软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(1):135−153 [doi: 10.13328/j.cnki.jos.004445]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

高速网络流量测量方法*

周爱平[1,2], 程 光[1,2], 郭晓军[1,2]

[1](东南大学 计算机科学与工程学院,江苏 南京 211189)

[2](教育部计算机网络和信息集成重点实验室(东南大学),江苏 南京 211189)

通讯作者: 周爱平, E-mail: apzhou@njnet.edu.cn

摘 要: 高速网络流量测量是目前实施实时准确地监测、管理和控制网络的基础.基于网络流量测量的应用,将网络流量测量分为抽样方法和数据流方法.从不同的层次,将抽样方法分为分组抽样和流抽样,分别介绍了两类抽样方法;从测度角度介绍了数据流方法.详细介绍了高速网络流量测量的常用数据结构,以及抽样、数据流方法在高速网络流量测量中的应用,比较了各种方法的优劣.概述了高速网络流量测量技术的研究进展.最后,就现有的网络流量测量方法的不足,对网络流量测量的发展趋势和进一步的研究方向进行了讨论.

关键词: 网络流量测量;分组抽样;流抽样;数据流

中图法分类号: TP393 **文献标识码**: A

中文引用格式: 周爱平,程光,郭晓军.高速网络流量测量方法.软件学报,2014,25(1):135−153. http://www.jos.org.cn/1000-9825/4445.htm

英文引用格式: Zhou AP, Cheng G, Guo XJ. High-Speed network traffic measurement method. Ruan Jian Xue Bao/Journal of Software, 2014,25(1):135−153 (in Chinese). http://www.jos.org.cn/1000-9825/4445.htm

High-Speed Network Traffic Measurement Method

ZHOU Ai-Ping[1,2], CHENG Guang[1,2], GUO Xiao-Jun[1,2]

[1](School of Computer Science and Engineering, Southeast University, Nanjing 211189, China)

[2](Key Laboratory of Computer Network and Information Integration, Ministry of Education (Southeast University), Nanjing 211189, China)

Corresponding author: ZHOU Ai-Ping, E-mail: apzhou@njnet.edu.cn

Abstract: Traffic measurement in high-speed network is essential for network monitoring, management, and control. Based on the applications of network traffic measurement, this study divides the measurement into sampling methods and data stream methods. Sampling methods are partitioned into packet sampling and flow sampling, both are introduced. Data stream methods are introduced from different metrics. This study introduces in detail the common data structure and applications based on sampling and data stream methods in high-speed network. Drawbacks of different methods are analyzed and compared. The research progress of high-speed network traffic measurement technology is summarized. Finally, the limitation of recent network traffic measurement methods, the evolving trend of network traffic measurement, and some possible directions of future research are discussed.

Key words: network traffic measurement; packet sampling; flow sampling; data stream

近年来,随着互联网的快速发展和新应用的不断出现,很多研究机构和学者致力于研究开发对互联网实施监测和测量的技术,网络流量测量已被广泛应用于网络计费、流量工程、网络安全等领域[1].随着网络链路速率

* 基金项目: 国家自然科学基金(60973123); 国家重点基础研究发展计划(973)(2009CB320505); 江苏省科技支撑计划——工业部分(BE2011173); 江苏省"六大人才高峰"

收稿时间: 2013-03-22; 修改时间: 2013-05-13; 定稿时间: 2013-06-26; jos 在线出版时间: 2013-07-25

CNKI 网络优先出版: 2013-07-25 14:03, http://www.cnki.net/kcms/detail/11.2560.TP.20130725.1403.003.html

的不断提高和网络数据流的急剧增加,当前在高速骨干网链路上,网络流量测量需要极高的计算和存储资源,从而给网络流量测量研究开发带来了技术挑战.在高速链路上,处理每个分组需要纳秒级时间,例如在 OC-768 (40Gbps)链路上,设分组的平均大小为 40B,则分组的平均处理时间为 8ns.传统的网络流量测量方法面临的主要问题是可扩展性,不能够适应高速网络环境.美国已在高速网络实验床上开展了下一代高速计算机网络及其典型应用的研究,其他国家和地区也相继开展了下一代高速互联网络及其应用的研究,如英国、加拿大.与国际同类研究相比,我国的下一代互联网的研究内容涵盖了高速互联网络涉及的大部分研究领域,包括基础设施、网络服务与网络应用,并取得了一定的理论与应用成果[2].高速网络流量测量技术是下一代互联网研究的重要组成部分,成为网络测量的发展趋势之一.目前,高速网络流量测量问题主要有 3 种解决方案:利用高性能的专用硬件,如 TCAM,ASIC 等,实现高速链路上网络流量的数据处理.然而,高性能的硬件设备极其昂贵;利用抽样技术只对部分有代表性的网络流量数据进行采集处理,虽然降低了系统的负荷,但却存在较大的误差[3];利用数据流技术[4]对所有网络流量数据进行处理,有效地减少存储资源的需求,同时保持一定的准确性.

高速链路上持续到达的海量网络流量给网络流量测量与分析带来了极大的困难,因此需要采取一些可行的措施,既能够对网络流量数据进行缩减,又能够保留网络流量数据的特征信息.根据网络流量测量的应用需求,数据缩减技术主要分为抽样和数据流两种.抽样技术的目的是选择具有代表性的网络流量数据分组子集,通过该子集推断网络流量总体数据分组的特征信息.数据流技术是将庞大的信息压缩到较小的存储空间并保持一定的精确度,数据流技术具有在线实时处理和有限存储空间的特性.各种网络流量测量方法是针对具体的应用需求提出来的,具有一定的局限性,目前还没有一种通用的网络流量测量方法.

本文详细综述了近年来国内外主要的高速网络流量测量方法的研究进展.第 1 节介绍网络测量方法的主要评价指标.第 2 节介绍网络流量测量的抽样方法.第 3 节介绍网络流量测量的数据流方法.第 4 节介绍高速网络流量测量的数据结构.第 5 节介绍抽样方法与数据流方法在网络流量测量中的应用.第 6 节概述高速网络流量测量技术的研究成果.第 7 节讨论现有的网络测量方法的不足、网络测量的发展趋势和可能的下一步的研究方向.最后对全文进行总结.

1 网络测量方法的评价指标

高速网络测量技术主要从以下几个方面进行评估:

- 实时性:反映网络测量技术能够在线地、快速地处理网络数据流的能力;
- 准确性:反映网络测量技术能够估计网络数据流的能力;
- 可扩展性:反映网络测量技术能够处理大量的网络数据流的能力;
- 存储复杂性:反映网络测量技术准确估计网络数据流所需存储空间;
- 计算复杂性:反映网络测量技术准确估计网络数据流所需处理开销,如内存访问、CPU.

目前,对这些评估指标进行量化还存在一定的困难.为了能够有效验证现有的网络测量方法,本文主要介绍误报率、漏报率和检测率、无偏估计和相对误差、平均相对差和加权相对差以及熵和标准熵评价指标.

1.1 检测率、误报率和漏报率

误报(false positive)是指正常事件被检测为异常事件.真阴性(true negative)是指正常事件被检测为正常事件.令 FP 为误报数,TN 为真阴性数,则误报率(false positive rate)为

$$R_{f+}=\frac{FP}{FP+TN}.$$

漏报(false negative)是指异常事件被检测为正常事件.真阳性(true positive)是指异常事件被检测为异常事件.令 FN 为漏报数,TP 为真阳性数,则漏报率(false negative rate)为

$$R_{f-}=\frac{FN}{FN+TP}.$$

检测率(detection rate)是指被检测为异常事件的异常事件数与实际异常事件总数的比率,则检测率为

$$R_d = \frac{TP}{FN + TP}.$$

由于 R_d+R_{f-}=1,因此实际应用中仅需考虑误报率与检测率.

1.2 无偏估计和相对误差

网络测量中常用流长估计的无偏性评价估计精度.如果 $E[\hat{n}] = n$, 则 $\hat{n}$ 是 n 的无偏估计(unbiased estimation).

相对误差(relative error)表示为 $|\hat{n} - n|/n$, 而实际应用中,常用离差系数表示相对误差,离差系数定义为流长估计的标准差与真实流长之比,即 $\sqrt{\mathrm{var}(\hat{n})}/n$.

1.3 平均相对差和加权平均相对差c

网络测量中常用流长分布估计的平均相对差和加权平均相对差评价估计精度.令 n_i 为大小为 i 的流数, $\hat{n}_i$ 为 n_i 的估计,则相对差(relative difference)表示为

$$relative\ difference = |n_i - \hat{n}_i| \Big/ \left(\frac{n_i + \hat{n}_i}{2}\right).$$

从而,平均相对差(mean relative difference)表示为

$$mean\ relative\ difference = \frac{1}{z}\sum_{i=1}^{z} \frac{|n_i - \hat{n}_i|}{\frac{n_i + \hat{n}_i}{2}}.$$

平均相对差不适用于评价具有重尾特性的流长分布.给每个流长估计的相对差分配一个权重 $\frac{n_i + \hat{n}_i}{2}$, 则加权平均相对差(weighted mean relative difference)为

$$weighted\ \ mean\ \ relative\ \ difference = \frac{\sum_{i=1}^{z} \frac{|n_i - \hat{n}_i|}{\frac{n_i + \hat{n}_i}{2}} \cdot \frac{n_i + \hat{n}_i}{2}}{\sum_{i=1}^{z} \frac{n_i + \hat{n}_i}{2}} = \frac{\sum_{i=1}^{z} |n_i - \hat{n}_i|}{\sum_{i=1}^{z} \frac{n_i + \hat{n}_i}{2}}.$$

加权平均相对差适用于评价网络流长分布估计.

1.4 熵和标准熵

在信息论中,熵是不确定性的度量.令数据集 X={x_1,x_2,…,x_N},它的熵表示为

$$H(X) = -\sum_{i=1}^{N} p_i \log_2(p_i),$$

其中,N 表示数据集 X 中元素的个数,p_i 表示第 i 个元素发生的概率.网络测量中,常用熵表示数据流中分组的随机性或差异性.若数据流中分组是相同的,则数据流获得最小熵 0;若数据流中所有分组是不同的,则数据流获得最大熵 $\log_2 N$.为了比较熵估计,定义标准熵为

$$H_n(X) = -\frac{\sum_{i=1}^{N} p_i \log_2(p_i)}{\log_2 N}.$$

标准熵的取值范围为[0,1].

