文章编号: 1001-0920 (2013) 06-0801-07

大 流 识 别 方 法 综 述

夏靖波, 任高明

(空军工程大学 信息与导航学院，西安 710077)

摘 要: 首先介绍大流识别方法的研究动机和国内外发展现状, 根据采用技术的差异, 将现有方法分为基于抽样的方法、基于计数的方法、基于最近最少使用算法的方法和基于哈希的方法, 简要分析了几类方法的优缺点; 然后分别选取经典算法进行剖析, 并给出大流识别的典型应用; 最后探讨了值得进一步研究的问题和可能的发展方向.

关键词: 流量测量；大流；网络管理；流量工程

中图分类号: TP393 **文献标志码:** A

Survey on elephant flow identifying methods

XIA Jing-bo, *REN Gao-ming*

(Institute of Information and Navigation，Air Force Engineering University，Xi'an 710077，China. Correspondent: REN Gao-ming，E-mail：gaomingren_928@126.com)

Abstract: Research motivation and current trends on elephant flow identifying are firstly introduced. New classification of existing elephant flow identifying approaches are proposed, which are sampling methods, count-based methods, LRU-based methods and hash-based methods, and the disadvantages and advantages are compared simply. Then the classic algorithms are analyzed respectively, and typical applications are given. Finally, several problems and their research tendencies in this field are presented.

Key words: network flow measure；elephant flow；network management；traffic engineering

0 引 言

近年来, 计算机网络呈现出向高速化、大规模、复杂化方向发展的趋势[1], 显著特点是产生的数据量大、数据分组到达频率高, 导致单位数据分组的处理时间越来越短, 处理难度越来越大[2], 这便对网络流量测量设备的处理能力提出了更高的要求. 例如, 对于 10 Gbit/s 的链路, 处理一个数据分组需要 32 ns[3]. 但当前用于网络流量测量的硬件或者速度太慢, 或者成本昂贵. 基于成本优化和实际硬件水平的考量, 传统全数据采集的网络流量测量手段已不再适用[4], 如何测量高速网络链路的流量, 成为亟待解决的问题[5].

当前, 网络流量测量通常以流为单位进行, 不同文献中对流的定义不尽相同. 通常定义给定时间内五元组 (源 IP 地址、目的 IP 地址、源端口、目的端口、协议类型) 相同的数据分组为流[6-7]. 研究发现互联网中流的大小服从重尾分布[8-10], 即少数字节数较大的流占据了大部分流量, 而其余的流量由大量字节数较小的流构成[11]. 在实际应用中, 多数情况下只需掌握占据大部分流量的大流信息即可满足需要[12-15]. 因此, 利用有限的硬件资源关注大流, 尽可能收集大流的信息成为一个较好的选择. 如何获取这些关键的大流信息, 即为大流识别问题. 大流识别是计算机网络技术高速发展的必然选择[16], 对网络计费、带宽规划和 Dos 攻击检测[17-19]等网络管理应用意义重大[20].

大流识别问题的理论研究在近几年备受关注. Estan 等首次提出了在网络流量测量中采取“抓大放小”的策略[21], 以放弃小流信息为代价换取存储空间和准确性上的优势[22-23]. 不同文献中的大流称谓有所不同, 网络研究中称为大流量对象、大象流[15,24]或大业务流[25], 流数据研究中称为 heavy-hitter 或频繁项[26]. 目前出现的大流识别方法门类繁多, 经过分析总结, 将其分为 4 类: 基于抽样的大流识别算法、基于计数的大流识别算法、基于 LRU(least recently used) 的大流识别算法和基于哈希的大流识别算法. 基于分位数的大流识别方法受关注较少, 本文不作介绍.

基于抽样的方法通过抽取部分有“代表性”的数

收稿日期: 2012-07-02；**修回日期:** 2012-10-21.

基金项目: 陕西省自然科学基金项目(2012JZ8005).

作者简介: 夏靖波(1963−), 男, 教授, 博士生导师, 从事网络管理、网络测量等研究；任高明(1986−), 男, 博士生, 从事网络测量的研究.

