

Super point detection based on sampling and data streaming algorithms

Cheng Guang Q iang Shiqing

(School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

(Key Laboratory of Computer and Network Technology of Jiangsu Province, Southeast University, Nanjing 210096, China)

Abstract: In order to improve the precision of super point detection and control measurement resource consumption, this paper proposes a super point detection method based on sampling and data streaming algorithms (SDSD), and proves that only sources or destinations with a lot of flows can be sampled probabilistically using the SDSD algorithm. The SDSD algorithm uses both the IP table and the flow bloom filter (BF) data structures to maintain the IP and flow information. The IP table is used to judge whether an IP address has been recorded. If the IP exists, then all its subsequent flows will be recorded into the flow BF; otherwise, the IP flow is sampled. This paper also analyzes the accuracy and memory requirements of the SDSD algorithm, and tests them using the CERNET trace. The theoretical analysis and experimental tests demonstrate that the most relative errors of the super points estimated by the SDSD algorithm are less than 5%, whereas the results of other algorithms are about 10%. Because of the BF structure, the SDSD algorithm is also better than previous algorithms in terms of memory consumption.

Key words: super point; flow sampling; data streaming

The problem of super point detection arises in network monitoring and security applications, such as DDoS attacks, worm attacks and network scan events, which cause some IP addresses to produce and send a large number of flows to distinct destinations or receive a large number of flows from different sources in a given measurement interval. For example, Slammer's scanner^[1] could produce more than 30 000 scans per second. For a lightly loaded OC-48 with a favorable traffic mix, a measurement system with a memory of hundreds of megabytes and efficient algorithms for counting flows can afford to keep an entry for each source and destination IP. However, under adverse traffic mixes such as massive DoS attacks with source addresses faked at random or worms aggressively probing random destinations, keeping even a small entry for each unique IP address will consume too much memory of measurement monitors, so we cannot afford to save the states of all aggregation points.

This problem of super point detection has been studied in recent years. For example, Snort^[2] and Flowscan^[3] keep records for each source and destination. There is no memory-eff-

icient implementation in this straightforward approach, since the hash table typically requires large quantities of DRAM for operation. As a result, this approach is not feasible for monitoring high-speed links. Venkataraman et al.^[4] proposed two flow sampling techniques to detect super points, where both one-level and two-level filter schemes used a hash-based flow sampling technique for estimating fan-outs. When these schemes are used in high-speed links, the sampling rate is typically low due to the traffic burst problem. Zhao et al.^[5] proposed two algorithms to solve the problems using the data streaming algorithm which focuses on estimating the number of flows in the source/destination IP, but they did not give a method concerning how to retain the source/destination IP records. Noriaki et al.^[6] proposed an adaptive method of identifying super points by flow sampling in order to satisfy the given memory size and the requirements for the processing time, so it can adaptively optimize parameters according to changes in traffic patterns.

This paper proposes a super point detection method based on sampling and data streaming algorithms (SDSD). In this method, only sources or destinations with a lot of flows can be sampled probabilistically using sampling and data streaming techniques. The contribution of the SDSD algorithm is to use both the IP table and the flow bloom filter (BF)^[7] data structures to maintain the IP and flow information.

1 SDSD Algorithm

The SDSD algorithm structure is shown in Fig 1. After a packet arrives in the monitor, the algorithm checks the IP table and judges whether the IP which the packet includes has existed in the IP table. If the IP is found in the IP table, then the flow which the packet belongs to will be checked and the IP entry in the IP table is updated. Otherwise, the SDSD will sample the flow to decide whether to add a new IP record into

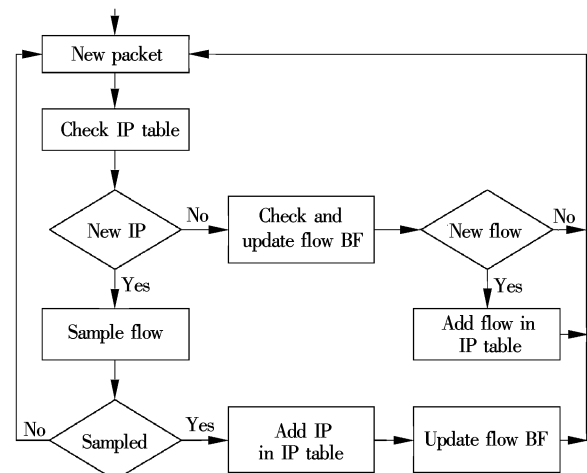


Fig 1 SDSD structure

Received 2008-09-22.

