

空间高效的数据包公平抽样算法^{*}

张进¹⁺, 邬江兴², 钮晓娜²

¹(解放军理工大学 通信工程学院, 江苏 南京 210016)

²(国家数字交换系统工程技术研究中心, 河南 郑州 450002)

Space-Efficient Fair Packet Sampling Algorithm

ZHANG Jin¹⁺, WU Jiang-Xing², NIU Xiao-Na²

¹(Institute of Communication Engineering, PLA University of Science and Technology, Nanjing 210016, China)

²(National Digital Switching System Engineering and Technology Research Center, Zhengzhou 450002, China)

+ Corresponding author: E-mail: boost_zj@163.com

Zhang J, Wu JX, Niu XN. Space-Efficient fair packet sampling algorithm. Journal of Software, 2010,21(10): 2642–2655. <http://www.jos.org.cn/1000-9825/3667.htm>

Abstract: Fair packet sampling can obtain a higher packet sampling ratio of short flows by sacrificing the packet sampling of long ones; thus, ensuring better fairness among all flows than uniform random sampling does. However, the previously proposed fair sampling algorithm of Sketch Guided Sampling (SGS) has the drawbacks of poor space efficiency and large estimation error for short flows. In this paper, a space-efficient fair packet sampling (SEFS) algorithm is proposed. The key innovation of SEFS is a multi-resolution d -left hashing schema for flow traffic estimation. The performance of SEFS is compared to that of SGS in contexts of both flow traffic measurements and a long flow identification process that uses real-world traffic traces collected from OC-48 and OC-192 backbone network. The experimental results show that the proposed SEFS is more accurate than SGS in both application contexts, while a reduction of 65 percent in space complexity can be achieved. The improvement of estimation accuracy of SEFS is remarkable, especially for short flows, which comprise as past of a large percentage of whole network traffic flows.

Key words: network traffic monitoring; packet sampling; d -left hashing

摘要: 数据包公平抽样通过牺牲长流的包抽样率以换取更高的短流包抽样率,因而比均匀随机包抽样更能保证数据流之间的公平性.现有的公平抽样算法 SGS(sketch guided sampling)存在空间效率低、短流估计误差大的问题.提出了一种空间高效的数据包公平抽样算法 SEFS(space-efficient fair sampling).SEFS 算法的新颖之处在于采用多解析度抽样统计器对数据流流量作近似估计,各个统计器由 d -left 哈希表实现.采用在 OC-48 和 OC-192 骨干网采集的真实流量数据,在数据流流量测量以及长流检测的应用背景下,对 SEFS 算法和 SGS 算法的性能进行了比较.实验结果表明,与 SGS 算法相比,SEFS 算法在空间复杂度降低 65%的前提下,仍具有更高的估计精度.特别是对于占网络数据流绝大多数的短流而言,SEFS 算法估计精度高的优势更为明显.

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2008AA01A323 (国家高技术研究发展计划(863))

Received 2008-06-18; Revised 2009-03-30; Accepted 2009-06-01

关键词: 网络流量监测;数据包抽样; d -left 哈希

中图法分类号: TP393 文献标识码: A

对网络流量的测量与分析是流量计费、流量工程、网络安全监测与异常检测等网络运营管理工作的前提.随着网络数据率的不断提高,当前在高速骨干网链路上,流量测量与分析对于计算和存储资源的需求极高.例如,在 OC-768 链路上,设平均包长为 1 000bits,则为了实现线速处理,所能允许的平均包处理时间为 25ns,这就要求采用专门的硬件,如 TCAM,ASIC 等实现流量测量与分析设备,代价极其高昂.为了降低骨干网流量测量与分析的实现代价,必须进行有效的数据约减.数据包抽样是一种有效的数据约减的方法^[1].与分析完整的链路流量相比,分析包样本通常可以在所能获取的信息量和所需付出的处理代价之间获得更好的折衷.基于上述原因,数据包抽样在高速骨干网链路数据流的流量测量^[2-5]、流量异常检测^[6-8]等领域被广泛采用.

当前,应用最广泛的抽样方法是均匀随机抽样和固定周期抽样.均匀随机抽样^[1]以概率 p 对任意数据包进行抽样.固定周期抽样的性能和均匀随机抽样的性能相当^[1],但更易于硬件实现,因此,Cisco 路由器中采用固定周期抽样代替均匀随机抽样^[2].然而,由于网络数据流流量分布的不均匀性,均匀随机抽样虽然能够很好地保留数据包级流量信息,但却无法较完整地保留数据流级的流量信息,如并发流数目、流长分布、各条数据流的流量,等等^[9-11].对于网络安全监测、网络异常检测以及业务流分类等需要较完整的数据流级信息的应用而言,均匀随机抽样引起的估计误差较大^[6,7].

文献[12]提出了一种草图指导的抽样算法(sketch guided sampling,简称 SGS),其核心思想是设置包抽样比为该数据包所属数据流的当前流量的单调递减函数.SGS 抽样算法通过牺牲长流的包抽样率以换取更高的短流包抽样率,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,更完整地保留数据流级的流量信息.然而,SGS 抽样算法需要实时获知每条数据流的流量.文献[12]采用简单的哈希表对各条数据流的流量作近似估计,该方法的主要缺点是空间效率低,且由于哈希冲突导致对短流流量的估计误差较大.尽管文献[12]认识到此问题,并且提出通过采用两级存储结构的统计计数器^[13-15]或者文献[16]提出的 MRSCBF(multi-resolution space-code Bloom filter)算法来测量各数据流的流量,以减小对于大容量高速存储器的需求.然而,文献[12]提出的方案并不可行,因为采用两级存储结构的统计计数器和 MRSCBF 虽然能够支持线速更新,却不能支持线速读取.SGS 算法的高空间复杂度导致其实现代价高,并影响了其部署灵活性.此外,考虑到短流占据着网络流量的绝大多数^[3],因此 SGS 算法对于短流的估计误差较大也严重影响了其应用效果.

本文提出了一种空间高效的数据包公平抽样算法 SEFS(space-efficient fair packet sampling).SEFS 采用多解析度抽样统计器近似的估计各条数据流的流量.多解析度抽样统计器由 n 层独立的抽样统计器构成,第 i 层抽样统计器以概率 $p_i=P^i$ (P 为抽样常数且有 $0<P<1$) 独立地对链路数据包进行抽样,并统计所抽取的包样本中各条数据流的流量大小.各层抽样统计器有一定的统计范围,多个抽样统计器的统计范围的并集覆盖整个流量区间.各个抽样统计器采用 d -left 哈希表^[17]实现,在有效地降低哈希冲突概率的同时,仍具有较高的空间效率.采用真实的骨干网流量数据进行仿真实验的结果表明,SEFS 算法在空间复杂度比 SGS 算法降低了 65%的前提下,仍具有更小的估计误差.特别是对于短流而言,SEFS 算法估计误差小的优势更为明显.

本文第 1 节总结相关工作,并对照说明本文工作的创新之处.第 2 节首先描述数据包公平抽样算法,然后给出公平抽样的包抽样比和流抽样比的解析结果.第 3 节给出流量的多解析度抽样估计算法,并分析多解析度抽样估计算法的估计误差、空间复杂度以及时间复杂度.第 4 节通过仿真实验,在流量测量和长流检测的应用背景下,对 SEFS 算法和 SGS 算法的估计误差进行比较.第 5 节给出算法实现的结果.第 6 节是结论.