2 抽样方法

抽样技术是指从原始流量数据中选择有代表性的分组子集,通过该分组子集推断原始流量数据的特征.随着链路速率的提高和应用的多样化,巨大的网络流量给流量采集、传输、存储、分析都带来了巨大的压力.为了解决高速网络被动测量问题,将抽样技术应用于高速网络流量测量,可在满足问题统计精度的条件下,减少用于测量、存储和处理的数据量.

在高速网络流量测量中,抽样方法实现受到技术和资源的限制,往往需要在抽样率和估计精度之间加以折

中.抽样采集使得系统的处理负荷大为减轻,具备较好的可扩展性,而且还能从样本特征参数反映出原始流量特征参数,具有一定的测量精度.抽样数据除了可以对流量特征进行分析外,还在流量计费、性能特征测量、异常检测等领域广泛应用.对于互联网中的流量,从分组和流的层次,抽样方法主要分为分组抽样和流抽样.本节主要介绍这两类抽样方法.

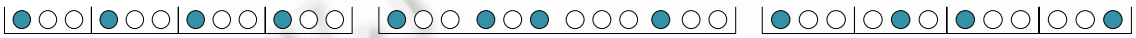
2.1 分组抽样

分组抽样(packet sampling)是指对构成网络流量的分组进行抽样,每个分组都是独立的,不考虑分组之间的相关性.常用的分组抽样方法包括系统抽样、简单随机抽样和分层随机抽样:

(1) 系统抽样(systematic sampling)是指以固定的间隔抽取对象,在选择抽取第 1 个对象后,每隔 N 个对象选择下一个对象,如图 1(a)所示.系统抽样方法是一种广泛应用的抽样方法,但是系统抽样存在一定的周期性;

(2) 简单随机抽样(simple random sampling)是指以一定的概率抽样对象,如图 1(b)所示.每个对象被抽样的概率可以是相同的也可以是不同的,这种概率一般会遵循某种概率分布函数.在流量测量中,常用的随机抽样方法分为简单随机抽样和随机增量抽样.这两种随机增量抽样方法可以避免系统抽样的同步问题;

(3) 分层随机抽样(stratified random sampling)是指首先把总体分成若干层次或类型组,然后从各个层次中按一定的比例随机抽样.这种分层可以是按照元素的排列顺序进行划分,如图 1(c)所示,也可以按照元素的某个特征,如分组长度、协议类型等进行分层,然后分别进行抽样.在流量测量中,常用的分层抽样为均匀分层随机抽样(uniform stratified radom sampling).该方法可以保证抽样相对于元素的属性是无偏的,减少分组统计的误差,使得估计结果更接近于原始数据.



(a) 系统抽样 (b) 简单随机抽样 (c) 分层随机抽样

Fig.1 Three kinds of sampling methods

图 1 3 类抽样方法

由于网络流量具有自相似性特征,He 等人[5]提出改进的系统抽样 BSS(biased systematic sampling),与静态的系统抽样和简单随机抽样相比,提高了均值的估计精度,同时降低了抽样开销.然而,大部分分组抽样方法均匀地选择分组,而没有考虑到分组的大小,从而使得网络测量获得一些不确定性,如流检测、估计精度、处理负荷等.为了克服分组大小给抽样带来的不利影响,Raspall[6]提出了 EBS(efficient byte sampling),以提高测量质量,使得测量精度更少地依赖于流量的特征,降低了测量开销.

2.2 流抽样

流抽样(flow sampling)是指在测量时间内对网络流进行抽样,构成网络流量的分组并不是孤立的,它们是为了完成具体的应用而产生的,它们之间存在着一定的关联,流是体现这种关联的一种方式.流抽样主要有两种抽样方式:先对分组进行抽样,再对分组进行流归并;先对分组进行流归并,再对流进行抽样.

流、分组的统计特性存在着完全不同的特点,流抽样和分组抽样的需求也不一样.由于分组的大小是受传输技术限制的,其最大长度不会超过网络能够支持的最大值,但是流的大小却是不受影响的.对于流量测量,采用哪种测量和抽样方法是由网络测量的目的决定的.对于流量计费,关注流量的长度、大小,大流丢失会导致大量信息损失,因此需要保证大流被抽样;如果对所有的流按照相同的概率进行抽样,少量的大流信息很容易被漏掉.然而对于异常监测,需要保留尽量多的流信息,如 SYN Flooding,DoS 攻击等通常由大量的小流构成.

抽样是降低内存消耗和分组处理时间的最广泛采用的方法之一.由于最坏情况下资源使用是平均情况下资源使用的多个数量级,抽样率的静态选择存在一些问题.Sampled NetFlow 需要配置一个静态的抽样率,该方

法的主要问题在于选择安全的参数,确保网络设备在不利的流量环境下持续地运行.因此,抽样率是在最坏情况下设定的.许多研究已经解决了动态选择抽样率的难题,通过自适应网络环境,克服设定静态抽样率的缺陷[7].然而,自适应的抽样方法,如 Adaptive NetFlow(ANF)[7],还没有广泛地使用,主要原因在于,自适应的抽样方法需要消耗大量 CPU 资源,依赖于复杂的数据结构与算法,使得它们在网络硬件中的实施缺乏吸引力.Sanjuàs-Cuxart 等人[8]提出一种基于自适应流抽样的测量方法,该算法基于一个简单、随机的数据结构,该结构需要很小的分组开销,很容易参数化.与之前的方法相比,该算法基于更加简单的算法,而且需要更少的硬件资源,因此适合于硬件实施.为了降低小流的估计误差,概率计数器更新算法被提了出来,如 ANLS[9],这些算法利用计数器值 c 的函数 $p(c)$代替静态的抽样率 p,因此抽样率根据抽样的分组数而变化.然而,由于测量精度不仅受到抽样函数的影响,而且受到流长分布的影响,选择一个独立的静态抽样函数也是不够的.因此,Ma 等人[10]提出了 Smart Selection Sampling(S^3)方法,通过利用流长分布信息选择合适的抽样函数,能够调整抽样函数,获得更高的测量精度.

3 数据流方法

高速链路上网络流具有实时性、连续性、无界性等特点,从而决定了处理网络流的算法只对网络流执行一趟计算,而且只需要使用有限的计算和内存资源.对这种网络流进行计算的算法必须满足以下条件:算法需要使用的空间必须足够小;处理和更新必须迅速、简单;对于查询必须有一定的准确度保证.抽样技术在网络流量测量与分析中得到广泛应用.尽管抽样方法产生了一个原始数据的代表子集,但是从抽样数据推断得到的网络流量统计信息存在一定的误差,不能确切地反映原始流量的特征.数据流方法具有单遍扫描、有限的计算和内存资源等特点,是高速网络流量测量的重要方法.随着计算机网络和通信技术的迅猛发展,数据流方法广泛应用于网络流量监控、商业交易和分析、传感器网络等领域.

数据流方法应用于近似测量高速链路上网络流量统计信息,如活跃流的总数、大流识别、流长分布、节点连接度和熵估计等.本节从熵估计、流量和流矩阵估计、连接度估计测度方面介绍了数据流方法.

3.1 熵估计

熵是网络测量中一个重要的测度.网络流量的熵有助于许多网络监测应用,如异常检测[11].在高速链路上流量的熵的测量需要低 CPU 和存储要求的准确算法.由于处理能力和存储的限制,传统方法不适用于高速链路.抽样方法能够降低处理和存储要求,适用于捕获一些流量统计信息,然而从抽样数据获得的估计可能存在较大的误差.数据流算法对计算和存储要求相对较低,适用于高速链路上估计网络流量的熵[12].

在数据流算法中,流量的熵定义为

$$H=-\sum_{i=1}^{n}\frac{m_i}{m}\log_2\left(\frac{m_i}{m}\right)=-\frac{1}{m}\left[\sum_{i=1}^{n}m_i\log_2 m_i-\sum_{i=1}^{n}m_i\log_2 m\right]=\log_2 m-\frac{1}{m}\sum_{i=1}^{n}m_i\log_2 m_i,$$

其中,m_i 表示流中第 i 项的频数,m 表示流的总项数,$m=\sum_{i=1}^{n}m_i$.

令 $S=\sum_{i=1}^{n}m_i\log_2 m_i$,熵估计与频数矩估计[13]具有相似的结构,基于此,Lall 等人[14]提出 S 的一个(ε,δ)近似算法,表示至少以 $1-\delta$概率获得相对误差至多为ε的估计,即:

$$\Pr(|X-\hat{X}|\leqslant X\varepsilon)\geqslant 1-\delta,$$

其中,$\hat{X}$ 是 X 的估计.该算法利用著名的 Alon-Matias-Szegedy 频数矩估计算法的思想[13].该数据流算法未考虑网络流量分布特征,然而网络流量分布具有重尾特性.在此基础上,Lall 等人[14]提出了另一种数据流算法,利用区分网络流中大流与小流的思想,通过分别估计大流、小流对熵的贡献,进一步提高熵估计的精度,同时减少存储空间.前一种数据流算法使用的空间直接与估计量的方差成正比,通过筛选出高计数项,能够显著地减小估计量的方差和存储空间.后一种数据流算法对抽样方法作了微小的改进,以小概率对每个位置抽样,而不是事先计算流中位置.在流中某项被抽样之后,为该项维护准确的计数,类似于 Sample and Hold 算法[15].如果流中某项被抽样 1 次,则认为是小流,通过以前的算法计算小流的熵;如果流中某项被抽样多次,则认为是大流,估计大流的熵.

前一种数据流算法优于传统的抽样方法,给出熵的无偏估计,获得更低的估计误差,同时使用类似的存储开销;后一种数据流算法有效地分离大流和小流,提高了熵估计的精度.

OD(origin-destination)流的熵也是网络测量中一个重要的测度,该熵有助于掌握 ISP 网络内流量动力学.估计网络内所有流的熵是非常有帮助的.网络性能下降和服务中断,可能是由多种事件引起的,包括网络异常,如 DDoS 攻击、网络故障、flash crowds 以及计划的网络维护任务,如路由器 IOS 更新、客户迁移,这些事件以分布式方式发生.检测这些事件和评价它们对网络服务的影响,需要从不同的位置来监控网络流量.更重要的是,流量分布的变化在传统的流量矩阵上可能是完全不可见的.然而,通过检查网络中每个 OD 流的熵,能够实时地捕获这些事件.Zhao 等人[16]提出一种数据流算法,解决了估计网络内所有 OD 流的熵难题.