Biography: Cheng Guang (1973—), male, doctor, associate professor, gcheng@njnet.edu.cn.

Foundation items: The National Basic Research Program of China (973 Program) (No. 2009CB320505), the Natural Science Foundation of Jiangsu Province (No. BK2008288), the Excellent Young Teachers Program of Southeast University (No. 4009001018), the Open Research Program of Key Laboratory of Computer Network of Guangdong Province (No. CCNL200706).

Citation: Cheng Guang, Q iang Shiqing. Super point detection based on sampling and data streaming algorithms [J]. Journal of Southeast University (English Edition), 2009, 25 (2): 224-227.

the IP table

The SDSD algorithm directly submits it to the BF process whether the source/destination IP in the packet belongs to an entry in the source/destination IP memory or not; otherwise, the flow sampling process randomly samples this flow with a probability p . Let a flow identifier of a packet be x , and a hash function h produce a hash value $h(x)$. Let the maximum of the hash value be $h_{\max}$ and the sampling probability be p . If $h(x)/h_{\max} < p$, then the source or destination IP of this packet is sampled, and the source/destination IP is added in the source/destination IP table. After the IP is added, the sample and hold process will transfer all its subsequent flows whose source/destination IP is equal to the IP in the flow BF. When a packet passes through the sample and hold process, the BF will be detected to judge whether the destination/source IP of this packet belongs to a new or a existing flow. If it has no record in the bloom filter, then its information is added to the BF, and the source/destination IP table is also updated at the same time.

The SDSD defines an IP table used in keeping the IP identifier and its flow number, and sets a BF structure to record those flows which have been written in the IP table. Let the size of the BF be w (bits), and the BF, which is used to record whether the flow exists, is initialized all to '0' at the beginning of the measurement interval. We set a hash function h that maps a flow label to a value uniformly distributed in $[1, w]$. As soon as a packet pkt arrives, the SDSD hashes its flow label($\text{pkt sourceIP, pkt destIP, pkt sport, pkt dport}$) using the h hash function, so $r = h(\text{pkt sourceIP, pkt destIP, pkt sport, pkt dport})$. The result r is treated as an index into the BF. If $B[r] = 1$, then the flow has been processed with a collision probability, and the SDSD simply misses processing the packet. Otherwise (i.e. $B[r] = 0$), the flow label of the packet is a new one, and the IP record is updated in the IP table.

2 Theoretical Analysis

We formally quantify the probability that a super point with a certain number of flows is not detected. Let F be a threshold of the flow number of a defined super point and p be the flow sampling probability. Then the probability p' that the super points with F flows are missed is $p' = (1 - p)^F \approx e^{-Fp}$. Let the probability of a missed super point with F flows be less than or equal to δ , that is, $p' = e^{-Fp} \leq \delta$, then the sampling probability p should satisfy $p \geq \frac{\ln(1/\delta)}{F}$.

to detect the super point with F flows. For example, if we need to detect a super point whose flow number is greater than 100 and $\delta \leq 0.0001$, then $p > 0.092$.

Let a super point have s flows. When the first flow of the super point is sampled, the number of passed flows in the s flows is X . Here X is a geometric probability distribution, and its probability mass function (PMF) is $P(X = k) = (1 - p)^{k-1} p$. Where $X = k$ means that when the first flow of the super point is sampled, the number of passed flows in the s flows is $k - 1$. $E(X) = 1/p$, $\text{var}(X) = (1 - p)^2/p$. The estimated value of x is $\hat{x} = 1/p$, and its variance is $(1 - p)^2/p$. Assuming that ε is the relative error threshold of the flow number estimator of a super point, $\varepsilon = 0.1$. Ac-

cording to $(\sqrt{1 - p/p})/F \leq \varepsilon$ and $(\sqrt{1 + 4\varepsilon^2 F^2} - 1)/(2\varepsilon^2 F^2) \leq p$, the minimal probability for flow sampling is 0.095. In this example, the sampling probability is set to 10% to satisfy the two conditions.

