Rule-set Database Inspection: Towards Data Utilization in Packet Processing

Mahmood Ahmadi, S. Arash Ostadzadeh, and Stephan Wong
Computer Engineering Laboratory
Microelectronics and Computer Engineering Department
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
{mahmadi, arash, stephan}@ce.et.tudelft.nl

***Abstract*— A critical task in network processing is packet analysis that includes operations like packet classification, filtering, and inspection. These operations are commonly based on matching headers and/or data within packets to rules inside a rule-set database. Consequently, the matching procedure determines how packets are classified. When matched, the same database contains the actions to be taken on the matched incoming packets. Even though much research has been performed in improving the performance of these packet processing operations, a thorough inspection of rule-set databases into their characteristics is still left open for further investigation. In this paper, we describe our inspection of real rule-set databases in order to determine the properties that can be exploited in future packet processing techniques or used to optimize current ones. As a result, different graphs are presented and discussed based on extracted information from various fields of rule-set database records.**

Keywords: Packet processing, packet classification, filtering, rule-set database, network processing

I. Introduction

Traditional routers solely utilize the packet destination address header field to forward packets towards their destinations. Since then, speed improvements in packet processing systems led to the inclusion of many other header fields or even packet data to determine the action to be taken on the packets, e.g., Deep Packet Inspection (DPI) [1]. In turn, this led to many new services including policy-based routing, firewalls, and VPN. The header fields on which the actions are based together with the action themselves are combined into rules that in turn are collected in rule-set databases. Commonly utilized header fields are source/destination IP addresses, protocol types, and service types.

Strictly speaking, a packet classification system compares header fields of every incoming packet against a rule-set database containing rules which act as filters to identify a flow [2][3][4]. Packet classification is generally an exhaustive multi-dimensional matching problem on the packet header. Particularly, the classification consists of the closest matches on the source/destination IPs, range matches on the source/destination port numbers, and an exact match on the protocol field, which indicates five fields in total. Although more complicated packet classification may be performed in some cases, the conventional 5-tuple classification is the most utilized one. Packet processing is expected to continue to grow in complexity and, therefore, it will constitute the performance bottleneck of future high performance routing systems [5]. As a result, it is logical and inevitable that future packet processing needs to exploit intrinsic characteristics of rule-set databases.

In this paper, we take an initial step to make a succinct yet comprehensive inspection of the data extracted from real rule-set databases in order to spot useful characteristics and set directions to improve the efficiency of packet processing techniques. Different graphs based on the extracted information from various fields of rule-set database records are depicted and discussed to enrich and support the arguments presented. These graphs address various distributions of source/destination IP addresses, specific or range port addressing, and strictly defined rules.

The subsequent sections of the paper are organized as follows. In Section 2, a brief survey of the related work is presented. The overall structure and properties of the examined rule-set databases is discussed in Section 3. Different kinds of experiments and the corresponding result analysis and discussions are presented in Section 4, and finally, Section 5 presents our concluding remarks.

II. Related Work

As stated before, there are not many works specifically addressing the rule-set database inspection to discover applicable information for optimized packet processing in networks. On the other hand, there exists rather considerable research on packet processing algorithms and techniques, and their performance evaluations. In the absence of publicly available real rule-set databases, it is difficult to perform experimental tests for new algorithms.

Gupta and McKeown [6] obtained access to several real rule-set databases through confidentiality agreements. 793 packet classifiers from 101 different ISPs and enterprise networks and a total of 41,505 rules were collected. They extracted a number of useful statistics which are widely employed. In addition, they generated synthetic two-dimensional rule-sets with the source/destination pairs by randomly selecting address prefixes from publicly available databases [7]. This technique was also utilized in [2][8]. A similar method for synthetic two-dimensional rule-set

generation with some modifications regarding wildcards controlling and prefix nesting is presented in [9]. Generating an augmented rule-set database from a sample collection of rules is also presented in [10]. The technique merely reproduces the IP prefix for rules keeping the remaining fields within the rules intact. The first work towards creating a benchmark for different application environments modeling is initiated in [11]. Later, based on the properties discussed in [11], several packet classification techniques are compared and evaluated in [12]. A thorough survey and framework is presented in [4] regarding a taxonomy based on high-level approaches to the packet classification problem.