3.2 流量与流矩阵估计

流量矩阵表示测量区间内网络中每个OD对之间的分组数或字节数,流量矩阵的估计困难已经受到相当多的关注.流量矩阵的准确估计有助于网络管理,如容量规划与预测、网络故障与可靠性诊断以及路由配置.有时,流量矩阵对于一些流级应用仍是不足够的,如推断 ISP 的使用模式、检测路由摆动、链路故障、DDoS 攻击以及 Internet worms.流矩阵表示网络中每个OD流之间的流量大小,与流量矩阵相比,流矩阵是更细粒度的,且更有助于流级应用.流矩阵估计是另一个重要的难题,基于统计推断或分组抽样的流量矩阵估计算法不能获得高精度的估计.为了满足高速链路上流量与流矩阵估计,Zhao 等人[17]提出了两种数据流算法,即基于 Bitmap 的数据流算法和基于计数器数组的数据流算法.基于 Bitmap 的数据流算法能够获得至少比之前的算法高一个数量级的流量矩阵估计;基于计数器数组的数据流算法获得比流量矩阵更细粒度的流矩阵估计.这两种数据流算法能够处理高速链路(如 40Gbps)上的网络流,产生比网络流小多个数量级的流量概要.

3.3 连接度估计

主机连接度是与某台主机相连的其他主机的数量,它是网络流量测量与监控的一个重要测度.超连接度主机是指在短时间内主机与其他主机之间存在大量不同的连接.对快速网络安全监控而言,检测超连接度主机是最重要的任务之一.例如,识别超连接度主机有助于检测端口扫描、蠕虫传播以及 DDoS 攻击,因为端口扫描和蠕虫传播是由在短时间内主机与不同目的主机建立大量的连接引起的,而DDoS攻击是大量的主机泛洪到一个目的主机所引起的.在两个相连的区间内主机连接度的显著变化,也是监控网络流量的一个重要测度.

由于在高速网络环境下大量的网络流量数据和有限的处理能力,很难准确、实时地测量和监控高速链路上的网络流量.高速链路上准确、实时地检测超连接度主机,是网络测量与网络安全中一个重要的难题,已经得到广泛的研究.维护每个流状态的简单方法,不适用于高速链路上检测超连接度主机.基于 Hash 的流抽样技术提供了一种分析与处理大量数据的可能有效的方法[18],然而该算法的准确性依赖于抽样率,高速链路上抽样率受到存储器的限制,从而不可能获得准确的主机连接度估计.同时,在两个相连的区间内仅仅能够估计抽样主机的连接度变化,抽样技术不能准确地测量主机连接度的变化,从而不能准确地检测连接度发生显著变化的主机.

数据流方法广泛应用于主机连接度估计.Guan 等人[19]利用 RCDS(reversible connection degree sketch)实时地测量和监控主机连接度和主机连接度的动态变化,能够准确、有效地检测超连接度主机或连接度发生显著变化的主机.Wang 等人[20]对该数据结构进行了改进,提出了一种检测超连接度节点或连接度发生显著变化节点的有效保留算法.为了提高空间的利用率,Yoon 等人[21]利用一组 Hash 函数从共享的位数组中随机地选择位,为每个节点建立虚拟 Bitmap,通过每个主机对应虚拟 Bitmap 估计节点连接度.虚拟索引方法 VCDS(virtual connection degree sketch)[22]估计高速链路上的节点连接度.为了减少因共享而对主机的虚拟 Bitmap 所产生的噪声污染,通过过滤 Bitmap 来估计节点连接度.

4 高速网络流量测量的数据结构

数据结构是高速网络流量测量的重要组成部分,优化的数据结构有助于提高算法的执行效率和估计精度,降低计算和存储开销.现有的数据结构主要包括 Bitmap,Hybrid SRAM/DRAM Counter,Bloom Filter,Count-Min

Sketch,Counter Braids,BRICK.本节主要介绍这些数据结构及其应用.

4.1 Bitmap

Bitmap 是一个简单的数据结构,将某个域映射到位数组.直接的 Bitmap[23]是一种流数估计算法,利用 Hash 函数将流标识映射到 Bitmap 中的一位.Bitmap 初始化为 0,当分组到达时,将该分组的流标识映射到 Bitmap 中的一位,并置该位为 1.属于同一流的所有分组映射到 Bitmap 中的同一位置,因此,无论每个流发送多少分组,每个流至多对应于 Bitmap 中的一位.Bitmap 中为 1 的位数作为流数的估计,由于存在 Hash 冲突,流数估计是不准确的.

基于 Bitmap 算法低估了实际的流数,使用离散区间的主要缺陷在于可能低估和不能频繁地报告.基于 Timestamp Vector 算法[24]是基于 Bitmap 算法的扩展,保持了基于 Bitmap 的流数估计算法的速度快和内存小的优点.在基于 Timestamp Vector 算法中,允许频繁报告实现了报告区间的分离,避免了流数低估问题,有效地提高了流数估计的精度.由于上述两种流数估计算法中每个分组到达时需要多次访问内存、创建新的流记录、处理冲突需要消耗大量的内存资源,在高速网络环境中需要存储大量的流标识,从而需要使用大量的内存资源.Hash 表必须存储在 DRAM 中,访问 DRAM 的时间长于分组相继到达的时间间隔,流数估计算法必须能够及时处理高速网络中的每个到达的分组.在直接的 Bitmap 的基础上,基于 Countdown Vector 算法[25]在滑动窗口上估计流数,显著地减少了所需的内存和 CPU 资源,提高了流数估计的精度.

4.2 Hybrid SRAM/DRAM Counter

在高速网络环境中如何有效地存储和维护大量的计数器,已经成为一个重要的研究方向,在网络性能监控、网络管理、入侵检测及流量工程等应用中也显得尤为重要.随着数据流技术的发展,大量高速计数器的维护引起学者的广泛关注.数据流算法将工作内存组织为一个概要数据结构(sketch),用来捕获尽可能与统计估计相关的信息.对不同的统计估计需要不同的 sketch,sketch 由计数器数组构成,有一个共同的在线操作(hash and increment).在高速链路上,巨大的网络流量使得数据流算法需要大量的计数器,某些计数器取值较大,因此,计数器在低速 DRAM 的存储和维护不适用于高速链路,而计数器在高速 SRAM 中的存储和维护满足高速链路.Shah 等人[26]提出了 Hybrid SRAM/DRAM Counter 结构,在此基础上,该计数器结构在文献[27]中得到进一步的改进.基于 Hybrid SRAM/DRAM 结构的两种算法在 SRAM 的使用上均获得了显著的减少,后者明显优于前者.尽管后者比前者更简单、高效,但在计数器管理算法[27]的实施上比较复杂.Zhao 等人[28]所提出的新的 Hybrid SRAM/DRAM Counter 结构在 SRAM 使用上是最优的,具有极为简单的控制逻辑,该算法满足高速链路的速度和存储要求.

4.3 Count-Min Sketch

Count-Min Sketch[29]是一个次线性空间数据结构.Count-Min Sketch 由二维数组构成,它的宽为 w,深为 d,数组的每个元素表示一个计数,即 $count[1,1],\ldots,count[d,w]$.数组的每个元素初始化为 0,$d$ 个相互独立的 Hash 函数被均匀、随机地选择.当更新(i_t,c_t)到达时,表项 a_{i_t} 被更新,c_t 被增加到每行的一个计数,如图 2 所示.计数器是由 Hash 函数 h_j 决定的,表示为

$$count[j,h_j(i_t)]\leftarrow count[j,h_j(i_t)]+c_t.$$

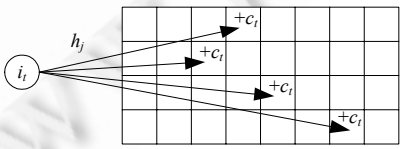


Fig.2 Structure of Count-Min Sketch

图 2 Count-Min Sketch 的结构

Count-Min Sketch 所需要的存储空间由一个二维数组和 d 个 Hash 函数构成,二维数组需要 wd 个字的存储

空间,每个 Hash 函数需要 2 个字的存储空间.Count-Min Sketch 允许在数据流概要中进行基本的查询,如点查询、范围查询和内积查询,同时也可以用于解决数据流中重要的难题,如查找分位数、识别大流.利用 Count-Min Sketch 解决这些难题,所需要的时间和空间界限显著提高.Count-Min Sketch 相当简单,已经应用于数据流的变化检测之中.Count-Min Sketch 的不足之处在于,无法用来计算数据流的熵.

4.4 Bloom Filter

Bloom Filter[30]是一种简单、高效的随机数据结构,利用一个 m 位的数组表示一个集合 $S=\{x_1,x_2,\ldots,x_n\}$,初始化为 0.Bloom Filter 使用 k 个独立的 Hash 函数 $h_1,h_2,\ldots,h_k$,Hash 函数的取值范围为 $\{1,2,\ldots,m\}$,主要包括初始化、元素插入和元素查询过程,具体实施方法如图 3 所示.对任意一个元素 $x\in S$,Hash 函数 h_i 映射到数组的位置 $h_i(x)$ 就会被置为 1($1\leqslant i\leqslant k$).如果一个位置多次被置为 1,那么只有第 1 次会起作用.在查询过程中,对 y 进行 k 次 Hash,如果数组中所有 $h_i(y)$的位置都是 1($1\leqslant i\leqslant k$),则认为 $y\in S$;否则,认为 y 不属于 S.对于一些应用,只要误报率足够低,则误报是可接受的,如图 2 所示中的查询过程,y_1 不是集合中的元素,y_2 属于这个集合或是一个误报.随着网络测量中数据量的飞速增长和有限的计算空间,Bloom Filter 及其变体在网络测量中得到广泛应用[30].Bloom Filter 有一些变化形式,如 Space-Code Bloom Filter[31],Counting Bloom Filter[32],Compressed Bloom Filter[32],Spectral Bloom Filter [33],Generalized Bloom Filter[34].