1 相关工作

根据网络流量特性和测量可用的资源状况进行自适应抽样的思想,在文献[18-20]中均有所体现.文献[12]首次考虑到抽样对于数据流之间的公平性的影响,并提出了 SGS 公平抽样算法.本文的工作和文献[12]基于同一个出发点,即保证数据流之间的抽样公平性.公平抽样需要实时获知各条数据流的流量.文献[12]提出采用简

单哈希表估测数据流流量的方法存在空间效率低以及对于短流的估计误差较大的缺陷.本文提出了一种新颖的多解析度抽样统计器以估计链路中各条数据流的流量.与文献[12]提出的采用简单哈希函数进行流量估计的方法相比,本文提出的多解析度抽样统计器在空间效率和估计准确度,特别是对于短流的估计准确度方面有明显的提升.此外,文献[12]没有分析公平抽样的包抽样比和流抽样比,本文给出了当估计误差一定时,公平抽样的包抽样比和流抽样比的解析结果.这一结果对于实际应用公平抽样算法进行网络流量分析时,在估计精度和处理代价之间进行合理权衡提供了理论准则.

公平抽样的前提是需要实时获知各条数据流的流量.在高速骨干网链路中,由于包到达间隔短且并发流数量巨大^[2-5],因此直接进行流量测量的代价极高.文献[16]采用多解析度的空码布鲁姆过滤器 MRSCBF 估计各条数据流的流量.虽然 MRSCBF 更新操作的时间复杂度低,但其查询操作的时间复杂度较高,因而难以实现高速流量条件下的线速查询,从而无法支持公平抽样.此外,文献[16]并没有对 MRSCBF 的估计误差进行理论分析.本文提出的基于 *d-left* 哈希表的多解析度抽样统计器可以实现线速查询,且空间复杂度明显低于 MRSCBF 算法.此外,本文还对多解析度抽样统计器的估计误差进行了理论分析,并通过仿真实验对所得出的理论结果进行验证.

文献[21]提出了采用 Time-Out Bloom Filter 进行包抽样,以期在一定的抽样比下能够比均匀随机抽样抽取到更多的数据流.文献[21]的核心思想是记录每条数据流的第 1 个数据包.该方法的缺陷在于其丢失了流的流量信息,从而极大地限制了其应用范围.

2 包公平抽样

设数据流 F 的流量(本文采用的流量单位为数据包数,而非字节数)为 $f(f \in [1, M])$,均匀随机抽样依概率 p 独立地对每个数据包进行抽样,设抽样到 f' 个数据包,则 $\hat{f} = f'/p$ 为 f 的无偏估计.由于 f' 服从参数为 (f, p) 的二项分布,因此其方差为 $\text{Var}(f') = fp(1-p)$,于是估计误差标准差为

$$\text{Std}(\hat{f}) = \frac{\sqrt{\text{Var}(\hat{f})}}{f} = \frac{\sqrt{\frac{1}{p^2} \text{Var}(f')}}{f} = \sqrt{\frac{1-p}{fp}} \quad (1)$$

可见,在一定的抽样概率 p 下,流量估计误差随着 f 单调递减.对于长流(如 $f \gg p^{-1}$)而言,其流量的抽样估计误差相对较小;对于中流和短流(如 $f < p^{-1}$)而言,其流量的抽样估计误差相对较大.由于网络流的流量大小服从幂律分布,因此,均匀随机抽样从占网络流总数的极少部分的长流中抽取了过量的数据包,而占网络流总数的绝大多数中、短流则被抽取到很少甚至 0 个数据包.对于网络流量测量^[2-5]、流量异常检测^[6-8]等需要数据流级流量信息的应用而言,均匀随机抽样无法提供足够完整的信息.

与均匀随机抽样不同,公平抽样可以保证对于任意 $f \in [1, M]$, $\text{Std}(\hat{f})$ 恒为某一常数 ε .公平抽样算法的思想是,对于数据流 F (设其流量为 f)的第 $i(i \in [1, f])$ 个数据包,设置其抽样比 $P(i)$ 为 i 的单调递减函数.这样,对于长流,随着它的流量的增加,其包抽样比逐渐减小,从而避免了被过多抽样.不难证明,当

$$P(i) = 1/(1 + \varepsilon^2(2i - 1)) \quad (2)$$

时, $\forall f \in [1, M]$, $\text{Std}(\hat{f}) = \varepsilon$ ^[12].文献[12]为了弥补由于 Hash 冲突引起的抽样不足而采用下列公平抽样函数

$$P(i) = 1/(1 + \varepsilon^2 i) \quad (3)$$

2.1 公平抽样的抽样比分析

在数据流的流量分布特性一定时,公平抽样的包抽样比和流抽样比由期望的估计误差 ε 来决定.在实际应用中,不仅需要关注公平抽样的估计误差,还需要关注公平抽样的包抽样比和流抽样比,以便在估计精度和处理代价之间进行合理的权衡.文献[12]没有从理论上分析公平抽样的估计误差和抽样比之间的关系,本节给出当网络数据流的流量服从 Zipf 分布^[3]时,在一定的估计误差水平下,公平抽样的包抽样比和流抽样比.

定理 1. 设流量在区间 $[1, M]$ 上服从参数为 z 的 Zipf 分布.当流量估计误差的标准差为 ε 时,公平抽样的包抽样比为

$$p_p = \frac{\sum_{i=1}^M i^{-z} \sum_{j=1}^i P(j)}{\sum_{i=1}^M i^{1-z}}.$$

公平抽样的流抽样比为

$$p_f = \sum_{i=1}^M \frac{\theta}{i^z} \left(1 - \prod_{j=1}^i (1 - P(j)) \right),$$

其中, $P(j)$ 由公式(2)确定, $\theta = \left(\sum_{j=1}^M j^{-z} \right)^{-1}$.

证明: 设链路上共有 N 条数据流, 因为流量大小在区间 $[1, M]$ 上服从参数为 z 的 Zipf 分布, 因此, 对于任意一条数据流, 其流量为 i 的概率为

$$P(f=i) = \frac{\theta}{i^z},$$

其中, θ 为规一化系数且有 $\theta = \left(\sum_{j=1}^M j^{-z} \right)^{-1}$. 于是, N 条数据流中, 流量大小为 i 的数据流的数目为 $\frac{N\theta}{i^z}$.

首先, 分析公平抽样的包抽样比. 对于流量大小为 i 的数据流, 公平抽样抽取到的数据包数的期望为 $\sum_{j=1}^i P(j)$, 其中, $P(j)$ 由公式(2)确定. 因此, 对于链路总流量, 公平抽样抽取到的数据包总数的期望为 $\sum_{i=1}^M \frac{N\theta}{i^z} \sum_{j=1}^i P(j)$. 又因为链路数据包总数为 $\sum_{i=1}^M \frac{N\theta}{i^z} \cdot i$, 于是公平抽样的包抽样比为

$$p_p = \frac{\sum_{i=1}^M \frac{N\theta}{i^z} \sum_{j=1}^i P(j)}{\sum_{i=1}^M \frac{N\theta}{i^z} \cdot i} = \frac{\sum_{i=1}^M i^{-z} \sum_{j=1}^i P(j)}{\sum_{i=1}^M i^{1-z}}.$$

其次, 分析公平抽样的流抽样比. 由大数定律可知, 流抽样比等于对于任意一条数据流, 其被抽样到的概率. 对于流量大小为 i 的数据流, 其被抽样到的概率为 $1 - \prod_{j=1}^i (1 - P(j))$, 其中, $P(j)$ 由公式(2)确定. 于是, 对于任意一条数据流, 其被抽样到的概率, 也即流抽样比为

$$p_f = \sum_{i=1}^M P(f=i) \left(1 - \prod_{j=1}^i (1 - P(j)) \right) = \sum_{i=1}^M \frac{\theta}{i^z} \left(1 - \prod_{j=1}^i (1 - P(j)) \right),$$