Kounavis, et. al. [13] carried out an analysis of several rule-set databases and proposed a general framework for packet classification in network processors. Based on their analysis, observations were made about the dependency between the size of the rule-set database and the number of "partially specified" rules which utilize wildcards in the source/destination IPs. They determined that partially specified rules comprise a smaller proportion of the rule-set as the number of rules increases. In addition, observations were also made about the structure of partially specified rules, the number of matching rules for an incoming packet, and the proportion of application specific rules.

ClassBench [14] comprises a suite of tools for benchmarking packet classification algorithms and devices. It was developed due to the absence of a standard performance evaluation tool for the research community, serving to facilitate future research with a common basis for meaningful benchmarking. ClassBench consists of three tools: a filter set analyzer, a filter set generator, and a trace generator. The suite constructs a set of benchmark parameter files that describe relevant characteristics of a real rule-set database, generates a synthetic rule-set from a chosen parameter file and small set of high-level inputs, and provides the option to generate a sequence of packet headers to probe the synthetic rule-sets using the trace generator.

III. Rule-set Databases

In this section, we present the structure of rule-set databases and their properties. Generally, rule-set databases have fewer than a thousand predefined rules that reside in firewalls or routers. As network processing tends to move into core networks, it is expected that these databases expand to contain tens of thousands of rules or even more [6]. We performed our experiments with several real rule-set databases from Applied Research Laboratory in Washington University in St. Louis [14] provided by Internet Service Providers (ISPs), a network equipment vendor, and other researchers working in the field. The rule-set databases are different in size ranging from tens to thousands of rules. They are in one of the following formats:

- ACL Access Control List - standard format for security, VPN, and NAT rule-sets for firewalls, and routers (enterprise, edge, and backbone)
- FW FireWall - proprietary format for specifying security rule-sets for firewalls.
- IPC IP Chain - decision tree format for security, VPN, and NAT filters for software-based systems.

In rule-set databases, each rule contains 5 fields defined as "*[Source IP address, Destination IP address, Source port, Destination port, Protocol]*" and the format is "*@[Source IP address prefix in dot-decimal notation]/[Prefix length] [Destination IP address prefix in dot-decimal notation]/[Prefix length] [Low source port] : [High source port] [Low destination port] : [High destination port] [Protocol value in hexadecimal]/[Protocol mask in hexadecimal]*". As an example, a rule in the rule-set database is defined like: *@204.152.188.80/28 204.152.188.64/28 67 : 67 67 : 67 0x11/0xff*.

Table I lists the fifteen rule-set databases used in our experiments and the number of rules in each of them. The rule-set databases *fw1*, *acl1*, and *ipc1* were extracted from real rule-sets and the others were generated by the ClassBench benchmark tool.

IV. Experimental Results and Analysis

We conducted several experiments based on different packet header fields and criteria found in the rule-set databases to extract appropriate information to be utilized in packet processing techniques. The observations and discussions presented here are general and one is expected to incorporate them according to particular requirements and preferences taking into account their applications in various areas. We classified the relevant inspections which are presented in the following subsections. It should be noted that in the depicted diagrams and graphs, the observed values have been normalized according to the number of rule-set entries in the corresponding databases. A general discussion of the results can be found at the end of this section.

A. Source/Destination IP Addresses

The distributions of source/destination IP address prefixes are depicted in Figure 1. From Figure 1 (a), we can observe that most of source IP addresses in the rule-set databases have long prefixes with 23 valid bits and up, except the 0-bit prefix which is quite common in rule-sets. The source IP addresses with 32 valid bits have the highest rank since most of the rules for the source IP addresses are applied with a specific IP address. Additionally, it is apparent that certain rules do not utilize any bits in their source IP address fields which means that they are employed to state general policies.