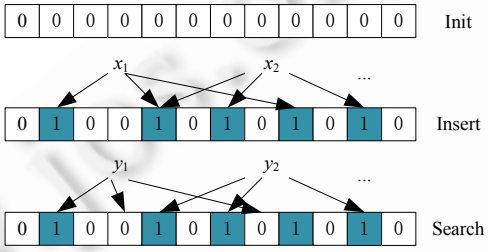


Fig.3 Demonstrations of Bloom Filter

图 3 Bloom Filter 示例

Space-Code Bloom Filter(SCBF)利用 Bloom Filter 对数据进行大量的压缩以降低存储要求,同时,它通过多个解析度的设计来保证根据压缩后的数据能够估计流量数据中每个流包含的分组数.SCBF 以低存储、计算复杂性获得了合理的测量精度.Space-Code Bloom Filter 采用多组 Hash 函数,每组包含多个 Hash 函数,并通过这些 Hash 函数对流的关键字进行 Hash 计算.根据计算结果,Bloom Filter 中对应的位被置为 1.由于 Hash 函数是随机选择的,对于一个流,它的每个分组到来时可能会选择同一组 Hash 函数或不同组的 Hash 函数.但从概率上看,如果一个流包含的分组越多,那么被它选择的 Hash 函数组越多,Bloom Filter 中也就有更多的对应位被置为 1.为了解决这个问题,可以采用多个 Space-Code Bloom Filter,每个具有不同的解析度,即 Multi-Resolution Space-Code Bloom Filter(MRSCBF),每个 SCBF 对某一范围值(流的大小)有较高的精度.因此,对于任意大小的流,都有一个适合的 SCBF,使得对它的估计达到一定精度.

在 Bloom Filter 中,插入元素是容易的,而不能通过逆过程删除一个元素.如果对元素进行 k 次 Hash 运算,将对应的位置为 0,其他的元素可能也被 Hash 到该位置,因此,Bloom Filter 不再正确地反映集合的所有元素.Counting Bloom Filter[32]克服了 Bloom Filter 的不足,在 Counting Bloom Filter 中,每个记录不是一个单独的位,而是一个小的计数器.当一个项被插入时,相应的计数器增加;当一个项被删除时,相应的计数器减小.

研究表明,宽度为 4 位的计数器应该足够满足大部分应用.Compressed Bloom Filter[35]降低了 Bloom Filter 的误报率,同时减少了每个项传输的位数.Spectral Bloom Filter[33]使得 Bloom Filter 存储近似的多重集,并且支持频数查询.有效负载分配是 Bloom Filter 在网络测量中的另一个应用领域,有效负载系统的优点直接与有效负载的实际源、目的的不确定性的减少量有关.当前的互联网架构允许恶意的主机伪装源地址发动 DoS 攻击,IP 回溯法是鉴别恶意主机的有效方法.IP 回溯法主要包括两种类型:以概率标记具有部分路径信息的分组;以

Bloom Filter 的形式存储分组概要,通过迭代检查邻近的路由器重建攻击路径.Generalized Bloom Filter(GBF)[34]解决了无状态的单包 IP 回溯问题,以牺牲漏报率为代价,利用内置的保护抵制 Bloom Filter 被篡改.Bloom Filter 及其变体广泛应用于多种网络系统,如 Web 代理与缓存、数据库服务器、路由器[30].

4.5 Counter Braids

细粒度的网络测量要求网络设备以高速链路速率更新大量计数器.简单的方法需要 SRAM 存储计数器和流-计数器关联规则,使到达的分组能够以链路速率更新相应的计数器,导致准确的流测量变得复杂且昂贵,促进了检测与测量大流的近似算法.统计计数器设计的应用和困难已经引起研究人员的广泛关注.两种主要方法是:利用 Hybrid SRAM/DRAM 结构准确地计数[27,28];利用流长分布的重尾特性近似计数[15].Lu 等人[36]提出一种计数器架构,即 Counter Braids.Counter Braids 有一个分层的结构:第 l 层由深度为 d_l 位的 m_l 个计数器构成.令总层数为 L,在实际应用中,L=2.状态位位于第 1 层计数器,对应的计数器首次溢出,状态位被置为 1.状态位占据额外的空间,但为信息传输解码器提供了有用的信息.进一步减少了第 2 层计数器的数量,在空间上获得一种均衡.在计数器与第 1 层计数器之间,以及第 1 层计数器与第 2 层计数器之间,利用相同的随机映射,如图 4 中虚线箭头所示.

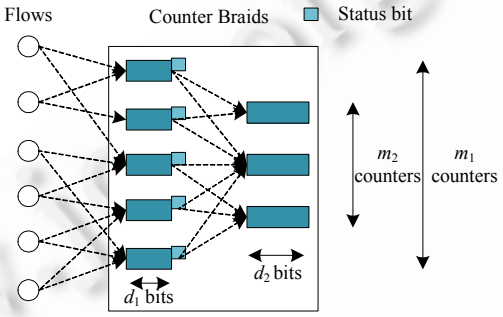


Fig.4 Two-Layer Counter Braids

图 4 两层 Counter Braids

通过随机图编织分层的计数器,解决了流测量的计数器空间和流-计数器的关联问题.通过共享流间计数器,显著减少了存储空间.利用随机图避免了流-计数器关联的存储.Counter Braids 是渐近最优的,该算法能够获得最大的压缩率.一种低复杂度信息传输解码算法,能够以零误差恢复流长,从而可以在硬件中实施.Braids Counter 的缺点在于不支持流长的瞬间查询.

4.6 BRICK

为了能够适应高速网络,Ramabhadran 等人[27]提出了 Hybrid SRAM/DRAM 计数器架构,显著减少了 SRAM 开销,但同时也导致了通过系统总线 SRAM 与 DRAM 之间流量增加的问题.被动计数器满足许多网络监控应用,还有许多应用需要活跃计数器,计数器需要频繁地读出.在网络数据流算法[15,17,29,37]中,当分组到达时,需要读出计数器值,然后采取下一步的操作.为了有效地维护准确的活跃计数器,Zhao 等人[17]提出一种新的计数器架构,即 BRICK(bucketized rank indexed counters),完全在 SRAM 中构建,每秒能够处理大量的分组,也不会产生 SRAM 与 DRAM 之间的流量.该架构在 SRAM 中有效地存储宽度变化的计数器数组,支持快速更新和查询,也能够在硬件或软件中实施.被动计数器对许多网络监控应用是足够的,而一些应用需要维护活跃计数器.例如,如果 Count-Min Sketch[29]应用到大流检测,对每个分组需要读出计数器值,因为该读数将决定一个流是否需要插入到优先队列中.Stanojevic 等人[38]鉴别维护活跃计数器的数据流算法,包括用于大流检测的 Multistage Filters[15]、在线分层的大流识别算法[39].准确的活跃计数器将会节省这些应用的存储开销.BRICK 的基本思想是基于统计复用.把计数器数组分成计数器数目相等的组,每组的计数器随机地从计数器数组中选择.因此,每组的计数器具有变化的宽度.假设计数器数组中计数器的平均宽度为 γ,根据大数定律,在绝大多数分组中,计数

器的总宽度接近γ与每组计数器数目的乘积.实际上,基准方法很难在硬件中实施,主要有两个原因:能够容易随机访问任意的计数器;非前缀编码技术用变长度的符号代替计数器值,使得存储空间更小,同时导致访问与修改数据的开销更大.BRICK 以稍微多一点的总 SRAM 费用克服了这些困难.BRICK 的关键技术是索引策略,即 rank indexing[40]与更新该数据结构对 ASIC 实施不仅是简单的,而且通过内置的指令得到当前处理器的支持,使得软件实施是有效的.因此,该方法能够在硬件或软件中得以有效实施.

5 基于抽样与数据流方法的应用

抽样与数据流方法是高速网络流量测量的重要方法,广泛应用于大流识别、流长分布估计等.本节主要从大流识别、流长分布估计、异常检测和超点检测应用角度介绍高速网络流量测量方法,并比较不同方法的优劣.

5.1 大流识别

在互联网中,少量的大流占据了网络流量的大部分,它们对于网络计费、流量工程等应用非常重要.在高速网络环境下,存在巨大的网络流,缓存所有流信息需要较大的内存.由于在路由器等网络设备中 SRAM 非常昂贵,而 DRAM 又无法达到线速的要求,因此,抽样与数据流技术应用在高速链路上的大流识别显得尤为重要.

抽样技术已经广泛应用于大流识别.Duffield 等人[41]提出 Smart Sampling 算法,该算法是一种针对流记录的非均匀抽样.Smart Sampling 的基本思想是:对象 x 的抽样概率函数 $p_z(x)=\min\{1,x/z\}$,流长 x 大于阈值 z 的流以概率 100%被抽样,而流长 x 小于阈值 z 的流以概率 x/z 被抽样.小流被抽样的概率与其大小有关,同时通过抽样阈值 z 控制 Smart Sampling.重正化函数 $r_z(x)=\max\{x,z\}$,其对小流产生偏大的估计 z,对于一些应用,可能认为是不利的,如流量计费,过高地估计了用户使用的网络流量.然而,若计费策略与阈值抽样相结合,则能够提高流量估计的精度.z 越大,被抽样的对象数越少,误差越高;z 越小,被抽样的对象数越多,误差越低.与均匀抽样相比,Smart Sampling 对大流的估计具有更高的准确性.

Cristian 等人[15]提出两种大流识别方法,即 Multistage Filters,Sample and Hold.Multistage Filters 和 Sample and Hold 具有相似的性能,Sample and Hold 的优势在于实施简单;而 Multistage Filters 的优势在于高精度,但却更加复杂.与 Sampled NetFlow[7]相比,在相同大小的 SRAM 下,Multistage Filters 和 Sample and Hold 的相对误差与 SRAM 的大小成反比,而 Sampled NetFlow 的相对误差与 SRAM 大小的平方根成反比,因此,Multistage Filters 和 Sample and Hold 具有更高的精度,Sampled Netflow 具有更少的访问内存的次数.它们的不足之处在于:被识别大流的数量受到 SRAM 存储大小的限制.Mori 等人[42]提出一种大流识别方法,该方法由两个阶段组成:首先,利用截断 Pareto 分布从抽样流推断原始流长分布;然后,通过 Bays 定理识别大流.该方法提供一个灵活的架构,在给定的抽样率下,使得误报与漏报达到一个合理的均衡.与 Sample and Hold 和 Multistage Filters 相比,该方法不需要处理每个分组,因此适合于部署在高速网络中.多数大流识别方法的不足表现在:不能准确地估计流长或不能维护所有大流的流记录.Lall 等人[43]提出一种 Hybrid SRAM/DRAM 算法,通过 SRAM 中的 Spectral Bloom Filter 数据结构维护每个流的近似计数,利用该近似值,以更大的概率对中流和大流进行抽样,将抽样的分组存储到 DRAM 中的流表.与 Sample and Hold 和 Multistage Filter 相比,该方法能够精确地识别所有的大流和中流.