据分组, 计算其统计特性, 推断总体数据的统计特征, 这类用部分反映总体的方法有效地缓解了网络链路高速化带来的流量测量压力, 但同时也引入了误差, 抽样率越高, 测量结果越准确, 可适用的链路速率越低. 换言之, 可适用链路速率和流量测量准确性相互制约, 在实际应用中, 需根据具体情况在二者之间寻找平衡; 基于计数的方法简单且易实现, 通过快速丢弃占据小部分流量却占用大量计数单元的小流进而降低内存消耗, 但在处理速度和内存占用上尚存在改进的空间, 不适用高速网络数据流的分析需求; 基于哈希的方法在解决数据取值范围内发生转化的问题时效果较好, 但散列方法作为随机算法, 要获得高置信度的近似解需占用大量内存, 而且为降低冲突所构建的一组散列函数必须互相独立, 不易实现[27]; 基于LRU的方法处理速度快、识别效率高、硬件实现简单, 但若大量小流突发到达, 则会造成某些大流被替换出LRU缓存, 从而引起漏检, 同时LRU方法属于启发式算法, 很少有人对其进行理论推导, 参数设置更多的是依靠经验, 这给方法增加了不确定性[28].

本文的目的是尝试对当前与大流识别专题有关的技术进行分类, 并选取典型的算法进行简单地剖析.

1 大流识别方法

1.1 基于抽样的方法

网络流量抽样是统计学方法在网络流量测量领域的应用. 通过抽取部分有“代表性”的流量数据, 推断原始流量数据的特征. 流量抽样是数据量缩减和保留原始数据细节的折衷[2].

流量抽样从级别上可以分为字节级别抽样、数据分组级别抽样和流级别抽样; 从方法上可以分为系统抽样、随机抽样和分层抽样. 选取3种典型的基于抽样的大流识别方法予以介绍.

1.1.1 Sample and hold

Estan等[14,29]提出了sample and hold算法, 处理流程如图1所示. 用概率p对每个字节进行抽样, 则大小为s的数据分组被抽取的概率为$p_s = 1-(1-p)^s$. 可以发现, 包含字节数多的大流被抽取的概率更大; 当某个流所属数据分组被抽样, 属于该流的后续数据分组到来时, 流的信息都会被更新, 因此保证了测量的准确性[10]. 该方法从抽样级别看, 属于字节抽样; 从抽样方法看, 属于分层抽样.

设流F为大流, 含字节流量为T, 则流F在固定时间段内不能被识别为大流的概率为$(1-p)^{\mathrm{T}}$. Sample and hold方法较易实现, 所需缓存空间小, 准确率高, 相比于普通抽样的错误率$1/\sqrt{M}$, 该方法错误率更低, 为$1/M$ (M为缓存大小), 并可以通过提高

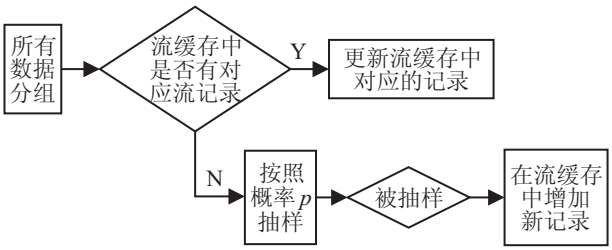

图 1 sample and hold 算法处理流程

抽样概率p保证准确率. 但由于需要对每个字节进行处理, 开销较大.

1.1.2 Smart sampling

Smart sampling[1,30]的核心思想是保证大于预定阈值z的流总能被抽取到, 而小于阈值z的流被抽取的概率与流的大小成正比关系. 具体操作流程为: 设x为抽样对象的大小(例如流所含数据分组的多少), 定义任意对象x的抽样概率函数为$p_z(x) = \min\{1, x/z\}$, 其中正数z为预定阈值, 大于z的对象以概率1被抽样, 小于z的对象以概率x/z抽样[31]. 该方法使用时较为灵活, 可根据实际情况调整阈值z的大小, 在测量精度和资源占用之间找到平衡点, 且准确性较高, 可用于计费等关注流量大小的测量[10].

