An Adaptive Fair Sampling Algorithm Based on the Reconfigurable Counter Arrays

Jing Wang(✉), BingQiang Wang, Xiaohui Zhang, and YunZhi Zhu

National Digital Switch System Engineering & Technology Center,
Zhengzhou, 450002, China
Wangjingniu_2003@sina.com,
{wangbingqiang,zhangxiaohui}@ndsc.com.cn, zhuwangzilz@163.com

Abstract. At present, how to trade off the balance between the memory resources and sampling accuracy balance has become one of the most important problems focused on by the network packet sampling algorithms. This paper discusses a novel adaptive fair packet sampling algorithm (AFPS) to solve the above problem by improving the use ratio of memory resources. The key innovation of AFPS is the reconfigurable counter structure composed of two counter arrays, by which the AFPS count the small flow and large flow in a differential way and the size of two arrays can be adjusted adaptively according to the dynamic flow size distribution. The reconfigurable counter structure ensures not only a high memory use ratio value under different network conditions but also accurate estimation of small flows so that the overall sampling accuracy of AFPS is improved. The theoretical analysis and evaluation on real traffic traces show that AFPS can estimate the small flows accurately and the estimation error of the large ones' equals to SGS. Besides AFPS keeps the memory resource use ratio on almost 0.952 under different conditions so that it can use the memory resource efficiently.

Keywords: Network traffic measurement · Packet sampling · Estimation error · Reconfigurable parameter

1 Introduction

Network traffic measurement is essential for network routing, management and security. As the rate of networks links increases rapidly, the confliction between limited measurement resources and measurement accuracy becomes more and more serious. In order to improve the measurement accuracy, packet sampling [1] is usually used. Packet sampling can reduce the number of flow records as well as keep the traffic's primitive characters.

The traditional packet sampling algorithms often keep the flow records in memory resources such as SRAM, DRAM or other fabric structures. The size of memory resources used to store the information of sampled flows affect the accuracy of measurement result directly. Due to the limited memory resources, the information of sampled flows can only be recovered partly. So how to make a trade-off between memory

V.C.M. Leung et al. (Eds.): TridentCom 2014, LNICST 137, pp. 371–378, 2014.
DOI: 10.1007/978-3-319-13326-3_35

resource and measurement accuracy is the most important problem to be solved. Some algorithms only focus on parts of flows which they are interested in just like SH and MBF [2]. Others adjust sampling rates adaptively according to the flow size or some other conditions [3]. Although these improved sampling algorithms can ensure the accuracy of traffic measurement results partly, they do not use the memory resources efficiently. In the future, the high-speed network bandwidth and the various new network applications will need more and more memory resources to store sampled flows. So a good sampling algorithm in the future network must try to use the memory resources enough as well as ensure a high sampling accuracy.

Recently SDM(software defined measurement) is proposed and discussed [7][8]. In SDM, the tradeoff between resource usage and accuracy is one of the important problems to be solved. Kinds of resources (CPU, memory, network) are be orchestrated by a controller. The memory usage becomes a function of measurement requirement in different spatial and time granularity.

This paper proposes a novel adaptive fair packet sampling algorithm (AFPS). The key innovation of AFPS is the reconfigurable counter structure, which structure is composed of two counter arrays counting packets in a differential way. One counter array named C_m is used to count packets of small flows one by one. The other named C_e is used to count large flows by counting sketch [4]. The two counter arrays share one memory space and the size of each counter array is decided by a reconfigurable parameter TEF (the Threshold to judge Elephant Flow). TEF can be adjusted adaptively according to the dynamic change of the maximal flow size on the link so that the memory resources can be used efficiently under different network conditions. Counting small and large flows in a differential ways ensures that the estimation error of small flows is 0 and the estimation error of large ones' equals to SGS. The dynamic adjustment of TEF makes AFPS estimate more small flow while the maximal flow size becomes small so that AFPS can get small average standard error.

2 Analysis of the Problem

Packet sampling can reduce the number of flow records stored in memory resources. But with the rapid development of network bandwidth and the appearance of some new network applications, existing sampling algorithms can not ensure high sampling accuracy within the constrain of the memory resource.

Especially to the network application such as anomaly detection, collecting as much as traffic information is very important. The information loss on small flows will affect the estimation accuracy of such network application's statistics. Fair sampling is a kind of methodology to settle the above problem. SGS[4] is a classic fair sampling algorithm. It makes the packet sampling probability as a decreasing function of the size of the flow which the packet belongs to. In this way, the packet belongs to the small flows can be sampled with high probability while low sampling probability of the elephant flows will not decrease the estimation accuracy, resulting in much more accurate statistic results.