其中, $\theta = \left(\sum_{j=1}^M j^{-z} \right)^{-1}$. 证毕. \square

定理 1 为在实际应用公平抽样算法时, 在抽样比和估计误差之间进行合理权衡提供了理论依据. 图 1(a) 为 $z=1.5$ 时**, 不同 M 和 ε 值下公平抽样的包抽样比(图 1 中, 为估计误差, 标注“Uni.”的曲线采用均匀随机抽样). 由图 1(a) 可见, 当 M 值较大时, 只需较低的包抽样比就能保证较小的估计误差. 例如, 当 $M=10^5$ 时, 包抽样比 $p_p=0.06$, 而估计误差为 $\varepsilon=0.1$. 图 1(b) 为不同 M 和 ε 值下公平抽样的流抽样比, 以及在相同包抽样比下均匀随机抽样的流抽样比(当以概率 p 进行包抽样时, 均匀随机抽样的流抽样比为 $\left(\sum_{j=1}^M j^{-z} \right)^{-1} \sum_{j=1}^M \frac{(1-(1-p)^j)}{j^z}$, 详见第 3.2 节定理 3 的证明过程). 由图 1(b) 可见, 在相同的包抽样比下, 公平抽样比均匀随机抽样能够抽取到更多的数据流. 例如, 当 $\varepsilon=0.1$

** 通过对实际采集的骨干网流量数据的分析表明, 参数 z 的取值在 1.5~2 之间^[22]. 本文第 4 节中对两组分别采集自 OC-48 和 OC-192 骨干网链路的流量数据的分析结果也验证了这一点.

且 $M=10^5$ 时,公平抽样能够抽取到 99.6%的数据流;而此时,在相同的包抽样率下,均匀随机抽样仅能抽取到 30.9%的数据流.

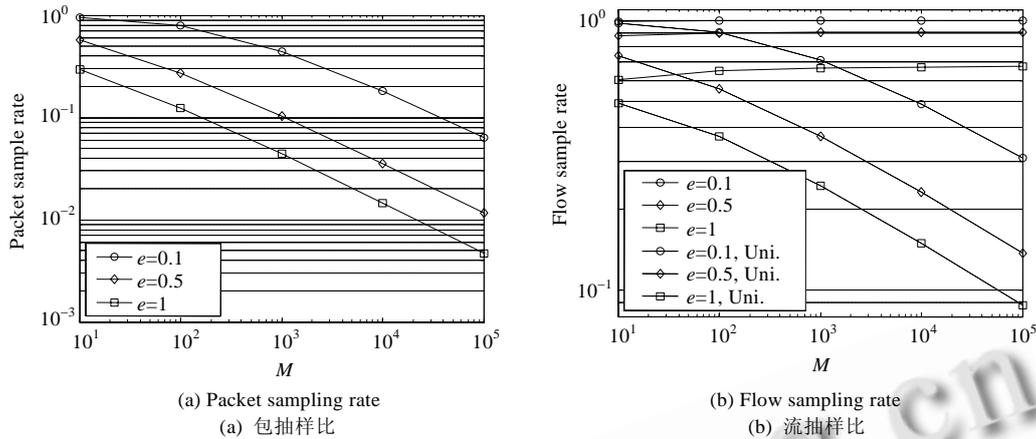


Fig.1 Sampling rate under various M ($z=1.5$)

图 1 $z=1.5$ 时,抽样比和 M 值的关系

3 流量的多解析度抽样估计算法

由公式(2)、公式(3)可见,公平抽样在决定每数据包的抽样比时,需要知道该数据包所属数据流的当前流量.在高速骨干网上,准确地获知每条数据流的流量代价极高^[2-5].文献[12]采用简单哈希表对各条数据流的流量作近似统计.由于存在哈希冲突,统计的结果存在一定误差,特别是对于短流而言,测量误差相对更大.为了减小统计误差,则需要增加哈希表空间,这又增大了实现代价,降低了部署灵活性.

本文提出了多解析度抽样估计算法用于流量的近似估计.多解析度抽样估计算法由 n 个独立的抽样统计器 C_1, C_2, \dots, C_n 构成,如图 2 所示.第 i ($1 \leq i \leq n$) 个抽样统计器 C_i 以概率 $p_i = P^i$ (P 为抽样常数且有 $0 < P < 1$) 对链路流量进行抽样,并统计各条数据流抽样后的流量.各个抽样统计器的计数区间为 $[0, L+1]$,计数值 $L+1$ 表示计数器的溢出状态(overflow, 记作 OV).在更新计数器时,若某条数据流的流量等于 OV ,则其计数值维持在 OV 而不再增加.在查询某条数据流 F 的流量时,根据 n 个抽样统计器的统计结果(设为 $c_1^F, c_2^F, \dots, c_n^F$),通过某种估计算法得出 F 的流量,即 $\hat{f} = estimate(c_1^F, c_2^F, \dots, c_n^F)$.由于查询过程是一个逐包操作,因此需要采用较为简单的估计算法.由公式(1)可知,流量大小一定时,估计误差和抽样比成反比.因此,我们选择所有有效计数值中抽样比最大的那个作为最终的有效计数,即 $\hat{f} = c_j^F / p_j$, 其中, $j = \arg \min_i (c_i^F \in [0, L])$.

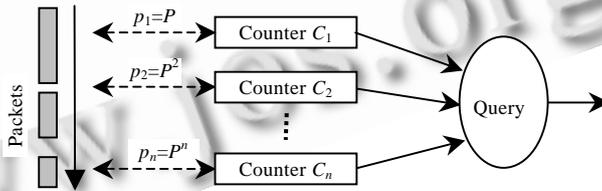


Fig.2 Multi-Resolution sampling and estimation

图 2 多解析度抽样统计

抽样统计器 C_1, C_2, \dots, C_n 采用 d -left 哈希表实现. d -left 哈希表的优点是空间效率高^[17,23].采用 d -left 哈希表实现各层统计器,可以在保证空间效率较高的前提下哈希冲突的概率足够低,以至于和抽样估计的误差相比,哈希冲突引起的误差可以忽略不计.因此,第 3.1 节在分析多解析度抽样估计算法的估计误差时仅考虑抽样估计

的误差,而忽略 d -left 哈希表的哈希冲突引起的误差.

3.1 估计误差分析

如上文所述,公平抽样依赖于对数据流流量的实时准确估计.因此,流量估计的误差对于公平抽样的性能有着重要影响.本节首先通过合理的假设对多解析度抽样统计器进行了理论建模,进而从理论上分析了多解析度抽样估计的估计误差上限;然后,将理论误差上限和通过仿真实验得到的估计误差进行了比较.

在分析多解析度抽样估计的误差之前,首先给出有效覆盖区间的定义.

定义 1(有效覆盖区间). 若抽样统计器 C_1, C_2, \dots, C_n 的计数范围为 $[0, L+1]$, 则定义 $\omega_i = \left[\frac{1}{p_i}, \frac{L}{p_i} \right] = P^{-i}[1, L]$ 为 C_i 的有效覆盖区间.

考虑到相邻的若干个抽样统计器的覆盖区间可能会部分重叠,定义交叠系数以刻画相邻若干个覆盖区间的重叠程度.

定义 2(交叠系数). $\forall i \geq 1, j \geq 2$, 若 $\omega_i \cap \omega_{i+j-1} \neq \emptyset$ 且 $\omega_i \cap \omega_{i+j} = \emptyset$, 则称 j 为交叠系数.

由公式(1)可知,当流量大小 f 一定时,估计误差和抽样比 p 成反比.由于相邻的 j 个覆盖区间存在交叠,某一数据流可能在多层抽样统计器中存在有效计数,为了便于分析,下面给出最佳抽样假设.

