2008 Second International Conference on Future Generation Communication and Networking

A Detecting Superpoint Algorithm on Multiple Sampling Technology

Cheng Guang, Gong Jian
School of Computer Science & Engineering, Southeast University, Nanjing, P.R.China, 210096
gcheng@njnet.edu.cn

Abstract

Super points are sources or destinations that connect to a larger number of distinct destinations or sources during a measurement time interval. High-speed monitoring of super points is a challenging problem with application to real-time attack detection using a limited memory space. In this paper, we propose a method for detecting super points, and prove guarantees on their accuracy and memory requirements. Our method is based on sampling and data streaming, and sampling technique can probabilistically assure to sample only large-flow sources or destinations. Data streaming technique sets an IP bitmap and flow bitmap to judge an existed IP. Our method are both theoretically and experimentally more efficient than previous approaches.

1. Introduction

The problem of detecting super points arises in network monitoring and security applications. Many network attacks, such as DDoS attacks, worm attacks, network scan events, cause some sources or destinations IP address to produce or receive a large number of messages to distinct destinations or sources in a given measurement interval. For a lightly loaded OC-48 with a favorable traffic mix, a measurement system with a few hundred megabytes of memory 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.

This problem of detecting super points has been studied in recent years. Snort [1] and Flowscan [2] keep record for each source and destination. This straightforward approach doesn't have memory-efficient implementation, since the hash table typically requires large quantities of DRAM for operation, so this approach is not feasible for monitoring high-speed links. Venkataraman [3] proposes two flow sampling techniques for detecting super points. Qi Zhao [4] proposes two algorithms to solve the problems using data streaming algorithm and sampling technology. But his paper focused on estimating the number of flows in source or destination IP, and didn't give a method to keep the source/destination IP records. Noriaki [5] proposed an adaptive method of identifying super points by flow sampling.

Our method is based on sampling and data streaming, and can probabilistically sample only large-flow sources or destinations. After a packet arrives in the monitor, the algorithm checks the IP table to judge whether the IP which the packet belongs to 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 to judge whether the arrived flow is a new flow. If a new flow is arrived, then the IP entry in the IP table is updated. If the IP isn't found in the IP table, then we will sample the flow. If the flow is sampled, then the IP which the flow belongs to will be added into the IP table.

The rest of this paper is organized as follows. Section 2 analyzes our method in detail. Section 3 analyzes the accuracy of the algorithm. We evaluate this method and two other algorithms using the NLANR traces in Section 4, and end with conclusions in Section 5.

2. Super points Detection algorithm (SDMA)

After a packet arrives in the monitor, the algorithm checks the IP table to judge whether the IP which the packet belongs to 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 to judge whether the arrived flow is a new flow. If a new flow is arrived, then the IP entry in the IP table is updated. If the IP isn't found in the IP table, then we will sample the flow. If the flow is sampled, then the IP which the flow belongs to will be added into the IP table.

The algorithm submits it to the bloom filter process directly if the source or destination IP in the packet belongs to an entry in the source or destination IP memory, otherwise, the flow sampling process samples this flow random with a probability p. Let a flow identifier of a packet be x, a hash function h produce a hash value h(x), the maximum of the hash value be max, and p be sampling probability. If h(x)/max is less than p, then the source or destination IP of this packet is sampled, and the source or destination IP is added in the source or destination IP memory. After the IP is added, the sample & hold process will submit all its subsequent packets whose source or destination IP is equal to the IP. When a packet passes through the sample & hold process, the bloom filter will detect the bloom filter data memory to

978-0-7695-3431-2/08 $25.00 © 2008 IEEE
DOI 10.1109/FGCN.2008.111

judge whether the destination or source IP of this packet belongs a new or existing IP. If it has no record in the bloom filter data memory, then its information is added into the bloom filter, and the source or destination IP memory is also updated at the same time.

In order to save the source or destination IP memory, the early-removal process removes some small source or destination IP entries from the memory. The measured interval is divided into several sub-intervals and the early-removal process checks and removes some small entries from the source or destination IP memory based on some predefined thresholds in every sub-interval.

This method consists of three processes and two data structures. The three processes are flow sample & hold, bloom filter, and early-removal. The flow sample & hold process decides whether to measure the flow, the bloom filter judges whether the packet belongs to a new flow, and the early-removal process removes some small entries from the source/destination IP data memory. We will discuss and analyze the three processes as following.

