

## RES: A Resource-Efficient System for Monitoring Flow Distribution

Hong Ye and Weijiang Liu\*

*School of Information Science and Technology, Dalian Maritime University,  
Dalian, Liaoning, China, 116026  
yehong@dlmu.edu.cn, wjliu@dlmu.edu.cn*

### Abstract

*Knowledge of flow distributions from a stream of packets passing through a network link helps a network operator infer traffic demands, characterize source traffic, and detect traffic anomalies. Flow monitoring provides critical information for many network applications, and some approaches have been proposed to address flow monitoring. However, they either lack accuracy, or require intensive computing resources. In this paper, we propose a Resource-Efficient System (called RES) for monitoring flow distribution, which consists of Double sampling module, Online streaming module, and Offline processing module. Using real network traffic traces, we show that the RES indeed produces the desired accuracy in monitoring flow distribution.*

**Keywords:** *Double sampling, IP flows, Estimation, Flow distribution*

### 1. Introduction

The Internet has become an essential part of the daily life for billions of users worldwide. People are using and relying on a large variety of services built on the top of the Internet, such as web browsing, online banking, shopping, entertainment, VoIP, Video on demand, auction, social networks, *etc* [1]. However, some network attacks including DDOS, information phishing and email spamming are pervasive in the Internet, and often cause great financial loss [2]. Detecting anomalies, attacks, and faults, and determining network traffic properties require online monitoring support. Accurate monitoring also aids network operators in optimizing network performance in a timely manner [3]. Flow statistics from monitoring systems remain useful for characterizing traffic. For example, the measured numbers of flows and the distribution of their lengths have been used to evaluate gains in deployment of web proxies [4], and to determine thresholds for setting up connections in flow-switched networks [5].

In this paper, we propose a Resource-Efficient System (RES) for monitoring flow distributions. In RES we exploit a novel sampling method, *double sampling*, that combines packet sampling and flow sampling. Double Sampling consists of packet sampling and flow sampling:

- 1) First, employ packet sampling to obtain a substream.
- 2) Then, deploy flow sampling to select a small number of packets from the substream.

The main contributions of our work are summarized as follows:

---

\* Corresponding Author

- Double sampling (DSampling) is used to collect traffic data. DSampling combines the advantages of both packet and flow sampling to achieve a tradeoff of sampling efficiency against computational cost.
- RES is designed and an algorithm for estimating flow distributions from sampled data is given. This system is not only simple to implement but also scalable for high speed links.
- Evaluation based on real traces collected from distinct networks shows that RES is remarkably accurate and efficient.

The rest of this paper is organized as follows. In the next section, we review related work for monitoring flow. Some elementary concepts on flow and sampling are given in Section 3. In Section 4 we elaborate the RES with a detailed discussion. In Section 5 we make some experiments to observe estimation accuracy of the RES. We conclude in Section 6.

## 2. Related Work

Kumar et al proposed a novel SCBF that performs per-flow counting without maintaining per-flow state in [6] and an algorithm for estimation of flow size distribution in [7]. Their disadvantage is that all packets must be processed due to not using sampling. The Packet Sampling working group (PSAMP), which was founded by IETF in 2003, is chartered to define a standard set of capabilities for network elements to sample subsets of packets by statistical and other methods [8]. The method and application of Trajectory Sampling was considered in [9, 10]. The key idea of Trajectory Sampling is to sample packets based on a hash function computed over the packet content. Using the same hash function will yield the same sample set of packets in the entire domain, and enables us to reconstruct packet trajectories. Furthermore, sampling techniques have been employed in network products such as Cisco's Netflow [11] and NetranMet [12].

In [13], Duffield et al studied the statistical properties of packet-level sampling using real-world Internet traffic traces. This is followed by [14] in which the flow distribution is inferred from the sampled statistics. After showing that the naive scaling of the flow distribution estimated from the sampled traffic is in general not accurate, the authors propose an EM algorithm to iteratively compute a more accurate estimation. Scaling method is simple, but it exploits the sampling properties of SYN flows to estimate TCP flow frequencies; EM algorithm does not rely on the properties of SYN flows and hence is not restricted to TCP traffic, but its versatility comes at the cost of computational complexity. Two novel sampling methods were proposed in [15, 16].