Figure 1 (b) depicts the destination IP address prefixes for different rule-set databases. We can observe that most of the destination IP addresses have long prefixes associated with them. The characteristic of destination IP addresses is similar to source IP addresses, however, there is a trend towards strictly specified addresses compared to 0-bit entries which was more visible in the source counterparts.

Rule-set Database	fw1-100	fw1-1k	fw1-5k	fw1-10k	fw1	acl1-100	acl1-1k	acl1-5k
Number of rules	92	791	4653	9311	266	98	916	4415
Rule-set Database	**acl1-10k**	**acl1**	**ipc1-100**	**ipc1-1k**	**ipc1-5k**	**ipc1-10k**	**ipc1**	
Number of rules	9603	752	99	938	4460	9037	1550	

TABLE I

RULE-SET DATABASES SPECIFICATIONS

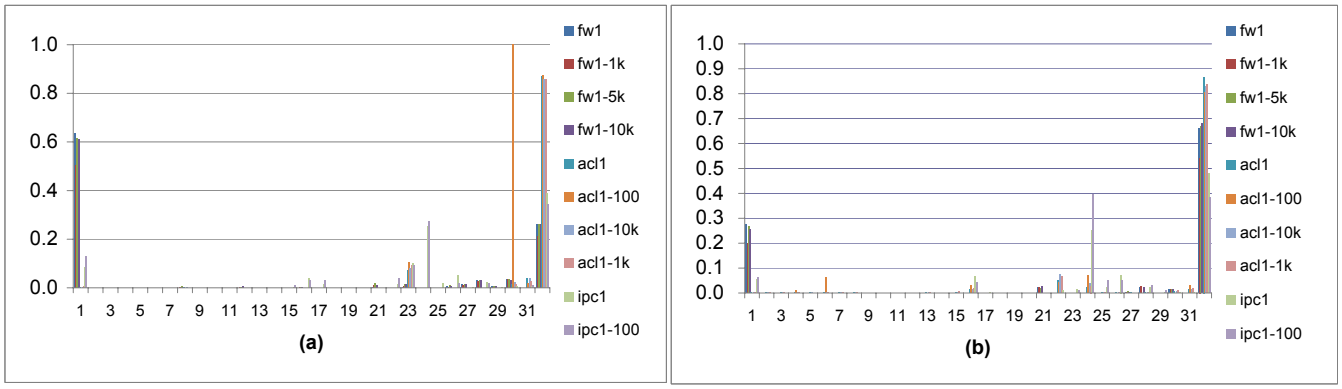

Fig. 1. IP address prefix distribution (a) Valid bits of source IP addresses (b) Valid bits of destination IP addresses

Figure 2 demonstrates the percentage of strictly defined IP addresses compared to partially defined ones in different rule-set databases. For each database, four different kinds of rules are considered: rules with specific source IP addresses (prefix value equal to 32), specific destination IP addresses, both specified source/destination IPs, and strictly defined. By strictly defined we refer to the rules that have all the five fields specifically defined, i.e., no wildcard or range specifications. It is inferred that in the *acl1* database series most of the source and destination IP addresses (75 percent up to 90) are well defined. This is similarly the case for the *fw1* and *ipc1* databases but with a lower ratio (25 up to 70). The difference between specific source and destination IP addresses is substantial in *fw1* databases (the factor in nearly a half) compared to *acl1* and *ipc1*. There is a common feature in all databases and that is the nearly zero or less than one percentage of strictly defined rules. This means that most databases have strictly defined IP addresses but specify a range for their source and/or destination ports rather than a port number.

