一种用于小流估计的数据包公平抽样算法

任高明，夏靖波，乔向东，杨 仝

(空军工程大学信息与导航学院 西安 710077)

【摘要】现有数据包公平抽样算法通常根据到达数据包所属流大小的估计值设置包抽样率，令大流所含数据包抽样率低，小流所含数据包抽样率高，缺点是算法的优劣依赖于不同方法对流大小估计的准确性；小流估计误差较大。针对此问题，利用大流持续时间长且到达速率高的特点，提出一种基于时间分片的用于小流估计的数据包公平抽样算法(MFEPS)。该算法将测量时间分割成片，抽取每个流在每个时间片内的第一个数据包，而不需要估计数据包所属流的大小。理论分析和实验结果均表明，与已有算法相比，对于小流估计，MFEPS算法在相同的CPU资源消耗条件下，具有更高的准确性和良好的扩展性。

关 键 词 重尾分布; 小流估计; 数据包抽样; 流量测量

中图分类号 TP393 文献标志码 A doi:10.3969/j.issn.1001-0548.2014.04.023

A Fair Packet Sampling Algorithm for Mice Flow Estimation

REN Gao-ming, XIA Jing-bo, QIAO Xiang-dong, and YANG Tong

(Information and Navigation College, Air Force Engineering University Xi'an 710077)

Abstract In most existing fair packet sampling algorithms, the sampling probability is usually set according to the estimation of the size of flow which the arriving packet belongs to, so the accuracy of the algorithm depends on the accuracy of the method to estimate the size of the flow and existing algorithms have a high estimation error for mice flow. To solve this problem, a new fair packet sampling algorithm which is based on time sectioning and used to estimate mice flow is proposed according to the characteristic that elephant flow has a high arrival rate and long alive time. The algorithm samples the first packet of every flow in a fixed time section while do not need to estimate the size of the flow. Theoretical analysis and experiments results show that packet sampling for mice flow estimation (MFEPS) method has a higher accuracy and a better scalability at the same CPU resource consumption in estimating the size of mice flow compared with existing sampling algorithms.

Key words heavy tailed distribution; mice flow estimation; packets sampling; traffic measurement

网络流量测量对网络工程、异常检测等网络运营管理意义重大。随着网络规模的不断扩大和链路速率的迅速提高，数据包到达频率愈来愈高，现有的网络流量测量硬件的处理速度难以满足需要[1]。这种情况下，传统的全流量测量方法已不再适用[2]。如何在有限的资源条件下，完成高速链路流量测量成为当前亟待解决的问题[3-4]。

根据不同应用需要，有选择地提取“有代表性”的流量信息是当前网络流量测量的主要解决方案之一[5]。不同的应用对流量数据的需求各不相同，如有的需要最详尽的分组信息；有的需要流层面的信息；有的只对特定的流量感兴趣等。常用的流量测量方案有大流提取技术[6]和公平抽样技术[7]。已有研究表明网络中的流服从“重尾分布”，即少数大流占据大部分流量，剩余的流量由大量小流构成[8]。对于某些应用，如流量计费、流量监控等，只关注占据大部分流量的大流就可以满足需要，因此，研究者提出“抓大放小”的策略，即大流提取技术。然而，对其他一些应用，如果网络异常检测、业务流分类等需要较完整的流级别信息的应用来说，丢弃占据绝大多数的小流信息，会引起很大的误差，为保证大流和小流之间的公平性，研究者提出了公平抽样技术。

但现有的公平抽样算法大多数根据到达数据包所属流大小的估计值设置包抽样率，令大流所含数据包抽样率低，小流所含数据包抽样率高。存在的不足有：1) 算法的准确性依赖于流大小估计的准确性。流大小估计准确，则算法的估计误差小；反之

收稿日期：2013 – 05 – 28；修回日期：2014 – 05 – 06

基金项目：国家自然科学基金(61202489)；陕西省自然科学基础研究计划(2012JZ8005)

作者简介：任高明(1986 –)，男，博士生，主要从事网络流量测量、网络管理方面的研究.