In fact, two different sampling rules are included in the above work: *packet sampling*, which acts directly on individual packets and is ignorant of flows, and *flow sampling*, where entire flows of packets are retained or discarded at once. *Packet sampling* without per-packet processing can be easily implemented. Hohn and Veitch in [17] discussed the inaccuracy of estimating flow distribution from sampled traffic, when the sampling is performed at the packet level. For flow sampling, its disadvantage is that all packets must be processed before deciding to be retained or discarded. This is a challenge on high speed links.

## 3. Some Elementary Concepts

This section considers packet sampling alone. Within the functional requirement of sampling packets at a given rate, a number of different implementations are possible. Implementations include independent sampling of packets with probability  $\frac{1}{N}$ , and periodic

selection of every  $M$ th packet from the full packet stream. In both cases we will call  $N$  the sampling period, i.e., the reciprocal of the average sampling rate. An IP flow is a set of packets, that are observed in the network within some time period, and that share some common property known as its key. The fundamental example is that of so-called raw flows: a set of packets observed at a given network element, whose key is the set of values of those IP header fields that are invariant along a packet's path. Examples are the raw flows observed at a router, where the flow key distinguishes individual source and destination IP address, and TCP/UDP port numbers. There are at least a few definitions for the term flow depending on the context of research. In this study, we employ the one adopted in [18] which stems from the packet train model by Jain and Routhier [19].

**Definition 1.** A flow is defined as a stream of packets subject to flow specification and timeout.

In most cases, we call flow specification as flow identifier. When a packet arrives, the specific rules of flow specification determine which active flow this packet belongs to, or if no active flow is found that matches the description of this packet, a new flow is created. In this paper, the flow interpacket timeout is 64 seconds. A general flow is a stream of packets subject to timeout and having the same source and destination IP addresses, same source and destination port numbers (not considering protocol). In this paper, we will use the term original flow to describe the above flow. A flow length is the number of packets in the flow. The frequency of flows with  $k$  packets is the number of flows that contain  $k$  packets.

**Definition 2.** A sampled flow is defined as a stream of packets that are sampled at probability  $p = \frac{1}{N}$  from an original flow.

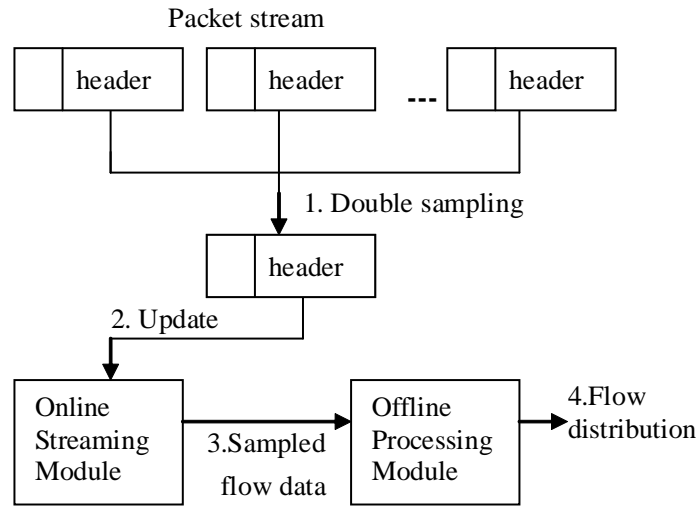
Sampling entails an inherent loss of information. We expect use statistic inference to recover information as much as possible. However, more detailed characteristics of the original traffic are not so easily estimated. Quantities of interest include the number of packets in the flow--we shall refer to this as the flow length--and the number of flows with fixed length.

## 4. A Resource-Efficient System

We propose a Resource-Efficient system, named RES, its purpose is to monitor the flow distribution.

