

## 基于 Counting Bloom Filter 的流抽样算法研究

翟金凤<sup>1</sup>, 孙立博<sup>1</sup>, 鲁 凯<sup>2</sup>, 林学勇<sup>2</sup>, 秦文虎<sup>1</sup>

(1. 东南大学 仪器科学与工程学院, 南京 210096; 2. 南京市计量监督检测院, 南京 210049)

**摘 要:** 为适应高速网络环境并实现对网络流量的准确测量, 提出一种将计数型布隆过滤器结构与基于报文的流抽样技术相结合的网络流等概率抽样算法。利用 4 bit 的 Counter 向量识别是否有新流出现, 通过实时调整抽样频率弥补新流判定中的错误率, 从而对网络流进行等概率抽样并获取较真实的网络流分布情况。实验结果表明, 该算法的测量结果与网络流真实值较接近, 且具有可扩展性, 可以满足当前复杂多变的高速网络环境下的流量测量需求。

**关键词:** 高速网络; 流抽样; 计数型布隆过滤器; 等概率抽样; 哈希函数; Counter 向量

**中文引用格式:** 翟金凤, 孙立博, 鲁 凯, 等. 基于 Counting Bloom Filter 的流抽样算法研究 [J]. 计算机工程, 2018, 44(8): 273-278.

**英文引用格式:** ZHAI Jinfeng, SUN Libo, LU Kai, et al. Research on flow sampling algorithm based on Counting Bloom Filter [J]. Computer Engineering, 2018, 44(8): 273-278.

## Research on Flow Sampling Algorithm Based on Counting Bloom Filter

ZHAI Jinfeng<sup>1</sup>, SUN Libo<sup>1</sup>, LU Kai<sup>2</sup>, LIN Xueyong<sup>2</sup>, QIN Wenhui<sup>1</sup>

(1. School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China;

2. Nanjing Metrology Supervision and Inspection Institute, Nanjing 210049, China)

**[Abstract]** In order to adapt to the high-speed network environment and realize the accurate measurement of network traffic, a network algorithm of flow equal probability sampling based on the combination of Counting Bloom Filter and packet-based flow sampling technique is proposed. It identifies whether there is a new flow by the 4 bit Counter vector, and realizes the equal probability sampling of the network flow by adjusting the sampling frequency constantly to compensate for the error rate of the new flow judgement. Then, it makes an equal probability sampling of network flow and obtains a more realistic distribution of network traffic. Experimental results show that the algorithm's measurement results are close to the real value of network flow, and it has scalability, which can meet the current demand of traffic measurement in the complex and changeable high-speed network environment.

**[Key words]** high-speed network; flow sampling; Counting Bloom Filter; equal probability sampling; hash function; Counter vector

**DOI:** 10.19678/j.issn.1000-3428.0048088

### 0 概述

网络流量测量作为一种用来掌握网络运行状况和行为特征的基本方法, 已被广泛应用于安全检测、运营管理和流量工程等领域。新一代互联网的规模更大、速率更快, 对系统计算和存储资源提出了更高的要求<sup>[1]</sup>, 且增加了传统的对流经链路所有数据包进行捕获和测量的难度。因此, 需要在流量测量中引进一些必要的测量策略, 用来对数据量进行缩减

并保留流量数据的特征信息。抽样技术作为一种可扩展的数据缩减技术<sup>[2]</sup>, 已被众多大规模网络引进到实际的网络流量测量中, 其能够有效保留原始流量行为的细节, 提高系统的处理效率和资源利用率。因此, 抽样测量已成为网络流量测量领域的重要研究方向之一。

但是, 当前网络的高速发展使得网络的异构性和复杂性在不断加强, 而系统处理分析能力的局限性和存储资源的缺乏, 导致基本的抽样方法已难以

**基金项目:** 国家质量监督检验检疫总局科技计划项目“网络公正性检测方法研究”(2015QK059); 中央高校基本科研业务费专项(2242017K40114); 江苏省重点研发计划项目“智能网联汽车车载网络架构设计及其信息安全防护关键技术研发”(BE2017035)。