亦然。2) 小流估计误差大[9]。

本文针对现有数据包公平抽样算法的缺点，提出一种新的用于小流估计的数据包公平抽样算法(packet sampling for mice flow estimation，MFEPS)，该算法牺牲大流数据包抽样率换取更高的小流数据包抽样率，在有限的资源条件下，提高小流的估计准确性，以满足高速网络流量测量的需要。

1 算法描述

SGS(sketch guided sampling)算法[7]是公平抽样算法的典型代表，其核心思想是设置包抽样率为该数据包所属流大小估计值的单调递减函数，使得大流所含数据包以较低的概率被抽样，小流所含数据包以较高的概率被抽样，通过牺牲大流的包抽样率以换取较高的小流包抽样率。与均匀随机抽样相比，在相同的包抽样比下，SGS抽样算法能够更好地保证数据流之间的公平性，更完整地保留了数据流级的流量信息。文献[10]在SGS算法的基础上，采用多解析度抽样统计器近似地估计各条数据流的流量，提出一种空间高效的数据包公平抽样算法(space-efficient fair sampling，SEFS)，用于大流检测和流量测量。其余的公平性抽样算法[11-12]和SGS算法类似，只是估计数据包所属流大小的方法不同。

以SGS算法为例进行分析，假定小流包含的数据包个数为N，大流所含的数据包个数为M(N<M)，根据SGS算法的抽样概率计算公式，有：

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

式中，i表示流的第i个数据包；ε为估计标准差。SGS算法中大流的前N个数据包和小流的N个数据包的抽样概率一一对应相等，需要的内存访问次数近似相等。由于大流所含数据包远多于小流所含数据包，对大流后续的M–N个数据包的抽样操作消耗了大部分系统资源，极大地影响了小流估计的准确性，并没有完全保证大流和小流之间抽样的公平性。若要进一步降低小流估计误差，则对系统资源的要求会更高，严重限制了算法的扩展性，在有限资源条件下，快速、准确地估计小流大小对于异常监测等应用意义重大。如网络中的异常流量通常是在短时间内，大量小流突然到达，在这种情况下，对网络流量测量的实时性和准确性提出了很高的要求。尽管现有公平抽样方法和均匀随机抽样相比，已经极大地提高了小流的抽样率，但面对网络攻击发生时，仍不能满足检测需要。

本文针对现有的数据包公平抽样算法的缺点，提出一种新的用于小流估计的数据包公平抽样算法MFEPS。相关的研究表明小流往往持续时间短或者分组到达速率低，大流持续时间长且分组到达速率高[13]。利用这一特点，算法基本原理是将测量时间分片，在每一时间片Δt内，只抽取每条流的第一个数据包。设置合理的时间片Δt，通常情况下，时间Δt内到达的属于大流的数据包较多，只抽取第一个数据包，有效地降低了大流的包抽样率。对于小流来说，Δt时间内，到达的数据包个数相对较少，抽取第一个数据包，和大流的包抽样率相比，相应地提高了小流所含数据包的抽样率。

算法流程图如图1所示，当有数据包到达时，首先在流缓存中查找流记录，如果不存在相应的流记录，则新建流记录，并开始计时；如果存在，判断距上一次抽取该流记录的数据包是否已经超过时间片Δt，若是，抽取该数据包；否则，只更新计时器。

当数据包到达时，首先查找是否存在相应的流记录，为提高网络数据包的处理速度，通常采用两种存储方案：使用内容寻址存储器(content addressable memory，CAM)对缓存中的流记录统一寻址和软件上采用Hash表存储。为方便仿真实验，本文采用Hash方案。根据Hash函数将数据包的五元组(源IP地址、目的IP地址、源端口、目的端口和协议)作为关键字，映射到相应的流记录上，Hash函数选择H_3函数，缓存中的流记录采用链表结构组织。