SGS uses the counting sketch to encode the approximate of all flows. The hash collision in counting sketch might cause more than one flows to be hashed to the same index, resulting in the increasing of estimation error. This inaccuracy has more impactions on small flows than elephant ones.

The existence of hash collision in SGS is due to the limitation of memory resources. Today's memory technique does not support allocate a single counter to each flow. Though the memory resources used as the counters are very limited, SGS wasted lots of memory spaces because of its counting sketch structure. So we try to change the counting structure and improved the use ratio of the limited memory resource.

3 Our Algorithm

To solve the problem that the counting sketch structure used by SGS results in the decreasing of estimation accuracy for flows, the accuracy of the approximation for small flows used to calculate the sampling probability must be improved. This paper proposed an adaptive fair packet sampling algorithm (AFPS) to increase the estimation accuracy of fair sampling algorithm. The key innovation is the reconfigurable counter structure composed of two counter arrays, in which the size of counter arrays can be adjusted according to the changes of dynamic traffic characters on the network. Introducing of this novel counter structure can not only eliminate the hash collision to small flows but also improve the memory use ratio. AFPS can provide better overall accuracy than SGS.

3.1 The Architecture of AFPS

The adaptive fair packet sampling algorithm (AFPS) is mainly composed of three modular: flow counting, packet sampling and adaptive adjusting. The overall architecture of AFPS is shown in figure 1. Once each packet arrives, the AFPS scheme firstly tries to count the size of the flow which the currant packet belongs to by the reconfigurable counter structure. AFPS use the value in counters directly as the unbiased estimation of flow size which is the parameter to calculate the sampling probability. Secondly, the packet sampling modular samples the packet with the probability which is calculated by the decreasing sampling function f of the flow size the packet belongs to. If the packet is sampled, the flow record which the packet belongs to will be update. Finally, at the end of each sampling cycle (a predefined period of time), AFPS adjusts the size of two counter arrays according to the estimation of the maximal flow size during the sampling cycle. The overall architecture is similar to SGS besides the additional adaptive adjusting modular.

The sampling function used by AFPS is :

$$P(i) = 1/(1+\varepsilon^2 i) \tag{1}$$

Where i is the value of flow size. AFPS simply uses the counter value in the reconfigurable counter structure as the approximation of the concurrent flow size so that AFPS can support full line speed processing. The small flow counter array eliminates the hash collision by counting packet one by one. So AFPS can estimate the small flows accurately. The adaptive adjustment of the counter arrays ensure a high efficiency usage of memory spaces resulting in the increasing of overall estimation accuracy.

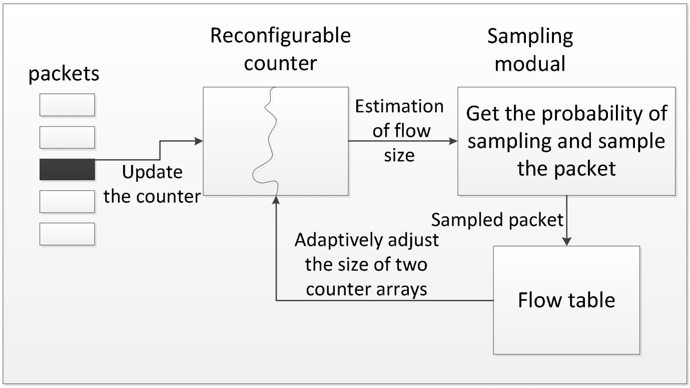


Fig. 1. The overall architecture of AFPS

3.2 The Sturcture of the Reconfigurable Counters

The most important part and the innovation of AFPS is the structure of the reconfigurable counter (RC) which can not only counts packet of small flows one by one but also ensure the high efficiency usage of memory spaces. The structure of RC is shown in figure 2.

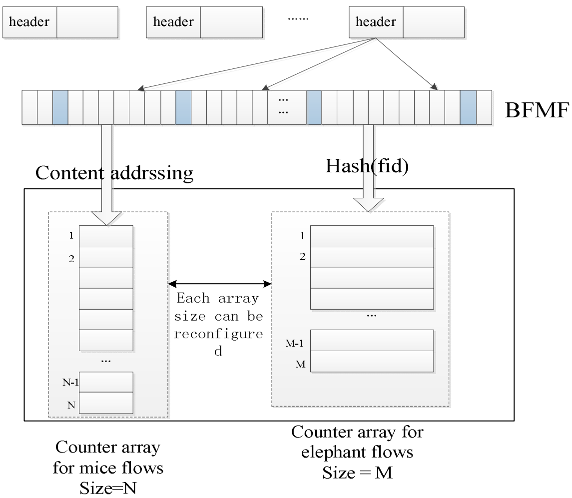


Fig. 2. The structure of RC

AFPS is composed of a Bloom Filter for small flows(BFMF), counter array for small flows C_m, counter array for elephant flows C_e and a reconfigurable parameter: the Threshold to judge Elephant Flow(TEF). The bit length of counters in C_m is smaller than in C_e while the number of counter in C_m is more than in C_e. C_m and C_e share the same memory space. The structure of RC is shown in figure 2.