1.1.3 基于Bayes定理的数据分组抽样方法

文献[32]提出根据Bayes定理估计周期性抽取的数据分组从而识别大流的方法, 主要思想是从N个数据分组中随机抽取n个, 抽样频率定义为$f = n/N$. 假设流j共有数据分组X_j个, Y_j为从中抽取的数据分组个数. 在已知$X_j = x$的前提下, Y_j满足$Y_j = y$的概率为

$$P_r[Y_j = y|X_j = x] = \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} N-x \\ n-y \end{bmatrix} \Big/ \begin{bmatrix} N \\ n \end{bmatrix},$$

即服从超几何分布. 根据Bayes定理, 已知$Y_j \geqslant y$, X_j满足$X_j \geqslant x$的概率$P_r[X_j \geqslant x|Y_j \geqslant y]$为

$$\frac{\sum\limits_{k=x} P_r[Y_j \geqslant y|X_j = k]P_r[X_j = k]}{\sum\limits_{k=1} P_r[Y_j \geqslant y|X_j = k]P_r[X_j = k]}, \tag{1}$$

其中

$$P_r[Y_j \geqslant |X_j = x] = 1 - \sum_{i=0}^{y-1} P_r[Y_j = i|X_j = x]. \tag{2}$$

式(1)表明: 在已知先验分布$P_r[X_j = x]$的前提下, 根据抽取流的数据分组个数y, 可以计算出流的数据分组大于x的概率. 假设$\hat{x}$为识别大流的阈值, 如果在给定$y = \hat{y}$的条件下, $P_r[X_j \geqslant \hat{x}|Y_j \geqslant \hat{y}]$足够接近于1, 则认为流所含数据分组个数大于$\hat{x}$, 即流$j$为大流.

该方法不需要对每个数据分组进行处理, 从而节省了资源开销, 可应用于大规模的高速网络, 但由于

周期性抽样和Bayes估计都会引入误差, 不适合准确性要求高的应用[33].

1.2 基于计数的方法

由Boyer等[34]提出的基于计数的方法, 使用固定或有限的计数器记录流数目. 当一个数据分组到达时, 先检查是否存在该数据分组所属流记录, 若存在, 则更新流记录, 否则创建新的流记录. 典型的算法有LC(lossy counting)[35]和PLC(probabilistic lossy counting)[36]两种方法.

1.2.1 lossy counting

LC是数据流上的频繁项挖掘方法之一, 将其应用于大流识别的具体操作为: 维护一个由目录(e, f, Δ)组成的数据结构D, e为流编号, f为估计流大小(数据分组数或比特数), Δ为f中的最大误差. 将输入数据分组分割成窗口大小为$1/\epsilon$的数据块, 依顺序处理每个数据块. 当有数据分组在窗口i中到达时, 如果该数据分组所属流记录已存在于D中, 则将相应的计数器加1; 否则创建一个新的流记录, 将f初始化为1, Δ初始化为$i-1$. 当到达窗口i的边界时, 线性扫描数据结构D, 从中删除$f+\Delta \geqslant i$的条目. 当有查询到达时, 输出D中所有$f \geqslant (\Phi-\epsilon)N$的条目. 该算法通过引入参数"最大误差", 较好地解决了因为过早删除表记录中没有结束的大流记录而造成的频数损失, 可操作性好, 准确性高.

LC算法能够在任意时刻输出网络流量中的大流, 由于在每个窗口的边界处需要扫描一遍数据结构D, 该操作的时间复杂度为$\Omega(1/\epsilon)$, 难以满足高速网络的要求, 并过高地评估了流的大小, 具有较高的假阳性误判率. 另外, 最大误差设定的好坏会直接影响存储资源的开销.

1.2.2 probabilistic lossy counting

在LC算法中, 最大误差反映了流大小估计的潜在错误, 这种潜在错误是由于流记录的提前移除导致的. LC算法中移除流记录的判断标准为: 在窗口结束时, 如果流大小与最大误差之和小于或等于给定的阈值, 则移除表中的流记录. 这便使得具有较大的最大误差的流记录在表中停留的时间更长. 根据Little定理, 流记录在表中停留的时间越久, 需要的内存越大, 即最大误差的值直接影响LC算法的内存消耗.

