Easily-Implemented Adaptive Packet Sampling for High Speed Networks Flow Measurement*

Hongbo Wang, Yu Lin, Yuehui Jin, and Shiduan Cheng

State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, 100876, Beijing, China
{hbwang, linyu, yhjin, chsd}@bupt.edu.cn

Abstract. An Easily-implemented Adaptive Packet Sampling (EAPS) is presented in this paper, which overcomes the shortcoming of NetFlow and Adaptive Netflow. EAPS is easy to be hardware-implemented and scalable for high-speed networks. Additionally, EAPS is automatically adaptive to traffic rate under resource constrains, thus it is convenient to be used by network operators. Experiments are conducted with the real network traces. Results show that EAPS is more accurate than ANF over links with light load or traffic fluctuations.

1 Introduction

Traffic measurement is the basis of IP network monitoring, management and controlling tasks. Particularly, flow-level measurements are widely used for various applications such as traffic profiling, dominant applications or users tracking, and traffic engineering. With the ever increasing speeds of transmission links and volume of network traffic, flow-level measurements face the formidable challenges of the scalability issues. On today's high-speed links, monitoring every packet and recording statistics of every flow consume too much resource at the routers or other network elements. Packet sampling has been suggested[1-5] to address this problem, which is a scalable alternative for network measurement.

Cisco's NetFlow[6] which adopts static packet sampling method are widely deployed by most major ISPs and becomes the most popular flow measurement solution. Though the wide deployment of NetFlow is a proof of its ability to satisfy the important needs of network operators, it has several shortcomings[5]: during flooding attacks, resource consumed by flow records may increase beyond what is available; selecting the right static sampling rate is difficult. Estan et al. proposed Adaptive NetFlow(ANF)[5] to address the problems of NetFlow. When the traffic volume increases, ANF can dynamically decrease the sampling rate until it is low enough for the flow records to fit into flow cache memory. However, the renormalization algorithm adopted by ANF is complex and heuristic, which hinder it from the implementation by hardware. Therefore, ANF itself may become the processing bottleneck when the link speed exceeds OC-48(2.5Gpbs). Furthermore, the maximal sampling rate of ANF

* This work was supported by the NSFC (No. 90204003 and 60502037), CNGI (No. CNGI-04-8-1D) and the 973 project of china (No. 2003CB314806).

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part IV, LNCS 3994, pp. 128 – 135, 2006.

(about 1/35) is computed under worst case conditions, but it is suboptimal and less accurate in the light loaded conditions. Additionally, once the sampling rate used by ANF is reduced, it can not increase again in one measurement bin. Consider the cases that the traffic rate fluctuates in the measurement bin: the traffic load is high in the fore-half bin, so ANF adopts small sampling rate; but as the traffic load decrease, ANF can not increase its sampling rate again. Thus ANF becomes less accurate when the traffic load decreases during the measurement bins.

In order to overcome aforementioned shortcomings of ANF, an Easily-implemented Adaptive Packet Sampling EAPS is proposed in this paper for flow measurement. With measurement buffer, EAPS samples a fixed number of packets in every measurement interval by adopting a reservoir sampling method. Since the number of sampled packets is constant in every measurement interval, the sampling rate of EAPS automatically decreases with the increasing of traffic rate. On the contrary, the sampling rate will automatically increase when the traffic rate decreases. Furthermore, the flow cache memory and bandwidth consumption is also constrained by the size of the measurement buffer [1]. Therefore, EAPS is robust during flooding attacks. We also show the upper-limit of the relative standard deviation of EAPS estimation through theoretical analyses. With the experiments of real network traces, the result demonstrates that EAPS is more accurate than ANF over links with light load or traffic fluctuations.

The rest of this paper is organized as follows. Section 2 describes the methodology of EAPS. Section 3 proposes the estimation error. In Section 4, EAPS is compared to ANF using real flow traces. Finally, the paper is concluded in Section 5.