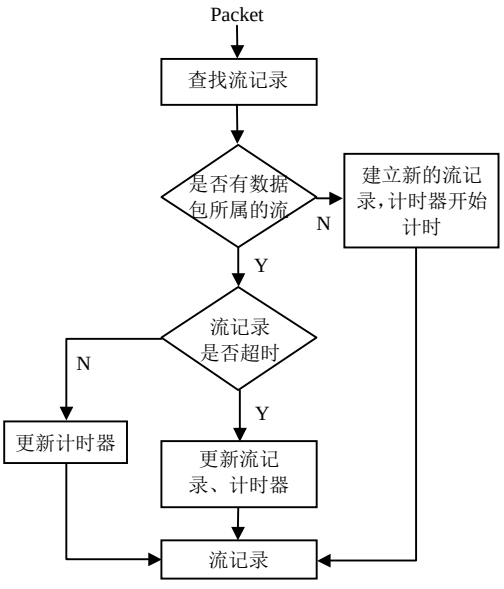


图1 算法流程图

2 理论分析

2.1 误差分析

假定流F的大小为s(包含数据包的个数)，持续时间为T，将持续时间T分为N段，则每个时间片$\Delta t=T/N$，设一共有k个数据包被抽取到($k\leqslant N$)，其中，

第i个数据包被抽取的概率为p_i，则流F包含的数据包个数评估值为$\hat{s}=\sum_{i=1}^{k}1/p_i$。可表示为$\sum_{i=1}^{s}u_i\times 1/p_i$，若第$i$个数据包被抽取，则$u_i$为1；否则为0。由于各数据包的抽样操作相互之间独立，可知$\hat{s}$的方差为：

$$\mathrm{Var}(\hat{s})=\sum_{i=1}^{s}\mathrm{Var}(u_i\times 1/p_i) \tag{2}$$

其中，

$$\mathrm{Var}(u_i\times 1/p_i)=(1-p_i)/p_i \tag{3}$$

$$\mathrm{Var}(\hat{s})=\sum_{i=1}^{s}(1-p_i)/p_i \tag{4}$$

当N>1时，即持续时间T分为多个时间片，因为有k个数据包被抽取，不同的时间片内最小的抽样概率为$1/(s-(k-1))$，即为$1/(s+1-k)$。进而有：

$$\mathrm{Var}(\hat{s})\leqslant\sum_{i=1}^{s}\left(1-\frac{1}{s+1-k}\right)\Big/\frac{1}{s+1-k}\leqslant$$
$$s\times\left(1-\frac{1}{s+1-k}\right)\Big/\frac{1}{s+1-k}\leqslant s^2-sk \tag{5}$$

设定方差为固定值ε^2，即：

$$s^2-sk=\varepsilon^2 \tag{6}$$

由上述假设可知，为保证时间分段有意义，则$N\leqslant s$，由此可得：

$$\mathrm{Var}(\hat{s})\leqslant s^2-Nk=s^2-\frac{T}{\Delta t}k \tag{7}$$

设定方差为固定值ε^2，即：

$$s^2-\frac{T}{\Delta t}k=\varepsilon^2 \tag{8}$$

则$\varepsilon=\sqrt{s^2-Tk/\Delta t}$，可知，当流$F$的大小s确定后，分段时间$\Delta t$越小，标准差$\varepsilon$越小。

本文统一定义，5 s内，五元组(源地址、目的地址、源IP、目的IP和协议)相同的数据包为流。流的大小指包含数据包个数的多少。定义数据包个数小于单位时间内数据包数0.1%的流为小流。

