A Resource-Efficient Flow Monitoring System

Guang Cheng, *Member, IEEE,* and Jian Gong, *Member, IEEE*

***Abstract*— Network flow monitoring provides critical information for many network applications, and several approaches have been proposed to address flow monitoring. However they either lack flexibility in adapting to ever-changing network traffic (e.g. NetFlow), or require intensive computing resources (e.g. ANF). In this paper, we propose a Resource-Efficient Flow Monitoring System (called REFMS), which consists of Pre-Sampling module, Sample & Hold module, and Flow Removal module. Our REFMS can improve sampling accuracy and effectively utilize measurement resource with multiple sampling algorithms.**

***Index Terms*— Sampling methods, Internet traffic, measurement, adaptive systems.**

I. Introduction

NETWORK flow monitoring and analysis is crucial for many network applications, such as network planning, network management, and network security applications [1]. Network packets passing through the monitoring system can be classified into flows, which can be further applied for many purposes, such as detecting DoS attack, Worm spreading [2], where suddenly a lot of small flows are generated. NetFlow [3], implemented in Cisco routers, can generate and output flow records, and keep them in a flow cache to describe the passing traffic. To avoid problems caused by router CPU exhaustion, Cisco provides a Sampled NetFlow. Rather than looking at every packet to maintain NetFlow records, the router looks at every nth packet. Estan has proposed an Adaptive NetFlow (ANF) [4], which can be operated within limited resource, and use renormalization to reduce the number of NetFlow records using a new sampling ratio. These flow monitoring approaches either lack flexibility in adapting to ever-changing network traffic (e.g. sNetFlow), or require intensive computing resources (e.g. ANF). Obviously, there is a trade-off between monitoring accuracy and limited system resources (e.g. memory size, CPU speed). To improve accuracy with a given system resource is a significant challenge. In this paper, we will try to tackle this issue.

Two flow-measuring solutions, namely Sample & Hold and Non-Uniform Sampling which can improve estimation accuracy and save measurement resource are used in this paper. Estan [5] proposed the Sample & Hold algorithm for identifying the large flows, and Ashwin Lall [6] also uses a similar Sample & Hold algorithm. Raspall [7] presents a Shared-State Sampling algorithm to detect a large flow in the high-speed networks, which is a generation of Sample & Hold algorithm. The Non-Uniform Sampling algorithm [8] can control estimation variance arising from the observed heavy-tailed distribution of flow lengths. All of these sampling techniques are valuable to the study of our algorithm.

In this paper, we propose a novel approach to tackle this issue by using a Resource-Efficient Flow Monitoring System (REFMS) for a better trade-off between monitoring accuracy and limited system resources. By caching estimated values rather than actual measured value, REFMS could adapt to the flow sampling in varying sampling ratio. Our main contributions are as follows. ***(1). Record the Estimated Value.*** REFMS records an estimated result of the measured value rather than the measured value itself, so it can use different sampling ratios in a measurement period. ***(2). Non-Uniform Sampling-Based Flow Removal Algorithm.*** The flow removal module manages the size of the recorded cache and keeps long flows into the recorded cache which can record more packets, so it can promise more accurate and efficient flow monitoring than ANF. ***(3). Multiple Sampling Modules.*** REFMS uses three kinds of sampling modules to control measured resources. Pre-Sampling module controls the consumption of CPU, and a Sample & Hold module and a Non-Uniform Sampling-Based Flow Removal module manage the flow cache.

The paper is organized as follows: Section 2 elaborates the REFMS method with a detailed discussion. Comprehensive experiments are conducted and the results are discussed in Section 3. Section 4 is the conclusion of this work.

II. A Resource-Efficient Flow Monitoring System

A. REFMS Architecture

Sampling measurement must fulfill two goals: improving estimation precision and reducing consumption of the resources. It is very hard to configure the sampling ratio to adapt to the CPU and memory resource. REFMS sets Pre-Sampling module to better manage CPU resource, and Sample & Hold module and Non-Uniform Sampling-Based Flow Removal module mange the size of the recorded flow cache.