### 4.1. RES Architecture

Sampling measurement must fulfill two goals: improving estimation precision and reducing consumption of the resources. RES Achieves two goals by using double sampling. Figure 1 is the RES architecture, which consists of three modules: Double sampling module, On-line streaming module and off-line processing module. A packet is sampled from packet stream by double sampling module with sampling period  $N=n*m$  (arc 1 in Figure 1). Then, the sampled packet is processed by online streaming module (arc 2 in Figure 1). If there is a flow entry in the flow cache, then the counter of the flow is incremented, otherwise a new flow is created and its counter is set to 1. The measurement proceeds in epochs. At the end of each measurement epoch, the counter values( the length of sampled flow), which we refer to as the sampled flow data, will be paged out from the online streaming module, and these flow entries will be removed for the next measurement epoch(arc 3 in Figure 1). This sampled flow data will be processed by an offline processing module that produces a final estimate of the flow distribution using statistical inference techniques.



**Figure 1. RES Architecture of Monitoring Flow Distributions**

#### 4.2. Double Sampling Module

In this section, we review the methodology of double sampling [20]. Double Sampling consists of packet sampling and flow sampling:

Step 1: Independent and identically distributed (i.i.d.) packet sampling consists of, for each packet in an independent manner, retaining the packet with probability  $\frac{1}{n}$  or discarding it with probability  $\frac{n-1}{n}$ .

Step 2: Independent and identically distributed (i.i.d.) flow sampling consists of, for each flow whose packet is selected in Step 1 in an independent manner, retaining the flow with probability  $\frac{1}{m}$  or discarding it with probability  $\frac{m-1}{m}$ .

There are several ways to implement Step 1. In probabilistic sampling, the router makes a pseudorandom decision whether to sample each packet. In implementations, the decision could, for example, be governed by a pseudorandom number generator with well-known properties (see *e.g.*, [21]). Periodic (or deterministic) sampling can be used too, *e.g.*, every  $n$ th packet is selected.

We use a hash function over flow identifier (the part of packet's header) to implement Step 2. The same hash function is used throughout measurement interval, so that we are ensured that all packets of a flow are either sampled, or discarded. The choice of an appropriate hash function will obviously be crucial to ensure that this subset is not statistically biased in any way. For this, the sampling process, although it is a deterministic function of the packet's header, has to resemble a random sampling process. We can choose a hash function that maps each item in the universe of flows to a random number uniform over the range  $1, 2, \dots, m$ . In practice, reasonable hash functions appear to behave adequately, *e.g.*, [22]. A packet is sampled if its hash value is equal to the specific integer, *e.g.*, 1. Because the hash function is perfectly random, flow sampling rate is  $1/m$ , *i.e.*, a flow in every  $m$  flows is sampled.

For double sampling, we denote sampling period  $N$  by  $n*m$ , written as  $N=n*m$ . For example, if  $n=5$ ,  $m=10$ , then  $N$  is written as  $5*10$ .

### 4.3. Online Streaming Module

Online streaming module processes sampled packets to update the flow cache. This module is like a Netflow or sampled Netflow. An important difference between this module and Netflow is that flow cache is updated by double-sampled packet. It maintains a flow cache containing flow records that describe the traffic forwarded by double sampling. These flow records are then exported to the offline processing module. To update the flow cache when a sampled packet is seen, it must look up the corresponding entry in the flow cache and update that entry's counter. It inserts a new flow record into the flow cache if the packet does not belong to an existing flow. Since double sampling is used, the processor and the memory holding the flow cache can keep up with the packet rate in high speed link. This module may be a hardware implementation that can work at line speeds.

### 4.4. Offline Processing Module

Offline processing module helps to infer the actual flow distribution from the sampled data. Let  $g = \{g_j : j = 1, 2, \dots, t\}$ , where  $g_j$  is the frequency of sampled flows with  $j$  packets, denote a set of sampled flow length frequencies after double sampling, and let  $f = \{f_i : i = 1, 2, \dots, t, \dots\}$ , where  $f_i$  is the frequency of original flows with  $i$  packets, denote a set of original flow length frequencies. Our objective is inferring  $\{f_i\}$  from  $\{g_j\}$ .