B. Source/Destination Ports

The source port number distribution is depicted in Figure 3. Figure 3 (a) depicts the distribution of source port numbers in the *fw1* and the *acl1* rule-set databases. The lowest value for the source port distribution belongs to *fw1-1k*. Based on the experiment, the distribution of source port numbers for *acl1* rule-set databases are all the same. This is due to the fact that *acl1* rule-set database has been extracted from Access Control List in the firewalls and routers, therefore packets coming through all of the incoming ports are processed. We only depicted the graph for the *acl1* (real rule-set database). Figure 3 (b) depicts a similar graph for the *ipc1* rule-set databases. From Figures 3 (a) and 3 (b), we can conclude that wide ranges of different port numbers appear in most of the rules. For a specific port number or a range of port numbers, the value 0.7 means that 70 percent of the total rules in the corresponding database cover that port number/range.

Moreover, we extracted the relevant data from the distribution of destination port numbers. It can be concluded that the values for destination port number distribution are lower than the source port number counterparts. In most network devices, the rules are defined considering a particular destination port number. On the contrary, the source ports are mostly defined as a range. As a result, the graphs show approximately double the value of source port number distribution compared to the destination port number one. All the databases exhibit more or less a similar behavior and their graphs are not depicted here to save space.

Subsequently, we investigated the range-size distributions of source/destination port numbers. The range-size specifies the distance between the starting and ending port numbers in a range. As an example, the range-size of the interval [10-110] with the starting value of '10' and ending value of '110' will be 101. Considering the observations, it was evident to conclude that there is a tendency towards wide ranges in source port numbers. The range-size '1' represents just a single port. The range-sizes greater than '2' and less than '536' were not recognized in any of the rule-set databases. Graph inspection for the range-size of different rule-set databases for the destination port numbers, shows that most range-sizes either have a very low or high values. Range-size with the value '1' appears more than other