Figure 1 is the REFMS architecture, which consists of three modules: Pre-Sampling Module, Sample & Hold Module, and Flow Removal Module. In this architecture, packets are sampled by one out of n, and sampled packets are processed by the Sample & Hold module. If there is a flow entry in the flow cache, then the flow entry is updated, otherwise the packet is sampled again using one out of r to decide whether a new flow is created. If the number of flows X in the flow cache is over a threshold R, then a flow removal process based on a

Manuscript received February 6, 2007. The associate editor coordinating the review of this letter and approving it for publication was Prof. Enzo Baccarelli. This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant No. BK2006092, the Key Project of Chinese Ministry of Education under Grant No. 105084, the National Grand Fundamental Research 973 program of China under Grant No. 2003CB314804, and the Excellent Youth Teacher of Southeast University Program under Grant No. 4009001018.

The authors are with the College of Computer Science and Engineering, Southeast University, Nanjing, P.R. China, 210096 (email: gcheng@njnet.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2007.070198.

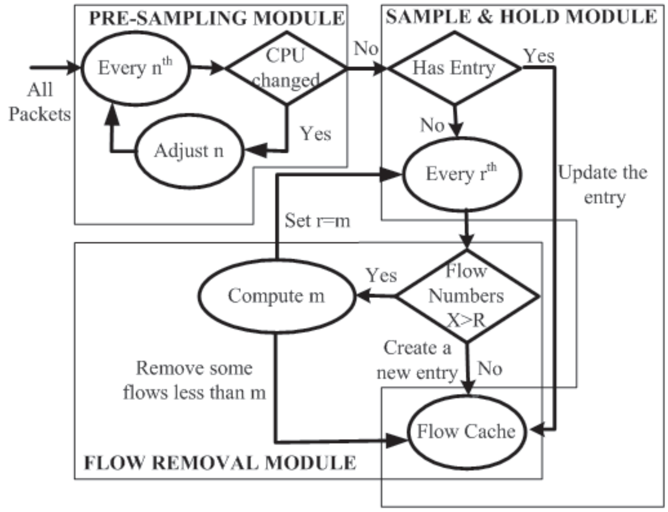

Fig. 1. the REFMS architecture.

non-uniform sampling algorithm will be started up to remove some flow entries from the flow cache.

B. Pre-Sampling Module

The Pre-Sampling module is a random process to sample packets. Let sampling ratio p be $1/n$, which is to sample 1 in n packets. Therefore to obtain an unbiased estimate $\hat{N}$ of the number of packets N using the sampled traffic, we should statistically compensate for the missed packets with probability $1 - p$. It is intuitive that if we add $n = 1/p$ to the estimator, the resulting estimator is unbiased. In a measurement period, if K packets is sampled with sampling ratio p_i for each sampled packets $\{pkt_i, p_i, i = 1, \ldots, K\}$, then the output of estimated packets is an unbiased estimator of sampling traffic $N = \sum_{i=1}^{K} 1/p_i$,where $1/p_i$ is an unbiased estimator of each sampled packet. Because $1/p_i$ is recorded directly, different sampling ratios can be used in a measurement period. The standard deviation of N, $SD(N)$, is in the equation 1.

$$SD(N) = \sqrt{\sum\nolimits_{i=1}^{K} 1/p_i^2} \tag{1}$$

C. Sample & Hold Module

The Sample & Hold module processes sampled packets to update the flow cache. After a flow is recorded, all its following packets are held, otherwise the packet will be sampled with a sampling ratio 1 in r, $p = 1/r$. If there are x packets which have been missed, before the first packet in the flow f is sampled, then these missed packets obey geometric probability distribution, and its probability distribution is $p(1 - p)^x$, so $E(x) = 1/p - 1$, $D(x) = (1 - p)/p^2$. When the first packet in flow f is collected, 1/p packets have passed, so we record $E(x) + 1 = 1/p$ as the initial value of flow f rather than 1, and its standard deviation $SD(x) = sqrt(1 - p)/p$. After the first packet in flow f is collected, all its following packets will be recorded in the flow entry without measurement errors. If its following packets are c, then an estimator of flow f is $c + 1/p$, so its standard deviation is in the equation 2.

$$SD(x) = \sqrt{(1 - p)/p} \tag{2}$$

D. Flow Removal Module

When the flow cache M is larger than a threshold R, some flow entries should be removed to evacuate some spaces to accept the new flows. Long-size flows can remember more packet information than small-size flows, so REFMS uses different sampling ratio to different size flows. Let a threshold be m, and size of a flow be k. If k is larger than m, then the flow is sampled by a probability 100%, and if k is less than m, then the sampling probability of the flow sampled is q=k/m. In this case, all flows larger than m will always be sampled 100% without any removal sampling error. On the other hand, sampling small flows with k/m probability can get some spaces from the recorded flow cache. Because these flows are small, so the error involved is small.