PLC算法是针对LC算法中存储开销较大和误报率较高等问题提出的. 与LC算法不同, 它采用基于概率误差区间的近似算法来估计误差. PLC算法利用LC算法中最大误差服从power-low分布的特性, 将最大误差补偿值合理地设置在大部分流能够得到足够补偿的范围(如保证被过早删除的流中95%可以得到充分补偿即可). 改进后的PLC算法在每一窗口结束的存储开销上有明显降低, 误报率也有所降低.

尽管PLC计算复杂度稍高, 但较好地减小了内存消耗. 因为模拟了"重尾分布"特性, 而重尾分布具有不稳定性, 所以导致假阴性误判. 与LC相比, 采用概率错误边界代替确定错误边界, 消减了资源消耗和假阳性误判.

文献[37]提出了MLC(mnemonic lossy counting)算法. 该算法是在LC算法的基础上改进得出的. 通过保持可能成为大流的流记录信息完成识别, 这些信息用来计算近似的最大误差从而评估流的大小. 张玉等[38]将高速网络中的大流量对象识别问题等价为带权值数据流中的频繁项挖掘问题, 提出一个新的频繁项挖掘算法WLC. 文献[39]基于网络数据流的幂律分布特性和连续性, 在PLC的基础上提出PFC(probabilistic fading counting)方法, 通过加大对表记录中非活动流的移除力度, 使这些流以更快的速度被删除, 从而有效降低了存储资源的开销.

1.3 基于LRU的方法

文献[40]从队列管理的角度出发, 提出了使用LRU(least recently used)缓存来识别大流(后文称为LRU_1方法). 基本思想是: 设置一个用于保存流记录的固定大小的缓存, 保持最新到达的数据分组所属流记录位于缓存最顶部, 则最久未到达数据分组所属流的记录位于缓存的最底部; 当有新流到达而缓存已满时, 将缓存最底部的流替换出去为新流腾出空间. 由于大流持续时间长且分组到达速率高, 使得其总能排在缓存的上部, 从而以较大的概率留在LRU缓存中.

LRU_1方法处理速度快, 识别效率高, 硬件实现简单. 但当有大量小流突发到达时, 会造成某些大流被替换出LRU缓存, 从而造成漏检. 文献[29]的实验结果中有10%～20%的大流被漏检. 此后, 研究人员围绕LRU_1算法展开了广泛研究. 文献[41]利用流的大小与速度强相关(即流越大其速率越高)的特性[42]提出一种新的基于LRU的大流检测算法(后文称为LRU_2). 基本思想是限制小流进入LRU缓存, 对有可能成为大流的流进行保护, 同时提供高准确度的字节数测量, 其模块图如图2所示.

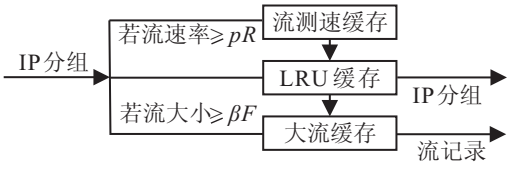

图 2　LRU_2算法模块

文献[18]采用串行的两级LRU缓存机制检测大流(后文称为LRU_3). 该算法通过增加一级过滤小

流的LRU缓存提高测量准确性, 可以有效克服大量小流短期到来时引发大流被漏检的缺点. 流量监控中某些应用只获知一种类型的流是不够的, 因此文献[43]提出了TS-LRU方法, 通过两级LRU缓存识别大流, 与LRU_3算法不同, TS_LRU方法在第2级LRU缓存中考虑了元素的大小. 文献[44]引入了流大小因子和调整因子, 通过调整链表内部流的排列方式识别大流. 王凤宇等[14]将LRU淘汰机制和LEAST(最少)淘汰机制相结合, 优势互补, 实现了固定大小空间内大流量对象的准确提取. 文献[31]提出一种S3-LRU(single step segmented LRU)方法, 将缓存区分成使用片和保护片, 与历史访问频率相结合, 克服了LRU_1方法难以应付大量小流突发到达所引起的误判.