从后文使用的互联网采集的流量集trace1中选取6个流数据，如表1所示。

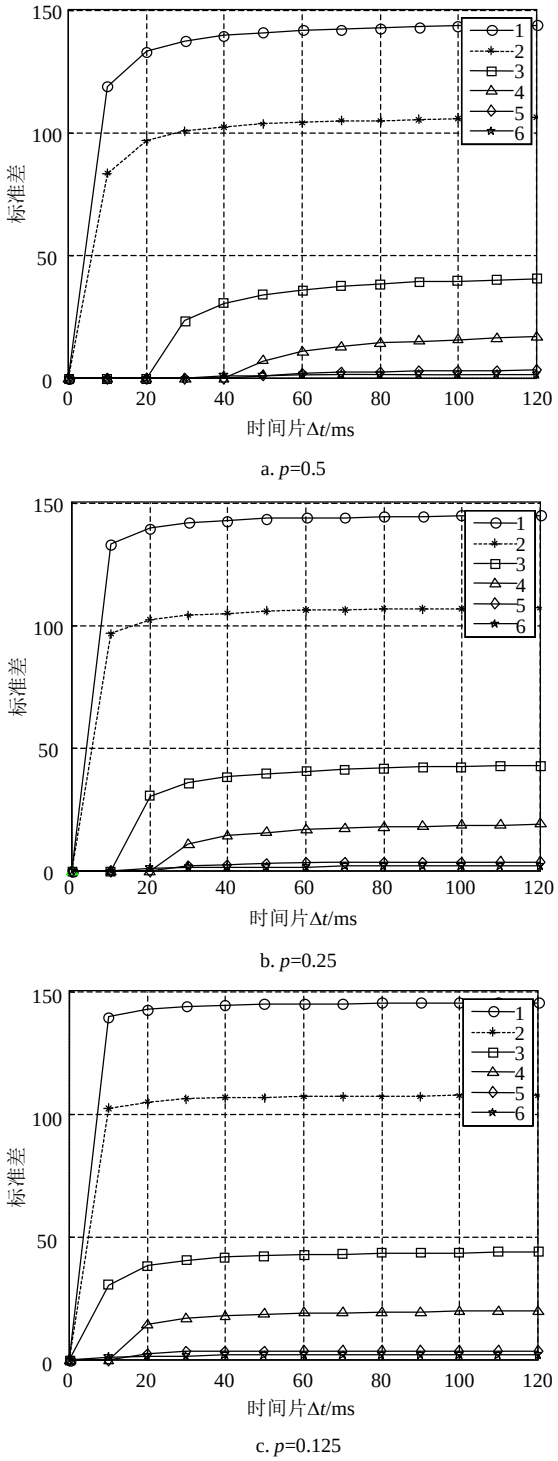
表1　流数据表

编号	数据包数	持续时间/ms
flow1	146	978.136 0
flow2	108	866.065 1
flow3	45	1 958.7
flow4	21	1 841.552 9
flow5	4	375.413 0
flow6	2	122.805 1

对于上述实测数据，假定每个流的总体抽样概率固定为$p=k/s$，则标准差为：

$$\varepsilon=\sqrt{s^2-\frac{Tsp}{\Delta t}} \tag{9}$$

式中，设定Δt的取值范围为0～120 ms。p分别取0.5、0.25、0.125时的标准差如图2所示。



a. p=0.5

b. p=0.25

c. p=0.125

图2　标准差示意图

从图2可以看出，当抽样概率p=0.5时，flow1和flow2的标准差随着时间分段Δt的变小，在Δt = 10 ms附近迅速变小；flow3和flow4的标准差分别在Δt =20 ms和Δt =40 ms处近似为0；flow5和flow6的

标准差全部近似为0。当抽样概率p=0.25时，即p变小，意味着抽样的数据包更少，flow1和flow2的标准差很高，随着时间分段Δt的变小，标准差有变小的趋势，但变化不大；flow3和flow4的标准差分别在Δt=10 ms和Δt=20 ms处近似为0；flow5和flow6的标准差略有变大。当抽样概率p=0.125时，即p更小，此时，flow1和flow2的标准差很高，且随着时间分段Δt的变小，标准差基本不变；flow5和flow6的标准差仍维持在较低范围。可以发现，随着抽样概率的降低，MFEPS算法对包含数据包个数较少的小流的估计误差变化很小，一直维持在较小范围内，估计准确性较高。

2.2 复杂度分析

2.2.1 时间复杂度

当一个数据包到达时，首先查找相应的流记录，MFEPS算法采用Hash方案，其计算复杂度为$O(1)$，后面的操作包括抽样、更新或者新建，因此，算法的处理时间由两部分组成：查找流记录和抽取数据包后的更新或者新建流记录。和SGS类算法相比，查找流记录是判断该数据包为大流还是小流的基础，两种算法均不可避免这一步，因此算法消耗的时间取决于总体抽样率。