最佳抽样假设. 设多解析度抽样统计器共包含 n 层统计器 $C_1, C_2, \dots, C_n, C_1$ 的抽样比为 $p_i = P^{i-1}$ ($0 < P < 1, 1 \leq i \leq n$), 将多解析度抽样统计器的覆盖区间 $[1, Lp_n^{-1}]$ 划分为 n 个子区间 $\omega_1, \omega_2, \dots, \omega_n$, 即 $\omega_1 = [1, Lp_1^{-1}], \omega_2 = (Lp_1^{-1}, Lp_2^{-1}], \omega_3 = (Lp_2^{-1}, Lp_3^{-1}], \dots, \omega_n = (Lp_{n-1}^{-1}, Lp_n^{-1}]$. 最佳抽样假设限定对于任意 $f \in \omega_i, f$ 的抽样比为 p_i .

最佳抽样假设实际上是假定了对于任意数据流,总是依据抽样比最小且统计结果未溢出的那个抽样统计器的统计结果来估计其流量.这一假设和多解析度抽样估计器的查询过程所采用的估计算法是一致的.引理 1 给出了在最佳抽样假设下,当交叠系数为 j 时,任意覆盖区间 $\omega_i (i \geq j+1)$ 内估计误差的标准差的上界.

引理 1. 设交叠系数为 $j, \forall f \in \omega_i (i \geq j), f$ 的估计误差标准差 $Std(\hat{f})$ 的上界为 $\sqrt{\frac{1-P^j}{LP}}$.

证明: 设数据流 F 的流量大小为 f , 以概率 p_i 抽样后,抽取到的数据包数为 f'_i . 取 $\hat{f}_i = f'_i / p_i$ 为 f 的估计值. 记 $Std^{(i)}(f) = Std(\hat{f}_i)$, 表示流量大小为 f 的数据流经抽样概率为 p_i 的抽样估计后,其估计误差的标准差. 由公式(1)可见,估计误差的标准差是 f 和 p_i 的单调减函数. 考虑到交叠系数为 j , 于是将区间 ω_i 分为 j 段, 即 $\omega_i = \omega_i^1 \cup \omega_i^2 \cup \dots \cup \omega_i^j$, 其中 $\omega_i^1 = [p_i^{-1}, Lp_{i-j+1}^{-1}], \omega_i^2 = [Lp_{i-j+1}^{-1}, Lp_{i-j+2}^{-1}], \dots, \omega_i^j = [Lp_{i-1}^{-1}, Lp_i^{-1}]$. 于是

$$Std^{(i-j+1)}(f) \leq Std^{(i-j+1)}(p_i^{-1}), \forall f \in \omega_i^1 \quad (4)$$

$$Std^{(i-j+s)}(f) \leq Std^{(i-j+s)}(Lp_{i-j+s-1}^{-1}), \forall f \in \omega_i^s, 2 \leq s \leq j \quad (5)$$

将公式(1)分别带入公式(4)和公式(5),得到

$$Std^{(i-j+1)}(f) \leq \sqrt{P^{j-1}(1-p_{i-j+1})}, f \in \omega_i^1 \quad (6)$$

$$Std^{(i-j+s)}(f) \leq \sqrt{\frac{1-p_{i-j+s}}{LP}}, f \in \omega_i^s, 2 \leq s \leq j \quad (7)$$

观察公式(6)和公式(7)的右侧表达式可知, $f \in \omega_i$, $Std(\hat{f})$ 的最大值为 $\max\left(\sqrt{P^{j-1}(1-p_{i-j+1})}, \sqrt{\frac{1-p_i}{LP}}\right)$. 由于 $j \geq 2$, 故 $p_{i-j+1} > p_i$, 于是

$$\left(\sqrt{P^{j-1}(1-p_{i-j+1})} / \sqrt{\frac{1-p_i}{LP}}\right) < \left(\sqrt{P^{j-1}(1-p_i)} / \sqrt{\frac{1-p_i}{LP}}\right) = \sqrt{LP^j} \quad (8)$$

又因为交叠系数为 j , 故 $P^{1-j} < L < P^{-j}$, 即有 $\sqrt{LP^j} < 1$, 因此有

$$\left(\frac{\sqrt{P^j(1-p_{i-j+1})}}{\sqrt{\frac{1-p_i}{LP}}} \right) < 1 \tag{9}$$

因此, $f \in \omega_i$, $Std(\hat{f})$ 的最大值为 $\sqrt{\frac{1-p_i}{LP}}$. □

定理 2. 设交叠系数为 j , 则 $f \in [1, P^j]$, 多解析度抽样估计的估计误差标准差的上界为 $\sqrt{(1-P)/P}$; $f \in [P^j, +\infty)$, 多解析度抽样估计的估计误差标准差的上界为 $1/\sqrt{LP}$.

定理 2 可以由引理 1 很方便地证明, 详细证明过程从略. 图 3 给出, 当 $L=30, P=0.25$ 时, 多解析度抽样估计的估计误差标准差的实验值和理论上限. 为便于观察, 右侧图为左侧图的局部放大. 图 3 中, Upper Bound 1 为 $\sqrt{(1-P)/P}$, Upper Bound 2 为 $1/\sqrt{LP}$. 由图 3 可见, 多解析度抽样统计器的估计误差的实验结果小于定理 2 所得出的理论上限. 由定理 2 可知, 为了减小多解析度抽样估计的估计误差, 需要增大 L 和 P . 但是, 增大 L 和 P 的同时也会增加算法的时间和空间复杂度. 因此, 在实际使用过程中, 应该根据具体需求, 在估计误差和复杂度之间进行权衡, 以选择合适的参数值.

定理 2 所得出的结论是建立在理想哈希函数的假设之下的. 而在实际中, 构建理想哈希函数的代价极高. 我们选择 d -left 哈希函数^[17]来构建几乎理想的哈希函数, 并考察了 d -left 哈希表的参数对于估计误差标准差的影响, 如图 3 所示. $L=30, P=0.25$, d -left 哈希表的选择数为 4, 桶深为 4, 额外深度为 1, 元素指纹的长度为 $Lf=10$. 从图 3 可见, 当元素指纹长度为 $Lf=10$ 时, 哈希冲突对于估计误差的影响即可忽略. 下文若不作特殊说明, d -left 哈希表的参数均如此设置.

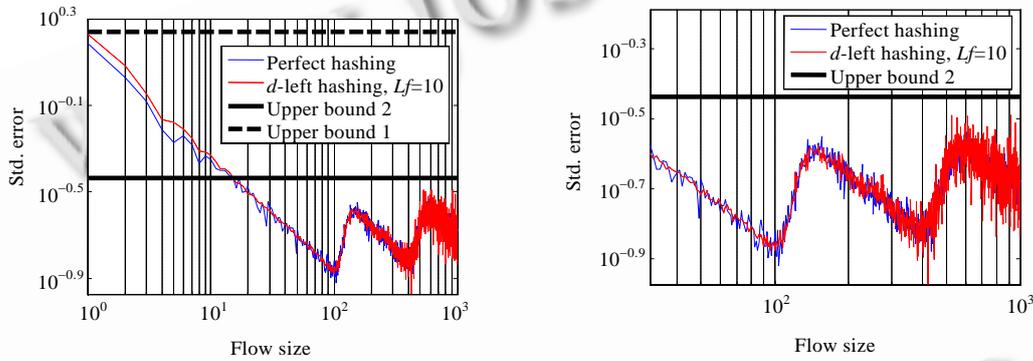


Fig.3 Impact of hash collision on the estimation error of multi-resolution sampling

图 3 哈希冲突对多解析度抽样估计的估计误差标准差的影响

3.2 空间复杂度分析

多解析度抽样统计器的层数 n 由最大流量 M 以及多解析度抽样统计器的参数 P 和 L 决定. 理想情况下, 当 $M \leq P^n L$ 时, 最高层的抽样统计器不会溢出, 故 n 的取值为

$$n = \lceil \log_P L - \log_P M \rceil \tag{10}$$

多解析度抽样统计器所需的存储空间为 n 层抽样统计器所需的存储空间之和. 定理 3 给出了多解析度抽样统计器的空间复杂度.