2 Methodology of EAPS

In this section, we will discuss the sampling and estimation methodology of EAPS. NetFlow uses four rules to decide when a flow has ended which then allows the corresponding record to be exported: 1) when indicated by TCP flags (FIN or RST), 2) 15 seconds(configurable) after seeing the last packet with a matching flow ID, 3) 30 minutes (configurable) after the record was created (to avoid staleness) and 4) when the flow cache is full. As shown in [5], most traffic analysis tools divide the traffic stream into fixed analysis bins. Unfortunately, NetFlow records can span bins, causing unnecessary complexity and inaccuracy for traffic analysis. Just as ANF, EAPS divide the NetFlow operation into short bins so that the bins used by traffic analysis are exact multiples of the measurement bins. Differing from ANF, EAPS retains Netflow's four rules during the measurement bins, but terminates all active flow records at the end of each measurement bin. In our experiments we used the more challenging one minute size for the measurement bin.

2.1 Random Sampling Algorithm with a Reservoir

Since the size of measurement buffer n is limited, EAPS can not record all the packets arrived within one measurement bin and then do sampling process. It must do sample at the same time one packet arrives. However, the challenging problem is how to select

[1] Our resource consumption controlling method is like to that of [7]. But our sampling is packet sampling at measurement point in stead of flow sampling at the collector of flow records[7].

a random sample of *n* packets from *N* successively arriving packets, where the value of *N* is unknown beforehand. This can be solved by reservoir random sampling algorithm in literature [8]. The naive algorithm work as follows: the first *n* packets arrived are stored in the reservoir and became the sample candidates; when the t^{st} ($t>n$) packet is arriving, it has a *n/t* probability of being a sample candidate, if it became a candidate, it will randomly replace one candidate in the reservoir. It is easy to see that the resulting set of *n* candidates in the reservoir forms a random sample of the first *t* packets. The computing complexity of this algorithm is *O(N)*, where *N* is the total number of packets arrived in one bin. In [8], Vitter proposed algorithm Z with the much less complexity $O(n(1+\log(N/n)))$. EAPS will adopt algorithm Z which is easy to be implemented by hardware with higher efficiency.

2.2 Design of Measurement Buffer

In order to further reduce the size of measurement buffer, we divide the measurement bin into *m* fixed intervals, and do reservoir sampling in each interval. In our scheme, the measurement buffer is divided into two reservoirs: reservoir A and B. As shown in figure 1, in the *i*th interval, when flow measurement software reads sampled packets from reservoir A to update flow entries, the packet arriving from network is buffered in another reservoir B. And in the (*i+1*)th interval, the function of reservoir A and B will be swapped ,and so on. The sampling will be done by the hardware implemented Algorithm Z.

In fact, only parts of packet fields (about 21 bytes) are needed to store in the measurement buffer for flow measurement, including TOS, packet length, protocol, source IP address, destination IP, source port, destination port, TCP flags, timestamp, and etc. For a reservoir containing 12000 packets information, a 3.85Mbit SRAM is needed, which is easily implemented by today's semiconductor technology.

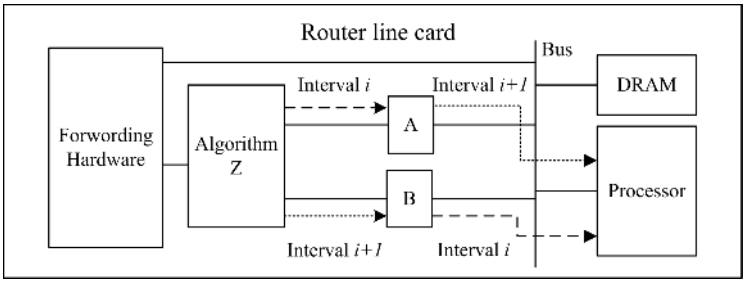

Fig. 1. Design of measurement buffer in router line card

2.3 Estimation Methodology

Denote *m* intervals in one measurement bin by $t_1, t_2, ..., t_m$. In the interval $t_i (i = 1,2,...,m)$, denote a flow appeared by $f_k (k = 1,2,...)$ and the total number of packets arrived of all flows by N_i. Since the number of sampled packets in the reservoir is constant *n*, the sampling rate in this interval is n/N_i. For a flow f_k in interval t_i, let N_i^k be the num-