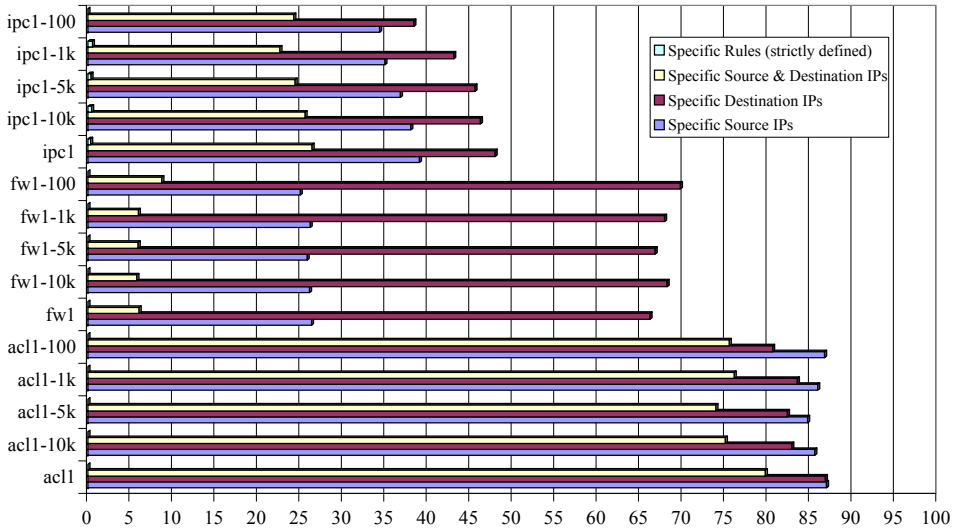

Fig. 2. Specifically vs partially defined rules in rule-set databases

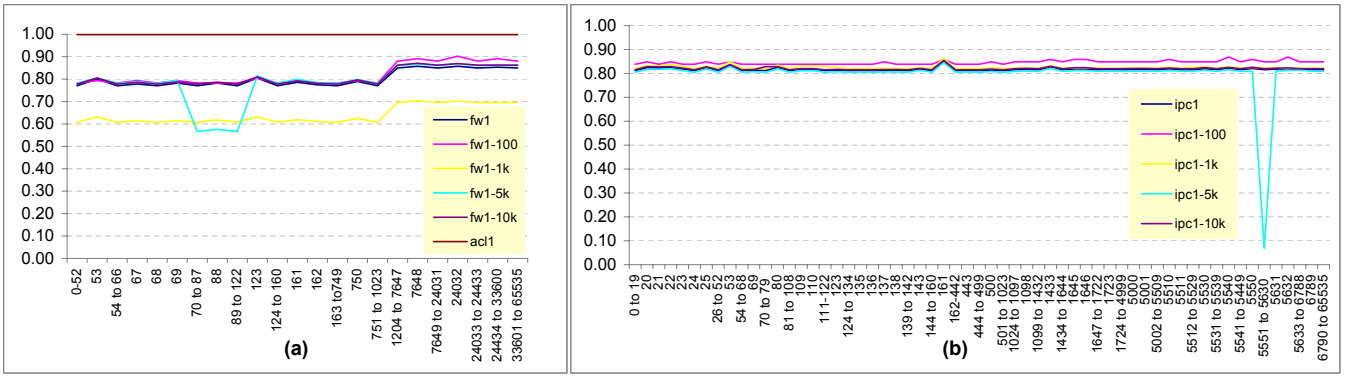

Fig. 3. Source port number distribution (a) Source Port number distribution for *fw1* database series and *acl1* (b) Source port number distribution for *ipc1* database series

range-sizes. In other words, most of the rules are defined as specific destination port numbers and the remaining rules are generally defined with ranges of high values. As a final deduction, it is inferred that in most of the rules, source port numbers are defined with wide ranges and most of the destination port numbers are specified as particular port numbers.

Figure 4 depicts strictly defined source port number distribution in *fw1* rule-set databases. Only for 10 particular port numbers there exists rules that strictly address these ports. For port numbers not included in the diagram there are no specific addressing, i.e., no rule is strictly defined for majority of the ports. As interpreted from Figure 4, this specific addressing only contains less that 5 percent of the total addressing with the port numbers 123 and 53 having the lead. A similar diagram for *ipc1* rule-set databases is depicted in Figure 5. Here we have 27 port numbers, however, the overall specific utilization compared to the total definitions remains approximately the same with port numbers 161 and 53 on top of the list. The average value remains about 1 percent for these port numbers which is quite low. For the *acl1* rule-set databases, there is no specific port addressing in the rules.

Figure 6 depicts strictly defined destination port number distribution in *fw1* rule-set database series. Compared to the counterpart diagram of source port numbers, it is not difficult to notice that the number of ports have increased substantially which is an indication of specific addressing of ports within destination IPs in firewall databases. 40 port numbers appear in the diagram. Not only the number of ports but the overall specific utilization is also increased considerable from less than 5 up to 30 percent.

C. Discussion

A well-known hash-based algorithm in packet processing is the tuple space search. This technique groups different rules using simple or complex mappings and searches the IP packets in the formed tuples [15]. In the mapping process, the source and destination ports are mapped onto the tuple space using Range-ID and nesting level concept. We presented various observations and assertions in the previous

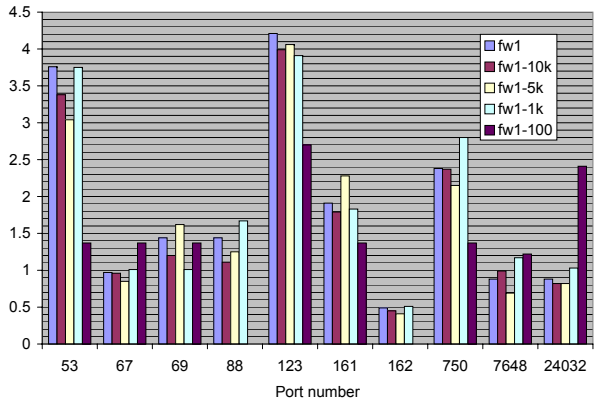

Fig. 4. Strictly defined source port number distribution in *fw1* rule-set database series