2.2.2 空间复杂度

无论是SGS算法还是MFEPS算法，均采集每个流的第一个数据包。在固定的测量时间片内，流的数目是确定的。而所需的存储空间就是测量时间片内流的数目，因而也是确定不变的，即为流记录缓存值。假设R为链路速率(单位为byte/s)，b为数据包平均值，n为每流平均数据包数，则在时间t内到达的流个数，即所需流记录缓存值为：

$$M=[t/(bn)]R \tag{10}$$

2.2.3 访问存储器次数

访问存储器次数，是影响一个算法能否处理高速网络数据的重要因素之一。MFEPS算法中，每个数据包到达时，查找是否存在相应的流记录，采用Hash表存储，需访问存储器1次。如果没有相应的流记录，则新建，此时，需要访问存储器2次，包括写操作和指针更改；否则，丢弃该数据包。因此，在每一时间片Δt内，MFEPS算法处理每个数据包需要访问内存的次数最多为3次，最少为1次。

对SGS方法进行类似分析可知，在固定的每一时间片Δt内，SGS方法最少需要访问内存$1+1/(1+\varepsilon^2 i)$次，最多需要访问内存3次。

两种方法的主要区别是：在时间片Δt内，对每条流第一个以后的数据包的处理方式不同，MFEPS方法直接丢弃，而SGS方法以一定概率抽取。正是这一区别致使MFEPS方法减小了存储器的访问次数，从而降低了系统消耗。根据式(1)，对大流来说，尽管随着数据包到达数目的增大，抽样概率变小，但由于大流包含数据包个数占网络数据的多数，SGS方法访问存储器的平均次数会明显大于MFEPS方法，直接增加了测量设备处理器的负担。

2.2.4 实现考虑

算法的实现复杂度主要取决于存储器的访问速度和大小。MFEPS方法中，流记录缓存最多需要访问内存3次。根据目前的半导体技术，SRAM的访问速度可以达到2 ns[4]，假设采用访问速度为4 ns的SRAM，最坏情况下MFEPS算法需要12 ns处理一个数据包。在OC-768链路上，设满速率传输包长为64 byte的数据包，则包到达间隔为12.5 ns[10]。因此，MFEPS算法完全可以满足OC-768链路的要求。

按每分组平均400 byte，每流平均10个分组计算，1 s内，所需空间为312.5k流记录。每个流ID最多104 bit，若字节总数、指针各需32 bit，则每个流记录最多需168 bit。因此，使用Hash表的情况下，共需52.5 Mb的SRAM。现在的半导体技术可以提供64 Mb单块的SRAM，因此MFEPS算法可以实现。

MFEPS算法用于持续测量过程时，借鉴文献[14]，使用SRAM和DRAM结合的存储机制。在单位测量时间结束后，将SRAM的数据转存至后台的DRAM，这样在保证数据包高速处理的同时，降低了存储代价。

3 实验结果及分析

本文使用互联网采集流量数据进行试验，并与SGS算法进行比较。对于同样的分组序列，总体抽样率直接决定了抽样方法的CPU资源消耗量。这是因为总体抽样率越高，抽取的数据包个数越多，内存访问次数越多。为保证算法比较之间的公平性，在估算误差相同的情况下，比较两种算法的总体抽样率，抽样率越高，说明消耗CPU资源越多。具体实施方法如下：给出MFEPS算法的时间分段Δt，抽样后，计算出MFEPS算法的总体抽样率p和误差ε；通过ε设定SGS算法中流所含数据包的抽样概率，最终得到总体抽样概率；对两种算法的总体抽样率进