We define a bitmap to record these flows which have been written into IP table and an IP table used to keep the IP records and its flow number. Let the size of the bitmap be w bits. The bitmap is initialized to all "0" at the beginning of the measurement interval. The bitmap is used to record whether the flow is existed. We set a hash function h that maps a flow label to a value uniformly distributed in [1, w]. As soon as the arrival of a packet pkt, we hash its flow label (pkt.sourceIP, pkt.destIP, pkt.sport, pkt.dport) using h hash function,

r= h(pkt.sourceIP, pkt.destIP, pkt.sport, pkt.dport) (1)

The result r is treated as an index into the bitmap B. if B[r] is equal to 1, then the flow has arrived and has been processed earlier, and we will miss to process the packet. If B[r]=0, then the flow of the packet belonging to is a new one. The IP record in the IP table is updated.

When a new packet arrived in the monitor, first we check the IP table. If the IP which belongs to the arrived packet has no entry in the IP table, then as with ordinary sampling, we sample the flow with a probability. If the flow is sampled, then a new entry is recorded in the hash table. If the IP has recorded into the IP table, we will check the bitmap to judge whether the flow is a new flow. If the flow is new one, then the IP entry is updated. By this method, after an IP entry is created for a source or destination IP, we will update the entry for every subsequent flow belonging to the source or destination IP.

Let a flow identifier of a packet be x, a hash function h produce a hash value h(x), the maximum of the hash value be max, and p be sampling probability. If h(x)/max is less than p, then the source or destination IP of this packet is sampled, and the source or destination IP (SDIP) is added in the SDIP memory. Because all packets of a flow have same hash values, the number of packets in a flow does not affect its probability to create of an entry in the SDIP memory.

3. Accuracy Analysis

We formally quantify the probability that a super points with a certain number of flows is not detected. Let F be a threshold of the number of flows of a defined super points and p be the flow sampling probability. The probability p' that the super points with F flows will not have an entry is $p' = (1-p)^F \approx e^{-Fp}$.

Let be a little number. We define p' as a probability that a super points with F flows is missed must be less than or equal to , i.e., $p' = e^{-Fp} \le \delta$. Thus, the sampling probability p should satisfy the equation (1) to detect the super point with F flows. For example, if we need to detect a super points whose flow number is larger than 100, and let $\delta \le 0.0001$. We can compute p >= 0.092 by the equation (2).

$$p \ge \ln(1/\delta)/F \qquad (2)$$

Let a super point has s flows, when the first flow of the super point is sampled, the number of passed flows in the s flows is X. X is a geometric probability distribution, it's 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, $\mathrm{var}(X) = (1-p)/p^2$. The estimated value of x is the equation (3),

$$\hat{x} = 1/p \qquad (3)$$

Its variance is $(1-p)/p^2$.

After the first flow of an IP is recorded into the IP table, all subsequent flows which belong to the IP will be measured, and its subsequent flow will be checked by the bitmap to judge whether the flow is a new one. When a new flow is judged in the bitmap, suppose the number of "0" entries in bitmap B (with size w) is u right before a packet pkt with source s arrives. Assume the flow is a new flow, and the flow identifier is mapped into B[r]. Then value of B[r] is "0" with probability u/w, and the probability of B[r]=0 is 1-u/w. So we can use w/u to update the IP flow numbers N(s).

Suppose in the measurement interval, we find k flows of belonging to s IP from the bitmap B. So we can get an unbiased estimator of s IP flow number after the first flow of s IP is recorded into the IP table

$$\hat{N}_s = \sum_{i=1}^{k} w/u_i \qquad (4)$$

Let $\delta = 0.0001$ be a probability threshold that a super points is missed and =0.1 be a relative error threshold which an error of an estimator of the number of super points is accepted. And let the threshold F of the flow minimum of super points be F=100. If the condition in the equation (2) is satisfied, then the minimal probability for flow sampling is 0.092, and if the condition in the equation (4) is satisfied, then the minimal probability for flow sampling is 0.095. If we set the sampling probability to 10%, then we can assure that the super points with over 100 flows is detected and evaluated with $\delta \le 0.0001$, and =10%.

As a second example, let F=1000, $\delta \le 0.0001$, =10%, Equation (2) needs the sampling probability larger than 0.0092, and equation (4) requires that it be larger than 0.0092. If we set the sampling probability to 1%, then the two conditions can be satisfied.