ber of packets belonging to it; and denote the *j*th packet of the flow by $p_{ij}^k (j=1,2,...,N_i^k)$ and its packet size by x_{ij}^k. Similarly, denote the number of sampled packets belonging to flow f_k by n_i^k, and the *j*th packet in the sampled packets by $q_{ij}^k (j=1,2,...,n_i^k)$, its packet size by X_{ij}^k.

The total bytes of flow f_k in the interval t_i, i.e., $x_i^k = \sum_{j=1}^{N_i^k} x_{ij}^k$, is estimated as

$$\hat{x}_i^k = \frac{N_i}{n} \sum_{j=1}^{n_i^k} X_{ij}^k \tag{1}$$

The total packet number of flow f_k in the interval t_i, i.e. N_i^k is estimated as

$$\hat{N}_i^k = \frac{N_i}{n} n_i^k \tag{2}$$

The total bytes of flow f_k in the whole measurement bin, i.e., $x^k = \sum_{i=1}^{m} x_i^k$, is estimated as

$$\hat{x}^k = \sum_{i=1}^{m} \hat{x}_i^k \tag{3}$$

The total packet number of flow f_k in the whole measurement bin, i.e., $N^k = \sum_{i=1}^{m} N_i^k$, is estimated as

$$\hat{N}^k = \sum_{i=1}^{m} \hat{N}_i^k \tag{4}$$

3 Estimation Error

In this section, we firstly prove that $\hat{x}^k$ and $\hat{N}^k$ is the unbiased estimation of x_i^k and N^k, then give upper-limit of their relative errors. Consider a flow f_k during the measurement interval t_i. Let $\mu_i = \frac{1}{N_i^k} \sum_{j=1}^{N_i^k} x_{ij}^k$, $\sigma_i^2 = \frac{1}{N_i^k} \sum_{j=1}^{N_i^k} (x_{ij}^k - \mu_i)^2$. The average size of packets during the measurement bin is denoted as $\mu^k = \frac{x^k}{N^k}$.

Lemma 1. Let n_i^k be the number of sampled packets, belonging to flow f_k which have N_i^k packets during interval t_i. Then n_i^k is a binomial random variable, and its mean and variance are $E(n_i^k) = N_i^k \frac{n}{N_i}$, $Var(n_i^k) = \frac{n}{N_i}(1 - \frac{n}{N_i}) N_i^k$ respectively.

Proof. Since the buffer size is *n* and the total packet number in interval t_i is N_i, the probability that any packet is sampled is n/N_i. Thus, n_i^k is a binomial random variable with parameters N_i^k and $\frac{n}{N_i}$, so $E(n_i^k) = N_i^k \frac{n}{N_i}$, $Var(n_i^k) = \frac{n}{N_i}(1 - \frac{n}{N_i}) N_i^k$.

Theorem 1. Consider a flow f_k, $\hat{N}^k$ is an unbiased estimation of N^k.

Proof. From equation (2) and Lemma 1, $E(\hat{N}_i^k) = E(\frac{N_i}{n} n_i^k) = \frac{N_i}{n} E(n_i^k) = N_i^k$. Also from equation (4), we have $E(\hat{N}^k) = E(\sum_{i=1}^{m} \hat{N}_i^k) = \sum_{i=1}^{m} E(\hat{N}_i^k) = \sum_{i=1}^{m} N_i^k = N^k$ which proves the theorem.

Theorem 2. Consider a flow f_k, $\hat{x}^k$ is an unbiased estimation of x^k.