**作者简介:** 翟金凤(1995—), 女, 硕士研究生, 主研方向为无线传感网络、网络安全、虚拟现实; 孙立博, 副教授、博士; 鲁 凯, 工程师; 林学勇, 高级工程师; 秦文虎, 教授、博士。

**收稿日期:** 2017-07-24 **修回日期:** 2017-09-15 **E-mail:** qinwenhu@seu.edu.cn

满足实际测量的需求。为解决该问题,同时保证测量具备一定的高效性和准确性,以及系统资源具有一定的可控性,有研究者考虑在测量中引入一些其他的关键技术。目前,应用较广泛的技术主要有哈希算法、超时策略和概要数据结构<sup>[3]</sup>,并由此衍生出如流抽样、抽样与哈希结合的测量以及基于 Bloom Filter 的抽样测量等<sup>[4]</sup>。其中, Bloom Filter 凭借其简便易实现、资源利用率高以及处理速度快等优点,吸引了众多研究者的关注,如何将其与抽样技术进行高效地结合,成为近年来网络流量测量领域的热点论题<sup>[5]</sup>。

文献[6]较早将哈希函数引进抽样测量中。文献[7]使用 Time-Out Bloom Filter 来缓解对长流的抽样偏差,以提高对短流的抽样概率。该方法与随机抽样相比,能够在相同的抽样率下记录并识别出更多的短流,但是因为未考虑具体的流信息,所以其存在局限性。文献[8]提出一种基于 Dynamic Count Filter 的公平抽样算法,可以高效地获取全面的流信息,但 Dynamic Count Filter 存在误判率,导致算法会出现一定的测量误差。文献[9]基于改进型 Bloom Filter 数据结构,提出一种简单实用的流抽样算法,其通过调整抽样频率来实现对网络流的等概率抽样。其中,改进型 Bloom Filter 设置 3 层位向量,通过对 2 层位向量的判定结果取交集来实现对新流的识别,但该算法在 3 层位向量中交替使用时,会将某些已识别的流再次判断为新流,导致某些网络流的重复抽样。

针对上述方法存在的不足,本文将基于报文的流抽样技术和 Bloom Filter 相结合,提出一种基于 Counting Bloom Filter 的流抽样算法,用以实现对网络流的等概率抽样,然后通过实验验证该算法的性能。

## 1 流量抽样测量技术

### 1.1 基于报文的流抽样

报文抽样不考虑报文之间的相关性,其相对平等地对组成网络流量的报文进行抽样。常用的报文抽样方法主要有 3 种:系统抽样,随机抽样,分层抽样。然而,组成网络流量的报文并非独立的,它们是为实现特定的行为应用而产生的,彼此间有着一定的相关性,流则是反映这种相关性的一种途径。基于报文的抽样忽略了报文之间的相关性,不能体现更高层面的信息,从而无法实现掌控和提升网络性能的目的<sup>[10]</sup>。为解决该问题,衍生出基于报文的流抽样测量,其先对报文实施抽样,再对具有相同特征的报文进行流归并操作。这样可以将属于特定流的分组综合起来进行分析,从而缩减需要处理的数据量。此外,将部分重要特征信息保留下来,具有一定程度的可扩展性,更适用于当前的高速网络环境。

基于报文流抽样的普遍实现策略是在测量设备中,为抽中的流维护一块存储空间,先按某种频率对报文实施随机抽取,再按报文的特征信息将其归并为不同的流,并以流的形式存储在开辟的缓存中,最终构成抽中的流集合。对于在链路上传输的每一条报文,都将其与抽中的流集合中的属性信息作比对,如果其与某条流的属性相匹配(如具有相同的源 IP 地址、目的 IP 地址、源端口、目的端口和协议类型),则该报文被抽中,更新与该报文具有相同属性的相关流信息;如果其与集合中任何流的属性都不匹配,则该报文被丢弃。对网络流进行测量和分析,有助于了解网络的流量行为细节,且能在很大程度上释放系统存储资源。

