Computer Communications 37 (2014) 64–76

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# Source address filtering for large scale networks ☆

Mingwei Xu [a,b], Shu Yang [a,b,*], Dan Wang [c], Fuliang Li [a,b], Jianping Wu [a,b]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[b] Tsinghua National Laboratory for Information Science and Technology, Beijing, China
[c] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, KL, Hong Kong

ARTICLE INFO

*Article history:*
Received 26 May 2013
Received in revised form 21 September 2013
Accepted 28 September 2013
Available online 9 October 2013

*Keywords:*
Source address filtering
Distributed filtering
Network security

ABSTRACT

Source address filtering is very important for protecting networks from malicious traffic. Most networks use hardware-based solutions such as TCAM-based filtering, however, they suffer from limited capacity, high power consumption and high monetary cost. Although software, such as SRAM, is larger, cheaper and consumes less power, the software-based solutions need multiple accesses in memory, which as a result bear much more additional lookup burden.

In this paper, we propose a new software-based mechanism. In our mechanism, routers cooperate with each other, and each only checks a few bits rather than all bits in source addresses. Our mechanism can guarantee the correctness, i.e., filtering all malicious traffic. We formulate it as an optimization problem where the loads across the network can be optimally balanced. We solve the problem by dynamic programming.

With the increasing number of filters, storage could also become a bottleneck for source address filtering. Our mechanism improves this by distributing filters among different routers. We re-formulate the problem by adding an additional storage constraint. Then we prove that the problem is NP-Complete, and propose a heuristic algorithm to solve it.

At last, using comprehensive simulations with various topologies, we show that the mechanism greatly improves both lookup burden and storage space. We conduct a case study on China Education and Research Network 2 (CERNET2), the largest pure-IPv6 network in the world. Using CERNET2 configurations, we show that our algorithm checks less than 40 bits on each router, compared with 128 bits in IPv6 addresses.



## 1. Introduction

Despite a significant breadth of research, malicious traffic problems such as DDoS attack and scanning, remain an important problem today [1]. Packet filtering is a prevalent mechanism for preventing malicious traffic. Due to the importance of source addresses, source address filtering is widely adopted in current ISPs. Traditionally, ingress routers will maintain a blacklist, which is the set of source addresses that should be filtered. During the past years, the blacklist has increased explosively, largely enabled by botnets and other platforms for launching attacks. This situations are even worse in large scale networks. In 2003, more than 20,000 sources appeared in an attack against an online betting site [2]. In 2007, a storm botnet was reported to include 50 million sources [3]. In 2008, more than 800,000 unique malicious IP sources addresses were reported everyday [4]. Under the devastating security crisis, the blacklists in ISPs are continually expanding to defend against malicious traffic from possible attackers.

To implement the IP blacklist, TCAM is currently the de facto industry standard. TCAM can achieve wire speed as it enables parallel matching [5]. However, TCAM stoarge space is limited due to its high cost and power consumption. The line-card in Cisco 12000, which is a typical core router, can only accommodate 20000 entries. With more and more malicious traffic, TCAM-based solutions can not accommodate so many entries, and can not defend against today's most severe attacks, not to mention larger attacks in the near future, where millions of sources are expected [6]. Limited storage even makes some TCAM-based solutions allow part of the malicious traffic for better aggregation [7]. Even worse, the growth of TCAM size can not keep pace with the explosively increasing number of filters in the foreseeable future.

Although software-based solutions can provide larger space and accommodate more filters, they are not widely used currently because they need multiple accesses during a single lookup. For

☆ The research is supported by the National Basic Research Program of China (973 Program) under Grant 2009CB320502, the National Natural Science Foundation of China (61073166), the National High-Tech Research and Development Program of China (863 Program) under Grants 2011AA01A101.

* Corresponding author. Address: Room 9-402, East Main Building, Tsinghua University, 100084 Beijing, China. Tel.: +86 (0) 10 62785822; fax: +86 (0) 10 6260364.

*E-mail address:* yangshu@csnet1.cs.tsinghua.edu.cn (S. Yang).

0140-3664/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2013.09.013

example, current high performance routers usually use SRAM, and the largest SRAM chip has 144 Mb (288 Mb SRAM are on the roadmap of major vendors) [8], thus a large fraction of software-based solutions are using lookup tries [9]. Thus, software-based solutions may introduce large latencies and serious congestions, especially when facing burst traffic, despite their fast speed.[1]

Traditionally, the filters are stored in border routers, which is the choke point that transit traffic is sure to pass by. As a result, the border routers have to bear the additional processing burden. In this paper, we try to balance the load by designing a distributed mechanism where more routers can share the lookup burden. All routers along a path can work cooperatively to handle the source-IP filtering correctly. Such a design scales better facing increased filtering requirements, especially in large scale networks.

Although many distributed solutions already existed [12,6], they distribute tasks to routers by filters, that is, different routers checks different address blocks. Unlike previous solutions, our mechanism assigns different bits to routers. Such that each router only checks a few bits rather than all in source addresses. In this way, the load is balanced across the network, each router bears less additional lookup burden, and achieve fast lookup speeds more easily. The mechanism guarantees correctness, i.e., filtering all malicious traffic, by letting all routers along a path cooperatively check all bits in source addresses.