Proof. Suppose n_i^k packets belonging a flow f_k are sampled during interval t_i. From the attributes of simple random sampling[9], the sizes of each sampled packets, i.e. $X_{ij}^k (j = 1,2,...,n_i^k)$, is random variable, and $E(X_{ij}^k) = \mu_i$. From equation (1),

$$E(\hat{x}_i^k \mid n_i^k) = E(\frac{N_i}{n} \sum_{j=1}^{n_i^k} X_{ij}^k) = \frac{N_i}{n} n_i^k \mu_i \tag{5}$$

Then $E(\hat{x}_i^k) = E(E(\hat{x}_i^k \mid n_i^k)) = E(\frac{N_i}{n} n_i^k \mu_i) = \frac{N_i}{n} E(n_i^k) \mu_i = N_i^k \mu_i = x_i^k$. From equation (3), we have $E(\hat{x}^k) = E(\sum_{i=1}^{m} \hat{x}_i^k) = \sum_{i=1}^{m} E(\hat{x}_i^k) = \sum_{i=1}^{m} x_i^k = x^k$ which proves the theorem.

Theorem 3. For each flow f_k, the standard deviation of $\hat{N}^k$ is up-bounded by $1 \Big/ \sqrt{n \frac{N^k}{N}}$, where N is the total packet number in one measurement bin.

Proof. From equation (2) and Lemma 1, the variance of $\hat{N}^k$ is $Var(\hat{N}_i^k) = Var(\frac{N_i}{n} n_i^k) = \frac{N_i^2}{n^2} Var(n_i^k) = \frac{N_i^2}{n^2} (\frac{n}{N_i}(1 - \frac{n}{N_i}) N_i^k) = \frac{N_i}{n}(1 - \frac{n}{N_i}) N_i^k < \frac{N_i}{n} N_i^k$

Since the sampling processes of different measurement intervals are independent with each other, $\hat{N}_i^k (i = 1,2...,m)$ are independent random variances. From equation (4), $Var(\hat{N}^k) = Var(\sum_{i=1}^{m} \hat{N}_i^k) = \sum_{i=1}^{m} Var(\hat{N}_i^k) < \sum_{i=1}^{m} \frac{N_i}{n} N_i^k = \frac{1}{n} \sum_{i=1}^{m} N_i N_i^k$. Note that $N_i > 0$ and $N_i^k \geq 0$, then $\sum_{i=1}^{m} N_i N_i^k < \sum_{i=1}^{m} N_i \sum_{i=1}^{m} N_i^k = N N^k$, we thus have $\frac{\sqrt{Var(\hat{N}^k)}}{N^k} < 1 \Big/ \sqrt{n \frac{N^k}{N}}$ which proves the theorem.

Theorem 4. For each flow f_k, the standard deviation of $\hat{x}^k$ is up-bounded by $1 \Big/ \sqrt{n \frac{N^k}{N} \frac{\mu^k}{b_{\max}}}$, where $b_{\max}$ is the maximal size of IP packet, N is the total packet number in an measurement bin.

Proof. From formula of variance of the sample mean in simple random sampling[9],

$$Var(\hat{x}_i^k \mid n_i^k) = Var(\frac{N_i}{n}\sum_{j=1}^{n_i^k} X_{ij}^k) = \left(\frac{N_i}{n}n_i^k\right)^2 Var(\frac{1}{n_i^k}\sum_{j=1}^{n_i^k} X_{ij}^k) = \left(\frac{N_i}{n}n_i^k\right)^2 \frac{\sigma_i^2}{n_i^k}\left(1-\frac{n_i^k-1}{N_i^k-1}\right) < \frac{N_i^2}{n^2}n_i^k\sigma_i^2$$

