Using a fast one-dimensional searching strategy might potentially shorten the search time. In this work, we reduce the size of the cross-product database by merging rows and columns that are equal using an equivalence class.

ISSN NO: 0364-4308
Vol 13 Issue 02,April 2023

Product enhancement

The prior cross-product method requires K times as many linear searches for K-dimensional spaces. Instead of using a fast one-dimensional search strategy, which might result in address range overlap, a linear search is performed over all dimensions. A single IP packet address may represent numerous units in different dimensions. Since a database search will likely yield several matching rules, it is necessary to identify the rule with the greatest priority. This leads to inefficient searching. The criteria for classification are provided in Table 1. The S,T-type of address is reflected in Table 2's cross-product table.

Table 1 classification rules example

The source address (S)	The destination address (T)	Rules
00*	001*	R1
00*	01*	R2
011*	10*	R3
01*	*	R4
*	10*	R5
*	*	R6

The Cross-product method requires a linear search in S to find the location P (0110, 1000), and the pattern match is 01 *, 011 *, *. Then, 10 *, * is matched after a linear search in T is performed. According to Table2, we may derive a total of six matching rules: R3, R4, R5, and R6. According to the rule's serial number, R3 has the greatest priority and is thus accepted as the optimal matching rule. As a result of the preceding steps, we get insight into both spatial and temporal dimensions (S,T), despite the fact that the rule-defined ranges are incompatible.

S \ T	10*	01*	001*	*
01*	R6	R6	R6	R4
011*	R3	R6	R6	R6
00*	R6	R2	R1	R6
*	R5	R6	R6	R6

Figure 1 shows the area graph of the rules base C. R3, R4, R5, and R6 have overlapping areas (the rectangular areas where R3 is in).When searching for the match rules of H, four matching rules will be found. At the same time, R6 has overlapping parts with all other areas. So R6 would be matched at any time when a point matches other rules (R1-R5).

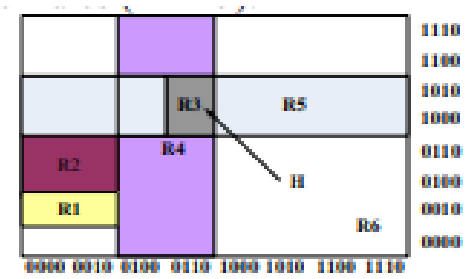

Fig 1 areas of Cross-product

In order to get rid of this overlapping area, we adopt a segmentation method. Figure 2 is the areas graph of cross-product without overlapping area. The X axis is divided into four intervals: X1, X2, X3, and X4. The Y axis is divided into five sections: Y1, Y2, Y3, Y4, and Y5. Then this plane is divided into 20 areas. The 20 areas don't overlap with each other. The points in each area have the same properties: either all are matched with R or all are not matched at all. For matching rules, due to the different points of the area have the same properties, so their best match rules are the same. Before the look-up table is created, the best matching rule should be calculated firstly. Only the best matching rules may be stored.

ISSN NO: 0364-4308
Vol 13 Issue 02,April 2023

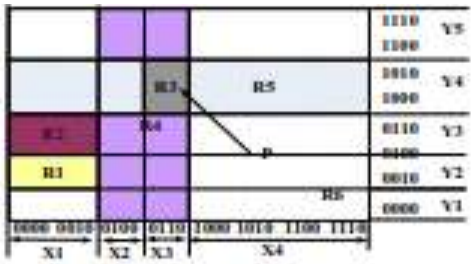

Fig 2 areas of Cross-product without overlapping

Because no two intervals in a matching search will ever touch, we can utilize quick one-dimensional algorithms like the binary searching algorithm or the binary searching tree method. As a result, we can reduce the time complexity from O(KN) to O(Klongs). There is also no need to prioritize the matching rules since there is just one in the look-up database. The optimal matching rule is identified. As an illustration of the rule set, consider Table 1. Using the non-overlapping regions shown in figure2, a cross-product table is constructed from the segmented address S,T. Only a binary search in the source address S is required to find the matching source address, X3, when searching for the point H (0110,1000).Then, we compare T with the destination address, and because Y4 is a perfect match, we get the best set of matching rules, R3.

Table 3 Cross-Product without overlapping

S \ T	Y1	Y2	Y3	Y4	Y5
X1	R6	R1	R2	R5	R6
X2	R4	R4	R4	R4	R4
X3	R4	R4	R4	R3	R4
X4	R6	R6	R6	R6	R6

But the above method's main defect is increasing the size of Cross-product table. In order to reduce the table size, this paper proposes a solution based on the equivalent class.

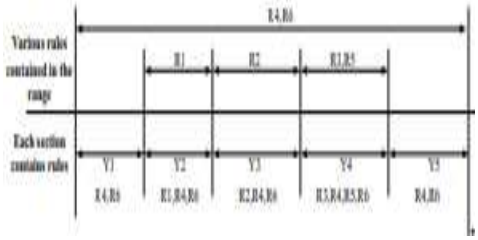

Fig.3 projection on Y axis

The Y-axis projection of the final destination T is first determined, as seen in Figure 3. Rule ranges are shown above the Y axis, with an example being R1's range of 0001-0011.R3 and R5's ranges overlap, and both are the same (0000-1111). After being partitioned along the Y axis, the rules for each subset are shown under the Y axis. For example, the Y1 subset includes Rules 4 and 6, whereas the Y2 subset also includes Rules 1, 4, and 6. Figure 3 demonstrates that Y5 and Y1 follow identical principles. It is assumed that Y1 and Y5 have the same effect arrangement. To make the Cross-product table more manageable, we might think about combining them. Table 4 displays the resulting one-dimensional table after the merge.