### 1.2 Bloom Filter 技术

#### 1.2.1 标准 Bloom Filter

Bloom Filter 是概要数据结构中最具代表性的结构之一,是由 Howard Bloom 于 1970 年提出的二进制向量数据结构<sup>[11]</sup>,其在高速网络流量测量中发挥着巨大作用。Bloom Filter 使用长为  $m$  的位向量  $V$  表示含有  $n$  个元素的集合  $S = \{s_1, s_2, \dots, s_n\}$ 。初始化时 Bloom Filter 为空,位向量全部设为 0,同时定义  $k$  个相互独立的具有分布均匀特性的哈希函数  $h_1, h_2, \dots, h_k$ <sup>[12]</sup>,且其值域均为  $[1, m]$ 。计算每个元素  $s_i \in S$  的哈希值  $h_1(s_i), h_2(s_i), \dots, h_k(s_i)$ 。将位向量  $V$  中对应于哈希映射的  $k$  个位置设为 1。标准 Bloom Filter 原理如图 1 所示。

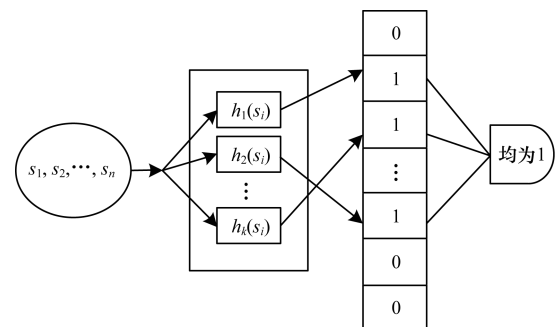


图 1 标准 Bloom Filter 原理

与其他数据结构相比, Bloom Filter 可以充分利用有限的存储资源,有效提高数据查询和处理的速率,且可以表示全集。但是,由 Bloom Filter 原理可以看出,在哈希映射过程中其可能会将位向量中的某一位重复置 1,这会导致在判断一个元素是否属于某个集合时,存在一定的概率将不属于这个集合的元素误判为属于这个集合,本文定义这样的误差概率为误判率<sup>[13]</sup>。下面对误判率的大小进行分析估计。

为简化分析模型,假设各哈希函数是完全随机的,且  $kn < m$ 。在 Bloom Filter 算法中,当集合  $S =$

$\{s_1, s_2, \dots, s_n\}$  中的  $n$  个元素经过哈希映射对应到向量中后, 这个位向量中某位还是 0 的概率为:

$$p_0 = \left(1 - \frac{1}{m}\right)^{kn} \quad (1)$$

其中,  $\frac{1}{m}$  表示某一位被哈希映射为 1 的概率,

$\left(1 - \frac{1}{m}\right)$  表示某一位被哈希映射为 0 的概率。

则误判率为:

$$P = (1 - p_0)^k = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx (1 - e^{-\frac{kn}{m}})^k \quad (2)$$

### 1.2.2 Counting Bloom Filter

标准 Bloom Filter 仅支持对元素的插入和查询操作, 当对静态集合进行表示时, 其能够呈现出较好的工作性能, 但是当要表示的集合经常变动时, 因为它不支持删除操作, 所以会产生一定的误差。

由于 Bloom Filter 存在误判率, 以及不能用来表示动态集合, 因此诸多研究人员对 Bloom Filter 提出了新的改进。目前, Bloom Filter 的经典改进结构主要有: Counting Bloom Filter, Compressed Bloom Filter, Spectral Bloom Filter 和 Dynamic Count Filter 等。其中, 文献[14]提出的 Counting Bloom Filter 是最具代表性的改进结构之一。

Counting Bloom Filter 解决了标准 Bloom Filter 无法删除元素的问题。对于标准 Bloom Filter, 当要从集合中删除元素时, 它不能简单地将位向量中的对应位置设为 0, 而 Counting Bloom Filter 则将标准 Bloom Filter 位向量的每一位扩展为一个小的 Counter(计数器), 并赋初值为 0。Counting Bloom Filter 将元素插入操作扩展为给对应的  $k$  个 Counter 值分别加 1; 元素查找操作扩展为检验对应的  $k$  个 Counter 值是否为非零; 元素删除操作扩展为将对应的  $k$  个 Counter 值分别减 1。Counting Bloom Filter 原理如图 2 所示。