在流量测量中,每种流方法缺乏可扩展性.在高速链路上,网络流数是巨大的,绝大部分网络流量是由少数大流产生的,而少部分网络流量由多数小流产生.因此,减小小流的抽样率可能是避免误报的一种合理的策略.Sample and Hold 的不足在于准确性和存储要求的均衡,为了获得合理的准确性,必须提高抽样率,从而导致小流识别增加了存储开销.在 Sample and Hold 和 Multistage Filters 中,需要的缓存依赖于它们阻止进入流缓存的小流数量.因此,Raspall 等人[44]提出 S^3(shared-state sampling),S^3 是 Sample and Hold 和 Multistate Filters 的扩展.S^3 的优势在于:适应于现有的存储技术,允许在 DRAM 中部分实施.S^3 的关键问题在于:如何降低小流的检测率,同时不影响大流的检测率.S^3 的不准确性来源于抽样的不确定性和冲突的影响.SHa[45]与 S^3 的不同之处在于:SHa 仅更新每个抽样分组的流记录.该方法有两个显著的特性:其一,不需要处理每个分组,根据抽样率可以调整计算开销,使得该方法具有轻量级的特性及可扩展性和灵活性,从而可以在 DRAM 中实现;其二,存储大小不受流量变化的影响,使得该方法适合于流量工程.另外一个优势在于容易配置.该方法能够有效地识别大流和准确地

估计其大小.基于 Bays 定理,Mori 等人[46]提出基于周期分组抽样的大流识别方法,与其他大流识别方法的不同之处在于,它的通用性和不需要处理每个分组,这样可能减少实施费用和操作开销.在给定的抽样率下,该方法使得误报率和漏报率达到合理的均衡.该方法的不足之处在于:无法获得原始流长的概率分布.

在高速网络环境下,由于受到计算和存储资源的限制,准确、实时地识别大流对于检测大规模网络安全事件具有重要的意义.随着骨干网链路带宽的增加和应用类型的多样化,海量的网络流给网络流量的测量与分析带来极大的困难.抽样技术成为减少存储和时间复杂性的有效方法.抽样技术在大流识别方面已取得一些研究成果.为了进一步提高大流识别的精度和实时性,数据流技术在大流识别中也取得了一些研究成果.Zhang 等人[47]提出一种新的加权数据流频繁项挖掘算法,能够提供单数据项最坏处理时间为 $O(1)$的处理速度.采用一个部分排序的数据结构 POSS(partially-ordered-stream-summary),能够在保证处理速度的同时,尽量降低算法的存储开销.Alon 等人[48]的研究结果表明:利用次线性于数据流的不同元素个数的存储空间中,不能准确地得到大流的大小;然而,有限的存储能够近似估计大流的流长[15].Lossy Counting[49]是一种基于计数器的大流识别算法,误差界限对于表中不同元素的流标识是该算法的一个重要参数,重要性在于小误差界限的元素移走的可能性高于大误差界限的元素.基于该重要性和网络流长的重尾分布特征,Dimitropoulos 等人[50]提出 Probabilistic Lossy Counting 算法,存储开销低于 Lossy Counting 与 Multistage Filters,减小了 Lossy Counting 的误报率,尽管估计误差稍微高于 Lossy Counting,但仍然较低.Babcock 等人[51]从分布式环境角度提出大流识别方法,在不同的位置观察多个数据流,找到全局上排序靠前的大流,同时使得不同位置之间的通信开销最小化.表 1 比较了大流识别算法的性能,其中,$\hat{n}_e$ 表示识别的大流数,n_e 表示识别的大流中真实流数,N_e 表示实际的大流总数.

Table 1 Comparisons of performance for heavy hitter identification algorithms

表 1 大流识别算法的性能比较

References	Heavy hitter identification algorithms	Performance evaluation	
		Accuracy	Memory consumption
Ref.[15]	Multistage Filters, Sample and Hold, Sampled NetFlow	MF and S&H has higher accuracy than Sampled NetFlow	MF and S&H reduces memory overhead
Ref.[41]	Smart Sampling, Uniform Sampling	S&S has higher accuracy than Uniform Sampling	–
Ref.[44]	Shared-State Sampling, Sample and Hold, Multistage Filters, Sticky Sampling, Lossy Counting	LC and S^2 has higher accuracy than S&H, MF and S^3	MF or S^3 requires the minimal flow memory
Ref.[45]	Sample and Hash	S&H identifies heavy hitters with high probability and accurately measuring their size	S&H requires constant memory
Ref.[46]	Periodic Sampling	$R_{f+}=1-n_e/\hat{n}_e, R_{f-}=1-n_e/N_e$	–
Ref.[47]	WLC, OWLC, Space Saving, Lossy Counting	WLC has similar accuracy with other algorithms	WLC requires much less memory than the theoretical bound in practice
Ref.[50]	Probabilistic Lossy Counting, Lossy Counting, Multistage Filters	PLC has similar accuracy with MF; PLC has slightly worse accuracy than LC	PLC uses fewer memory than LC and MF
Ref.[52]	CCBF, Multistage Filters	CCBF has higher accuracy than MF under the same memory	–

5.2 流长分布估计

流长分布对于流量建模和网络管理是非常重要的测度.流长分布有助于服务提供商推断网络的使用模式;流长分布有助于检测引起全局的网络动力学模式变换的事件;流长分布有助于检测各种互联网安全攻击.对于流长分布估计,简单的方法是使用一个由每个流的计数器构成的 Hash 表,跟踪所有的活跃流,通过计数器值估计流长分布.虽然该方法比较简单,但不适用于高速链路.另外一种方法是选择少量的分组,然后从抽样的流量推断流长分布[53],该算法能够从抽样数据获得尽可能多的信息,而它的估计精度受到低抽样率的影响.

基于分组抽样的流长分布估计精度不高,虽然 FS(flow sampling)获得大量的统计信息,但需要消耗大量的存储和处理资源.Dual Sampling[54]在使用相同的存储处理资源下,能够提供类似于 FS 的网络流统计信息.该抽

样方法获得优于其他分组抽样方法的流长分布估计精度,但却不如流抽样算法.由于通过分组抽样方法获得的所有数据中大部分分组来源于大流,仅有少部分分组来源于小流和中流,小流和中流的信息丢失显著地影响各种网络统计信息估计的精度.然而,只有储存每个流的信息才能获得每个流的准确大小,在高速链路上所付出的代价是相当高的.SGS(sketch-guided sampling)[39]通过牺牲大流的抽样率来提高小流和中流的抽样率,获得了更准确的各种网络统计信息估计.该方法的不足之处在于:采用简单的 Hash 表对所有流长作近似估计,导致空间效率低,小流的流长估计误差较大.虽然利用两级存储结构的统计计数器[26]和 MRSCBF[31]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施代价较高,并影响了其部署的灵活性.

多数抽样方法存在小流估计精度不高的问题.最简单的分组抽样方法 Static Sampling(SS)对所有流使用相同的抽样率 p 对分组进行抽样.研究结果表明:抽样率 p 越小,相对误差越大,因此,小流不能够准确地被估计;而抽样率越大,相对误差越小,导致存储空间增加与抽样的目的相矛盾.根据计数器值调整抽样率,Adaptive Non-Linear Sampling(ANLS)[9]能够解决小流估计精度不高的问题.该抽样方法显著地提高了流长估计精度(尤其是对于小流而言),节约了存储开销;同时,在相同的估计精度下使用较小的存储空间,流长分布对估计精度没有影响.Yang 等人[55]提出基于分组抽样的流长分布估计算法,该算法有效地克服了抽样带来的有偏性,实现了流长分布与活跃流数估计的分离.Ribeiro 等人[56]利用抽样分组的 Fisher 信息测度估计流长分布.网络流长分布具有重尾特性,流长分布依赖于其尾指数,绝大部分算法通过简化假设仅获得近似估计.Loisean 等人[57]利用抽样数据获得准确的流长分布尾指数的极大似然估计,从而有利于流长分布的精确估计.

Kumar 等人[37]提出适用于高速链路的数据流算法,该算法由在线流模块和离线处理模块构成,如图 5 所示.在线流模块中,使用由一个计数器数组构成的数据结构,具有较低的存储和计算复杂性;在离线处理模块中,利用在线流模块获得的计数器值能够准确地估计流的总数,而因 Hash 冲突难以准确估计流长分布.因此,利用 Bays 统计方法推断流长分布估计.与基于分组抽样方法相比,该方法获得高于一个数量级的估计精度.在此基础上,提出了流长分布的多分辨率估计方法,当流的总数远多于计数器数组的大小时,该方法获得的流长分布估计精度稍微下降,从而使得在平均情况下,流长分布估计准确且存储有效;而在最坏情况下,流长分布精度仅稍微有所下降.

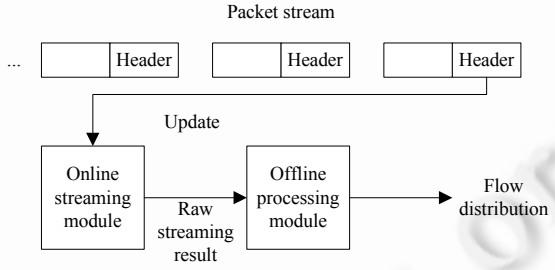


Fig.5 Flowchart of flow size distribution estimation based on data streaming

图 5 基于数据流的流长分布估计的流程图

5.3 异常检测

互联网规模的增大、应用类型的多样化以及网络安全事件都会造成网络流量异常.由于互联网流量每年呈上升趋势,如何对高速骨干网络流量进行实时监测、及时地发现网络流量异常并追踪定位异常源,并做出合理的响应,是保证网络有效运行的重要措施.异常检测主要分为两类:基于特征的检测和基于统计的检测.两类检测方法各有优、缺点:基于特征的检测方法的优点是能够准确地检测已知的异常,其缺点是不能检测未知的异常,同时需要预先设定特征库,特征库的规模将影响检测性能;基于统计的检测方法的优点是能够准确地检测已知的和未知的异常,同时不需要预先设定特征库.因此,基于特征的检测不适用于高速骨干网络[58].为了满足在高速网络环境下实时处理的要求,抽样与数据流技术已经成为可扩展互联网流量测量与异常检测的重要组成

部分.

