A New Algorithm for Long Flows Statistics—MGCBF

Zhou Mingzhong, Gong Jian, and Ding Wei

Department of Computer Science and Engineering,
Southeast Univercity, Nanjing, Jiangsu, China 210096
{mzzhou, jgong, wding}@njnet.edu.cn

Abstract. Long flows identification and characteristics analysis play more and more important role in modern traffic analysis because long flows take main traffic payload of network. Based on the flows length distribution and long flows characteristics of the Internet, this paper presents a novel long flows' counting and information maintenance algorithm called Multi-granularity Counting Bloom Filter (MGCBF). Using a little fix memory, the MGCBF maintains the counters for all incoming flows with small error probability, and keeps information of long flows whose length are bigger than an optional threshold set by users. This paper builds up an architecture for long flows' information statistics based on this algorithm. And the space used, calculation complexity and error probability of this architecture are also discussed at following. The experiment applied this architecture on the CERNET TRACEs, which indicates that the MGCBF algorithm can reduce the resource usage in counting flows and flows information maintenance dramatically with losing little measurement's accuracy.

1 Introduction

Flow-based measurement is wildly used in network usages just like accounting, bandwidth measurement and network security, etc. A flow is defined as a stream of packets subject to flow specification and timeout. The flow definition can be changed according to its usage, but recent studies show that a very small percentage of flows carry the majority of the packets and bytes [1][2][3][8] regardless of their definition. And so it is very important to improve the network performance by finding out these heavy-hitter flows (called long flows).

In the following of this section, we will introduce the recent related works on long flows identification and statistics, and indicate the main contributions of the Multi-granularity Counting Bloom Filter (MGCBF) algorithm introduced by this paper in flow identification briefly. In the next section, the algorithm is presented in detail, and its performance and error probability are anatomized. And then the optimizations are also expressed. It is proved that the error probability can be controlled through parameters adjusting. In Section 3, some experiments based on the TRACEs from CERNET are employed to illustrate and evaluate the performance and error probability of MGCBF comparing with traditional hash method and CBF method. At last, we discuss the usages of MGCBF in other domains and present the future works.

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part IV, LNCS 3994, pp. 112 – 119, 2006.

1.1 Related Works

As the increasing needs of flow-based traffic measurement, Long flows identification and counting are widely studied as one of its main branch [2][3][4][7].

As most widely used way of flow identification, the method presented by IETF RTFM group can gather total or parts of flows information that transmitting by the router. But sampling is widely used in these supports because of the resource restriction of routers in flow identification and exporting according to the RTFM criterions. And sampling satisfies the needs of performance by losing the precision. A. Shaikh, J. Rexford and K. G. Shin[3] applied a method to realize the load balance by keeping the information of every flow, and judged the flow belonging to long or not by the packet number it arrived in a fixed time unit. This is a direct but not efficient way. Smitha, I. Kim, and A. L. N.Reddy.[2] provided an algorithm called Least Recent Used (LRU). This algorithm can be used to identify long existing and heavy-hitter flows for load balance, but it can only maintain flows information in a short time and must refresh frequently, and so long flows with large duration will not be kept. C. Estan and G. Varghese [4] used two methods to find out long flows: *sample and hold* and *multistage filters*. And both of them resolve the problem of storing flows information by packet sampling efficiently, but they are only used to find out and keep the information of those long flows, which take a very large ratio of total traffic (i.e. the flow volume is larger than 0.1% of total traffic). A. Kumar, J. Xu, et al.[7] provided an algorithm called SCBF for flow counting, which used limited resource to store all flows' length information with a little errors by sampling. This algorithm applied maximum likelihood estimation (MLE) and mean value estimation (MVE) to estimate the length of every flow. C. Estan, G. Varghese and M. Fisk [10] described a bitmap algorithm to take count of the flows with different length with little storage resource. But all those algorithms and methods do not maintain the detail information for flows, which can satisfy the needs of special applications (i.e. load balance, traffic accounting).

1.2 Main Contribution

This paper proposes a long flows counting and identification algorithm called Multi-granularity Counting Bloom Filter (MGCBF) based on standard bloom filter according to the study of flow distribution and characteristics in detail. This algorithm has the features of adaptability and expansibility because a threshold value can be set according to different usages. And the MGCBF takes advantage of the heavy-tailed distribution of flow length in Internet backbone, and uses a new structure to store flows information, which can save the resource dramatically. This algorithm does not maintain the flow information whose length less than the threshold, and its resource usage is less than other prevalent algorithms [1][11] that kept the flow information.