When the ith flow arrives the bloom filter (a bloom filter structure with m bits spaces, and k hash functions), the n_i bit positions are 1, so the probability that a hash value of a new flow entries into one 1 bit position is n_i/m. Because the probability that all k hash function entry into 1 bit positions is $(n/m)^k$, 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 SDIP flows on the sampled traffic, we should statistically compensate for the fact that with probability $1-p_i$, the bit in the bloom filter has value 1 and the flow will miss the update to the SDIP memory due to aforementioned hash collisions. It is intuitive that if we add $1/p_i = 1\big/(1-(n_i/m)^k)$ to the SDIP entry, the resulting estimator is unbiased, and its variance is $(1-p_i)\big/p_i^2 = (n_i/m)^k\big/(1-(n_i/m)^k)^2$.

To be more precise, suppose in a measurement epoch, the BF is updated by altogether T flows {flow$_j$, j=1, 2, ...,T} from a SDIP s. The output of the SDIP memory, which is an unbiased estimator of super points on the sampled traffic, is the equation (5), and its variance in equation (6).

$$E(\hat{s}) = \sum\nolimits_{i=1}^{T} \left(1\big/\left(1-(n_i/m)^k\right)\right) \qquad (5)$$

$$Var(\hat{s}) = \sum\nolimits_{i=1}^{T} \frac{(n_i/m)^k}{(1-(n_i/m)^k)^2} \qquad (6)$$

If we consider the FSH, and the BF at the same time, we can establish the following equation (7) and (8) to characterize an unbiased estimator $\hat{s}$, and the variance of the estimator $\hat{s}$ according to equation (3), (5), (6).

$$E(\hat{s}) = \sum\nolimits_{i=1}^{T} \left(1\big/\left(1-(n_i/m)^k\right)\right) + 1/p \qquad (7)$$

$$Var(\hat{s}) = \sum\nolimits_{i=1}^{T} \frac{(n_i/m)^k}{(1-(n_i/m)^k)^2} + (1-p)/p^2 \qquad (8)$$

We define a relative error , and assure the estimation error of the super points with F flows less than or equal to , i.e., $\left(\sqrt{1-p}\big/p\right)\big/F \le \varepsilon$, so the sampling probability p should satisfy equation (9). For example, let a super points have larger than 100 flows, and its relative error be 0.1, we can compute the p>=0.095 according to equation (9).

$$\left(\sqrt{1+4\varepsilon^2F^2}-1\right)\Big/2\varepsilon^2F^2 \le p \qquad (9)$$

4. Evaluations

We use two groups packet header traces gathered at NLANR [6] to test the model. The First group traces (Traces1 and Traces2) (IPLS-CLEV) were collected on two OC48c links at IPLS router node, on August 14, 2002, from 9:00 am to 9:10 am. The second group traces (Traces 3 and Traces 4) used OC192MON hardware to collect data on August 19, 2004, from 13:40pm to 13:50pm. Table 1 summarizes the information for threshold of F=100 and F=1000 in the data traces.

Data Name	# of flow	# of super point, F=100	# of super point, F=1000
A0S	8432	281	8
A0D	33760	103	8
A1S	11747	104	9
A1D	47232	190	1
C0S	26160	644	268
C0D	76526	359	15
C1S	18050	236	11
C1D	104333	143	4

In this table, A0S means the source IP flows in the No.1 traces; A0D is the destination IP flows in the No.1 traces; A1S means the source IP flows in the No.2 traces; A1D is the destination IP flows in the No.2 traces; C0S means the source IP flows in the No.3 traces; C0D is the destination IP flows in the No.3 traces; C1S means the source IP flows in the No.4 traces; C1D is the destination IP flows in the No.4 traces. "# of flow" means the number of flows, "# of super point, F=100" means the number of super points which is defined with over 100 flows, "# of super point, F=1000" means the number of super points which is defined with over 1000 flows.

We compare the three algorithms: Venkataraman, Zhao, SDMA. Venkataraman's algorithm [3] uses flow sampling algorithm that estimates the fan-outs of sources. Their algorithm randomly samples a certain percentage of source-destination pairs using a hashing technique. Zhao's algorithm [4] also uses a hash-based flow sampling algorithm to approximately count the fan-outs of the sampled sources. Its main difference is that the sampled traffic is further filtered by a simple data streaming module. This allows for much higher sampling rate than achievable with traditional hash-based flow sampling.