分组抽样对网络异常检测的影响已得到广泛研究.Mai等人[59]评价了分组抽样对3种端口扫描检测算法的影响,表明分组抽样降低了3种算法的检测率,增加了误报率.Brauckhoff等人[60]评价了分组抽样对异常检测测度的影响,表明通过分组抽样能够获得准确的字节数和分组数估计,而不能够获得准确的流数估计.然而,特征熵测度受到分组抽样的影响较小,甚至在高抽样率下能够有效检测Blaster worm.由于分组抽样降低异常检测率,Ali等人[61]提出渐进的安全感知分组抽样方法,通过抽样更多的恶意分组,使异常检测器获得更高的检测率.该算法具有较低的复杂性,没有通信开销.与随机分组抽样相比,该算法对所有的异常检测算法均有较高的检测率.程光等人[62]在高速网络环境下建立了基于抽样的实时异常检测模型,实现了在系统资源可控范围内检测异常行为.

在高速链路上,流抽样是非常有吸引力的、可扩展的获取流统计信息的测量方法.流抽样对网络异常检测的影响已经得到广泛的研究.研究表明,网络异常产生大量的小流,如network scans,SYN flooding,worms.Mai等人[63]分析了流抽样技术对流量异常与端口扫描异常检测算法的影响,表明Smart Sampling与Sample and Hold能够准确地估计大流,而显著地降低流量异常和端口扫描检测精度.Androulidakis等人[64]评价与分析了两级抽样技术[55]对网络异常检测的影响.抽样数据是不全面的、有偏的原始流量的近似.由此,提出增强的两级选择抽样方法,通过智能的流抽样优先选择小流.甚至在小攻击、小抽样率下,该算法显著地提高了异常检测精度,而在大部分情况下甚至优于未抽样算法.Androulidakis等人[65]提出的流抽样算法侧重于选择小流,提高了异常检测性能,同时减少了被选择的流数.Androulidakis等人[66]通过基于熵的异常检测算法,评价机会流抽样对不同网络异常的影响.因此,观察不同类型异常的网络流量特征、选择合适的抽样方法优先地抽取流量数据,进一步减少了抽样数据,能够有效地提高异常检测性能.

随着网络带宽的快速增长和新攻击、病毒、蠕虫的不断出现,传统入侵检测系统中的异常检测算法无法满足高速网络的要求.Li等人[67]通过数据流技术设计了一个流级入侵检测系统,与传统入侵检测系统相比,该系统可扩展到高速网络流级检测,能够识别SYN泛洪与各种端口扫描,能够进行基于多点的聚合检测,从检测中分离异常,减少误报.郑军等人[68]提出一种基于数据流的大规模网络异常发现算法,首次将数据流模型用于大规模网络的异常发现.罗娜等人[58]提出一种基于概要数据结构的异常检测方法.基于观测值与预测值之间的差异,sketch采用均值均方差模型建立网络流量变化参考模型.该方法能够检测DDoS、扫描等攻击行为,并能追溯异常的IP地址,利用少量的计算与存储资源,因此适用于高速骨干网络上的异常检测.郑黎明等人[69]利用数据流中概要数据结构的思想,提出Filter-ary-Sketch数据结构,在该数据结构上采用基于熵的异常检测算法在骨干网上进行异常检测.该算法能够检测多种类型的网络攻击,且能有效地进行恶意流量阻断,但缺乏理论上的精度保证.

5.4 超点检测

在高速链路上实时、准确地检测网络安全事件,如DDoS攻击、端口扫描、蠕虫传播等,对网络安全和网络管理具有重要意义.这些安全事件具有类似的行为特征,如:端口扫描和蠕虫传播是由在短时间内源主机与不同目的主机建立大量的连接引起的,而DDoS攻击是由大量的主机泛洪到一个目的主机引起的.它们的共同特点是:源(目的)主机发送或接收到大量来自于不同目的(源)主机的连接.超点是指在测量时间内与大量源(目的)主机连接的目的(源)主机,超点检测是指检测在测量时间内发送或接收大量流数的源(目的)主机.维护流存在的状态信息和地址的状态信息,是超点检测的难点.

抽样与数据流方法已经应用于高速链路上的超点检测.Venkataraman等人[18]通过存储与处理少量的网络流量,极大地减少了所需的存储空间并降低了计算复杂度,该抽样方法的准确性很大程度上依赖于抽样率.Kamiyama等人[70]提出基于流抽样的超点检测方法,在给定的内存大小和处理时间要求下,根据流量模式变化,该方法能够自适应地优化Bloom Filter和Host Table参数,从而有效地检测超点.研究发现,基于哈希流抽样算法不能有效地扩展到更高速的网络环境中(2.5Gbps以上).针对该问题,王洪波等人[71]提出一种基于Bloom Filter流抽样的超点检测算法,在高速网络环境下,能够快速、准确地检测超点.该算法的不足之处在于:需要专用的硬

件以及高速存储器.程光等人[72]提出一种具有自适应抽样功能的超点实时检测算法,该算法结合多种网络测量技术,表明在自适应性、资源可控性、测量精度等方面优于 Sampled 和 Bitmap 等算法,能够实现高速网络上超点高精度实时检测.Shi 等人[73]提出基于抽样与数据流技术的在线架构,用于检测 Top Spreaders 与 Top Scanners.该算法提高了检测精度,同时减少了内存使用和 CUP 处理时间.Zhao 等人[74]提出两种基于抽样与数据流方法的超点检测方法:第 1 种超点检测算法是在基于 Hash 的流抽样算法的基础上提出来的,通过数据流模块进一步过滤抽样流量,允许更高的抽样率,获得更高的检测精度;第 2 种超点检测算法结合了数据流在高效保存、估计与已知源/目的相关的扇出/扇入的能力和抽样在产生候选源/目的的列表的能力,虽然该算法更加复杂,但却获得了更好的检测精度.Wang 等人[20]对该数据结构[13]加以改进,提出了一种超点检测的有效保留算法.虚拟索引方法 VCDS(virtual connection degree sketch)[22]应用于超点检测.由于 VCDS 需要大量的额外内存来存储节点地址,而新的数据结构 RVCDS(reversible virtual connection degree skecth)识别超点地址,不需要额外的内存空间,只是估计误差略微有所增加.此外,将 VCDS、RVCDS 与均匀流抽样技术相结合,能够有效地减少内存复杂度.

6 高速网络流量测量技术的研究成果

本节概述了目前国内外高速网络流量测量技术的主要研究成果,并从评价指标方面比较了各种网络流量测量技术的测量精度和实施性.研究表明,绝大多数网络流量测量技术获得了较高的测量精度,具有可实施性.

随着带宽的快速增长和数据流的急剧增加,高速网络流量测量技术成为必然的发展趋势,已经引起许多研究机构和学者的广泛关注.抽样和数据流方法是高速网络流量测量技术的重要组成部分,广泛应用于网络管理、网络安全等应用.虽然分组抽样方法降低了存储和处理开销,但同时也给网络流量测量带来了许多不确定性,如降低异常检测的检测率、不准确的流的大小估计.由于构成网络流量的分组并不是孤立的,它们是为了完成应用功能而产生的,它们之间存在着一定的关联.因此,流抽样方法广泛应用于大流识别、流长分布、异常检测以及超点检测之中,均获得了较好的性能.尽管抽样方法产生了一个原始数据的代表子集,每个被抽样对象都确实存在于原始数据中,但是,从这些数据推断出的关于原始对象的结论却并不一定准确.如果能够从原始数据产生一些概要信息,这些信息能够支持一些常规的查询,而且其结果具有相同或者更好的准确性,同时,这种方法需要更小的存储或计算需求、具有更小的响应时间.于是,数据流方法应运而生.数据流方法具有执行一趟计算,而且只需要使用有限的计算和内存资源的特性,广泛应用于流长分布估计、熵估计、连接度估计等.

实际上,每种网络流量测量方法是针对具体的应用而提出的,具有一定的局限性,因而,多种网络流量测量方法的有效结合,有助于提高算法的测量精度.两种分组抽样方法的结合[54]、分组抽样和流抽样方法的结合[75]、抽样方法与数据流方法的结合[73,74,76],均有利于提高算法的性能.抽样方法和数据流方法相结合,已经应用于网络测量与网络安全之中.研究发现,抽样方法和数据流方法适用于捕获信息谱中不同和互补的区域,抽样方法和数据流方法的结合,能够恢复完整的信息.在此基础上,Zhao 等人[74]提出基于抽样与数据流方法的超点检测方法.许多研究主要集中在总体流的流长分布估计[37,53]上,而估计子总体流的流长分布是更具挑战性.子总体流的流长分布估计主要有两个方面的困难:其一,不能事先知道子总体流;其二,子总体流的数量是巨大的. Kumar 等人[76]提出任意子总体流的流长分布估计算法,该算法由两个并行的数据采集模块和一个统计估计模块构成.抽样模块类似于 NetFlow 的分组抽样,数据流模块由一个计数器数组构成.利用抽样模块和数据流模块采集的数据,通过统计估计模块获得准确的子总体流的流长分布估计.

7 讨 论

7.1 主要问题

虽然网络流量测量的研究取得了显著的进展,但大部分属于理论分析,与实际应用还存在一定的差距.目前,我们认为网络流量测量还存在以下几方面的问题:(1) 抽样方法对数据进行有效压缩,同时又能保留流量的原始特征信息,有效地缓解了网络测量的处理和存储困难,然而估计却存在较大误差;(2) 存储技术的发展滞后

于高速链路速率的增长,导致无法满足一些网络应用的实时性要求;(3) 网络流量测量的可扩展性问题,现有的许多网络测量方法往往是针对具体的应用需求,不能够扩展到高速网络环境,造成了许多资源的低效利用;(4) 网络流量测量方法评价指标量化的问题,评价指标的合理量化值得深入研究;(5) 小流的识别问题,在实际的网络环境中,许多攻击是由小流组成的,高速链路上网络流量中大部分是小流,无法实时地监测每个小流;(6) 高速网络环境下并发连接度检测问题;(7) 分布式网络测量问题;(8) 高速链路上数据流入侵检测系统的构建问题.