Upon the arrival of a packet, one counter in RC will be updated. The updating process of RC is described in table1.

Table 1. The updating process of RC

The updating process of RC:	
1.	Initialize the BFMF, C_m, C_e and TEF;
2.	Abstract the flow identification f_{id};
3.	Search the BFMF by f_{id} ,judge the bit $\phi(h_i(f_{id})), i = 1,...,k$;
4.	If $\forall i,\ i = 1,....,k$, $\phi(h_i(f_{id})) = 1$,then:
5.	The packet belongs to a small flow, and then get the counter address of the flow $Addr_m$ in C_m according to the f_{id} by content addressing;
6.	If $C_m[Addr_m] < TEF$,then $C_m[Addr_m] = C_m[Addr_m] + 1$;
7.	Else a new elephant flow appears, then;
8.	Get the counter address of the flow $Addr_e$ in C_e according to the f_{id} by $HASH(f_{id})$, update the counter in C_e , $C_e[HASH(f_{id})] \leftarrow C_e[HASH(f_{id})] + C_m[Addr_m] + 1$, set the counter in $C_m[Addr_m] = 0$;
9.	Remove the current flow record from TEF, Set the bit in TEF to 0;

Based on the above section, we know that the memory resource of AFPS is mainly used by the RC structure. Since the concurrent flow number on the link of back bone is about 0.5 millons or 1 millons[6] , the size of each counter arrays in RC is no more than 10^6bit. So the C_m and C_e can both be implemented on SRAM. Besides the maximal time spent to sample a packet equals $(2k + 4) * T$, where k is the number of hash functions in BFMF, T is a memory access time. On the 10Gbps links (OC-192), AFPS can support the line-speed processing.

4 Evaluation and Discussion

4.1 Theoretical Analysis

AFPS aims to improve the accuracy of the estimation of small flows. Since AFPS allocates one counter to each small flow, the hash collision is avoided completely. AFPS can ensure the absolutely accurate flow size estimation for small flows. On the other hand, AFPS use the same counting sketch structure as SGS to count the packet of elephant flows. Thus the estimation error of large flows in AFPS is equal to SGS. Here we define average standard error as the mean of standard error of all flows in one sampling cycle. In table 2, we show the average standard error of AFPS and SGS in different TEF values. In our analysis, we suppose $\varepsilon = 0.1$.

Table 2. Average standard error of the two algorithms under different parameters

	TEF=100	TEF =600	TEF=1000	TEF=1500
AFPS	0.0856	0.0829	0.0734	0.0723
SGS	0.0958	0.0958	0.0958	0.0958

As can be seen from Table 2, the average standard error of AFPS is smaller than SGS. With the increasing of TEF, the average standard error of AFPS becomes smaller and smaller but the average standard error of SGS is a constant. So the sampling accuracy of AFPS is better than SGS.

The use ratio of the memory resources is another important index to measure the performance of sampling algorithms. As discussed in the above sections, the AFPS can keep a high use ratio of memory resources by adjusting the TEF. Here we analyze the use ratio of AFPS theoretically and compare the results to the assumption that the TEF cannot be changed. Figure 3 is the evaluation result where M is the maximal size of flow in the sampling cycles.