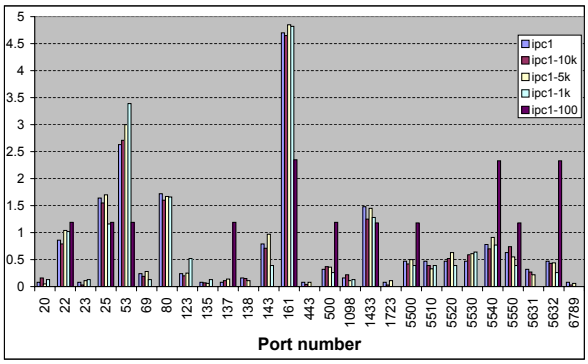

Fig. 5. Strictly defined source port number distribution in *ipc1* rule-set database series

subsections according to thorough inspection of data in rule-set databases. Utilization of the results of these analyses can lead to individual optimizations of the tuple space search algorithms. Based on particular distributions, these packet processing algorithms can be optimized using refined tuple space. In traditional tuple space, a search key is created using all of the fields in the rules or packet headers and this consequently increases the number of tuples created as well as the search time needed for optimal matching. By employing an even distribution of different fields and data grouping in a refined tuple space construction process, it is possible to take advantage of the range-formed data fields to efficiently perform the matching process in less time.

V. Conclusion

In this paper, we presented a detailed inspection of the data extracted from real rule-set databases in order to discover characteristics that can be properly utilized in future packet processing methods as well as optimizing current ones. Several graphs based on the extracted information from various fields of rule-set database records were presented and discussed. Future packet processing applications are expected to benefit from specific rule-set properties. Therefore, it makes sense that packet classification systems exploit these characteristics to optimize their performances and improve their efficiencies in different aspects, but only to the extent that they do not lose their general applicability.

References

[1] S. Dharmapurikar, P. Krishnamurthy, T. S. Sproull, and J. W. Lockwood, "Deep Packet Inspection Using Parallel Bloom Filters," *IEEE Micro*, vol. 24, no. 1, pp. 52–61, 2004.

[2] A. Feldmann and S. Muthukrishnan, "Tradeoffs for Packet Classification," in *Proc. Int'l IEEE Conf. INFOCOM*, vol. 3, 2000, pp. 1193–1202.

[3] P. Gupta and N. McKeown, "Algorithms for Packet Classification," *J. IEEE Network*, vol. 15, no. 2, pp. 24–32, March-April 2001.

[4] D. E. Taylor, "Survey and Taxonomy of Packet Classification Techniques," *ACM Comput. Surv.*, vol. 37, no. 3, pp. 238–275, 2005.

[5] D. Taylor and J. Turner, "Towards a Packet Classification Benchmark," Washington University, Department of Computer Science and Engineering, Tech. Rep. TR-WUCSE-2003-42, May 2003.

[6] P. Gupta and N. McKeown, "Packet Classification on Multiple Fields," in *Proc. Int'l Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 1999, pp. 147–160.

[7] ——, "Packet Classification Using Hierarchical Intelligent Cuttings," *IEEE Micro*, vol. 20, no. 1, pp. 34–41, 2000.

[8] F. Baboescu, P. Warkhede, S. Suri, and G. Varghese, "Fast Packet Classification for Two-dimensional Conflict-free Filters," *Comput. Networks*, vol. 50, no. 11, pp. 1831–1842, 2006.

[9] F. Baboescu and G. Varghese, "Scalable Packet Classification," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 2–14, 2005.

[10] F. Baboescu, S. Singh, and G. Varghese, "Packet Classification for Core Routers: Is There an Alternative to CAMs?" *22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, pp. 53–63, April 2003.

[11] T. Woo, "A Modular Approach to Packet Classification: Algorithms and Results," *19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, pp. 1213–1222, 2000.