2 Flow Identification and Statistics Based on MGCBF

This Section provides the prototype of MGCBF based on introducing the standard bloom filter. And the improving model of MGCBF introduced following can decrease its complexity and increase its precision. In the end of this section we will analyze the performance and error rate of this algorithm.

2.1 MGCBF Prototype

For large datasets querying, the time and space complexity are main problems that need to be solved [4][8][9]. MGCBF provided by this paper operates iceberg query to the datasets which items frequencies follow a heavy-tailed distribution. It employs a serial of CBFs (MGCBF=$\{cbf_0,cbf_1, \ldots, cbf_{h-1}\}$) which use different count spaces ($C=\{1,c_1,c_2, \ldots, c_{h-1}\}$) to count the frequency of different items in the set.The prototype of this algorithm is introduced as following:

1) When an item x wanted to add into MGCBF, the counters at positions $h_1^0(x)$, $h_2^0(x)$, …, $h_{k0}^0(x)$ in vector V_0 increase 1. (V_0 is the vector of MGCBF's first CBF, cbf_0. h_1, h_2, …, h_{k0} are the hash functions in cbf_0. Without the loss of generality, we suppose $h_1^0(x) \le h_2^0(x) \le \ldots \le h_{k0}^0(x)$).
2) Then check the value $h_1^0(x)$. If $h_1^0(x)=c_1$, the counters at positions $h_1^0(x)$, $h_2^0(x)$, …, $h_{k0}^0(x)$ decrease c_1, then values in the counters change to 0, $h_2^0(x)-c_1$, …, $h_{k0}^0(x)-c_1$; the counters at positions $h_1^1(x)$,$h_2^1(x)$, …, $h_{k1}^1(x)$ in V_1 which is the vector of cbf_1, that means $h_1^1(x)+1$, $h_2^1(x)+1$, …, $h_{k1}^1(x)+1$.
3) And then check the value $h_1^1(x)$. If $h_1^1(x)=c_2$, we operate the same action as 2) in cbf_1 and cbf_2. And the check does not stop until there is no carrying or the cbf_h are checked. If we suppose the set of counters' minimum value which is set by an item x is $M(x)=\{min_0(x),min_1(x), \ldots, min_{h-1}(x)\}$, then the frequency of x in S is:

$$Counter(x) = min_0(x)+min_1(x)*c_1+ \ldots + min_{h-1}(x)* \prod_{i=1}^{h-1} c_i \circ \quad (1)$$

The flow length follows a heavy-tailed distribution in Internet [2][3][4], which means very few long flows take most network traffic. And so it is more efficiency that we use the MGCBF algorithm to replace the traditional counting methods.

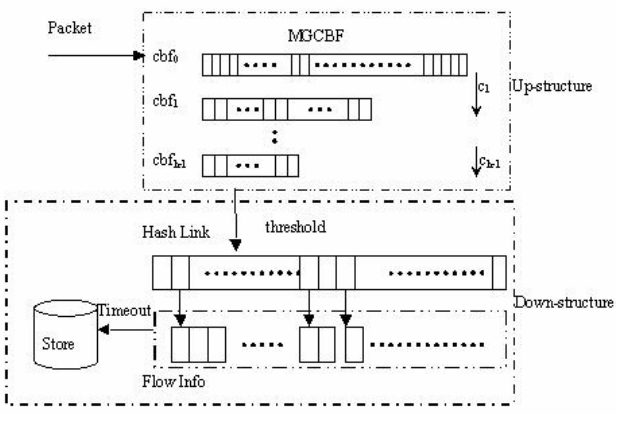

Fig. 1. Long flow information statistics model based on MGCBF

Fig. 1 illustrates the long flow information statistics model based on MGCBF. The data structure of MGCBF in the upside of this figure is used to maintain the packet number of every active flow, and the downside of this figure describes the data struc-

ture used to maintain the flow information by a hash table and several link tables. As a packet incoming, this structure upside creates a new flow or refreshes length of the flow which this packet belongs to; if a particular flow's number is bigger than the threshold, the flow number and other information (flow id, duration etc.) will be pushed into the downside structure.