### 1.1. Simple example

To illustrate our basic idea, we use an example in Fig. 1. In the network, packets will travel through ingress router $a$ towards egress router $d$, and there are three source prefixes to be filtered: 1**, 00* and 010. Conventionally, filters will be stored only at the ingress router $a$. Thus router $a$ needs to access memory up to 3 times when a packet arrives, which brings heavy burden on router $a$. In our mechanism, each router only checks 1 bit, i.e., $a$ checks the $0_{th}$,$b$ checks the $1_{st}$, and $c$ checks the $2_{nd}$ bit. When a packet with source 010 arrives at router $a$, it will be delivered towards the egress router along the path $\{a,b,d\}$. With the new mechanism, router $a$ checks the $0_{th}$ bit first, and moves the pointer from the root trie node to the $1_{st}$ level; then it passes the packet along with the intermediate pointer to router $b$, which checks the $1_{st}$ bit, moves the pointer to the $2_{nd}$ level and passes the information to $c$;$c$ will check the $2_{nd}$ bit, concludes that the packet falls in the blacklist and should be filtered. In this way, each router bears less burden, and the load is balanced across the network. The amortized burden on each router would be much lower.

In this paper, we generalize the example by formulating it as an optimization problem where we need to balance the load across the network, given that (1) the total bit set to be checked, which can be computed using the blacklist; (2) the network topology information, including the location of ingress and egress routers; (3) the spare capacity on each router for source address filtering, in other words, each router has a limitation on the extra burden. To solve the problem, we develop a dynamic programming based algorithm, which can find the optimal solution.

Although SRAM provides larger storage space than TCAM, it could still be subject to bottlenecks considering the rapidly increasing number of malicious sources. To mitigate this problem, we propose that storage (like the uni-bit trie in Fig. 1) can be divided among multiple routers, such that each router only stores one part of the total storage to be looked up. We introduce a new problem by adding an additional storage constraint, and then prove the problem to be NP-Complete and propose a heuristic algorithm to solve it.

[1] The maximum clock rate of SRAM is 400 MHz, while TCAM is 266 MHz, the price of SRAM is 10–100 times lower than TCAM [10,11].

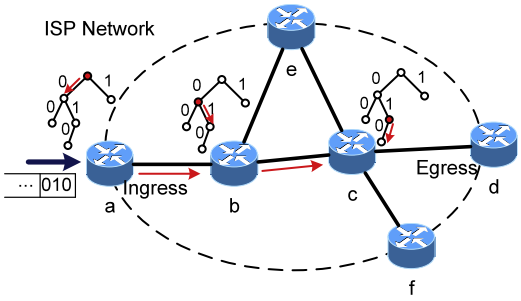


**Fig. 1.** Router $a$ is the ingress and $d$,$e$,$f$ are egress routers. The source address has 3 bits, and there exist 3 source filters: $1^{**}$, $00^{*}$, 010. The source filters are organized as a uni-bit trie.

To evaluate our mechanism, we conduct various simulations using both real and BRITE generated topologies. We show that our mechanism can balance the load across routers much better, and greatly reduce the number of bits that should be checked on each router. With storage constraint, our mechanism provides new room for storing large number of malicious sources. Using the real configurations from China Education and Research Network 2 (CERNET2, which is the world's largest IPv6 network, including 59 Giga-PoPs), we also conduct a case study. We show that, throught using our mechanism, each router in CERNET2 only needs to check at most 40 bits rather than whole 128 bits in IPv6 addresses. Using real data-traces, we also evaluate the overheads brought by our mechanism in both data and control planes. The results show that the overheads caused by our mechanism are quite low, this further prove that our mechanism is feasible in real networks.

The paper is organized as follows: We present the related work in Section 2. Section 3 is devoted to design overview of the new mechanism. We formulate the problem and present the optimal algorithm in Section 4. In Section 5, we take the storage constraint into consideration. Section 6 shows our implementation design. We evaluate our mechanism in Section 7, conduct a case study in Section 8, and conclude our paper in Section 9.

## 2. Related work

A significant body of research works have been devoted to battle against DDoS and spoof problems with filters. For example, most current networks use ingress access lists [13] or static ACLs (Access Control Lists) [14] to keep malicious traffic our of the networks. TCAM, which is a scarce resource, is the de facto standard for storing blacklist. However, with the exponentially increasing of blacklist, TCAM-based filtering fails to accommodate so many filters due to its limited storage space, high cost and high power consumption [15].

Due to lack of hardware memory space, many solutions have been proposed to reduce the number of filters. Pack et al. [16] reduces the number of source prefixes through aggregation. Yi et al. [17] utilizes bloom filters, which occupies much smaller storage space, to defend against malicious traffic. Some solutions [7,16] even allow part of the malicious traffic for better aggregation of source prefixes, which are likely to cause collateral damage. In [18], the bayesian decision theory based on attacking history is used to optimize the set of filters in blacklists.

In [6], a distributed filtering mechanism is studied to reduce collateral damage. To save the scarce TCAM resources, it resolves the problem as a resource allocation problem. With this mechanism, routers only stores part of all filters, and different routers defend against different IP address blocks along a path. However, the