From equation (5) and Lemma 1, we have

$$\begin{aligned} Var(\hat{x}_i^k) &= E(Var(\hat{x}_i^k \mid n_i^k)) + Var(E(\hat{x}_i^k \mid n_i^k)) < E(\frac{N_i^2}{n^2}n_i^k\sigma_i^2) + Var(\frac{N_i}{n}n_i^k\mu_i) \\ &= \frac{N_i^2}{n^2}E(n_i^k)\sigma_i^2 + \frac{N_i^2}{n^2}Var(n_i^k)\mu_i^2 = \frac{N_i}{n}N_i^k\sigma_i^2 + \frac{N_i}{n}N_i^k(1-\frac{n}{N_i})\mu_i^2 \\ &< \frac{N_i}{n}N_i^k(\sigma_i^2+\mu_i^2) \end{aligned}$$

Since the sampling processes of different measurement intervals are independent with each other, $\hat{x}_i^k (i = 1,2...,m)$ are independent random variances,

$Var(\hat{x}^k) = Var(\sum_{i=1}^{m}\hat{x}_i^k) = \sum_{i=1}^{m}Var(\hat{x}_i^k) < \frac{b_{\max}}{n}\sum_{i=1}^{m}(N_i\sum_{j=1}^{N_i^k}x_{ij}^k) < (\sum_{i=1}^{m}N_i)(\sum_{i=1}^{m}\sum_{j=1}^{N_i^k}x_{ij}^k) = Nx^k$. Therefore $\frac{\sqrt{Var(\hat{x}^k)}}{x^k} < \sqrt{\frac{b_{\max}N}{nx^k}} = \sqrt{\frac{b_{\max}N}{n\mu^k N^k}} = 1\Big/\sqrt{n\frac{N^k}{N}\frac{\mu^k}{b_{\max}}}$ which proves the theorem.

4 Experiment Results and Analysis

In our experiments we use real network traces from OC-48 links collected by CAIDA[10] project[2], and ten one-minute datasets were adopted. We run EAPS over these datasets using different parameters settings, including buffer size of 6k, 12k, 18k and 24k, and measurement interval of 6, 12 and 20 seconds. For same parameters settings, we run EAPS for 27 times.

4.1 Experiment Results of EAPS

Firstly, the relative estimated error of packet number is computed and shown in figure 2.a. We can see that the points are symmetrically distributed around coordinate y. Thus, the estimation is unbiased in accordance with theorem 1.

In practice, it is convenient for network analysis to aggregate individual flows into aggregate flow. In this paper, we define individual flows with same port number as an aggregate flow. As shown in figure 2.b, the upper line is the upper-limit of relative error given by theorem 3, and the spanned points below the line are the results of different experiments. This conforms to theorem 3.

For space limit, we omitted the corresponding traffic bytes results which are similar to figure 2.a. or figure 2.b.

---

[2] We thank CAIDA for providing us the real network traces data.

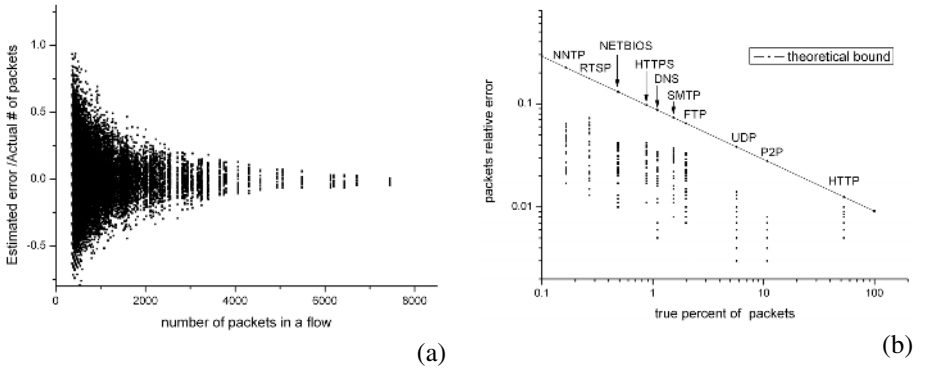

Fig. 2. Experiment Results of EAPS

4.2 Comparison with ANF

We then compare EAPS with ANF proposed in [5]. For same parameter setting, we run EAPS and ANF respectively.