7.2 发展趋势和未来的研究方向

综合上述讨论,我们认为,高速网络流量测量技术的发展趋势和未来的研究方向包括:(1) 可重构硬件成为网络设备未来的发展趋势,可重构技术使得网络流量测量与分析功能以组件的形式集成到网络设备中成为可能,目前,Cisco 的 NetFlow、InMon 的 sFlow 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,而可重构硬件组件能够满足高性能的需求;(2) 无线网络的流量测量方法研究,现有的有线网络测量方法在无线网络测量中的适用性、可扩展性以及针对无线网络特性的新测量方法研究;(3) 随着移动通信网络和 Web 技术的发展,以微博为代表的在线社交网络已成为人们信息共享和舆论传播的重要媒介,利用网络测量方法了解其网络拓扑特征与用户行为特征等基本属性,已成为网络流量测量研究与发展的趋势之一;(4) 利用云计算平台强大的数据处理能力来处理海量网络流量,提高网络流量测量的效率,已成为网络流量测量研究与发展趋势之一;(5) 网络流量测量方法应具有可扩展性的能力,以满足多种网络应用需求;(6) 针对高速网络中节点连接度检测问题,在深入分析高速链路上数据流特性的基础上,设计出更高效的数据结构来组织数量庞大的并发连接记录,以支持更快的查询等操作;(7) 针对现有的大部分抽样方法对小流估计存在的较大误差,自适应的抽样方法动态地调整抽样率,有利于提高估计精度;(8) 抽样方法和数据流方法通常适合于获取信息谱中不同的和互补的信息域,抽样方法与数据流方法相结合,能够有助于恢复完全的信息;(9) 针对小流的检测问题,可以通过流抽样和数据流方法减少处理和存储开销;(10) 分布式网络测量实施;(11) 构建适用于高速链路上的数据流入侵检测系统.

8 总 结

近年来,随着链路速率的提高和网络应用的多样化,巨大的数据流给网络流量测量与分析带来挑战,高速网络流量测量方法逐渐成为研究的热点之一.高速网络流量测量方法大致经历了一个“报文抽样-流抽样-数据流”的发展历程.高速网络流量测量方法的主要目标是:在保证一定准确性的前提下降低所需要的处理和存储开销.本文将高速网络流量测量方法分为抽样方法和数据流方法,在此基础上,介绍了抽样方法和数据流及其应用,利用统计学方法对获得的流量信息进行推断,分析了它们的各种网络统计信息量的精度、计算和存储复杂度,指出了它们的不足之处.提出了高速网络流量测量方法存在的主要问题、发展趋势以及未来的研究方向.

References:

[1] Cheng G, Gong J. Internet Flow Measurement. Nanjing: Southeast University Press, 2008 (in Chinese).

[2] http://www.edu.cn

[3] Cheng G, Tang YN. Estimation algorithms of the flow number from sampled packets on approximate approaches. Ruan Jian Xue Bao/Journal of Software, 2013,24(2):255−265 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4316.htm [doi: 10.3724/SP.J.1001.2013.04316]

[4] Muthukrishnan S. Data streams: Algorithms and applications. Foundations and Trends in Theoretical Computer Science, 2005,1(2): 117−236. [doi: 10.1561/0400000002]

[5] He GH, Hou JC. On sampling self-similar Internet traffic. Computer Networks, 2006,50(16):2919−2936. [doi: 10.1016/j.comnet.2005.11.009]

[6] Raspall F. Efficient packet sampling for accurate traffic measurements. Computer Networks, 2012,56(6):1667−1684. [doi: 10.1016/j.comnet.2011.11.017]

[7] Estan C, Keys K, Moore D, Varghese G. Building a better netflow. ACM SIGCOMM Computer Communication Review, 2004, 34(4):245−256. [doi: 10.1145/1030194.1015495]

[8] Sanjuàs-Cuxart J, Barlet-Ros P, Duffield N, Kompella R. Cuckoo sampling: Robust collection of flow aggregates under a fixed memory budget. In: Proc. of the 31st Annual IEEE Int'l Conf. on Computer Communications (Mini-Conf.). Orlando: IEEE, 2012. 2751−2755. [doi: 10.1109/INFCOM.2012.6195693]

[9] Hu CC, Liu B, Wang S, Tian J, Cheng Y, Chen Y. ANLS: Adaptive non-linear sampling method for accurate flow size measurement. IEEE Trans. on Communications, 2012,60(3):789−798. [doi: 10.1109/TCOMM.2011.112311.100622]

[10] Ma XY, Hu CC, Jiang JC, Wang J. S3: Smart selection of sampling function for passive network measurement. In: Proc. of the 36th Annual IEEE Conf. on Local Computer Networks. Bonn: IEEE, 2011. 416−423. [doi: 10.1109/LCN.2011.6115368]

[11] Carela-Espanñol V, Barlet-Ros P, Cabellos-Aparicio A, Sol-Pareta J. Analysis of the impact of sampling on netflow traffic classification. Computer Networks, 2011,55(5):1083−1099. [doi: 10.1016/j.comnet.2010.11.002]

[12] Chakrabarti A, Do Ba K, Muthukrishnan S. Estimating entropy and entropy norm on data streams. Internet Mathematics, 2011,3(1): 63−78. [doi: 10.1080/15427951.2006.10129117]

[13] Alon N, Matias Y, Szegedy M. The space complexity of approximating the frequency moments. Journal of Computer and System Sciences, 1999,58(1):137−147. [doi: 10.1006/jcss.1997.1545]

[14] Lall A, Sekar V, Oginara M, Xu J, Zhang H. Data streaming algorithms for estimating entropy of network traffic. ACM SIGCOMM Computer Communication Review, 2006,34(1):145−156. [doi: 10.1145/1140103.1140295]

[15] 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]

[16] Zhao HC, Lall A, Ogihara M, Spatscheck O, Wang J, Xu J. A data streaming algorithm for estimating entropies of OD flows. In: Proc. of the 7th ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2007. 279−290. [doi: 10.1145/1298306.1298345]

[17] Zhao Q, Kumar A, Wang J, Xu J. Data streaming algorithms for accurate and efficient measurement of traffic and flow matrices. ACM SIGMETRICS Performance Evaluation Review, 2005,33(1):350−361. [doi: 10.1145/1071690.1064258]

[18] Venkataraman S, Song D, Gibbons PB, Blum A. New streaming algorithms for fast detection of superspreaders. In: Proc. of the Network and Distributed System Security Symp. San Diego: the Internet Society, 2005. 149−166. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.133.1592

[19] Guan XH, Wang PH, Qin T. A new data streaming method for locating hosts with large connection degree. In: Proc. of the IEEE GLOBECOM Telecommunications Conf. Honolulu: IEEE, 2009. 1−6. [doi: 10.1109/GLOCOM.2009.5426280]

[20] Wang PH, Guan XH, Qin T, Huang QZ. A data streaming method for monitoring host connection degrees of high-speed links. IEEE Trans. on Information Forensics and Security, 2011,6(3):1086−1098. [doi: 10.1109/TIFS.2011.2123094]

[21] Yoon MK. Li T, Chen SG, Peir JK. Fit a spread estimator in small memory. In: Proc. of the 28th Conf. on Computer Communications. Rio de Janeiro: IEEE, 2009. 504−512. [doi: 10.1109/INFCOM.2009.5061956]

[22] Wang PH, Guan XH, Towsley D, Tao J. Virtual indexing based methods for estimating node connection degrees. Computer Networks, 2012,56(12):2773−2787. [doi: 10.1016/j.comnet.2012.03.025]

[23] Estan C, Varghese G, Fisk M. Bitmap algorithms for counting active flows on high-speed links. IEEE/ACM Trans. on Networking, 2006,14(5):925−937. [doi: 10.1109/TIFS.2011.2123094]

[24] Kim HA, O'Hallaron DR. Counting network flows in real time. In: Proc. of the IEEE Global Telecommunications Conf. 2003. 3888−3893. [doi: 10.1109/GLOCOM.2003.1258959]

[25] Sanjuàs-Cuxart J, Barlet-Ros P, Solé-Pareta J. Counting flows over sliding windows in high speed networks. In: Proc. of the Networking. LNCS 5550, 2009. 79−91. [doi: 10.1007/978-3-642-01399-7_7]

[26] Shah D, Iyer S, Prahhakar B, McKeown N. Maintaining statistics counters in router line cards. IEEE Micro, 2002,22(1):76−81. [doi: 10.1109/40.988692]

[27] 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]

[28] 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]

[29] Cormode G, Muthukrisnan S. An improved data stream summary: The count-min sketch and its applications. Journal of Algorithms, 2005,55(1):58−75. [doi: 10.1016/j.jalgor.2003.12.001]

[30] Tarkoma S, Rothenberg CE, Lagerspetz E. Theory and practice of bloom filters for distributed systems. IEEE Communications Surveys & Tutorials, 2012,14(1):131−155. [doi: 10.1109/SURV.2011.031611.00024]

[31] Kumar A, Xu J, Li L, Wang J. Space-Code bloom filter for efficient traffic flow measurement. In: Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2003. 167−172. [doi: 10.1145/948205.948226]

[32] Broder A, Mitzenmacher M. Network applications of bloom filters: A survey. Internet Mathematics, 2004,1(4):485−509. [doi: 10.1080/15427951.2004.10129096]

[33] Cohen S, Matias Y. Spectral Bloom Filter. In: Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2003. 241−252. [doi: 10.1145/872757.872787]

[34] Laufer RP, Velloso PB, Cunha DO, Moraes IM, Bicudo MDD, Moreira MDD, Duarte OCMB. Towards stateless single-packet IP traceback. In: Proc. of the 32nd IEEE Conf. on Local Computer Networks. Dublin: IEEE, 2007. 548−555. [doi: 10.1109/LCN.2007.15]

[35] Mitzenmacher M. Compressed Bloom Filters. IEEE/ACM Trans. on Networking, 2002,10(5):604−612. [doi: 10.1109/TNET.2002.803864]

[36] Lu L, Montanari A, Prabhakar B, Dharmapurikar S, Kabbani A. Counter braids: A novel counter architecture for per-flow measurement. ACM SIGMETRICS Performance Evaluation Review, 2008,36(1):121−132. [doi: 10.1145/1384529.1375472]

[37] Kumar A, Sung M, Xu J, Wang J. Data streaming algorithms for efficient and accurate estimation of flow size distribution. ACM SIGMETRICS Performance Evaluation Review, 2004,32(1):177−188. [doi: 10.1145/1012888.1005709]

[38] Stanojevic R. Small active counters. In: Proc. of the 26th IEEE Int'l Conf. on Computer Communications. Anchorage: IEEE, 2007. 2153−2161. [doi: 10.1109/INFCOM.2007.249]

[39] Kumar A, Xu J. Sketch guided sampling—Using on-line estimates of flow size for adaptive data collection. In: Proc. of the 25th IEEE Int'l Conf. on Computer Communications. Barcelona: IEEE, 2006. 1−11. [doi: 10.1109/INFOCOM.2006.326]

[40] Hua N, Zhao HQ, Lin B, Xu J. Rank-Indexed hashing: A compact construction of Bloom Filters and variants. In: Proc. of the IEEE Int'l Conf. on Network Protocol. Orlando: IEEE, 2008. 73−82. [doi: 10.1109/ICNP.2008.4697026]