定理 3. 设测量周期内链路上共有 N 条并发流, 流量大小在区间 $[1, M]$ 内服从参数为 z 的 Zipf 分布. 又设多解析度抽样统计器的各统计层由理想哈希函数构成, 各层的有效计数区间为 $[0, L]$, 第 i 层的抽样比为 P^i , 则多解析度抽样统计器所需空间为

$$S = \theta N \log_2(L+1) \sum_{i=1}^n \sum_{j=1}^M \frac{1-(1-P^i)^j}{j^z}$$

其中, $\theta = \left(\sum_{j=1}^M j^{-z} \right)^{-1}$, n 由公式(10)确定.

证明:当以概率 p 进行包抽样时,对于流量为 f 的数据流,至少有一个数据包被抽样到的概率为 $1-(1-p)^f$.因此,当以概率 p 进行包抽样时,对于任意数据流,至少有一个数据包被抽样到的概率为 $\sum_{j=1}^M P(F=j) \cdot (1-(1-p)^j)$, 其中, $P(F=j)$ 为对于任意数据流,其流量大小为 $j(1 \leq j \leq M)$ 的概率.因此,第 i 层抽样统计器所需的空间为

$$S_i = N \log_2(L+1) \sum_{j=1}^M P(F=j) \cdot (1-(1-p_i)^j) \tag{11}$$

由于流量大小在区间 $[1, M]$ 内服从参数为 z 的 Zipf 分布,故

$$P(F=j) = \frac{\theta}{j^z} \tag{12}$$

其中, $\theta = \left(\sum_{j=1}^M j^{-z} \right)^{-1}$ 为归一化系数.于是, n 层抽样统计器所需空间之和为

$$S = \sum_{i=1}^n S_i = \theta N \log_2(L+1) \sum_{i=1}^n \sum_{j=1}^M \frac{1-(1-P^i)^j}{j^z} \tag{13}$$

证毕. □

定理 3 中假设多解析度抽样统计器的各统计层由理想哈希函数构成.考虑到我们采用 d -left 哈希函数实现几乎理想的哈希函数,因此每元素所需空间为 $\log_2(L+1)+10$,增加的 10 比特为 d -left 哈希函数元素指纹的长度.因此,SEFS 算法的空间复杂度为 $S = \sum_{i=1}^n S_i = \theta N (\log_2(L+1)+10) \sum_{i=1}^n \sum_{j=1}^M \frac{1-(1-P^i)^j}{j^z}$.易知,SGS 算法所需空间复杂度为 $S' = M \log_2(M+1)$.表 1 为 $z=1.5, M, L, P$ 取几种典型值时,SEFS 和 SGS 算法所需的空间之比.

Table 1 Ratio of space complexity of SEFS to that of SGS

表 1 SEFS 和 SGS 算法所需空间之比

	$M=10^3$	$M=10^4$	$M=10^5$	$M=10^6$
$L=62, P=1/8$	0.88	0.71	0.62	0.53
$L=126, P=1/16$	0.50	0.46	0.41	0.35
$L=62, P=1/16$	0.56	0.43	0.38	0.33
$L=126, P=1/32$	0.35	0.31	0.26	0.23

3.3 时间复杂度分析

我们采用每数据包所需的访存次数作为衡量算法时间复杂度的性能指标.在高速链路中,要求每数据包所需的访存操作的次数尽量少.由于各层抽样统计器彼此独立地进行统计,因此每一个抽样统计器可以采用独立的存储单元实现.对于每一个抽样统计器 C_i 而言,每数据包所需的访存操作为一次读操作和一次以概率 p_i 进行的更新操作.更新操作可看作是一次读操作和一次写操作.考虑到每个抽样统计器的计数上限为 $L+1$,因此达到计数上限后,更新操作不需要进行写操作而仅仅包含读操作.于是,对于每数据包而言,抽样统计器 C_i 所需的访存次数小于 $1+2p_i$.因此,多解析度抽样统计器的时间复杂度小于 $O\left(n + \frac{2P(1-P^n)}{1-P}\right)$,其中, n 由公式(10)确定.虽然本文提出的多解析度抽样统计器的时间复杂度高于文献[12]采用的简单哈希函数,但是由于各统计层的并行性和空间高效性,可以利用多个片内的小容量高速存储器实现多解析度抽样统计器,因而从实现的角度而言是完全可行的(详细的实现可行性分析见第 5 节).

4 仿真实验

包抽样的结果可以用于估计各条数据流的流量、估计数据流的流量分布、长流检测以及业务流分类等^[12].

本文通过仿真实验,分析了当抽样结果用于流的流量估计以及长流检测时 SEFS 算法的性能,并将 SEFS 算法和 SGS 算法的性能进行了比较.首先,我们采用模拟流量数据,在数据流的流量估计的背景下考察了抽样常数 P 和计数上限 L 对于 SEFS 算法性能的影响.然后,我们采用实际采集的骨干网流量数据,在流的流量估计以及长流检测的应用背景下考察 SEFS 算法的性能,并与 SGS 算法进行比较.

衡量流的流量估计的性能指标测量误差的标准差(Std. error).假设在测量周期内,链路上共有 N 条并发流 $F_i(1 \leq i \leq N)$, F_i 的流量大小为 $f_i(1 \leq f_i \leq M)$, f_i 的估计值为 \hat{f}_i .

$$\text{令 } e_i = \frac{|\hat{f}_i - f_i|}{f_i}, E_j = \{e_i | f_i = j\} (j \in [1, M]), \text{定义流量大小为 } i \text{ 的数据流的测量误差的标准差为 } \bar{e}_j = \frac{1}{|E_j|} \cdot \sum_{e_i \in E_j} e_i.$$

衡量长流检测的性能指标有识别率、误判率以及平均流量测量误差.定义流量大于 T 的数据流为长流.识别率是指在所有的真实流量大于 T 的数据流中,其测量值也大于 T 的数据流所占的比例;误判率是指在流量的测量值大于 T 的数据流中,真实流量小于 T 的数据流所占的比例;平均流量测量误差是指所有被成功识别的长流的流量测量误差的标准差的平均值.

4.1 模拟数据仿真

我们首先采用人工产生的流量数据进行仿真实验.设流量服从参数为 z 的 Zipf 分布,共有 N 条并发数据流,每条数据流的取值区间为 $[1, M]$.实验中,取 $z=1.5, M=1000, N=200000$.仿真程序重复运行 100 次,取其平均值.

图 4 为 ϵ 分别取 0.1, 0.5, 1 和 2 时, SEFS 和 SGS 算法用于流的流量估计时的测量误差标准差.