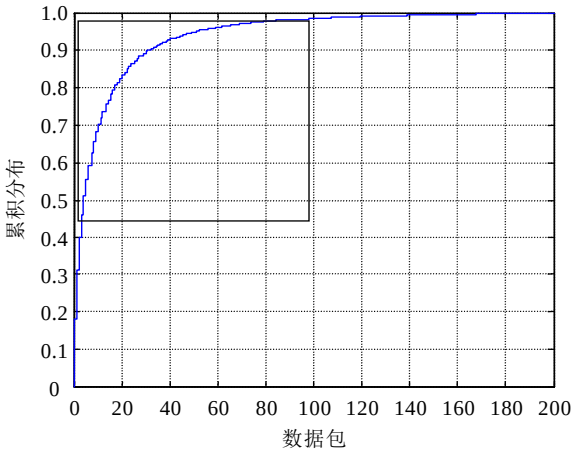
行比较，可以得出处理器CPU的消耗量。

实验所用数据分别来自于CAIDA和MAWI，具体数据如表2所示。

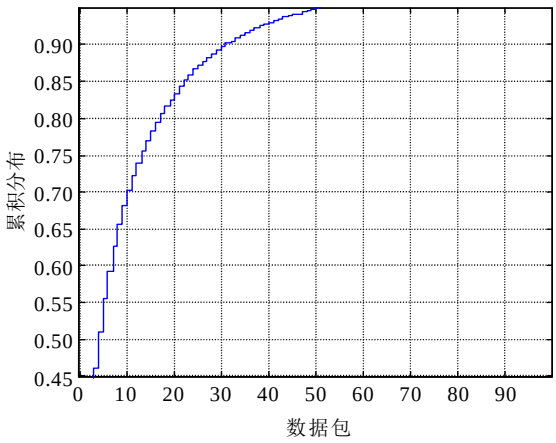
表2　实验数据表

编号	实验数据	时间	链路速率/ Gb·s^{-1}	持续时间/s
trace1	oc48-mfn	2011.1.15	2.5	300
trace2	oc12	2011.8.11	0.622	5 400

根据前文定义，trace1中数据包个数小于31的流为小流，trace2中数据包个数小于9的流为小流。为计算方便，trace1和trace2的小流分界值分别记为30和10。图3为trace1中前5 s数据的流大小的累积分布，图3b为图3a的放大。从图中可以看出，数据包个数在30以内的流占了总流数目的89%。类似地，计算得trace2中数据包个数小于10的流占总数的84.3%。



a. tracel流大小累积分布



b. a图的放大图

图3　trace1流大小累积分布图

试验中，取时间分段分别为80、100和120 ms。试验中分别选择10,15,…,40个数据包为小流和大流的分界值，得到的试验结果分别如表3和表4所示。

可以发现，当时间分段确定后，MFEPS算法的总体抽样率确定；在小流分界线确定后，抽样标准差确定，在同样标准差条件下，总体抽样率越高，所需资源越多。对于trace1，流所含数据包个数在30以下时，MFEPS算法的抽样率低，而在流分界线为35以上时，MFEPS算法抽样率高于SGS方法。对于trace2，流所含数据包在10以内时，MFEPS算法抽样率较低，而在10以上时，SGS方法的抽样率较低，即消耗资源低。

上述实验结果表明，在相同的误差条件下，对于小流的估计，MFEPS方法所需CPU资源更少，换言之，在相同的资源条件下，对于小流的估计，MFEPS方法准确性更高。

表3　trace1实验结果

Δt/ms	流界值	ε	p	p_{sgs}
80	10	0.207 3	0.496 2	0.919 4
	15	0.343 2		0.782 3
	20	0.455 1		0.663 8
	25	0.556 4		0.570 4
	30	0.634 0		0.507 2
	35	0.703 3		0.459 9
	40	0.762 7		0.425 2
100	10	0.222 2	0.466 5	0.913 9
	15	0.368 2		0.764 3
	20	0.488 3		0.642 8
	25	0.596 2		0.549 8
	30	0.679 1		0.487 4
	35	0.753 0		0.441 3
	40	0.815 8		0.405 8
120	10	0.237 5	0.441 9	0.900 3
	15	0.392 2		0.747 8
	20	0.519 4		0.623 6
	25	0.633 0		0.534 3
	30	0.720 0		0.470 1
	35	0.797 9		0.425 6
	40	0.864 1		0.392 7

表4　trace2实验结果

Δt/ms	流界值	ε	p	p_{sgs}
80	10	0.132 4	0.815 9	0.944 0
	15	0.393 4		0.670 0
	20	0.501 2		0.571 3
	25	0.533 3		0.542 9
100	10	0.159 3	0.783 9	0.922 2
	15	0.468 2		0.615 6
	20	0.594 5		0.537 5
	25	0.631 8		0.491 0
120	10	0.183 8	0.755 2	0.902 8
	15	0.536 6		0.570 0
	20	0.679 2		0.469 5
	25	0.720 7		0.451 3