从近几年的相关文献可以发现, 利用LRU机制开展的大流识别方法研究可以简单地分为以下4类: 在数据分组进入LRU缓存之前进行处理, 包括提前过滤掉小流和提前对有可能成为大流的流进行保护; 在LRU之后进行处理, 对已经被LRU机制淘汰的流对象再次筛选, 提供重新进入LRU缓存的机会, 提高识别大流的准确性; 采用LRU淘汰机制识别大流的同时, 引入其他参数帮助识别大流[45], 这些参数包括流速度、流存活时间、流包含数据分组的大小等; 综合使用以上方法. 这些方法从节省资源和提高准确性角度促进了大流识别技术的发展, 但仍存在一定缺点, 例如有的方法需要人为设置相关参数, 在一定程度上限制了方法的推广; 有的方法实现困难, 难以用于工程中.

1.4 基于哈希函数的方法

定义一组哈希函数, 将数据从一个范围映射到另一个范围, 是计算机领域的一种常用手段[46]. 根据所采用数据结构的不同, 又可以将利用哈希函数识别大流的方法分为bloom filter方法和Sketch方法, 本文分别选取一种具有代表性的方法进行简要分析.

1.4.1 Multistage filter 方法

Bloom filter[47]的本质是将集合中的元素通过哈希函数映射到向量中, 常用于数据库查询和数据存储. 与传统的哈希查询算法和树型查询算法相比, bloom filter所需存储空间与元素自身大小和集合规模无关, 仅与元素映射到向量的位数相关, 可以极大地节约存储空间, 详细内容参见文献[36].

文献[48]提出了一种基于loop bloom filter的大流计数方法, 并引入“time out”机制, 该方法只能进行粗粒度的计算, 操作复杂, 不具备扩展性. 文献[49]提出一种基于多维计数型bloom filter的大流检测机制, 将一位计数型bloom filter结构扩展到支持多业务流表示、查询和统计计数的MDCBF结构.

Multistage filter[13]是基于bloom filter大流识别方法中的典型代表, 处理流程如图3所示. 基本思想是: bloom filter中每一向量位关联一个计数器; 根据关键字将数据分组映射到bloom filter的向量位中, 并将数据分组的大小累积到对应计数器上. 由于同一个流F所含数据分组会映射到同一个向量位中, 当流F对应的计数器值达到或超过阈值T时, 将流F识别为大流. 考虑到哈希碰撞会引起误判, multistage filter采用多级bloom filter, 每级使用独立的哈希函数, 只有各级相应向量位的计数器值均大于阈值的流才被识别为大流, 从而大大减少了假阳性误判.

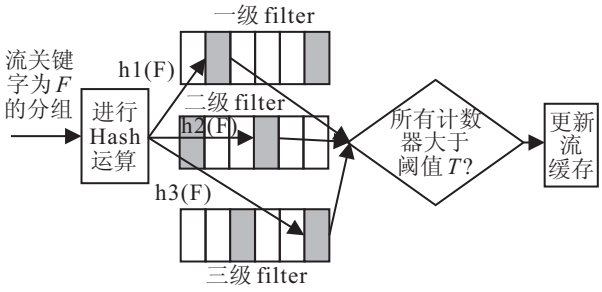

图 3 multistage filter 算法处理流程

文献[13]分析表明, 在4级过滤器分配大约4 000个计数器的情况下, 小流通过multistage filter的概率不足0.1%, 且只需约1 015个缓存.

Multistage filter作为bloom filter的一种变化形式, 在高效、简洁地识别大流的同时, 存在假阳性误判(将小于阈值T的流误判为大流), 而不存在假阴性误判(将大于阈值T的流误判为小流).

Multistage filter识别大流较为准确, 占用空间少, 误差随着过滤器级数的增加呈指数减少. 由于哈希算法会将不同的流映射到相同的向量位上, 导致高估大流的实际流量; 硬件实现复杂, 算法启动较慢, 初期会丢失采样数据分组, 难以应用于实时测量.