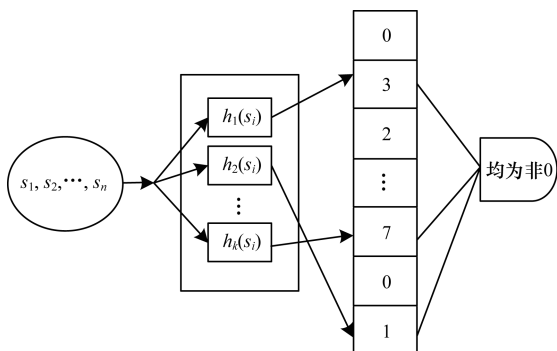


图 2 Counting Bloom Filter 原理

此外, 有研究表明, 若为 Counting Bloom Filter 中的每个计数器分配 4 bit, 则当计数器值达到 16 时, Filter 会溢出, 该概率为:

$$F \leq m \times 1.37 \times 10^{-15} \quad (3)$$

其中,  $m$  为 Counter 向量的大小, 即向量空间的大小。此时的溢出率已经微乎其微, 可以满足大部分应用程序的需求。在实际应用中, 使用 Counting Bloom Filter 时也可以从实际需求的角度为 Counter 分配合适的位数。

## 2 基于 Counting Bloom Filter 的流抽样算法

### 2.1 算法描述

如图 3 所示, 基于 Counting Bloom Filter 的流抽样算法由 Counting Bloom Filter 模块、误差吸收模块和随机抽样模块组成。Counting Bloom Filter 模块用于判定到来的数据分组所属网络流是否为新流; 由于 Counting Bloom Filter 同样存在误判率, 误差吸收模块就用来记录当前到达的流数量, 同时根据抽样过程中产生的误判率调整抽样频率; 随机抽样模块则以调整后的抽样频率对经 Counting Bloom Filter 认定的新流进行抽样, 从而实现对网络流的等概率随机抽样, 该模块可以有效反映网络流的真实分布情况, 具有较高的测量精度。

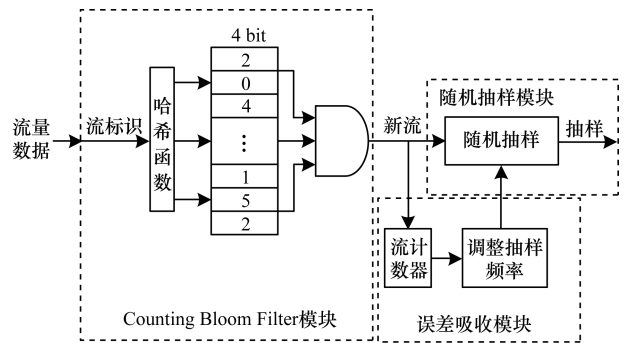


图 3 基于 Counting Bloom Filter 的流抽样算法结构

基于 Counting Bloom Filter 的流抽样算法设计流程如图 4 所示。

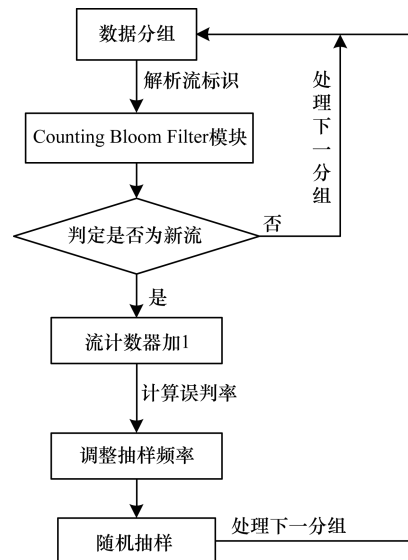


图 4 基于 Counting Bloom Filter 的流抽样算法设计流程

算法具体实现步骤为:

**步骤 1** 对 Counting Bloom Filter 结构的参数、哈希函数个数和向量空间大小进行合理配置,为 Counting Bloom Filter 结构中的每个 Counter 分配 4 bit。

**步骤 2** 当一个分组到达时,先解析其流标识,然后计算其流标识的  $k$  个哈希函数值,若 Counting Bloom Filter 结构中对应的  $k$  个 Counter 值均大于或等于 1,则判定为没有新流到达;否则,判定为有新流到达。

**步骤 3** 如果到来的数据分组被判定为新流,先将该流插入到 Counting Bloom Filter 中,对应的  $k$  个 Counter 值分别加 1,流计数器加 1。再根据已插入的流个数按照式(2)计算误判率  $P$ ,然后调整抽样频率为  $\frac{f}{1-P}$  ( $f$  是整个算法的抽样频率),确保对任意新流的抽样频率都等于  $f$ ,最后随机抽样模块以调整后的抽样频率对新流实施随机抽样。完成抽样后继续处理下一分组,重复步骤 2 并继续循环。

**步骤 4** 如果判定为没有新流到达,则继续处理下一分组,重复步骤 2 并继续循环。

在步骤 3 中,因为 Counting Bloom Filter 存在误判率  $P$ ,即一个新流被成功识别出的概率为  $1-P$ ,所以当 Counting Bloom Filter 识别出一个新流时,如果将随机抽样模块的抽样频率调整为  $\frac{f}{1-P}$ ,因为  $(1-P) \times \frac{f}{1-P} = f$ ,所以能够确保对任意新流的抽样频率都等于  $f$ ,进而满足对网络流实施等概率随机抽样的要求。

## 2.2 哈希函数个数和向量空间大小选择

在使用 Counting Bloom Filter 进行流抽样测量时,有一个较重要的问题,就是如何根据网络情况确定合适的哈希函数个数和向量空间大小,这对算法的测量效果有很大影响。使用较少的哈希函数时,会造成较大的误判率;使用较多的哈希函数时,则会引起计算复杂度的上升以及向量空间消耗的增大。而向量空间选择过大会浪费较多的存储资源,向量空间过小则会导致误判率的增加<sup>[10]</sup>。

文献[15]表明,当哈希函数个数  $k = \ln 2 \cdot \frac{m}{n}$  ( $m$  为向量空间的大小,  $n$  为所表示集合元素的个数)时,Counting Bloom Filter 可以获得最小的误判率。假设在规定误判率不能大于  $E$  的前提下,为保证 Counter 向量里至少一半为 0,向量空间的大小  $m$  应该满足:  $m \geq n \cdot \text{lb}\left(\frac{1}{E}\right) \cdot \text{lb} e$ ,  $m$  约为  $n \cdot \text{lb}\left(\frac{1}{E}\right)$  的 1.44 倍。

误判率在  $\frac{m}{n}$  固定时与  $k$  的关系如图 5 所示(其中,  $\frac{m}{n}$  取 32)。由图 5 可以看出,随着哈希函数个数

的增加,误判率呈现出先减小后增大的变化趋势。尤其是  $k > 10$  时,误判率变化缓慢,且  $k$  达到一定数量后,会使计算复杂度和误判率增大。

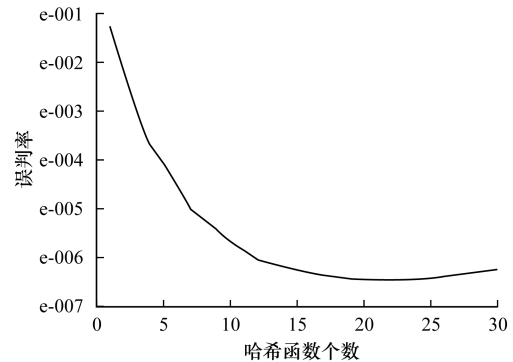


图 5 误判率随哈希函数个数的变化关系

图 6 所示为  $k = 4, 8, 16, 32$  时误判率随  $\frac{m}{n}$  的变化曲线。由图 6 可以看出:

1) 误判率与  $\frac{m}{n}$  成反比,因此,对于给定的  $n, m$  取值越大,测量中产生的误判率就越小。

2)  $k = 8, 16, 32$  的 3 条误判率变化曲线较接近,即当哈希函数达到一定数量后,继续增加  $k$  值对误判率的影响极其微小。因此,要根据实际情况对误判率与计算复杂度<sup>[16]</sup>进行权衡,从而选取恰当的  $k, m$  值。

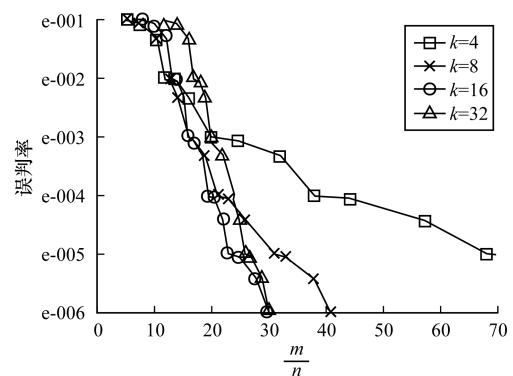


图 6 误判率随  $\frac{m}{n}$  的变化关系

## 3 实验结果与分析

### 3.1 实验数据

本次实验使用的流量数据来自互联网数据分析合作组织 CAIDA 公开提供的真实被动测量数据 Traces。Trace 1 为 2011 年 2 月 17 日在芝加哥采集得到的数据,Trace 2 为 2012 年 6 月 21 日在圣约瑟采集得到的数据,数据详情如表 1 所示。实验平台为 Visual Studio 2013 和 MATLAB R2014b。由于硬件性能限制,本文选取 2 组 Traces 数据的前  $10^4$  个数据分组进行仿真分析。

表 1 Trace 1 和 Trace 2 流量数据信息

数据组	链路速率/(GB · s <sup>-1</sup> )	分组数量
Trace 1	2.468	344 736 805
Trace 2	3.537	1 337 466 050

### 3.2 算法性能分析

使用报文头中的源 IP 地址和目的 IP 地址作为网络流的流标识, 为 Counting Bloom Filter 结构中的每个 Counter 分配 4 bit, 向量的长度即 Counter 向量的大小设为实际流总数的 20 倍, 先后使用 3 个、10 个哈希函数进行仿真比较。

图 7、图 8 分别为使用 Trace 1 和 Trace 2 数据时, 基于 Counting Bloom Filter 的流抽样算法使用 3 个、10 个哈希函数的测量值与流量数据真实值的对比结果。

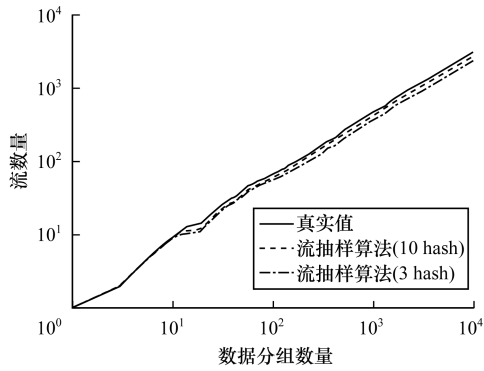


图 7 Trace 1 流数量测量结果比较

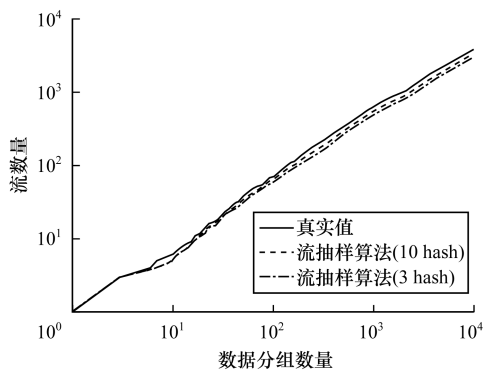


图 8 Trace 2 流数量测量结果比较

由图 7、图 8 可以看出, 本文网络流等概率抽样算法得到的流数量与真实值较接近, 算法测量精度较高。误判率的存在会导致部分新流数据不被识别, 从而使得 Counting Bloom Filter 模块识别出的新流数量小于实际流数量, 引起测量误差。但是, 本文算法中的随机抽样模块将会以实时调整抽样频率的形式吸收该误差, 从而降低误判率对最终测量结果的影响。