4　结 束 语

小流的估计和监测对于异常检测、安全监控等应用意义重大，尤其在当前高速网络中，在有限的资源条件下准确、快速估计小流显得尤为重要。本文针对现有数据包公平抽样算法的不足，利用大流

持续时间长且到达速率高，小流往往持续时间短或者分组到达速率低的特点，提出了一种基于时间分片的数据包公平抽样算法，用于小流估计。和现有算法相比，MFEPS算法的创新之处在于其不需要估计数据包所属流的大小，具有实现简单、小流估计准确率高和扩展性强等优点。理论分析和实验结果表明，在相同的资源消耗条件下，MFEPS算法提高了小流的抽样率，估计小流准确性更高。

参 考 文 献

[1] TAMMARO D, VALENTI S, ROSSI D, et al. Exploiting packet-sampling measurements for traffic characterization and classification[J]. International Journal of Network Management, 2012, 22(6): 451-476.

[2] REN Wu-yue, LI Rui-ying, LI Mei-nan. The applicability of traditional sampling techniques in the measurement of LAN availability[C]//2012 International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE). Chengdu: IEEE, 2012: 83-88.

[3] DUFFIELD N. Fair sampling across network flow measurements[C]//Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems. [S.l.]: ACM, 2012: 367-378.

[4] 裴育杰，王洪波，程时端. 基于两级LRU机制的大流检测算法[J].电子学报，2009，37(4): 684-691.
PEI Yu-jie，WANG Hong-bo，CHENG Shi-duan. A dual-LRU based algorithm for identifying and measuring large flows[J]. Acta Electronic Sinica ,2009 ,37(4): 684-691.

[5] RASPALL F. Efficient packet sampling for accurate traffic measurements[J]. Computer Networks, 2012, 56(6): 1667-1684.

[6] CRISTIAN E, GEORGE V. New direction in traffic measurement and accounting[J]. SIGOMM Computer Communication Review, 2002, 32(4): 323-336.

[7] KUMAR A, XU J. Sketch guided sampling-using on-line estimates of flow size for adaptive data collection[C]//Proc IEEE Infocom. [S.l.]: IEEE, 2006.

[8] FANG W, PETERSON L. Inter-AS traffic patterns and their implications[C]//Global Telecommunications Conference, GLOBECOM'99. [S.l.]: IEEE, 1999.

[9] HU C, LIU B, WANG S, et al. ANLS: Adaptive non-linear sampling method for accurate flow size measurement[J]. IEEE Transactions on Communications, 2012, 60(3): 789-798.

[10] 张进，邹江兴，钮晓娜. 空间高效的数据包公平抽样算法[J]. 软件学报, 2010, 21(10): 2642-2655.
ZHANG Jin, WU Jiang-xing, NIU Xiao-na. Space-efficient fair packet sampling algorithm[J]. Journal of Software, 2010, 21(10): 2642-2655.

[11] KUMAR K, XU J, WANG J, et al. Space-code bloom filter for efficient per-flow traffic measurement[C]//Proceedings of the Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies. [S.l.]: IEEE, 2004, 3: 1762-1773.

[12] KONG Shi-jin, HE Tao, SHAO Xiao-xin, et al. Time-out Bloom filter: a new sampling method for recording more flows[M]//Information Networking, Advances in Data Communications and Wireless Networks. Berlin Heidelberg: Springer, 2006: 590-599.

[13] 王洪波，韦安明，林宇，等. 流测量中基于测量缓冲区的时间分层分组抽样[J]. 软件学报, 2006, 17(8): 1775-1784.
WANG Hong-bo, WEI An-ming, LIN Yu, et al. Time stratified packet sampling based on measurement buffer for flow measurement[J]. Journal of Software, 2006, 17(8): 1775-1784.

[14] LIEVEN P, SCHEUERMANN B. High-speed per-flow traffic measurement with probabilistic multiplicity counting[C]//Proceedings of IEEE INFOCOM. [S.l.]: IEEE, 2010: 1-9.

编 辑 漆 蓉