Before we begin to examine the measured accuracy of different algorithms, two error metrics are defined. The error metrics in the equation (10) evalues the estimated error of the ith super point. Where Xi is the actual flow numbers of the ith super point, and $\hat{X}_i$ is the estimated value of the ith super point. The avg_error metrics in the equation (11) is a average estimated error of all n estimated super points, where n is the number of super points.

$$error_i = (X_i - \hat{X}_i)/X_i \times 100\% \qquad (10)$$

$$avg\_error = \sum_{i=1}^{n} error_i / n \qquad (11)$$

The measured average error of all super points, which is larger than 100 flows, is compared among the three algorithms using the sampling rate 10% in the figure 1. X axis is the No. of traces in the table 1, and y axis is the average error.

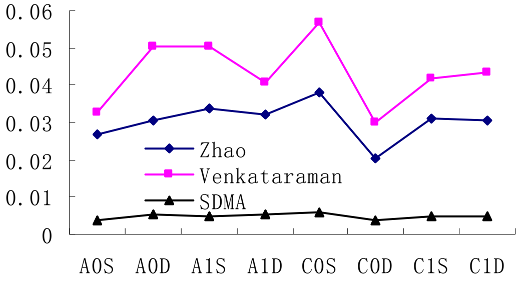

Figure 1 Average Error of Super Points

The figures 1 shows that the average accuracy of SDMA's algorithm in our paper is better than that of Zhao, and Venkatarman when the super point is defined over 100 flows and the sampling rate in all algorithms is 10%.

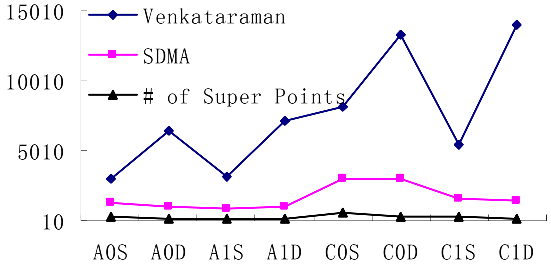

Figure 2 Cached IP number and the number of super points

Figure 2 is the cached aggregation point entries using the SDMA algorithm and Venkataraman algorithm to detect the super points which the IP has over 100 flows. Venkataraman algorithm and Zhao algorithm have almost the same cached aggregation point entries, so we only give the Venkataraman's result. In the figure 2, we also give the number of super points in these traces. Figure 2 shows that the SDMA can save more memory than the Venkataraman because a removal process which is used in the SDMA algorithm can reduce the aggregation point entries in the aggregation point memory.

5. Conclusion

It is a significant challenge in network management and security to detect super points in the high-speed network links efficiently and accurately. In this work, we propose a new method for detecting super points that guarantees accurate and exhibits a realistic memory requirement. Our method is based on sampling and data streaming, and sampling technique can probabilistically assure to sample only large-flow sources or destinations. Data streaming technique sets an IP bitmap and flow bitmap. We show experimental results on real network traces.

Acknowledgments

This work has been supported by the National Grand Fundamental Research 973 program of China under Grant No. 2009CB320505, the Excellent Youth Teacher of Southeast University Program under Grant No. 4009001018, and the Free Research Program of Key Lab of Computer Network in Guangdong Province under Grant No. CCNL 200706, and the 2008 Natural Science Fundamental Program of Jiangsu Province "the Adaptive Sampling Measurement Method of the Anomaly Hotspots in the Next Generation Network".

References

[1] M.Roesch. Snort-lightweight intrusion detection for network. In proc. USENIX Systems Administration Conference, 1999.

[2] D. Plonka. Flowscan: A network traffic flow reporting and visualization tool. In USENIX LISA, Dec. 2000.

[3] S. Venkataraman, D. Song, P. Gibbons, and A. Blum. New streaming algorithms for fast detection of superspreaders. In Proc. NDSS, 2005.

[4] Qi Zhao, Abhishek Kumar, and Jun Xu, Joint Data Streaming and Sampling Techniques for Detection of Super Sources and Destinations. IMC 2005, Pages:77 – 90.

[5] Noriaki Kamiyama, Tatsuya Mori, Ryoichi Kawahara: Simple and Adaptive Identification of Superspreaders by Flow Sampling. 2481-2485, INFOCOM 2007.

[6] NLANR PMA: Special Traces Archive, http://pma.nlanr.net/Special/.