Firstly, we make recovery of flow sampling as follows:

$$g'_j = m * g_j, \quad j = 1, 2, \dots, t. \quad (1)$$

We regard  $g'_j = \{g'_j : j = 1, 2, \dots, t\}$  as sampled flow length frequencies with packet sampling rate  $\frac{1}{n}$  from original flows. In practice, measured sampled flow length distributions are smoother, so some effective manner of smoothing would be required for long flows. According to Eq.(1), the inferred distribution of flow lengths would be concentrated on length  $j$ . However, we know that flow length distributions have the property of being heavy-tailed, *i.e.*, the number of long flows is very few. For example, for flow sampling rate  $1/m=1/100$ , if we only sample a flow of length  $j=10000$  in length interval  $[9900, 11000]$ , *i.e.*,  $g_{10000} = 1$ , and  $g_j = 0$ , for  $j = 9900, \dots, 11000$ , we shouldn't think that there are 100 flows of length 10000, rather than think that there are 100 flows in length interval  $[9900, 11000]$ . So we estimate as 100 flows of different lengths, not 100 flows of the same length. After they are smoothed, we obtain:

$$\bar{g} = \{\bar{g}_j, j = 1, 2, \dots\} \quad (2)$$

Under independent sampling of packets with probability  $p = 1/n$ , the number of packets  $j$  sampled from an original flow of  $i$  packets follows the binomial distribution

$$B_p(i, j) = \binom{i}{j} p^j (1-p)^{i-j}. \text{ Let } \gamma = \sum_j \bar{g}_j, \text{ and } \phi_i \text{ denote the frequencies of original}$$

flows of length  $i$  conditional on at least one of its packets being selected, and  $\sum_i \phi_i = 1$ . Our

aim is to estimate  $\phi = \{\phi_i\}$  from the frequencies  $\{\bar{g}_j\}$ . We now derive an expression for log-

likelihood  $L(\phi)$  to obtain  $\bar{g}_i$  given  $\phi$ . Here,  $c_{ij} = B_p(i, j) / (1 - B_p(i, 0))$  is the probability

that  $j$  packets are sampled from a flow of length  $i$ , conditional on  $j \geq 1$ , i.e., that the flow is sampled. For any  $j$ , its probability function is  $(\sum_{i \geq j} \phi_i c_{ij})^{\overline{g_j}}$ . Hence we obtain the likelihood

function  $\prod_{j \geq 1} (\sum_{i \geq j} \phi_i c_{ij})^{\overline{g_j}}$ . Therefore the logarithm of likelihood function is:

$$L(\phi) = \sum_{j \geq 1} \overline{g_j} \log \sum_{i \geq j} \phi_i c_{ij} \quad (3)$$

Now we adopt a standard iterative approach: the Expectation Maximization (EM) algorithm [14], the standard form is as follows.

Starting with an initial value  $\phi^{(0)}$ , for example,  $\phi^{(0)} = \{\frac{\overline{g_i}}{\gamma}\}$ , the algorithm finds  $\sup\{L(\phi) : \phi \in \Delta\}$ , by iterating between the following two steps ( $k = 0, 1, \dots$ ):

E step. Let  $f_{ij}$  denote the frequencies of original flows of length  $i$  from which  $j$  packets are sampled. Thus  $\overline{g_j} = \sum_i f_{ij}$ , while  $\overline{f_i} = \sum_{j \geq 1} f_{ij}$  is the frequency of original flows of length  $i$  at least one of whose packets is sampled. Form the complete data likelihood function assuming known  $f_{ij}$ :

$$L_c(\phi) = \sum_{i \geq j \geq 1}^{N_j \text{ bond}} f_{ij} \log \phi_i c_{ij} \quad (4)$$