After the first flow of an IP is recorded in the IP table, all the subsequent flows which belong to the IP will be measured, and their subsequent flows will be checked by the BF and judged whether the flow is a new one or not. Let the number of '0' bits in BF B (with size w) be u when a new flow with source IP S arrives. If the flow identifier is mapped into $B[r]$, then $B[r] = 0$ with probability u/w , so we can use w/u to update the IP flow number $N(S)$ in the IP table. In the measurement interval, if k flows of IP S are found in the BF B , we can obtain an unbiased estimator of the IP S flow number after the first flow of IP S

is recorded into the IP table. $N_s = \sum_{i=1}^k \frac{w}{u_i}$. Assuming that

the number of '1' bits in the BF with m -bit spaces and k hash functions is n_i when the i -th flow arrives, the collision probability that a hash value of a new flow enters into '1' bit position is n_i/m . Because the probability that all k hash functions enter into '1' bit positions is $(n_i/m)^k$, the probability that a new flow can be detected is $p_i = 1 - (n_i/m)^k$. Therefore, to obtain an unbiased estimator of the IP flows from the sampled traffic, we should statistically compensate for the fact that with probability $1 - p_i$, the bit in the BF has a value of 1 and the flow will miss the update in the IP table due to aforementioned hash collisions. It is obvious that if we add $1/p_i = 1/(1 - (n_i/m)^k)$ to the IP table, the resulting estimator is unbiased, and its variance is $(1 - p_i)/p_i^2 = (n_i/m)^k / (1 - (n_i/m)^k)^2$. To be more precise, supposing, in a measurement period, the BF is updated by an IP S with T flows $\{\text{flow}_j, j = 1, 2, \dots, T\}$. An un-

biased estimator of T is $E(\$) = \sum_{i=1}^T \frac{1}{1 - (n_i/m)^k}$, and its

variance is $\text{var}(\$) = \sum_{i=1}^T \frac{(n_i/m)^k}{(1 - (n_i/m)^k)^2}$. If the sam-

pling and the BF are considered at the same time, then $E(\$) = \sum_{i=1}^T \frac{1}{(1 - (n_i/m)^k)} + \frac{1}{p}$, $\text{var}(\$) = \sum_{i=1}^T \frac{(n_i/m)^k}{(1 - (n_i/m)^k)^2} + \frac{1 - p}{p^2}$.

The IP table size is determined according to the number of identified super points. The actual number of sampled flows is an upper bound on the number of entries in the IP table because new entries are created only for sampled flows. The expected number of the sampled flows is Np , where N is the number of flows, and p is the sampling probability. Since the number of sampled flows is a binomial probability distribution, we can use a normal distribution curve to fit the number of sampled flows during the measurement interval with high probability. The actual flow number with a probability of 99% will be at most 2.33 standard deviations and above the expected value. The standard deviation of the number of sampled flows is $\sqrt{Np(1 - p)}$. The entries in the IP table are at most $Np + 2.33 \sqrt{Np(1 - p)}$, and the relative error of the super point with F flows is $(\sqrt{1 - p/p})/F$.

3 Experimental Analysis

Here the SSD algorithm will be compared with Venkataraman's sampled algorithm (VSAM), and Zhao's bitmap algorithm (ZBIT). The VSAM algorithm randomly samples a certain percentage of source-destination pairs using a hashing technique to estimate the fan-outs of sources, which are counted and scaled by $1/p$ to obtain an estimate of the fan-out of the sources in the original traffic. This counting process is typically performed using a hash table to store the fan-out values of all the sampled sources so far, and a newly sampled flow will increase the fan-out counter of the corresponding hash node. The ZBIT algorithm also uses a hash-based flow sampling algorithm to approximately count the fan-outs of the sampled sources. The main contribution of the SSD algorithm is that the sampled traffic is further filtered by a simple data streaming module, which guarantees that at most one packet from each flow is processed, so it allows for a much higher sampling rate than the traditional hash-based flow sampling.

The following equations show the flow number unbiased estimators of the three algorithms and their variances. The flow number unbiased estimator of the VSAM algorithm is $E(x) = n/p$, that of the ZBIT algorithm is $\frac{1}{p} \sum_{i=1}^T \frac{m}{m - n_i}$, and that of the SSD algorithm is $\sum_{i=1}^T \frac{m}{m^k - n_i^k} + \frac{1}{p}$. The variance of the VSAM algorithm is $\text{var}(x) = n(1-p)/p^2$, that of the ZBIT algorithm is $\frac{1}{p^2} \sum_{i=1}^T \frac{n_i}{m - n_i} + \frac{E(x)(1-p)}{p}$, and that of the SSD algorithm is $\sum_{i=1}^T \frac{(mn_i)^k}{(m^k - n_i^k)^2} + \frac{1-p}{p^2}$. Now we can define a theoretical error metric $e_{v/E} = \text{var}(x)/E(x)$ to compare the three algorithms. So the error metric of the VSAM algorithm is $e_{v/E} = (1-p)/p$, that of the ZBIT is $e_{v/E} \approx (2T/m + 1 - p)/p$, and that of the SSD is $e_{v/E} \approx mT^2/(2m - T)^2 + (1-p)/p^2$ (k is set to 1), where T is the flow number.