If a flow is sampled with sampling ratio $q = k/m < 1$, then this sampled flow is compensated and is recorded as $k/q = m$. All sampled small flows are m after they are compensated, and the new flow sampling ratio of the Sample & Hold module is set to $1/m$. Let an aggregated flow H have n flows and the size of a flow be $c + 1/p$, $p >= 1/m$, and its variance be $1/p^2$, then the removal sampling ratio is $q = (c + 1/p)/m$. Its unbiased estimator is $(c + 1/p)/q = m$, and its variance is the equation 3.

$$\frac{c}{q^2} + \frac{1}{q^2 p^2} = \frac{(cp^2 + 1)m^2}{(cp + 1)^2} \leq \frac{m^2}{cp + 1} < m^2 \tag{3}$$

If a flow is larger than threshold m during the removal process, then its sampling error only comes from the Sample & Hold module. Let the sampling ratio of Sample & Hold process be p, $p >= 1/m$, so its variance is $1/p2 \leq m^2$. We can draw the conclusion that the variance of all flows in the recorded flow cache is less than m^2. An unbiased estimator of the aggregated flow H is $\sum_{i=1}^{n} x_i$, and its variance is $n * m^2$. If the actual size of the aggregated flow H is x, then the relative error is $m\sqrt{n}/x$. In fact, we can only know a estimated value $\hat{x}$ of x, so the relative error can be approximate to $m\sqrt{n}/\hat{x}$.

Now we discuss a quick search algorithm to find the threshold m. Firstly, a dimension D with d items is set. If the flow cache is updated, the dimension D is also updated. Let the length of a flow s be x, and $x < d$. If a packet which belongs to s is sampled, then D can be updated $D[x] = D[x] - 1$, $D[x + 1] = D[x + 1] + 1$. If $x = d$, then $D[x] = D[x] - 1$. If a flow size is larger than d, then the flow won't be recorded in the dimension D. If a new flow is created in the flow cache, then $D[1] = D[1] + 1$. αM flow records should be removed from the recorded flow cache. The removal threshold m can be computed by the equation 4.

$$\sum_{i=1}^{m-1} D[i](1 - \frac{i}{m - 1}) < \alpha M \leq \sum_{1=1}^{m} D[i](1 - \frac{i}{m}) \tag{4}$$

E. Theoretical Error

Table I shows the theoretical error between ANF and REFMS. p_A is the last sampling ratio of Estan's ANF. p_pre is the smallest pre-sampling ratio of REFMS, and p_s is the smallest sampling ratio of Sample & Hold module. k is the flow number of a aggregated flow in the flow cache. t is the packet number of the aggregated flow. M is the size of the flow cache.

TABLE I

Theoretical Error of Aggregated Flows

Measure	Relative Error	M-Based Relative Error
ANF [4]	$\sqrt{1/p_A t}$	$\sqrt{1/(Mf)}$
REFMS	$\sqrt{k}/(p_{pre}p_s t)$	$\sqrt{k}/(Mf)$

TABLE II

Error of Aggregated Flows $\leq 0.1\%$ of Total Packets.

Percentile	5%	25%	50%	75%	95%
ANF-TE	0.0446	0.0945	0.1361	0.1747	0.2128
ANF-ME	0.0044	0.0221	0.0514	0.0992	0.1975
REFMS-TE	0.0034	0.0105	0.0119	0.0298	0.0612
REFMS-ME	0.0005	0.0025	0.0066	0.0141	0.0395

Proof. It is easy to get the variance of REFMS less than $k/(p_{pre}^2 p_s^2)$ according to the equation (1-3), so its relative standard deviation is $\sqrt{k}/(p_{pre}p_s t)$. Let T be the total number of packets sent during the measurement interval with a sampling ratio M/T, the expected number of packets is M, and the expected number of entries is at most M. Thus the sampling ratio, at which the expected number of entries is M, will be $p_s \leq M/(Tp_{pre})$, and its relative standard deviation is $\sqrt{k}/(p_{pre}p_s t) \leq \sqrt{k}/(p_{pre}tM/(Tp_{pre})) = \sqrt{k}/(Mf)$.