1.4.2 Count-min Sketch 方法

Count-min(CM) Sketch通过d个HASH函数将向量映射到wd个计数器. CM Sketch由两维数组构成, 数组的每项是一个计数器, 宽为w, 深为d, 可以表示为$\mathrm{count}[1,1],\mathrm{count}[2,2],\cdots,\mathrm{count}[d,w]$, 每行对应一个HASH函数.

当更新(i_t,c_t)到来时, c_t被分别增加到每行的一个计数器中, 并更新对应的计数器. 计数器由第j行对应的HASH函数h_j决定, 可以规范地表示为$\forall 1\leqslant j\leqslant d$, 存在

$$\mathrm{count}[j,h_j(i_t)]\leftarrow \mathrm{count}[j,h_j(i_t)]+c_t. \tag{3}$$

CM Sketch通过$\widehat{a}_j=\min_j \mathrm{count}[j,h_j(i)]$得到估算值, 即取所有HASH对应项中的最小值. 对于估算

得到的$\widehat{a}_i$, 可以保证$a_i \leqslant \widehat{a}_i$, 且至少有$1-\delta$的概率使得$\widehat{a}_i \leqslant a_i + \epsilon\|a\|_1$, 其中$\|a\|_1 = \sum_{i=1}^{n}|a_i(t)|$. 如果估算值$\widehat{a}_{i_t}$大于阈值$\oslash\|a\|_1$, 则将其放入缓存中, 否则从堆中删除. 在输出时扫描缓存, 大于阈值的值被输出. CM Sketch 方法应用广泛, 可对向量进行各种基本查询, 需同时进行哈希运算, 空间复杂度较高.

采用哈希技术识别大流, 可以有效利用哈希查找时间短、存储空间开销小、计算速度快的特点, 但不同的元素可能会映射到同一向量位发生碰撞, 导致假阳性误判发生.

2　典型应用

大流识别技术打破了传统流量测量中全数据采集的思想, 根据实际需求, 提出“抓大放小”的策略, 极大地缓解了高速网络链路条件下流量测量系统负载的压力, 是在网络流量测量准确性和处理效率之间的一种折衷. 开展高速网络中的大流识别研究, 对于网络行为和流量工程技术具有重要意义, 下面列出几种大流识别技术的具体应用.

1) 网络计费. 按照用户使用流量进行网络计费具有很多优点, 尤其在当前使用手机等移动终端上网异常普遍的情况下, 采用按流量计费更是一个好的选择. 但网络链路速率的提高使得对每个数据分组都进行准确计费存在较大难度, 且需要投入大量的资源, 因此, 将大流识别技术应用于网络计费, 保证大流被记录, 小流被忽略, 既能保证测量的准确性, 又能降低投入的成本.

2) 异常检测. 通过流量测量进行安全监测是网络安全管理的常用手段. 很多研究通过总体流量的变化来发现异常, 并不能定位异常的来源. 若要定位异常来源, 则不仅需要总体流量的变化信息, 还需要知道导致流量异常变化的对象[15]. 对于一些大规模网络安全事件, 只需关注大流即可. 例如 DDOS 攻击固定的目标地址.

3) 网络管理. 通过大流信息可以估算网络的延迟、丢包、带宽等性能参数, 除了可以分析流量的长度、大小等一般特征外, 还可以分析流量的应用类型分布、协议类型分布、在 AS 之间的分布等参数[50-51]. 这些数据对于网络管理意义重大[52].

3　结　论

虽然大流识别是网络流量测量问题中的一个分支, 但其应用非常广泛, 尤其在网络链路速率不断提高的今天, 显得尤为重要. 因此, 大流识别作为一种具有可扩展性的解决方案, 是近年来网络流量测量领域的研究热点之一[53].

本文首先根据所采用技术的不同对现有大流识别方法进行了分类; 然后, 简单分析了各类方法的优缺点, 并选取其中的经典算法进行剖析; 最后给出了大流识别技术的几种典型应用.