Fig 2 is the simulation comparison among the three algorithms according to their theoretical error equations. In the example, T is set to 100 000, and m is set to 1 048 576. From the simulation results, we know that the theoretical error of the SSD is less than those of both the VSAM and the ZBIT.

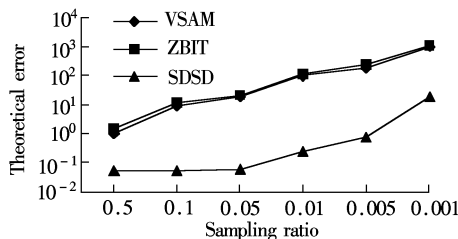


Fig 2 Theoretical error comparison

We also use packet header traces in this paper gathered at an OC-48 backbone link of CERNET at 18:48 pm on November 10, 2005^[8]. The CERNET trace is 60 s, with 128 350 source IP addresses, 759 041 flows, and 37 008 564 packets. If the super point threshold is set at 0.1% of the total flow

number, then the number of super points is 85 in the trace, and its threshold is 759 flows. In the experiment, the flow label consists of the four tuples: source IP, destination IP, source port, destination port, and the IP label is source IP. The super point is the IP label whose flow exceeds a predefined threshold of 0.1% of the total flow number.

In the experiments, both the SSD algorithm and the ZBIT algorithm use a bit array structure to record if the flow has arrived. The size of the bit array in the two algorithms is set at 128 KB, that is $128 \times 8 \times 1\,024 = 1\,048\,576$ bits. The flow sampling rates of all the three algorithms are set at $p = 1/8$. Before we begin to measure the accuracies of the different algorithms, an error metric e_{rel} is defined. The e_{rel} metric to estimate the average error of all n estimated super points is $e_{rel} = \sum_{i=1}^n \frac{|X_i - \hat{X}_i|}{X_i}$. Where X_i is the actual flow number of the i -th super point, and $\hat{X}_i$ is the estimated value of the i -th super point.

The estimated results of the super points using the three algorithms are compared with each other in Fig 3, where the X-axis is the true flow number of the super points, and the Y-axis is the estimated flow number of the super points. Let the super point threshold be T , and the range of the X-axis be defined as $[T, 2T]$ such that the range of the X-axis is $[759, 1\,518]$ in the CERNET trace. The diagonal line is used as a standard line to compare the detection performance. If these points are nearer to the diagonal line, then the estimated super point value is closer to the actual value. The dotted line above the diagonal line shows that the estimated values are 5% greater than the actual values, and the dotted line below the diagonal line shows that the estimated values are 5% less than the actual values. The points inside the two dotted lines have relative errors of less than 5%. Fig 3 shows that nearly all points of the SSD detection are inside the two dotted lines, while lots of points of the other two algorithms are outside the two lines. It means that the relative errors of detecting super points using the SSD algorithm are less than 5%, while the relative errors of the other two algorithms are more than 5%. The greater the distance between the point and the diagonal line, the greater the estimated error. Fig 4 explains the relative error distribution of the super points. The X-axis is the range of the relative error of the super point, and the Y-axis is the ratio of the relative errors of the super points in the range of the X-axis. As shown in Fig 4, most relative errors of the super points estimated by the SSD algorithm are less than 5%, which are better than the detected values of the VSAM algorithm and the ZBIT algorithm.