Using a prearranged ***threshold*** for flow length, this algorithm can maintain all kinds of flows information, whose length are bigger than 1 packet. With reducing of the threshold value, the stages of MGCBF are reducing, and this will reduce the spending of this algorithm. When the threshold is reduced to a value small enough (i.e. threshold≤10), MGCBF will degrade to CBF.

2.2 Optimizations

Because the number of different items in the set is very large while the volumes of MGCBF's vectors are small relatively, the conflictions of hash functions in MGCBF are inevitable. This section will introduce two methods to improve the performance and reduce the error rates of this algorithm.

(1) Periodical Refreshing

When the items number of a set is very large (i.e. the packets number measured in some point in 24 hours), the volume of the set space V is too large to be stored in the memory of a measurement system when MGCBF is employed to analyze this set. Even though the measurement system can maintain this huge structure, it can't be applied because of the low cost performance.

The set $\boldsymbol{S}$ is divided into several subsets using a method called periodical refreshing, $S=\{S_1,S_2,\ldots,S_\gamma\}$, and that means the original set space is divided γ equal subspace. For every subset S_i, we use MGCBF to calculate and statistical analysis. When the first subset is finished, we initialize the MGCBF structure for next subset but unchanged the down structure in Fig. 1. Because in flow measurement the subsets are defined by the packets passing from a measurement points in the network for a fixed period, this method is called ***periodical refreshing***. The cost of this method is breaking the relationship between the subsets S_i and S_{i+1}, which may introduces errors to the measurement and increases the costs of calculation. The error analysis and calculation cost are illustrated in Sect. 2.4.

(2) Recurring Minimum

About the original errors coming from Bloom Filter, B.Bloom illustrated them in detail as he introduced his algorithm in [5]. The following is the probability error of Bloom Filter when the input and the parameters of Bloom Filter are fixed.

m/n=8 k=6 E=0.0215
m/n=12 k=8 E=0.00314

Recurring minimum method proposed by Cohen [6] uses an affiliated CBF called CBF_t to reduce the progressing counting errors. An example presented in [6] shows that $E_{RM}<E/18$ when the parameters are set as following: k=5, n=1000, m/n= k*ln(2)=0.7k, $m^s=m/2$ (m^s is the vector space of the CBF_t). It is to say that the error probability can be reduced to 1/18 by using the recurring minimum method, which only increase 1/2 storage space and 1- $P(R_x)$ calculating cost.

On realization of Long flow information statistics model by MGCBF, only the high-stage CBFs using recurring minimum is in the balance of the counting error influence and the compute complexity.

2.3 Performance Analysis and Error Estimation of MGCBF

In this section, we evaluate the algorithm efficiency and estimate the probability errors of the long flow information statistical model by introducing MGCBF used in this model. The maximal counting value in MGCBF is the threshold denoted long flow because when the counting value of some flow is added to the threshold, it will be submitted to downside structure illustrated in Fig. 1. But it can be proved that this above analysis will not lose its generality if only the items sequence in the set needed checking follows heavy-tailed distribution.

(1) Performance

Firstly, we suppose that flows information whose packet number bigger than 1000 (***Threshold***=1000) needs to gather. Considering the performance and costing, we set the MGCBF two stages (h=2), and the second stage CBF uses recurring minimum method to reduce the error probability. Because the flows whose length is smaller than 16 packets take above 90 percents of total flows [3]. If the first stage CBF used the counter whose maximum value is 16 (c_1=16), the second stage CBF vector V_2 can be as small as 1/10 of the first stage CBF vector V_1. Referring to§2.2 equation (1) and with the parameters: Threshold=1000, c_1=16, we can calculate the value c_2=64 by $MAX(min_1)=(Threshold-min_0)/c_1 = (1000-16)/16=61.5<64=2^6$. According to the suggestion L. Fan, P. Cao, et al. proposed in [11], when a serial of unrepeated items insert into one CBF, if the counter length is set to 4 bit, the possibility of counter overflowing by adding can be ignored.

$$Pr(\max(c)\geq 16) \leq 1.37\times 10^{-15}\times m$$

the left-side of inequation means the possibility of counter overflowing, m is the vector space. Then we can define the counter volume of first-stage CBF is $\log_2(16)+4=8$ bit, and the counter volume of second-stage CBF is $\log_2(2^6)+4=10$ bit, then we can calculate the space of MGCBF structure in the next equation.

$$M_{MGCBF}=8\times m_1+10\times m_2+1/2\times(10\times m_2)=9.5m_1$$

While using traditional CBF to store the flow number information, the space needed can be calculated as following:

$$M_{CBF} = (\log_2(1000)+4)m_1=14m_1$$

When we set threshold 1000, MGCBF can save 1/3 storage space than CBF; and when the threshold changes to 65536, the space saving ratio will change to 1/2. The storage space needed is reducing rapidly as the threshold increasing because of the heavy-tailed distribution of flow length.