Form the expectation  $Q(\phi, \phi^{(k)})$  of  $L_c(\phi)$  conditional on the known frequencies  $\overline{g_j}$ , according to a distribution  $\phi^{(k)}$ :

$$Q(\phi, \phi^{(k)}) = \sum_{i \geq j \geq 1} E_{\phi^{(k)}}[f_{ij} | \overline{g}] \log \phi_i c_{ij} \quad (5)$$

M step. Define  $\phi^{(k+1)} = \arg \max_{\phi \in \Delta} Q(\phi, \phi^{(k)})$ . From the Legendre equations in the maximization of  $Q(\phi, \phi^{(k)})$  we have:  $\phi_i^{(k+1)} = \frac{E_{\phi^{(k)}}[\overline{f_i} | \overline{g}]}{\gamma}$ . Through direct computation of the above conditional expectation we obtain:

$$\phi_i^{(k+1)} = \frac{1}{\gamma} \sum_{i \geq j \geq 1} \frac{\phi_i^{(k)} c_{ij} \overline{g_j}}{\sum_{l \geq j} \phi_l^{(k)} c_{lj}} \quad (6)$$

Iterate steps E and M until some termination criterion is satisfied. Let  $\overline{\phi}$  denote the termination point. We write our estimation of original small flows as  $f_i = \overline{\phi_i} \gamma / (1 - B_p(i, 0))$ .

**Table 1. Traces Used in our Experiments**

Note that flow label is 4-tuple <srcIP, dstIP, srcPort, dstPort>

Trace	# of flows	# of packets
IPKS0	164147	3453219
IPKS1	132139	3453219
MAVI	267984	2453203
CERNET	147561	4032981

## 5. Evaluation

In this section we apply the estimators derived in the previous section to experimental traffic traces. We infer the flow statistics from the sampled versions of the traces, and we compare them with the unsampled flow statistics of the original traces.

We adopt the Weighted Mean Relative Difference (WMRD) from [14] as our evaluation metric. Suppose the number of original flows of length  $i$  is  $n_i$  and our estimation of this

number is  $\hat{n}_i$ . The value of WMRD is given by: 
$$WMRD = \frac{\sum_i |n_i - \hat{n}_i|}{\sum_i \left(\frac{n_i + \hat{n}_i}{2}\right)}$$

### 5.1. Data Considerations

Trying to make the experimental data as representative as possible, we use packet header traces gathered at three different locations of the Internet, namely, the MAWI Working Group of the WIDE Project(MAWI) [23], Jiangsu provincial network border of China Education and Research Network (CERNET) [24], and NLANR [25]. The trace from MAWI was collected on a trans-Pacific line (150Mbps link), on March 30, 2009 at 00:00 am. The IPv6 packets of MAWI are filtered out in following experiments.

The CERNET were collected at Jiangsu provincial network border of China Education and Research Network (CERNET) on April 17, 2004. The backbone capacity is 1000Mbps; mean traffic per day is 587 Mbps. For this trace, we make double sampling at double sampling period  $n*m=2*2, 2*5, 2*10, 5*2, 5*5, 5*10, 10*2, 10*5, 10*10$ , respectively.

We also use a pair of unidirectional traces from NLANR: IPKS0 and IPKS1, collected simultaneously on both directions of an OC192 link on June 1, 2004. The link connects Indianapolis (IPLS) to Kansas City (KSCY) using Packet-over-SONET.