大流识别技术还在不断完善和向前发展, 在未来, 以下几方面的工作值得进一步探索:

1) 组合多属性研究算法. 流的基本属性包括大小、持续时间、速率和突发性等. 文献 [54] 研究表明, 流的大小与速率、突发性呈强相关关系, 现有的大流识别方法大多根据流的某一属性识别大流, 如流所含数据分组多少、流速率、流速率变化量、流速率的峰值等. 若将大流的两种或多种属性相结合设计算法, 将有助于提高大流识别的准确率和处理效率.

2) 自适应调整参数. 根据分析, 现有各种方法多数需要人为设置参数, 如抽样方法中的抽样率、计数方法中的窗口大小、页面置换算法中的 LRU 缓存大小等. 在实际网络链路中, 流量复杂多变, 参数的人为设置难以适应网络流量的变化, 且极大地限制了方法的扩展性. 因此, 研究根据网络流量的实时情况, 自适应地调整相关参数的大流识别方法具有重要意义.

3) 应用研究. 目前, 大多数的大流识别方法仍集中在理论研究方面, 已有的面向工程应用的方法准确性欠佳, 不能满足某些应用的需要, 将现有的大流识别方法应用于实际是有意义的工作.

参考文献(References)

[1] Hawa Mohammed, Rahhal Jamal S. Filesize models for shared content over the BitTorrent Peer-to-Peer network[J]. Peer-to-peer Networking and Applications, 2012, 5(3): 279-291.

[2] Yang L, Michailidis G. Sampled based estimation of network traffic flow characteristics[C]. The 26th IEEE Int Conf on Computer Communications. IEEE, 2007: 1775-1783.

[3] 杨家海, 吴建平, 安常青. 互联网络测量理论与应用[M]. 北京: 人民邮电出版社, 2009: 156-320.
(Yang J H, Wu J P, An C Q. Internet measurement theory and applications[M]. Beijing: Poste and Telecom Press, 2009: 156-320.)

[4] Hu Chengchen, Liu Bin, Wang Sheng. ANLS: Adaptive non-linear sampling method for accurate flow size measurement[J]. IEEE Trans on Communications. 2012, 60(3): 789-798.

[5] Carra D, Neglia G, Michiardi P, et al. On the robustness of BitTorrent swarms to greedy peers[J]. IEEE Trans on Parallel Distributed Systems, 2011, 22(12): 2071-2078.

[6] 程论, 王中杰. 基于数据流的 Internet 网络控制系统延时模型研究[J]. 控制与决策, 2011, 26(4): 513-518.
(Cheng L, Wang Z J. Research on time-delay model for

NCS based on data streams[J]. Control and Decision, 2011, 26(4):513-518.)

[7] Caviglionea L, Davolib F. Traffic volume analysis of a nation-wide eMule community[J]. Computer Communications, 2008, 31(10): 2485-2495.

[8] 潘乔, 裴昌幸, 朱畅华. 一种用于异常检测的网络流量抽样方法[J]. 西安交通大学学报, 2008, 42(2): 175-178.
(Pan Q, Pei C X, Zhu C H. Novel traffic sampling method for anomaly detection[J]. J of Xi'an Jiaotong University, 2008, 42(2): 175-178.)

[9] Zhang Y, Breslau L, Paxson V, et al. On the characteristics and origins of internet flow rates[J]. ACM Sigcomm Computer Communication Review, 2002, 32(4): 309-322.

[10] Feldmann A, Greenberg A, Lund C, et al. Deriving traffic demands for operational IP networks: methodology and experience[J]. IEEE/ACM Trans on Networking, 2001, 9(3): 265-280.

[11] Hua N, Xu J, Lin B. BRICK: A novel exact active statistics counter architecture[J]. IEEE-ACM Trans on Networking, 2011, 19(3): 670-682.

[12] Andrew R, Wonho K, Praveen Y. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection[C]. Int Conf on Computer Communication. Waterloo: IEEE, 2011: 1629-2637.