[12] V. Sahasranaman and M. Buddhikot, "Comparative Evaluation of Software Implementation of Layer-4 Packet Classification Schemes," in *ICNP '01: Proceedings of the Ninth International Conference on Network Protocols*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 220–228.

[13] M. Kounavis, A. Kumar, H. Vin, R. Yavatkar, and A. Campbell, "Directions in Packet Classification for Network Processors," *Second Workshop on Network Processors (NP2)*, february 2003.

[14] D. Taylor and J. Turner, "ClassBench: A Packet Classification Benchmark," Department of Computer Science And Engineering, Washington University in St. Louis, Tech. Rep. WUCSE2004-28, May 2004.

[15] V. Srinivasan, S. Suri, and G. Varghese, "Packet Classification Using Tuple Space Search," in *Proc. Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 1999, pp. 135–146.

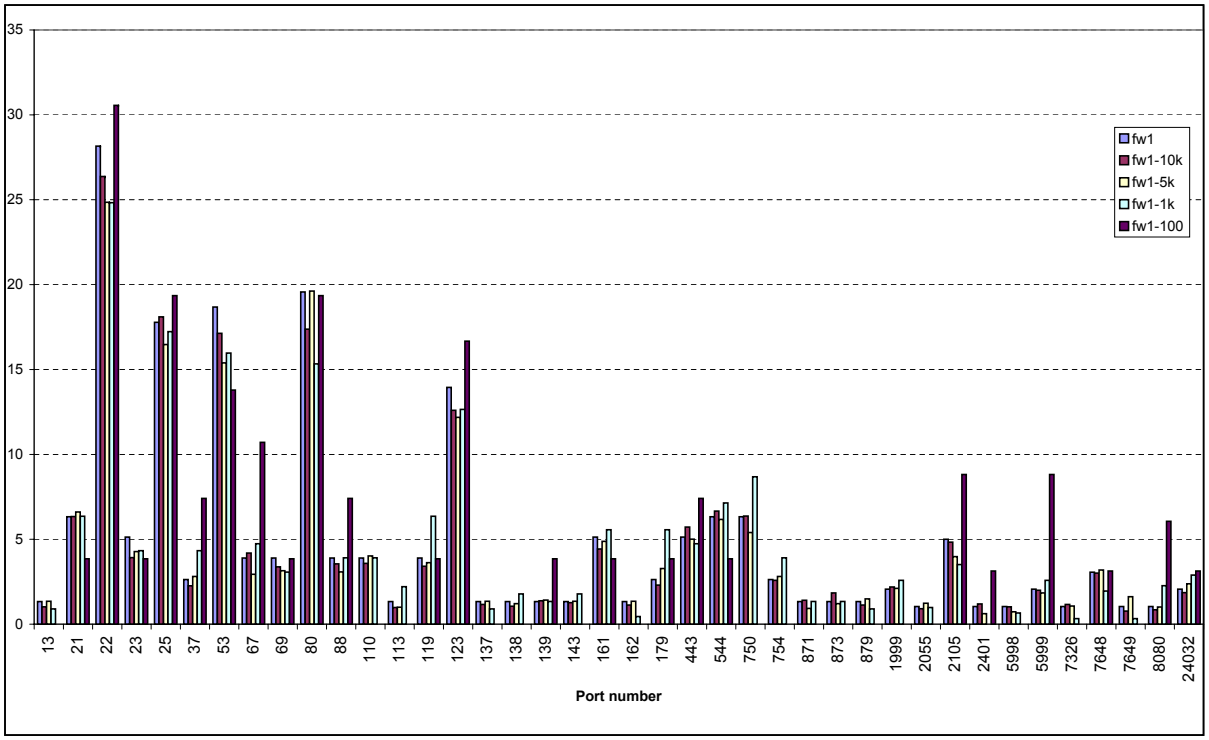

Fig. 6. Strictly defined destination port number distribution in *fw1* rule-set database series