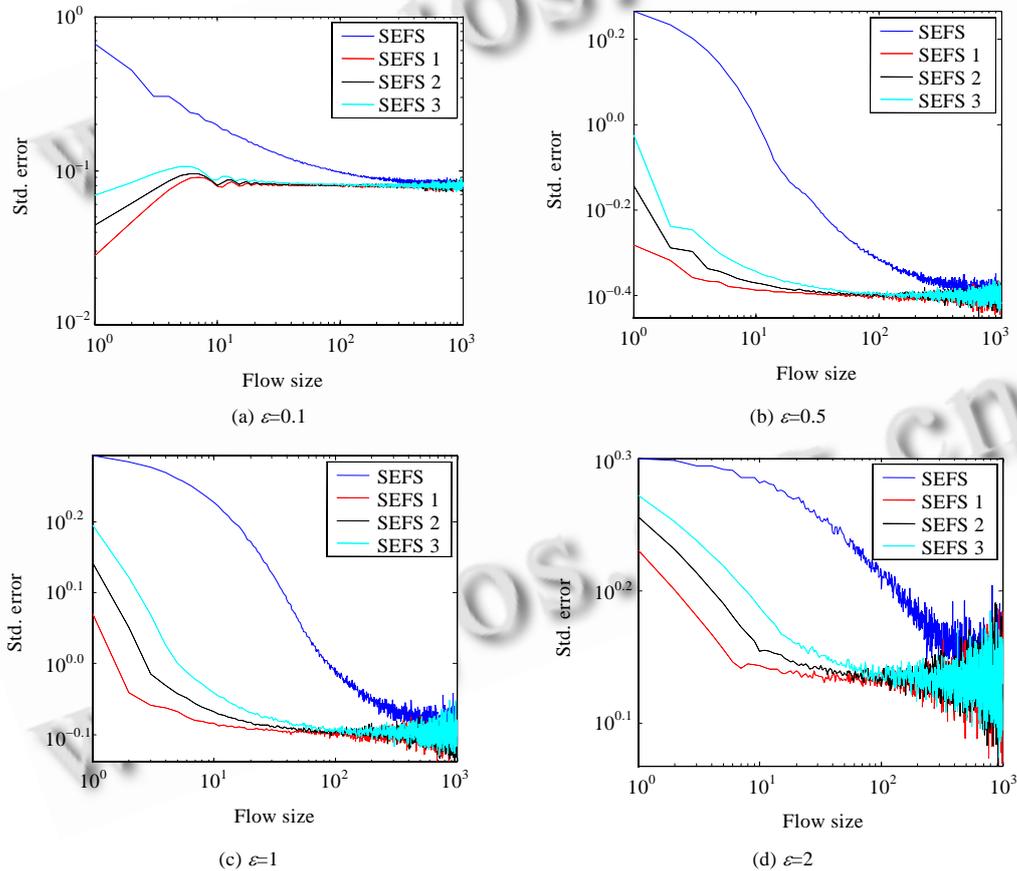


Fig.4 Standard error of flow traffic estimation using SEFS and SGS algorithm, with synthetical traffic trace

图 4 采用模拟流量数据时,利用 SEFS 和 SGS 算法进行流量估计的误差标准差

图 4 中,SEFS 1~SEFS 3 的参数设置分别为 $P=1/4, L=30; P=1/8, L=62; P=1/16, L=126$.从图 4 可见,对于短流,SEFS 的测量误差明显较 SGS 算法要小;对于长流,两种算法的测量误差较为接近,SEFS 的测量误差略小.由图 4 还可以看出,抽样常数越小,SEFS 的测量误差越大.但是,即使当抽样常数 $P=1/16$ 时,SEFS 算法的测量误差仍然小于 SGS 算法.

由于 SGS 算法采用简单哈希对流量进行估计,因此 SGS 算法的估计误差仅仅来源于 Hash 冲突,此误差随着流量的增大而递减;SEFS 算法采用多解析度抽样结合 d -left 哈希进行流量估计,其估计误差如图 3 所示.对于短流,简单哈希的估计误差显然要大于 $Lf=10$ 时 d -left 哈希的估计误差,因此,此时 SGS 算法的测量误差要明显大于 SEFS 算法.对于长流,虽然简单哈希的估计误差会接近甚至小于多解析度抽样统计器的估计误差,但是简单哈希的估计误差是单边的,即估计值肯定大于真实值,从而导致 SGS 算法抽样不足;而多解析度抽样统计器的估计误差却是双边的,流量估计值可能大于也可能小于真实值,于是过度抽样和抽样不足带来的结果相抵消,从而产生较小的测量误差.

4.2 真实流量数据仿真

实验所用的数据来源于 NLANR 公布的在 OC-48 和 OC-192 骨干网链路上采集的流量数据^[24].我们截取了 Abilene-I 数据集中 IPLS-KSCY-20020814-090000 数据文件的第 1 分钟和第 2 分钟的流量数据(下文分别称为 Trace 1 和 Trace 2)以及 Abilene-III 数据集中 20040601-193121-0 数据文件的第 1 分钟和第 2 分钟的流量数据(下文分别称为 Trace 3 和 Trace 4)作为实验数据源.采用五元组作为流标识.通常,流超时时间设置为 1 分钟^[3-5].本文中,由于所采用的流量数据均只持续一分钟,故没有考虑流的超时结束问题.Trace 1~Trace 4 的详细信息见表 2.

Table 2 Detailed information of Trace 1~Trace 4

表 2 Trace 1~Trace 4 的详细信息

Trace ID	Link type	Measurement interval	Number of packets	Data rate (Packets/S)	Number of flows	Length of longest flow (Packets)
Trace 1	OC-48	1 minute	3 541 483	59 025	45 807	112 008
Trace 2			3 526 723	58 779	43 347	76 954
Trace 3	OC-192		6 410 992	106 850	172 004	459 668
Trace 4			6 313 112	105 219	175 985	432 863

从第 4.1 节模拟数据仿真实验的结果来看,当设置抽样常数 $P=1/16$ 时,SEFS 算法的估计误差仍然小于 SGS 算法.因此,本节中我们设置 SEFS 算法的参数为 $P=1/16, L=126$.此外,SEFS 算法和 SGS 算法期望的估计误差均设置为 $\epsilon=0.5$.

4.2.1 流的流量估计

图 5 为将 SEFS 和 SGS 算法应用于数据流的流量估计时估计误差的标准差.图 5(a)~图 5(d)分别对应于采用 Trace 1~Trace 4 时的实验结果.由图 5 可见,采用真实流量数据仿真所得到的结果和图 4 中模拟数据所得结果类似,对于短流,SEFS 算法的测量误差明显小于 SGS 算法;对于长流,两者的测量误差较为接近.

当采用 Trace 1 和 Trace 2 时,SEFS 算法和 SGS 算法所需空间之比分别为 0.345 和 0.346;采用 Trace 3 和 Trace 4 时,SEFS 算法和 SGS 算法所需空间之比分别为 0.308 和 0.312.对比表 1 第 2 行可以发现,采用真实流量数据时,SEFS 算法的空间复杂度略低于 $z=1.5$ 的理想情况.这是因为当采用 Zipf 分布来拟合实际流量数据时,参数 z 的取值介于 1.5 和 2 之间,接近 1.8.又由公式(13)可知,在 M, P, L 一定时,SEFS 算法的空间复杂度随着 z 的增大而降低.因此在实际应用中,根据 $z=1.5$ 的理想结果来预先设定 SEFS 算法的存储空间会略有富余.

4.2.2 长流检测

流量工程等网络管理应用仅仅需要关注链路上的巨型流.因此,长流检测以及长流的流量测量成为公平抽样的一个重要应用.用于长流检测时,公平抽样的抽样函数调整为^[12]

$$P'(i) = \begin{cases} 0, & i < T \\ 1/(1 + \epsilon^2(2i - 1)), & i \geq T \end{cases} \quad (14)$$

表 3~表 6 为 $\varepsilon=0.5, P=1/16, L=126, T$ 分别取 8,16,32 时,SGS 和 SEFS 算法长流检测的性能比较.从表 3~表 6 可以看出,SEFS 算法的识别率高于 SGS 算法,但是 SEFS 算法的误判率明显比 SGS 算法要低.这主要是因为 SGS 算法采用简单哈希函数估计流量,由于哈希冲突,流量的估计值不小于其真实值.特别是对于短流,流量估计的相对误差较大,导致大量短流被误判为长流,从而增加了误判率.