We suppose that cbf_1 will refresh every c_1 packets coming on average. We denote τ as the time for inserting and/or extracting an item from the CBF, and then we can deduce that the mean calculating time for every packet in this MGCBF fitting with two-stages is $\tau_{MGCBF}<\tau_0+1/c_1\times\tau_1$. For assuring the precision of CBF in high-stage, we set the parameter $k_1=\alpha k_0>k_0$; And the calculation complexity introduced by using

recurring minimum in cbf_1 is about 20% [6], that means $\tau_1=1.2\alpha\tau_0$. Then we can deduce the calculation costing of every packet in MGCBF is $\tau_{MGCBF}<(1+1.2\alpha/c_1)\tau_0$. When the parameters are set as following $c_1=16$, $\alpha=4/3$, we can get the result $\tau_{MGCBF}<1.1\tau_0$, which means that every packet increases only 1/10 calculation costing on average. It is also can be proved that the increasing scope of calculation costing decreases with the stages of MGCBF increasing.

(2) Error Analysis

The errors in flow statistical model used MGCBF can be divided into two types: (1) the error of MGCBF; (2) the error introduced by periodical refreshing.

The error ratio of the MGCBF algorithm can calculate individually in different stages: cbf_0 set as E_0, cbf_1 set as E_1. It only introduces error ratio E_1 in the cbf_1 of this MGCBF compared with traditional CBF. The error ratio increase about 1/288 when we compare the error ratio of MGCBF with that of CBF in the same scale because the total error ratio of MGCBF can be denoted as this equation: E_0+1/c_1*E_1. And so we can conclude that the MGCBF error probability is determined by its first stage (cbf_0). The error probability of high stages can be dismissal.

The measurement errors caused by the periodical refreshing is mainly caused by the truncation of the long flows whose lengths are bigger than threshold but not reaching the threshold, denoted as $\boldsymbol{E_t}$. It may cause some long flows or partial packets of long flows being discarded. If we suppose the long flows incoming rates is not changing badly, the discarded long flows number will be determined by the *threshold*, *flows' rate* and *flows' timeout value.* The long flows whose rate is v takes η percents of total flows, the flows number in time unit is s', flows timeout is To, then we can deduce the probability flows number that is influenced:

$$s'_f = \sum_{i=1}^{n} \int_0^{To} \eta_i(t) s' dt$$

$\eta_i(t)$ is the percents of flows in total flows whose rate is $v_i<\text{threshold}/(To-t)$. For most long flows, $\eta_i(t)$ is a very small value at $To-t<<To$, and $\eta_i(t)$ is almost 0 when this condition is changed. Then we can conclude that $s_f'<<(s'*To)$. In Sect. 3, the experiments using different TRACEs prove that the second type errors caused by periodical refreshing cannot be larger than 1%. The periodical refreshing method proposed by this paper can reduce the storage space dramatically by the costing of little compute complexity and little flow identification precision.

3 Experiments

The datasets used by the experiments are the TRACEs gathered from the CERNET backbone in different time: CERNET1 and CERNET2.

Table 1. Flow length distribution of the TRACEs

	Total flows number	Long flows number (threshold =1000)	percentage
CERNET1	17164783	30316	0.17%
CERNET2	59987620	80850	0.13%

The experiment results show that, the error probability difference between MGCBF and CBF in the long flow information maintenance and the flow number storage is no more than 1% (CERNET1 is 0.26%, and CERNET2 is 0.81%). This difference is caused by periodical refreshing and second-stage CBF, for the latter we can estimate this probability by calculating, and then we can get the error caused by periodical refreshing. The error probability difference between MGCBF and classical flow information method is about 2% (CERNET1 is 1.6%, and CERNET2 is 1.3%), so we can also estimate the error probability caused by the first-stage CBF in MGCBF.