III. Experiment Analysis

We use a group of packet traces gathered at NLANR [9]. NLANR uses OC192MON hardware to collect data on August 19, 2004, from 13:40pm to 14:40pm. Estan's ANF algorithm has the same estimatation precision with the Sampled FetFlow algorithm, except that Sampled NetFlow can't change its sampling ratio. So in the experiments, we only compare Estan's ANF [4] with the REFMS.

In the equation 5, accuracy metrics, errori, evalues estimated error of the i_{th} aggregated flow, where X_i is the actual packet numbers of the i_{th} aggregated flow, and $\hat{X}_i$ is its estimated value.

$$error_i = (X_i - \hat{X}_i)/X_i \qquad (5)$$

A. Estimated Aggregated Flows

In the table II, ANF_TE is the theoretical error of ANF, and ANF-ME is the measurement error of ANF. REFMS-TE is the theoretical error of REFMS, and REFMS-ME is the measurement error of REFMS. The 5% percentile is a error such that at most 5% of the estimation error of all aggregated flows are less than this error and that at most (100-5)% are greater. 25% percentile, 50% percentile, 75% percentile, and 95% percentile all can be understood in the same way. All aggregated flows larger than 0.1% total packets are analyzed, and the size of flow cache M=8192. Table II shows that the REFMS is more accurate than ANF algorithm in terms of both theoretical error and measurement error.

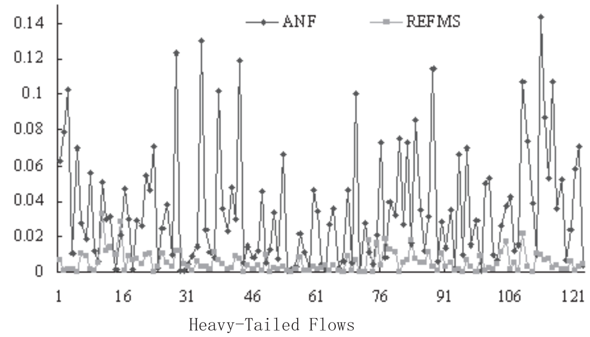

Fig. 2. Comparison of measurement error of heavy-tailed flows.

B. Estimated Heavy-Tailed Flows

Heavy-tailed flows are some largest flows, and every heavy-tailed flow only consists of one flow. Figure 2 is the comparison between the algorithms, where there are 123 heavy-tailed flows larger than 0.1% of total packets. This figure also shows that measurement errors of these heavy-tailed flows in the REFMS systems are better than that of ANF algorithm.

IV. Conclusion

This paper presents the REFMS to detect flow information, which includes Pre-Sampling module, Sample & Hold module, and Flow Removal module. REFMS can adapt to the flow sampling in varying sampling ratio by caching estimated values rather than actual measured value, and it removes flows from the flow cache using the non-uniform sampling-based algorithm to guarantee more accurate and efficient flow monitoring than ANF. We also present the NLANR data is used to compare the performance between REFMS with ANF, and analyze the application of aggregated flows and the heavy-tailed flows. The experiment shows that the REFMS has better precision than ANF under the same system resources.

References

[1] X. Li, F. Bian, M. Crovella, *et al. Detection and Identification of Network Anomalis Using Sketch Subspaces*, IMC Oct. 2006

[2] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. SIGCOMM*, Aug. 2005.

[3] Cisco IOS NetFlow Introduction, http://www.cisco.com/en/US/products/ps6601/ products_ios_protocol_group_home.html

[4] C. Estan, K. Keys, D. Moore, and G. Varghese, "Building a better Netflow," in *Proc. SIGCOMM*, Aug. 2004

[5] C. Estan and G. Varghese. "New directions in traffic measurement and accounting," in *Proc. SIGCOMM*, Aug. 2002.

[6] A. Lall, V. Sekar, M. Ogihara, J. Xu, and H. Zhang, "Data streaming algorithms for estimating entropy of network traffic," *ACM SIGMETRICS Performance Evaluation Review*, vol 34, no. 1, June 2006.

[7] F. Raspall, S. Sallent, and J. Yufera, "Shared state sampling," in *Proc. IMC*, Oct. 2006.

[8] N. G. Duffield, C. Lund, and M. Thorup, "Learn more, sample less: control of volume and variance in network measurement," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1756-1775, 2005.

[9] Abilene-V Trace Data, http://pma.nlanr.net/Special/ipls5.html