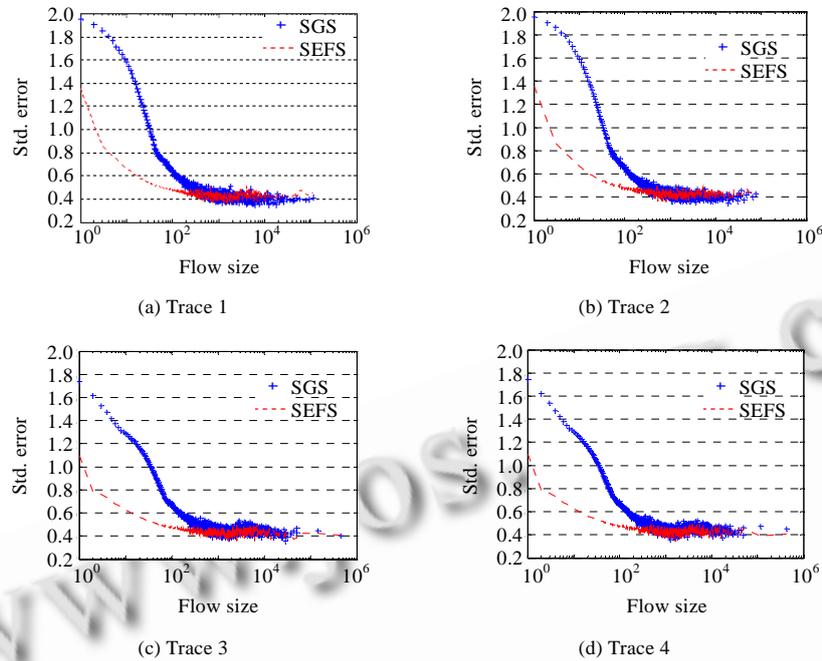


Fig.5 Standard error of flow traffic estimation using SEFS and SGS algorithm, with real-world backbone network traffic trace

图 5 采用真实骨干网流量数据时,利用 SEFS 和 SGS 算法进行流量估计的误差标准差

Table 3 Performance comparison of large flow identification of SGS and SEFS, with real-world Trace 1

表 3 采用 Trace 1 时,SGS 和 SEFS 算法长流检测的性能比较

	SGS			SEFS		
	$T=8$	$T=16$	$T=32$	$T=8$	$T=16$	$T=32$
Hit ratio	0.590 8	0.710 6	0.825 8	0.802 9	0.780 1	0.823 9
False positive ratio	0.271 5	0.351 4	0.450 3	0.085 4	0.042 7	0.077 5
Mean estimation error of the size of long flows	0.481 7	0.475 7	0.465 1	0.441 1	0.440 6	0.438 7

Table 4 Performance comparison of large flow identification of SGS and SEFS, with real-world Trace 2

表 4 采用 Trace 2 时,SGS 和 SEFS 算法长流检测的性能比较

	SGS			SEFS		
	$T=8$	$T=16$	$T=32$	$T=8$	$T=16$	$T=32$
Hit ratio	0.589 2	0.712 7	0.817 1	0.806 0	0.803 8	0.833 0
False positive ratio	0.259 9	0.337 9	0.426 1	0.076 8	0.058 3	0.074 1
Mean estimation error of the size of long flows	0.485 7	0.479 3	0.469 5	0.441 7	0.441 2	0.438 9

Table 5 Performance comparison of large flow identification of SGS and SEFS, with real-world Trace 3**表 5** 采用 Trace 3 时,SGS 和 SEFS 算法长流检测的性能比较

	SGS			SEFS		
	$T=8$	$T=16$	$T=32$	$T=8$	$T=16$	$T=32$
Hit ratio	0.538 4	0.649 4	0.731 3	0.744 9	0.751 4	0.785 8
False positive ratio	0.404 4	0.356 1	0.281 9	0.090 3	0.068 9	0.020 0
Mean estimation error of the size of long flows	0.492 5	0.489 4	0.483 3	0.437 3	0.436 3	0.439 3

Table 6 Performance comparison of large flow identification of SGS and SEFS, with real-world Trace 4**表 6** 采用 Trace 4 时,SGS 和 SEFS 算法长流检测的性能比较

	SGS			SEFS		
	$T=8$	$T=16$	$T=32$	$T=8$	$T=16$	$T=32$
Hit ratio	0.557 1	0.666 7	0.739 1	0.758 5	0.771 0	0.789 9
False positive ratio	0.426 3	0.350 1	0.256 4	0.100 2	0.062 6	0.017 3
Mean estimation error of the size of long flows	0.494 9	0.491 0	0.485 6	0.436 4	0.436 8	0.438 2

5 实现的考虑

随着可编程硬件技术,如 FPGA(field-programmable gate array)的飞速发展,硬件可重构将成为网络设备将来的发展趋势.可重构技术使得将流量测量与分析功能以组件的形式集成到网络设备中去成为可能.当前,Cisco 的 NetFlow^[2]以软件组件的形式提供给用户.软件组件的缺陷是性能较差,而基于可重构硬件平台的硬件组件在具备软件组件的部署灵活性的同时仍然能够保证较高的性能.FPGA 平台的特点是其片内集成的高速存储器的总容量小但并行度高.SGS 算法由于空间复杂度,无法基于 FPGA 实现.相反地,SEFS 算法采用多个并行的小容量存储器实现流量估计,与 SGS 算法相比,SEFS 算法在时间复杂度和空间复杂度之间进行了新的权衡,特别适于 FPGA 实现.基于 FPGA 实现的 SEFS 公平抽样算法可以以硬件组件的形式集成到网络设备中去,在具备高性能的同时,仍然具有高度的部署灵活性.本节讨论了基于 FPGA 的 SEFS 算法的实现方法.

由于多解析度抽样统计器的各层均需要线速读取,因此在实现时需采用多端口存储器^[25].考虑到各层独立地对流量进行抽样统计,于是也可以采用多个独立的单端口存储器来实现多解析度抽样统计器.例如,假设测量周期内共有 $N=10^6$ 条并发数据流,最大流量为 $M=10^6$,多解析度抽样统计器抽样常数取 $P=1/16$,各层的计数上限取 $L=126$,则由公式(10)可知,多解析度抽样估计器的层数为 $n=4$.由表 1 可知,SEFS 算法所需空间为

$$0.35 \times \lceil \log_2(M+1) \rceil \times 10^6 = 7\text{Mbits}.$$

各层抽样统计器采用 d -left 哈希函数实现, d -left 哈希函数在每次查找时需要访问 $d \times (b_m + b_e)$ 次存储器.其中, d 为并行哈希函数的个数, b_m 为平均桶深, b_e 为额外桶深.本文取 $d=4, b_m=4, b_e=1$,于是每次查找需要访问 20 次存储器,因此仍然需要采用多端口存储器或者多个独立的存储器以实现 d -left 哈希函数.考虑到 d -left 哈希函数的查找过程是并行进行的以及 FPGA 片内存储单元并行度高的特点,因此可以采用 20 个独立的存储器实现.这样, d -left 哈希函数每次查找操作只需串行地访问 1 次存储器.在 OC-768 链路上,设满速率传输包长为 64 字节的数据包,则包到达间隔为 12.5ns.于是,采用工作时钟频率为 80MHz 的 SRAM(static random access memory)存储器即可满足性能需求.

综上,针对 OC-768 链路实现多解析度抽样估计器共需要 80 个独立的存储单元,每个存储单元的工作时钟频率不小于 80MHz,80 个独立的存储单元的总容量为 7Mbits.对于今天的可编程器件而言,其片内的 SRAM 资源完全可以满足上述需求.例如,Xilinx 的 Virtex-5 系列的 XC5VFX200T 含有 912 个单片容量为 18Kbits 的 BlockRAM,总空间为 16.416Mbits,最高工作时钟频率可达到 550MHz^[26].可见,基于当前 FPGA 技术实现的 SEFS 算法可以支持 OC-768 链路中百万条并发数据流的公平抽样.