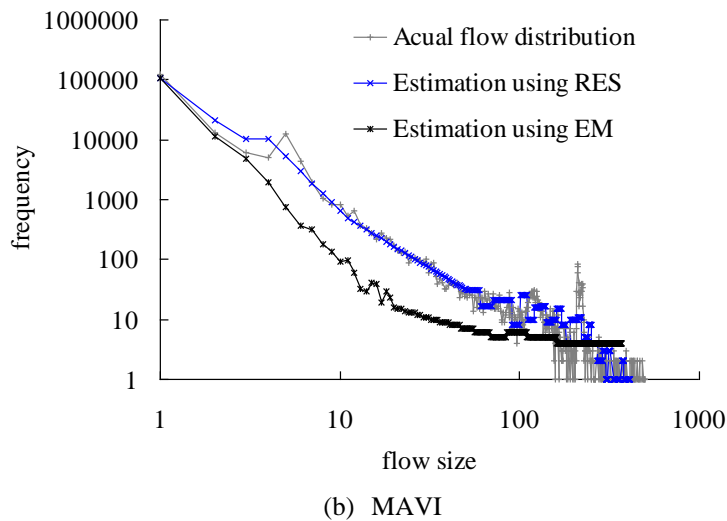
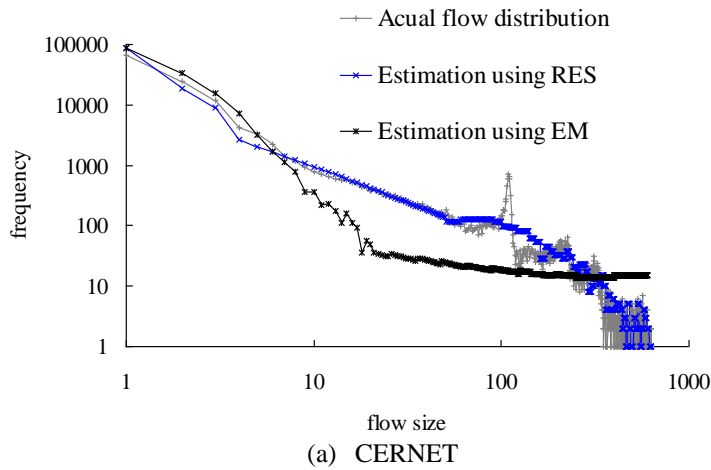
Table 1 summarizes all the traces used in the evaluation.

**Table 2. WMRD of Flow Distribution Estimation for Double Sampling and Packet Sampling Alone for CERNET Trace**

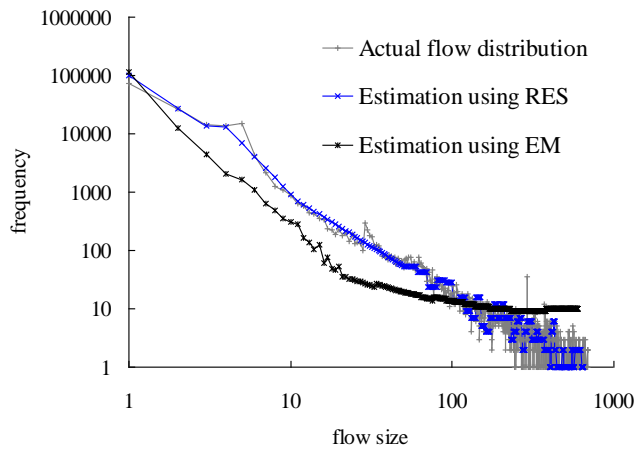
Sampling period n*m,N	WMRD of double sampling	WMRD of packet sampling alone
2*4,4	5%	12%
2*5, 10	8%	23%
5*2,10	10%	23%
5*5, 25	11%	28%
5*10, 50	13%	34%
10*10, 100	19%	39%

### 5.2. Estimation Comparison

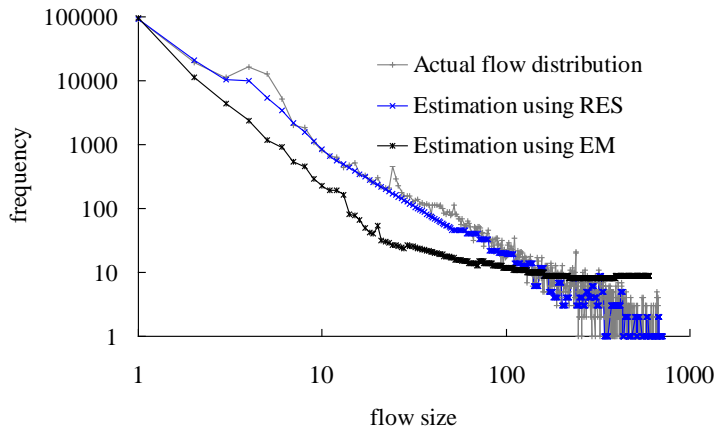
We then compare double sampling with packet sampling alone in estimating flow distributions. For same trace in same sampling period ( $N=n*m, nm$ ), we run double sampling and packet sampling respectively. Then we use EM algorithm to estimate flow distributions from sampled statistics, respectively. Comparing with the actual flow distributions, we find that the estimated results by double sampling are always more accurate than those by packet sampling. This conforms to the conclusion that inversion based on flow sampling performs well [17]. Figure 2 compares the two estimators derived by RES and EM, double sampling and packet sampling at sampling period  $N=10*10,100$ , respectively. Observe that estimated result by packet sampling is much worse. Table 2 shows the flow length distribution estimation of double sampling is much more accurate that of packet sampling alone.