由图 7、图 8 还可以看出, 算法中哈希函数个数设为 10 时的测量精度要高于哈希函数个数设为 3 的情况, 这与第 2.2 节得出的 Counting Bloom Filter 的误判率在一定范围内会随哈希函数个数的增加而减小的结论相一致。由于哈希函数个数大于 10 时误判率变化放缓, 且哈希函数达到一定数量后, 反而会使计算复杂度和误判率增大, 因此本文算法在实际应用中应尽量选择约 10 个哈希函数。

定义测量误差为  $e = |n' - n|/n$ , 其中,  $n'$  为通过算法测量出的网络流数量值,  $n$  为网络流数量的真实值。选取每 1 000 个数据分组作为记录点。图 9、图 10 分别为本文算法对 Trace 1 和 Trace 2 流量数据进行仿真时的测量误差。

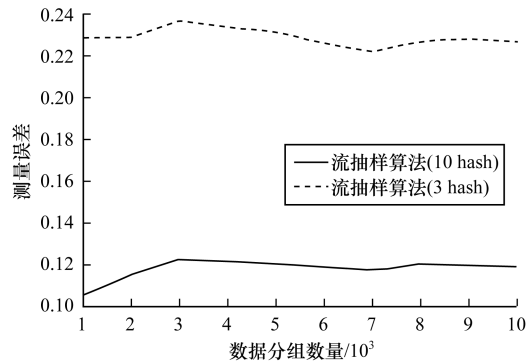


图 9 Trace 1 测量误差

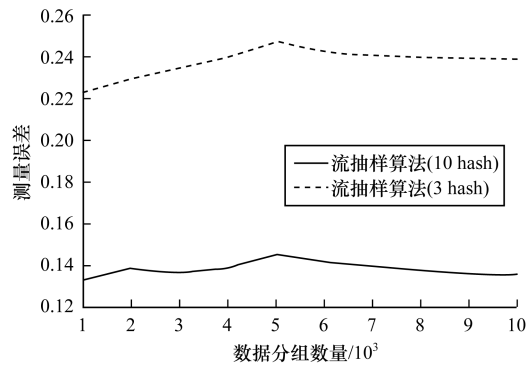


图 10 Trace 2 测量误差

由图 9、图 10 可以看出, 哈希函数个数设为 10 时, 本文算法能够有效提高对网络新流的识别率, 其测量误差明显低于哈希函数个数设为 3 时的情况, 测量结果较准确。因此, 哈希函数个数要尽可能地设为 10 左右。由图 9、图 10 还可以看出, 随着数据分组数量的增加, 算法结果的测量误差都呈现先上升后趋于平稳的趋势。算法结果控制在一定的误差范围内, 可以满足一定精度下的网络流等概率抽样, 并有效获得网络流的真实分布情况, 最终节省系统的处理和存储资源, 因此, 该算法适用于当前的高速网络环境。

#### 4 结束语

本文对基于报文的流抽样技术与 Bloom Filter 技术进行探究,将 Counting Bloom Filter 结构和流抽样技术相结合,并充分利用两者的优势,提出一种网络流等概率抽样算法。该算法为 Counting Bloom Filter 结构中的每个 Counter 分配 4 bit,通过 4 bit 的 Counter 向量识别是否有新流出现,并借助后续的随机抽样模块以实时调整抽样频率的形式吸收由 Counting Bloom Filter 的误判率引起的测量误差,最终实现对网络流的等概率抽样。仿真结果表明,本文算法测量结果趋近于网络流真实值,可以有效获得网络流的真实分布情况,测量精度较高,且具有可扩展性,能够适应当前的高速网络环境。下一步将考虑改进 Counting Bloom Filter,以使其计数器的位数能够动态适配高速网络环境。