6 结 论

本文提出了一种空间高效的数据包公平抽样算法(SEFS).与已有的公平抽样算法相比,SEFS 算法具有更低的空间复杂度和更好的性能.采用实际采集的骨干网流量数据对 SEFS 算法的流量测量和长流检测的性能进行了实验验证,结果表明,SEFS 算法在空间复杂度仅为现有算法 35%的前提下,仍具有更好的测量与检测性能.虽然 SEFS 算法的时间复杂度比现有算法要高,然而 SEFS 算法在时间和空间复杂度之间所进行的新的权衡使之特别适于 FPGA 实现,从而得以以硬件组件的形式集成到网络设备中去,在具备高性能的同时也具有高度的部署灵活性.下一步的研究工作将考察 SEFS 算法在流量异常检测以及业务分类等应用中的性能.

References:

- [1] Claffy KC, Polyzos GC, Braun HW. Application of sampling methodologies to network traffic characterization. *ACM SIGCOMM Computer Communication Review*, 1993,23(4):194–203. [doi: 10.1145/167954.166256]
- [2] Cisco System White Paper. NetFlow services solutions guide. 2007. <http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/netflsol/nfwhite.pdf>
- [3] Estan C, Varghese G. New directions in traffic measurement and accounting. *ACM SIGCOMM Computer Communication Review*, 2002,32(4):323–336. [doi: 10.1145/964725.633056]
- [4] Kodialam M, Lakshman TV, Mohanty S. Runs bAsed traffic estimator (RATE): A simple, memory efficient scheme for per-flow rate estimation. In: *Proc. of the IEEE INFOCOM 2004, Vol.3*. Washington: IEEE Computer Society Press, 2004. 1808–1818. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1354591
- [5] Raspall F, Sallent S, Yufera J. Shared state sampling. In: *Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement*. New York: ACM Press, 2006. 1–14. <http://portal.acm.org/citation.cfm?id=1177082>
- [6] Brauckhoff D, Tellenbach B. Impact of packet sampling on anomaly detection metrics. In: *Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement*. New York: ACM Press, 2006. 159–164. <http://portal.acm.org/citation.cfm?id=1177101>
- [7] Mai JN, Chua CN. Is sampled data sufficient for anomaly detection. In: *Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement*. New York: ACM Press, 2006. 165–176. <http://portal.acm.org/citation.cfm?id=1177080.1177102>
- [8] Chen G, Gong J, Ding W. A real-time anomaly detection model based on sampling measurement in a high-speed network. *Journal of Software*, 2003,14(3):594–599 (in Chinese with English abstract). http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no=20030340&journal_id=jos
- [9] Hohn N, Veitch D. Inverting sampled traffic. *IEEE/ACM Trans. on Networking (TON)*, 2006,14(1):68–80. [doi: 10.1109/TNET.2005.863456]
- [10] Duffield N, Lund C, Thorup M. Properties and prediction of flow statistics from sampled packet streams. In: *Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurement*. New York: ACM Press, 2002. 159–171. <http://portal.acm.org/citation.cfm?id=637201.637225>
- [11] Barakat C, Iannaccone G, Diot C. Ranking flows from sampled traffic. In: *Proc. of the 2005 ACM Conf. on Emerging network experiment and Technology*. New York: ACM Press, 2005. 188–199. <http://portal.acm.org/citation.cfm?id=1095921.1095947>
- [12] Kumar A, Xu J. Sketch guided sampling—Using on-line estimates of flow size for adaptive data collection. In: *Proc. of IEEE Infocom 2006*. Washington: IEEE Computer Society Press, 2006. 1–11. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4146979
- [13] Shah D, Iyer S, Prahakar B, McKeown N. Maintaining statistics counters in router line cards. *IEEE Micro*, 2002,22(1):76–81. [doi: 10.1109/40.988692]
- [14] Ramabhadran S, Varghese G. Efficient implementation of a statistics counter architecture. *ACM SIGMETRICS Performance Evaluation Review*, 2003,31(1):261–271. [doi: 10.1145/885651.781060]
- [15] Zhao Q, Xu J, Liu Z. Design of a novel statistics counter architecture with optimal space and time efficiency. *ACM SIGMETRICS Performance Evaluation Review*, 2006,34(1):323–334. [doi: 10.1145/1140103.1140314]
- [16] Kumar K, Xu J, Jia W, Spatschek O, Li L. Space-Code bloom filter for efficient per-flow traffic measurement. In: *Proc. of the IEEE INFOCOM 2003*. Washington: IEEE Computer Society Press, 2003. 167–172. <http://portal.acm.org/citation.cfm?id=948226>

- [17] Bonomi F, Mitzenmacher M, Panigrahy R, Singh S, Varghese G. An improved construction for counting bloom filters. In: Proc. of the 14th Conf. on Annual European Symp. New York: ACM Press, 2006. 684–695. <http://portal.acm.org/citation.cfm?id=1276191.1276252>
- [18] Duffield N, Lund C, Thorup M. Learn more, sample less: Control of volume and variance in network measurement. IEEE Trans. on Information Theory, 2005,51(5):1756–1775. [doi: 10.1109/TIT.2005.846400]
- [19] Estan C, Keys K. Building a better NetFlow. ACM SIGCOMM Computer Communication Review, 2004,34(4):245–256. [doi: 10.1145/1030194.1015495]
- [20] Wang JF, Yang JH, Zhou HX, Xie GG, Zhou MT. Adaptive sampling methodology in network measurements. Journal of Software, 2004,15(8):1227–1236 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/15/1227.htm>
- [21] Kong SJ, He T, Shao XX, Li X. Time-Out bloom filter: A new sampling method for recording more flows. Lecture Notes in Computer Science, 2006,3961:590–599. [doi: 10.1007/11919568_59]
- [22] Zhou MZ. Study of large-scale network IP flows behavior characteristics and measurement algorithms [Ph.D. Thesis]. Nanjing: Southeast University, 2006 (in Chinese with English abstract).
- [23] Zhang J, Wu JX, Lan JL, Liu JQ. Performance evaluation and comparison of three counting bloom filter schemes. Journal of Electronics, 2009,26(3):332–340. [doi: 10.1007/s11767-008-0031-x]
- [24] NLANR. Abilene-I data set. 2002. <http://pma.nlanr.net/Traces/long/ipls1.html>
- [25] Dharmapurikar S, Krishnamurthy P, Taylor DE. Longest prefix matching using bloom filters. In: Proc. of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications. New York: ACM Press, 2003. 201–212. <http://portal.acm.org/citation.cfm?id=863955.863979>
- [26] Xilinx. Virtex-5 family overview. 2009. <http://www.xilinx.com/support/documentation/>

附中文参考文献:

- [8] 程光, 龚俭, 丁伟. 基于抽样测量的高速网络实时异常检测模型. 软件学报, 2003, 14(3): 594–599. http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no=20030340&journal_id=jos
- [20] 王俊峰, 杨建华, 周虹霞, 谢高岗, 周明天. 网络测量中自适应数据采集方法. 软件学报, 2004, 15(8): 1227–1236. <http://www.jos.org.cn/1000-9825/15/1227.htm>
- [22] 周明中. 大规模网络 IP 流行为特性及其测量算法研究[博士学位论文]. 南京: 东南大学, 2006.



张进(1979—),男,江苏镇江人,博士,工程师,主要研究领域为网络测量,无线网络,片上系统,SoPC.



钮晓娜(1983—),女,助理工程师,主要研究领域为业务识别与控制,网络服务质量.



邬江兴(1953—),男,教授,博士生导师,中国工程院院士,主要研究领域为信息网络,交换技术.