Table 4 Y1 Y5 merged with one-dimensional table

The destination address (T)	Y1	Y2	Y3	Y4
range	0000-0001, 1100-1111	0010-0011	0100-0111	1000-1011

Unfortunately, we can't make use of the quick linear search strategy when dealing with these one-dimensional lookup tables. This means that there are two components to Y1. In such case, there are five columns in the table, but only four types. The new iteration creates a continuous, non-overlapping spectrum. To speed up the linear search, it may still employ a fast one-dimensional searching technique.

Analysis of Experiments

To put the method in the paper through its paces, we simulate a rule set. Although the worst-case search time for the cross-product technique is O(KlogN), this is still considered to be rather fast. However, its space complexity is O(N2), thus we primarily do simulations in storage space occupancy in order to assess the algorithm's true outcomes. Assume two autonomous systems, AS-A and AS-B, with physically connected border routers; each autonomous system uses a different set of prefix addresses. We model the border router of an autonomous system and simulate the package categorization process. To do this, we model our two-dimensional classification system by using AS-A as the source and AS-B as the target addresses[6]. Take AS-A and AS-B as an example; if each has 20 prefixes and AS-B has 40, multiplying the two together yields 800 rules in the base.

Table 5 Experiment result

number of rules	subsections of previous Cross-Product method	subsections of our Cross-Product method	equivalence class after merged	compression ratio relative to previous Cross-Product method	compression ratio relative to our Cross-Product method
100	10*10	21*21	11*11	121%	27.4%
200	10*20	19*39	11*21	116%	31.2%
300	15*20	28*39	16*21	112%	29.7%
500	20*25	39*43	21*26	109%	32.6%
800	20*40	39*79	21*41	108%	27.9%
1000	25*40	47*82	26*41	107%	46.8%
1500	30*50	57*98	31*51	105%	31.6%
2000	40*50	61*98	41*51	102%	35%
2500	50*50	93*74	51*51	104%	37.8%
3000	50*60	93*118	51*61	104%	28.3%
4000	50*80	74*112	51*81	103%	49.8%
5000	50*100	74*193	51*101	102%	26.1%

The experimental outcomes are shown in Figure 4 and Table 5. The Cross-product approach proposed in this work for modelling rule bases increases the granularity of each dimension. This is due to the fact that the worst-case subsection number for the algorithm is N2N2, whereas the prior technique just had N2. By combining equivalence classes, the Cross-product table may be made to take up just roughly a third of the area formerly occupied by it. After the merge, the space required for the cross-product table is comparable to that of the traditional cross-product approach. After equivalence class merging, the prefixes of the destination and source addresses do not exceed N+1 in any dimension. The size of the cross-product table decreases after the merge equivalence class. There aren't a lot of major overlaps. For instance, under the traditional approach, a set of 5,000 rules would yield 5,000 things, but the approach described in this article would yield just 5,151 items for the same set of rules. So, in this case, compress is unnecessary. Our strategy's significance increases search times by a large margin.

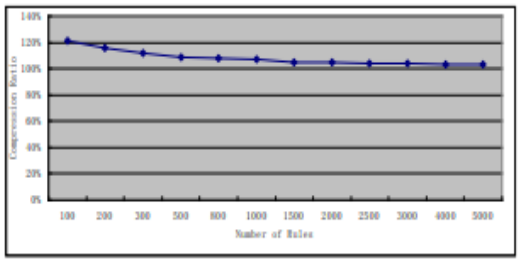

Fig.4 compression ratio relative to previous Cross-Product

Conclusions

In this study, we propose a few enhancements to the existing Cross-product algorithm. The first is to use Cross-product in a space where there are no overlaps. Each dimension is first partitioned into many groups before the cross-product is calculated. Both one- and two-dimensional spaces include overlapping regions, making the

employment of a rapid searching technique impossible. In this study, we use fast searching algorithms in all dimensions by adopting a categorization scheme based on space division that avoids overlap. The space need grows while the time complexity drops from O (KN) to O (KlogN). The second enhancement is using an analogous class, which helps with minimization of storage needs. The Cross-product table may be further compressed by joining rows and columns in an analogous segment. This technique not only expedites processing but also minimizes data storage needs.

References

[1] TianLiQin LinChuang, Xiao renyi, The design and realization based on IXP1200 quick message head classification algorithms. Computer researching and development. Volumn 40,no.11,pp 1292 – 1294,2003.

[2] T V Lakshman Stiliadis D J, High-speed policy-based packet forwarding using efficient multi-dimensional range matching [J].Journal of matching SIGCOMM ACM Comunication Computer, Volumn 28,no.4,pp 203-214,1998.

[3] Baboescu F, J G Varghese. Scalable packet classification [J]. IEEE Trans ACM on Networking, Volumn 14,no.1,pp 2- 14,2004.

[4] Gupta J P, N McKeown. Packet classfication on multiple fields [J]. Computer Communication, SIGCOMM ACM.191- 202,1998.

[5] V Srinivasan Suri, S, et al. Varghese G, Fast switching scalable and four layer [C]. Proc: drop In SIGCOMM ACM Conference participants' 98, ARCHITECTURES, as well as on Communication, integrity is available. New York: ACM Press, pp.191-202,1998.

[6] D.E.Taylor,J.S.Turner,"ClassBench:A Packet Classfication Benchmark",In Proceedings of IEEE Infocom 2005 Vol.3.March 2005,pp2068-2079.