[41] Duffield N, Lund C. Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure. In: Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2003. 179−191. [doi: 10.1145/948205.948228]

[42] Mori T, Takine T, Pan JP, Kawahara R, Uchida M, Goto S. Identifying heavy-hitter flows from sampled flow statistics. IEICE Trans. on Communications, 2007,E90-B(11):3061−3072. [doi: 10.1093/ietcom/e90-b.11.306]

[43] Lall A, Ogihara M, Xu J. An efficient algorithm for measuring medium to large-sized flows in network traffic. In: Proc. of the 28th Conf. on Computer Communications. Rio de Janeiro: IEEE, 2009. 2711−2715. [doi: 10.1109/INFCOM.2009.5062217]

[44] 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. [doi: 10.1145/1177080.1177082]

[45] Raspall F, Sallent S. Adaptive shared-state sampling. In: Proc. of the 8th ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2008. 271−284. [doi: 10.1145/1452520.1452552]

[46] Mori T, Uchida M, Kawahara R, Pan JP, Goto S. Identifying elephant flows through periodically sampled packets. In: Proc. of the 4th ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2004. 115−120. [doi: 10.1145/1028788.1028803]

[47] Zhang Y, Fang BX, Zhang YZ. Identifying heavy hitters in high-speed network monitoring. SCIENTIA SINICA Informationis, 2010,53(3):659−676. [doi: 10.1007/s11432-010-0053-5]

[48] Alon N, Matias T, Szegedy M. The space complexity of approximating the frequency moments. In: Proc. of the 28th Annual ACM Symp. on the Theory of Computing. New York: ACM Press, 1996. 20−29. [doi: 10.1145/237814.237823]

[49] Manku GS, Motwani R. Approximate frequency counts over data streams. In: Proc. of the 28th Int'l Conf. on Very Large Data Bases. Hong Kong: ACM Press, 2002. 346−357. http://dl.acm.org/citation.cfm?id=2367502.2367508

[50] Dimitropoulos X, Hurley P, Kind A. Probabilistic lossy counting: An efficient algorithm for finding heavy hitters. ACM SIGCOMM Computer Communication Review, 2008,38(1):5−16. [doi: 10.1145/1341431.1341433]

[51] Babcock B, Olston C. Distributed top-*k* monitoring. In: Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2003. 28−39. [doi: 10.1145/872757.872764]

[52] Wu H, Gong J, Yang W. Algorithm based on double counter bloom filter for large flows identification. Ruan Jian Xue Bao/Journal of Software, 2010,21(5):1115−1126 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3568.htm [doi: 10.3724/SP.J.1001.2010.03568]

[53] Duffield N, Lund C, Thorup M. Estimating flow distributions from sampled flow statistics. In: Proc. of the ACM SIGCOMM. New York: ACM Press, 2003. 325−336. [doi: 10.1145/863955.863992]

[54] Tune P, Veitch D. Towards optimal sampling for flow size estimation. In: Proc. of the 8th ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2008. 243−255. [doi: 10.1145/1452520.1452550]

[55] Yang L, Michailidis G. Sampled based estimation of network traffic flow characteristics. In: Proc. of the 26th IEEE Int'l Conf. on Computer Communications. Anchorage: IEEE, 2007. 1775−1783. [doi: 10.1109/INFCOM.2007.207]

[56] Ribeiro B, Towsley D, Ye T, Bolot J. Fisher information of sampled packets: An application to flow size estimation. In: Proc. of the 6th ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2006. 15−25. [doi: 10.1145/1177080.1177083]

[57] Loiseau P, Goncalves P, Girard S, Forbes F, Primet P. Maximum likelihood estimation of the flow size distribution tail index from sampled packet data. In: Proc. of the SIGMETRICS. New York: ACM Press, 2009. 263−273. [doi: 10.1145/1555349.1555380]

[58] Luo N, Li AP, Wu QY, Lu HB. Sketch-Based anomalies detection with IP address traceability. Ruan Jian Xue Bao/Journal of Software, 2009,20(10):2899−2906 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3995.htm [doi: 10.3724/SP.J.1001.2011.03995]

[59] Mai JN, Sridharan A, Chuah CN, Zang H, Ye T. Impact of packet sampling on portscan detection. IEEE Journal on Selected Areas in Communication, 2006,24(12):2285−2298. [doi: 10.1109/JSAC.2006.884027]

[60] Brauckhoff D, Tellenbach B, Wagner A, May M, Lakhina A. 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. [doi: 10.1145/1177080.1177101]

[61] Ali S, H IU, Rizvi S, Rasheed N, Sarfraz U, Khayam SA, Mirza F. On mitigating sampling-induced accuracy loss in traffic anomaly detection systems. ACM SIGCOMM Computer Communication Review, 2010,40(3):4−16. [doi: 10.1145/1823844.1823846]

[62] Cheng G, Gong J, Ding W. A real-time anomaly detection model based on sampling measurement in a high-speed network. Ruan Jian Xue Bao/Journal of Software, 2003,14(3):594−599 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/20030340.htm

[63] Mai JN, Chuanh CN, Sridharan A, Ye T, Zang H. 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. [doi: 10.1145/1177080.1177102]

[64] Androulidakis G, Papavassiliou S. Two-Stage selective sampling for anomaly detection: analysis and evaluation. Security and Communication Networks, 2011,4(6):608−621. [doi: 10.1002/sec.191]

[65] Androulidakis G, Papavassiliou S. Improving network anomaly detection via selective flow-based sampling. IET Communications Journal, 2008,2(3):399−409. [doi: 10.1049/iet-com:20070231]

[66] Androulidakis G, Chatzigiannakis V, Paravassiliou S. Network anomaly detection and classification via opportunistic sampling. IEEE Network, 2009,23(1):6−12. [doi: 10.1109/MNET.2009.4804318]

[67] Li ZC, Gao Y, Chen Y. HiFIND: A high-speed flow-level intrusion detection approach with DoS resiliency. Computer Networks, 2010,54(8):1282−1299. [doi: 10.1016/j.comnet.2009.10.016]

[68] Zheng J, Hu MC, Yun XC, Zheng Z. Anomaly detection of large scale network based on data streams. Journal on Communicaitons, 2006,27(2):1−8 (in Chinese with English abstract).

[69] Zheng LM, Zou P, Han WH, Li AP, Jia Y. Anomaly detection in backbone networks using filter-ary-sketch. Journal on Communicaitons, 2011,32(12):151−160 (in Chinese with English abstract).

[70] Kamiyama N, Mori T, Kawahara R. Simple and adaptive identification of superspreaders by flow sampling. In: Proc. of the 26th IEEE Int'l Conf. on Computer Communications. Anchorage: IEEE, 2007. 2481−2485. [doi: 10.1109/INFCOM.2007.305]

[71] Wang HB, Cheng SD, Lin Y. On flow sampling for identifying super-connection hosts in high speed networks. ACTA ELECTRONICA SINICA, 2008,36(4):809−818 (in Chinese with English abstract).

[72] Cheng G, Gong J, Ding W, Wu H, Qiang SQ. Super point detection based on adaptive sampling. SCIENTIA SINICA (E: Informationis), 2008,38(10):1679−1696 (in Chinese with English abstract).

[73] Shi XG, Chiu DM, Lui J. An online framework for catching top spreaders and scanners. Computer Networks, 2010,54(9): 1375−1388. [doi: 10.1016/j.comnet.2009.12.003]

[74] Zhao Q, Kuma A, Xu J. Joint data streaming and sampling techniques for detection of super sources and destinations. In: Proc. of the 5th ACM SIGCOMM Conf. on Internet Measurement. Berkeley: USENIX Association, 2005. 77−90. http://dl.acm.org/citation.cfm?id=1251086.1251093

[75] Liu WJ, Gong J. Double sampling for flow measurement on high speed links. Computer Networks, 2008,52(11):2221−2226. [doi: 10.1016/j.comnet.2008.04.003]

[76] Kumar A, Sung M, Xu J, Zegura E. A data streaming algorithm for estimating subpopulation flow size distribution. ACM SIGMETRICS Performance Evaluation Review, 2005,33(1):61−72. [doi: 10.1145/1071690.1064221]

附中文参考文献:

[1] 程光,龚俭.互联网流测量.南京:东南大学出版社,2008.

[3] 程光,唐永宁.基于近似方法的抽样报文流数估计算法.软件学报,2013,24(2):255−265. http://www.jos.org.cn/1000-9825/4316.htm [doi: 10.3724/SP.J.1001.2013.04316]

[52] 吴桦,龚俭,杨望.一种基于双重 Counter Bloom Filter 的长流识别算法.软件学报,2010,21(5):1115−1126. http://www.jos.org.cn/1000-9825/3568.htm [doi: 10.3724/SP.J.1001.2010.03568]

[58] 罗娜,李爱平,吴泉源,陆华彪.基于概要数据结构可溯源的异常检测方法.软件学报,2009,20(10):2899−2906. http://www.jos.org.cn/1000-9825/3685.htm [doi:10.3724/SP.J.1001.2009.03685]

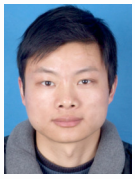
[62] 程光,龚俭,丁伟.基于抽样测量的高速网络实时异常检测模型.软件学报,2003,14(3):594−599. http://www.jos.org.cn/1000-9825/20030340.htm

[68] 郑军,胡铭曾,云晓春,郑仲.基于数据流方法的大规模网络异常发现.通信学报,2006,27(2):1−8.

[69] 郑黎明,邹鹏,韩伟红,李爱平,贾焰.基于 Filter-ary-Sketch 数据结构的骨干网异常检测研究.通信学报,2011,32(12):151−160.

[71] 王洪波,程时端,林宇.高速网络超连接主机检测中的流抽样算法研究.电子学报,2008,36(4):809−818.

[72] 程光,龚俭,丁伟,吴桦,强士卿.基于自适应抽样的超点检测算法.中国科学(E 辑:信息科学),2008,38(10):1679−1696.



周爱平(1982－),男,江苏泰州人,博士生,主要研究领域为网络测量,网络安全.
E-mail: apzhou@njnet.edu.cn

郭晓军(1983－),男,博士生,讲师,主要研究领域为网络测量,网络安全.
E-mail: xjguo@njnet.edu.cn

程光(1973－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络测量,网络安全,网络管理.
E-mail: gcheng@njnet.edu.cn