For space limit, we only show the bytes comparison results of EAPS and ANF based on two typical Datasets. In Dataset1, the traffic load is relatively light. In DataSet2, the traffic load is relatively high, and the traffic rate decreases (fluctuates) during the measurement bin. From table 2, we can see that, for both DataSet1 and DataSet2, EAPS is more accurate than ANF. It can be explained as follows: in the case of DataSet1(on 6s measurement interval), the traffic load is relatively light; and the automatically-adaptive sampling rate of EAPS is about 1/22, while the maximal sampling rate of ANF is 1/35; As to DataSet2(on 12s measurement interval), ANF choose a relatively small sampling rate (about 1/200) when the traffic load is high in the fore-half bin, and can not increase its sampling rate even when the traffic load decreases in the post-half bin; EAPS is automatically adaptive to the changes of traffic rate, it takes a sampling rate about 1/55. Thus EAPS is more accurate than ANF over links with light load or traffic fluctuations.

Table 1. Relative Error Comparison of EAPS with ANF

Aggregate flows		HTTP	P2P	FTP	SMTP	RTSP	HTTPS	DNS
Percent(%)		54.61	12.83	1.74	0.72	0.41	0.33	0.19
Dataset1 (6s)	EAPS	0.003	0.009	0.020	0.036	0.054	0.043	0.026
	ANF	0.007	0.016	0.041	0.070	0.068	0.074	0.048
Dataset2 (12s)	EAPS	0.005	0.014	0.031	0.042	0.073	0.059	0.046
	ANF	0.009	0.022	0.039	0.081	0.077	0.081	0.050

5 Conclusion

Adaptive NetFlow(ANF) has been proposed to overcome the shortcoming of Cisco's NetFlow by dynamically adjusting the sampling rate. However, the renormalization

algorithm adopted by ANF is difficult to be implemented by hardware, thus it is not scalable for higher speed links. In this paper, an Easily-implemented Adaptive Packet Sampling (EAPS) is presented. Compared with ANF, EAPS is easier to be hardware-implemented and used, as well as automatically adaptive to traffic rate with certain resource consumption. With the real network traces, the experiments demonstrate that EAPS is more accurate than ANF over links with light load or traffic fluctuations.

References

1. Claffy, K.C., Polyzos, G.C., and Braun, H.-W.: Application of sampling methodologies to network traffic characterization. In Proc. ACM SIGCOMM, (1993) 13-17
2. Hernandez, E.A., Chidester, M.C., George A.D.: Adaptive Sampling for Network Management. Technical Report, University of Florida, (2000)
3. Cheng Guang, Gong Jian, Ding Wei: Network traffic sampling measurement model on packet identification. Acta Electronica Sinica, December, (2002) 1986-1990
4. Choi, B.-Y., Park, J., and Zhang, Z.-L.: Adaptive Random Sampling for Traffic Load Measurement. IEEE International Conference on Communications (ICC '03), May 2003.
5. Estan, C., Keys, K., Moore, D., and Varghese, G.: Building a better netflow. In Proceedings of the ACM SIGCOMM (2004)
6. Cisco netflow. http://www.cisco.com/warp/public/732/Tech/netflow.
7. Duffield, N.G., Lund, C. and Thorup, M.: Flow sampling under hard resource constraints. In Proc. ACM SIGMETRICS 2004 85–96. ACM Press, New York(2004).
8. Vitter, J.S.: Random sampling with a reservoir. ACM Trans. Math. Software. (1985)1137–57.
9. Rice, J.A.: Mathematical Statistics and Data Analysis. Second Edition. Duxbury Press. USA(1995)
10. Cooperative association for internet data analysis.http://www.caida.org