(c) IPKS0



(d) IPKS1

**Figure 2. Comparing Performance of Different Traces for RES and EM Algorithm at Sampling Period  $N=10*10,100$ .**

### 5.3. Estimation Accuracy and Scalability

In this subsection, we consider how to choose  $n$  and  $m$  for fixed  $N=n*m$ . In fact, this is the choice on the estimation accuracy and on scalability with link speed. For fixed  $N$ ,  $n$  is decreasing as  $m$  increases, and vice versa. When  $n=1$ , double sampling becomes flow sampling alone. In this case, scalability with link speed is worst because each packet is processed, but the estimation of flow length distributions is best. On the contrary, double sampling becomes packet sampling alone when  $m=1$ . The scalability is best, but the estimation accuracy is worst. Therefore, choosing proper  $m$  and  $n$  is the trade-off between the accuracy and the scalability in practical application. Double sampling is very flexible in making appropriate trade-offs between the accuracy and the scalability.

## 6. Conclusions

This paper presents the RES to monitor flow distribution. Double sampling is used in RES to overcome the shortcomings of packet sampling alone and flow sampling alone. This

system is not only simple to implement but also scalable for high speed links. In experiments, the advantage of RES is shown in estimating the distribution of flow length; i.e., the accuracy of flow distribution measurement is dramatically improved.

## Acknowledgements

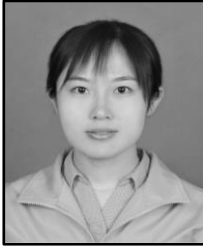
This work is supported in part by National Natural Science Foundation of China under Grant No.61173165, the Scientific Research Fund of Liaoning Provincial Education Department under Grant No.L2013195, and the Fundamental Research Funds for the Central Universities under Grant No.3132013040 and No.3132013039.