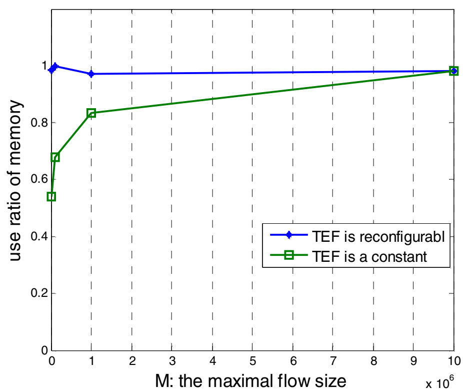


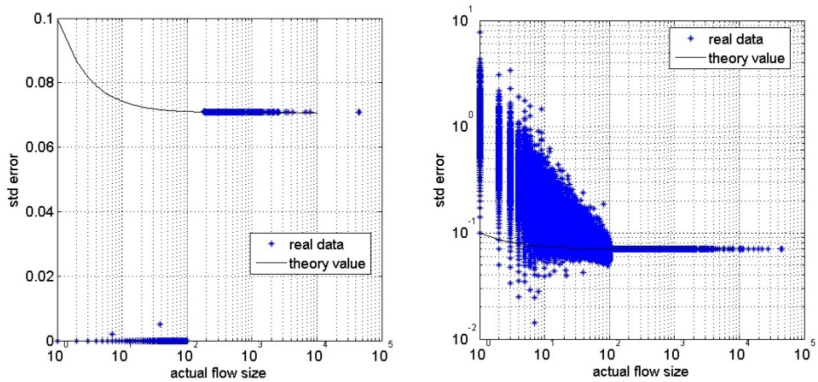
Fig. 3. The use ratio of memory in different schemes

4.2 Evaluation on Real Traffic Trace

The dataset used by the evaluation is from NLARN PMA's 2011[5], which is anonymized to protect the network users' privacy. The dataset file name is COS-1075142054-1.tsh.gz, and the detail information is shown in table 3.

Table 3. Detail information of the dataset

dataset	time	Flow numbers	Packet numbers	Speed of link	File name
NLARN PMA DATA-set	90s	162785	2268944	2.5Gbps	COS-1075142054-1.tsh.gz



(a) the std. error of ASPF (b) the std. error of SGS

Fig. 4. The standard error of ASPF and SGS

As can be seen from Fig. 4, the standard error of small flow size estimated by ASPF is 0 and the one of large flows is very close to the theory value. The flow size estimation of ASPF is more accurate than that of SGS. ASPF is very useful for the network applications which need the traffic information of small flows. The proposed algorithm is an effective way to improve the sampling accuracy with the memory resources constrains.

5 Conclusions and Future Work

In the proposed algorithm, the problem of confliction between memory resources constrain and sampling accuracy is resolved by a novel adaptive fair sampling method. ASPF introduces a reconfigurable counter structure to estimate the small flows and large flows in different way. The reconfiguration of the counter arrays ensures the use ratio of memory almost close to 1 under different network conditions. High use ratio of memory and different counting method for small flows and large flows result in the increasing accuracy of flow size estimation. AFPS is an absolutely effective fair packet sampling algorithm which can estimate small flows with 0 errors as well as do not increase the estimation error of large flows.

In the future, we will do deeper research on the direction of network measurement resource usage. We will study the influence on the measurement accuracy caused by other resource such as CPU or communication bandwidth. Then we want to explore a novel network measurement architecture which can allocate measurement resources between different measurement tasks so that the accuracy can be ensured and the resources can be utilized efficiently.

Acknowledgements. The research work was supported in part by a grant from the National Key Technology Support Program (No. 2012BAH02B01), the National Basic Research Program of China (973 Program)(No. 2012CB315902), the National Natural Science Foundation of China (No.61102074,61170215,61379120), Zhejiang Leading Team of Science and Technology Innovation(No. 2011R50010-03, 2011R50010-12, 2011R50010-19).

References

1. Fang, W., Peterson, L.: Inter-as: Traffic Patterns and their Implications. In: Proceedings of IEEE GLOBECOM (1999)
2. Estan, C., Varghese, G.: New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice. ACM Transactions on Computer Systems 21(63), 270–313 (2003)
3. Duffield, N.G., Lund, C., Thorup, M.: Flow sampling under hard resource constraints. In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, pp. 85–96. ACM Press, New York (2004)
4. Kumar, A., Xu, J.: Sketch guided sampling—Using on-line estimates of flow size for adaptive data collection. In: Proceedings of IEEE Infocom 2006, Washington, pp. 1–11 (2006)
5. NLANR. PMA DATA. 2011[OL] (May 2013) `http://labs.ripe.net/datarepository/data-sets/nlanr-pma-data`
6. Tammaro, D., Valenti, S., Rossi, D., Pescap, A.: Exploiting packet-Sampling measurements for traffic characterization and classification. Int. J. Netw. Manag. **22**(6), 451–476 (2012)
7. Masoud, M., Minlan, Y., Ramesh, G.: Resource/Accuracy tradeoffs in Software-defined measurement. In: HotSDN 2013, Hong Kong, China (August 16, 2013)
8. Minlan, Y., Jose, L., Rui, M.: Software Defined Traffic Measurement with OpenSketch. In: NSDI (2013)