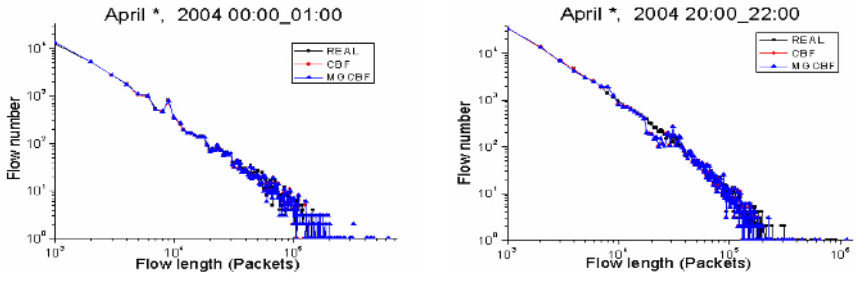

Fig. 2. The contraction of long flows distribution using different measurements

The experiment results illustrate that the long flows measurement based on MGCBF can improve the efficiency of flow information maintenance and save the storage space with little flow number measurement errors.

The method using the CBF takes more space in vector V for maintaining the flow information and it will increase as the incoming packet number increasing. Using classical flow maintenance method, it needs at least $29.05*10^6$ bytes to store the flow information; And using CBF method, the space is comprised of two parts: CBF vector space and flow information maintenance space, and it needs $224.03*10^6$ bytes; While using MGCBF method, it only needs $19.32*10^6$ for storing. The difference in structure between MGCBF and CBF makes MGCBF save only 34% space. The large difference in space between these methods we applied mainly caused by the periodical refreshing, which divides the dataset into several subsets and tackles them separately. Because calculation complexity of hashing is much less than that of numerical comparison, CBF and MGCBF are better than the traditional flow maintenance method in calculation complexity. And MGCBF's performance is much better than CBF when the distribution of items frequecy in the dataset follows heavy-tailed.

4 Conclusion and Future Work

This paper presents a long flow statistical and maintaining model based on MGCBF according to characteristics of flow length's heavy-tailed distribution in networks. This model can reduce the resource dramatically with little calculating cost, and maintain the flow information without losing the integrality of long flow information which does not exist in other flow length distribution and estimation algorithms. Fur-

ther more, this model also has excellent expansibility to maintain all flow information whose length is longer than 2 packets. And this model can also be widely used in other related domains if the frequency of items in the datasets follows heavy-tailed distribution.

Acknowledgement

This research is partially support by the National Basic Research Program (called 973 Program), No. 2003CB314803; Jiangsu Province Key Laboratory of Network and Information Security BM2003201 and the Key Project of Chinese Ministry of Education under Grant No.105084.

References

[1] N. Brownlee, C. Mills and G. Ruth. Traffic Flow Measurement: Architecture. RFC 2722. October, 1999.

[2] Smitha, I. Kim, and A. L. N. Reddy. Identifying Long Term High Rate Flows At A Router. In *Proceedings of High Performance Computing*. December, 2001.

[3] A. Shaikh, J. Rexford and K. G. Shin. Load-Sensitive Routing of Long-Lived IP Flows. In *Proceedings of the ACM SIGCOMM 1999*.Cambridge, M.A., USA. August, 1999.

[4] C. Estan and G. Varghese, New Directions in Traffic Measurement and Accounting, In *Proceedings of the ACM SIGCOMM 2002.* Pittsburgh, P.A., USA. August, 2002.

[5] B. Bloom, Space/Time trade-offs in hash coding with allowable errors. In *Commun.ACM*. Vol. 13, no.7, pp. 422-426, July 1970.

[6] S. Cohen, Y. Matias. Spectral Bloom Filters. In *Proceedings of the ACM SIGMOD 2003*. San Diego, C.A., USA. June, 2003.

[7] A. Kumar, J. Xu, et al. Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement. In *IEEE Infocom 2004*, Hongkong, China. March, 2004.

[8] R. M. Karp, S. Shenker, and C. H. Papadimitriou, A Simple Algorithm For Finding Frequent Elements In Streams And Bags, In *ACM Transactions on Database Systems (TODS)*. vol. 28, pp. 51–55, 2003.

[9] M. Charikar, K. Chen, and Farach-Colton, Finding Frequent Items In Data Streams, In *ICALP. Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, Germany*. 2002.

[10] C. Estan, G. Varghese and M. Fisk. Bitmap Algorithms For Counting Active Flows On High Speed Links. In *Proceedings of the ACM IMW,* 2003.

[11] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol. *IEEE/ACM Transactions on Networking*, 8(3):281-293, 2000.