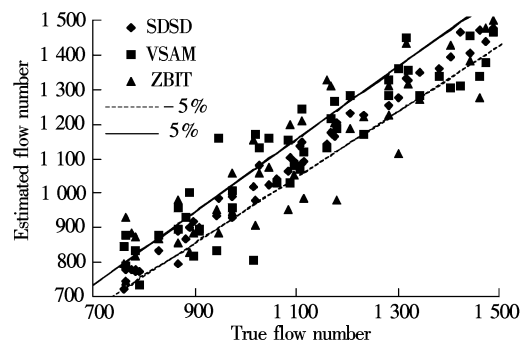


Fig 3 Comparison among three algorithms

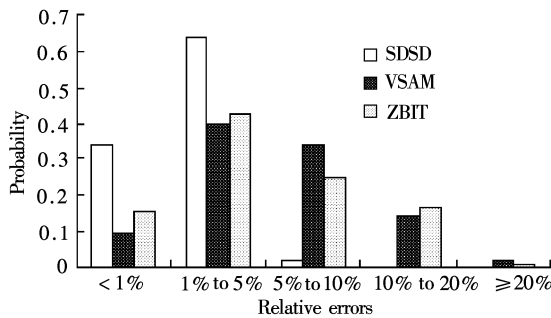


Fig 4 Relative error distribution of super points

4 Conclusion

It is a significant challenge in network management and security to detect the super points in high-speed network links efficiently and accurately. In this paper, we propose a new method for detecting super points to guarantee detection accuracy and memory requirements. Our method is based on sampling and data streaming algorithms, where the sampling technique can probabilistically guarantee to sample only super points and the data streaming technique sets a BF structure to save memory space.

The statistics method is adopted to analyze the memory space requirements and the estimated accuracy of super points, and a lower bound of sampling probability is deduced using a bounded variance estimator. The experiment with the CERNET traces shows that the SDSD can accurately detect super points, and efficiently save memory.

References

- [1] Moore D, Paxson V, Savage S, et al. Inside the slammer worm [J]. Security and Privacy Magazine, 2003, 1(4): 33-39.
- [2] Roesch M. Snort—lightweight intrusion detection for networks [C]//Proceedings of the 13th USENIX Conference on Systems Administration Berkeley, CA, USA: USENIX Association, 1999: 229-238.
- [3] Plonka D. Flow scan: a network traffic flow reporting and visualization tool [C]//Proceedings of the 14th USENIX Conference on Systems Administration Berkeley, CA, USA: USENIX Association, 2000: 305-317.
- [4] Venkataraman S, Song D, Gibbons P, et al. New streaming algorithms for fast detection of superspreaders [C]//Proceedings of the 12th Annual Network and Distributed System Security Symposium. San Diego, California, USA, 2005.
- [5] Zhao Q, Kumar Abhishek, Xu Jun. Joint data streaming and sampling techniques for detection of super sources and destinations [C]//Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement Berkeley, CA, USA: USENIX Association, 2005: 77-90.
- [6] Kamiyama N, Oriaki M, Ori Tatsuya, Kawahara Ryoichi. Simple and adaptive identification of superspreaders by flow sampling [C]//Proceeding of the 26th IEEE International Conference on Computer Communications Anchorage, AK, USA: IEEE INFOCOM, 2007: 2481-2485.
- [7] Kumar Abhishek, Xu Jun, Li Li, et al. Space-code BF for efficient traffic flow measurement [J]. IEEE Journal on Selected Areas in Communications, 2006, 24(12): 2327-2339.
- [8] Key Laboratory of Computer and Network Technology of Jiangsu Province. IP trace [EB/OL]. (2009-02) [2009-03]. <http://ntds.njnet.edu.cn/home/intra.php>. (in Chinese)

基于抽样和数据流算法的超点检测

程 光 强士卿

(东南大学计算机科学与工程学院, 南京 210096)

(东南大学江苏省计算机网络技术重点实验室, 南京 210096)

摘要: 为了提高超点检测的精度并控制测量资源的使用, 提出了一种基于抽样和数据流算法的超点检测方法. 该方法通过抽样从概率上保证发送或接收大量流的节点能被检测, 同时采用数据流技术建立了 IP table 和流 BF (BF) 两个数据结构. 其中 IP table 结构用于判断 IP 是否已经被创建, 如果已经被创建, 则将属于该 IP 的所有后续的流记录在流 BF 结构中; 如果 IP table 结构中不存在该 IP 记录, 则对属于该 IP 的流进行抽样. 对提出方法的精度和内存需求从理论上进行了分析, 并采用 CERNET 数据进行验证. 理论分析和实验测试表明, 提出的超点检测算法的测量误差基本控制在 5% 以内, 而其他算法的误差在 10% 左右. 另外, 由于使用 BF 数据结构, 提出的算法在使用空间上也优于其他算法.

关键词: 超点; 流抽样; 数据流

中图分类号: TP393.08