#### 参考文献

- [1] 林明方. 高速网络流量测量中抽样技术的研究[J]. 硅谷,2010(10):86,169.
- [2] 王 妮. 基于抽样和哈希技术的网络流量测量算法研究[D]. 长春:吉林大学,2012.
- [3] DUFFIELD N, LUND C, THORUP M. Flow sampling under hard resource constraints [C]//Proceedings of SIGMETRICS'04. New York, USA: ACM Press, 2004: 85-96.
- [4] 李雪梅,王洪源. 网络流量抽样测量技术综述[J]. 科技信息,2011(9):61,117.
- [5] 周爱平,程 光,郭晓军. 高速网络流量测量方法[J]. 软件学报,2014,25(1):135-153.
- [6] DUFFIELD N G, GROSSGLAUSER M. Trajectory sampling for direct traffic observation[J]. IEEE/ACM Transactions on Networking,2001,9(3):280-292.
- [7] KONG S, HE T, SHAO X, et al. Time-out bloom filter: a new sampling method for recording more flows [C]//Proceedings of International Conference on Information Networking. Berlin, Germany: Springer, 2006:590-599.
- [8] 王宜青,陈庶樵,张 震. 基于动态计数型过滤器的网络流公平抽样机制[J]. 计算机应用与软件,2014,31(11):139-142.
- [9] 王少龙,张毅卜,徐 敏,等. 基于改进型 Bloom Filter 的网络流抽样算法[J]. 电子设计工程,2015,23(24):122-125.
- [10] 孟金凤,高仲合. 基于 DCBF 的流抽样测量算法[J]. 计算机工程与应用,2015,51(17):92-95,162.
- [11] 张 震,汪斌强,张风雨,等. 基于 LRU-BF 策略的网络流量测量算法[J]. 通信学报,2013,34(1):111-120.
- [12] 张淋淋. 网络流量测量中的抽样算法研究[D]. 曲阜:曲阜师范大学,2015.
- [13] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization,1997,11(4):341-359.
- [14] FAN L, CAO P, ALMEIDA J, et al. Summary cache: a scalable wide-area Web cache sharing protocol [J]. IEEE/ACM Transactions on Networking, 2000, 8(3): 281-293.
- [15] 胡广昌. 基于 Bloom Filters 流抽样算法的研究[D]. 曲阜:曲阜师范大学,2010.
- [16] 郑先侠,王海泉,李 飞,等. 大规模天线系统中低复杂度的解码方法研究[J]. 计算机工程,2017,43(10):31-37.
- 编辑 吴云芳
- 
- (上接第 272 页)
- [9] YAO Bangpeng, KHOLSA A, LI Feifei. Combining randomization and discrimination for fine-grained image categorization[C]//Proceedings of International Conference on Computer Vision and Pattern Recognition. Washington D. C., USA:IEEE Press,2011:1577-1584.
- [10] SHARMA G, JURIE F, SCHMID C. Discriminative spatial saliency for image classification[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Computer Society, 2012: 3506-3513.
- [11] YANG Jianchao, YU Kai, GONG Yihong, et al. Linear spatial pyramid matching using sparse coding for image classification [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA:IEEE Press,2009:1794-1801.
- [12] WANG Jinjun, YANG Jianchao, YU Kai, et al. Locality-constrained linear coding for image classification[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press,2010:3360-3367.
- [13] CHANG Chihchung, LIN Chihjen. LIBSVM: a library for support vector machines [J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3):1-27.
- [14] LI Piji, MA Jun, GAO Shuai. Actions in stillweb images: visualization, detection and retrieval [C]//Proceedings of IEEE Conference on International Conference on Web-age Information Management. Berlin, Germany: Springer-Verlag, 2011:302-313.
- [15] 胡局新,张功杰. 基于 K 折交叉验证的选择性集成分类算法[J]. 科技通报,2013,29(12):115-117.
- [16] DELAITRE V, LAPTEV I, SIVIC J. Recognizing human actions in still images: a study of bag-of-features and part-based representations [C]//Proceedings of BMVC'10. Aberystwyth, UK:DBLP,2010:1-11.
- [17] 杨红菊,冯进丽,郭 倩. 基于多核学习的静态图像人体行为识别方法[J]. 数据采集与处理,2016,31(5):958-964.
- 编辑 刘 冰