## References

- [1] Y. Liu, W. Chen and Y. Guan, "A Fast Sketch for Aggregate Queries Over High-speed Network Traffic", INFOCOM, Orlando, USA, (2012) March 25-30.
- [2] M. Edman and B. Yener, "On Anonymity in an Electronic Society: A Survey of Anonymous Communication Systems", ACM Computing Survey, vol. 42, no. 1, (2009).
- [3] S. Gangma, P. Sharma, and S. Fahmy, "Pegasus: Precision Hunting for Icebergs and Anomalies in Network Flows", INFOCOM, Turin, Italy, (2013) April 14-19.
- [4] A. Feldmann, R. Caceres, F. Douglis, G. Glass and M. Rabinovich, "Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments", INFOCOM, (1999).
- [5] A. Feldmann, J. Rexford and R. Caceres, "Efficient Policies for Carrying Web Traffic over Flow-Switched Networks", IEEE/ACM Transactions on Networking, vol. 6, no. 6, (1998), pp. 673-685.
- [6] A. Kumar, J. Xu, L. Li and J. Wang, "Space Code Bloom Filter for Efficient Traffic Flow Measurement", IEEE INFOCOM, (2004), pp. 1762-1773.
- [7] A. Kumar, M. Sung, J. (Jim) X. and J. Wang, "Data Streaming Algorithms for Efficient and Accurate Estimation of Flow Size Distribution", ACM SIGMETRICS, (2004), pp. 177-188.
- [8] Packet Sampling (psamp), <http://www.ietf.org/html.charters/psamp-charter.html>, vol. 2, no. 2, (2005).
- [9] N. G. Duffield and M. Grossglauser, "Trajectory Sampling for Direct Traffic Observation", IEEE/ACM Trans. On Networking, vol. 9, no. 3, (2001), pp. 280-292.
- [10] N. G. Duffield, and M. Grossglauser, "Trajectory Sampling with Unreliable Reporting", IEEE Infocom 2004, (2004) March.
- [11] Sampled Cisco, [http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products\\_feature\\_guide09186a0080081201.html](http://www.cisco.com/en/US/products/sw/iosswrel/ps1829/products_feature_guide09186a0080081201.html), 2002.12.8.
- [12] NeTraMet Version 4.4 Now Available, <http://www2.auckland.ac.nz/net/Accounting/ntm.Release.note.html>, (2002).
- [13] N. G. Duffield, C. Lund and M. Thorup, "Properties and Prediction of Flow Statistics from Sampled Packet Streams", ACM SIGCOMM Internet Measurement Workshop, (2002), pp. 159-171.
- [14] N. G. Duffield, C. Lund and M. Thorup, "Estimating Flow Distributions from Sampled Flow Statistics", IEEE/ACM Transaction on Networking, vol. 13, (2005), pp. 933-945.
- [15] F. Raspall, S. Sallent, and J. Yuferra, "Shared-State Sampling", Proceedings of the 6th ACM SIGCOMM IMC, (2006).
- [16] C. Hu, B. Liu, S. Wang, J. Tian, Y. Cheng and Y. Chen, "ANLS: Adaptive Non-Linear Sampling Method for Accurate Flow Size Measurement", IEEE Transactions on Communications, vol. 60, no. 3, (2012), pp. 789-798.
- [17] N. Hohn and D. Veitch, "Inverting Sampled Traffic", Internet Measurement Conference, Miami Beach, Florida, USA, (2003) October 27-29.
- [18] K. C. Claffy, H. W. Braun and G. C. Polyzos, "A Parameterizable Methodology for Internet Traffic Flow Profiling", IEEE JSAC 13, (1995), pp. 1481-1494.
- [19] R. Jain and S.A Routhier, "Packet Trains-measurements and a New Model for Computer Network Traffic", IEEE JSAC 4, (1986), pp. 986-995.
- [20] W. J. Liu and J. Gong, "Double Sampling for Flow Measurement on High Speed Links", Computer Networks vol. 52, no. 11, (2008), pp. 2221-2222
- [21] P. L'Ecuyer, "Efficient and Portable Combined Random Number Generators", Communications of the ACM, vol. 31, (1988), pp. 742-749 and 774
- [22] M. V. Ramakrishna, "Practical performance of Bloom Filters and Parallel Free-text Searching" Communications of the ACM, vol. 32, no. 10, (1989), pp. 1237-1239.
- [23] WIDE, <http://tracer.csl.sony.co.jp/mawi/samplepoint-F/20090330/200903300000.html>, (2010).

- [24] JSLAB, (2010) October, <http://ntds.njnet.edu.cn/data/index.php>.  
[25] NLANR, (2010) October, <ftp://wits.cs.waikato.ac.nz/pma/long/ipls/3/>.

## Authors



**Hong Ye**, she received her master's degrees from Dalian University of Technology, China in 2003. Now she is full lecturer school of Information Technology, Dalian Maritime University, China. ince 2009, she has been pursuing the Ph.D. Degree at the Department of Computer Sciences, Dalian Maritime University. Her current research interests include different aspects of Distributed Systems.



**Weijiang Liu**, he received the Ph.D. degree in computation mathematics from Jilin University, Changchun, China, in 1998. From June, 2004 to June, 2006, he was a postdoctoral fellow of Post Doctoral Station for Computer Science and Technology, Southeast University, China. He is currently a professor in school of Information Technology, Dalian Maritime University, China. He has published more than 50 papers and his current research interests include network measurement and network security.