[13] 王宏, 龚正虎. Hits 和 Holds: 识别大象流的两种算法[J]. 软件学报, 2010, 21(6): 1391-1403.
(Wang H, Gong Z H. Hits and holds: Two algorithms for identifying the elephant flows[J]. J of Software, 2010, 21(6): 1391-1403.)

[14] 谢冬青, 周再红, 骆嘉伟. 基于 LRU 和 SCBF 的大象流提取及其在 DDoS 防御中的应用[J]. 计算机研究与发展, 2011, 48(8): 1517-1523.
(Xie D Q, Zhou Z H, Luo J W. An algorithm based on LRU and SCBF for elephant flows identification and its application in DDoS defense[J]. J of Computer Research and Development, 2011, 48(8): 1517-1523.)

[15] Z G, M K, F Z. Detecting heavy-hitters in a P2P network[C]. Int Conf on Network and Service Security. IEEE, 2009: 1-6.

[16] Curtis Andrew R, Kim Wonho, Yalagandula Praveen. Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection[C]. Int Conf on Computer Communication. Shanghai: IEEE, 2011: 1629-1637.

[17] Wang H, Gong Z H, Guan Q, et al. Detection network anomalies based on packet and flow analysis[C]. Proc of the 7th Int Conf on Networking. Cancun: IEEE Computer Society, 2008: 497-502.

[18] Agrawal S, Naidu K V M, Rastogi R. Diagnosing link-level anomalies using passive probes[C]. The 26th IEEE Int Conf on Computer Communications. IEEE, 2007: 1757-1765.

[19] He Ke-qiang, Hu Cheng-chen, Jiang Jun-chen, et al. Anti-attack counters for traffic measurement[C]. IEEE Global Telecommunications Conf. Miami, 2010: 1-5.

[20] Estan C, Varghese G, Fisk M. Bitmap algorithms for counting active flows on high speed links[C]. Proc of the 3rd ACM Sigcomm Conf on Int Measurement. 2003: 153-166.

[21] Cristian E, George V. New direction in traffic measurement and accounting[J]. Sigcomm Computer Communication Review, 2002, 32(4): 323-336.

[22] Li W, Canini M, Moore A W, et al. Efficient application identification and the temporal and spatial stability of classification schema[J]. Computer Networks, 2009, 53(1): 790-809.

[23] Wei G, Gu Y, Ge Y. Cluster: An effective solution to the problem of heavy-tailed distribution in P2P networks[C]. The 3rd Int Conf on New Trends in Information and Service Science. Beijing, 2009: 1397-1402.

[24] Otto John S, Sanchez Mario A, Choffnes David R. On blind mice and the elephant understanding the network impact of a large distributed system[J]. Computer Communication Review, 2011, 41(4): 110-121.

[25] Mondal Amit, Kuzmanovic Aleksandar. Upgrading mice to elephants: Effects and end-point solutions[J]. IEEE ACM Trans on Networking, 2010, 18(2): 367-378.

[26] 王风宇, 云晓春, 王晓峰, 等. 高速网络监控中大流量对象的提取[J]. 软件学报, 2007, 18(12): 3060-3070.
(Wang F Y, Yun X C, Wang X F, et al. Identifying heavy hitters in high-speed network monitoring[J]. J of Software, 2007, 18(12): 3060-3070.)

[27] 杜阿宁, 程晓明. 网络流量分析中的频繁项监测技术研究[J]. 通信学报, 2006, 27(2): 9-15.
(Du A N, Cheng X M. Frequent items maintaining algorithms in network traffic analysis[J]. J on Communications, 2006, 27(2): 9-15.)

[28] 裴育杰, 王洪波, 程时端. 基于两级 LRU 机制的大流检测算法[J]. 电子学报, 2009, 37(4): 684-691.
(Pei Y J, Wang H B, Cheng S D. A dual-LRU based algorithm for identifying and measuring large flows[J]. Acta Electronic Sinica, 2009, 37(4): 684-691.)

[29] Duffield N, Lund C, Thorup M. Charging from sampled network usage[C]. Proc of the 1st ACM Sigcomm Workshop on Int Measurement. San Francisco, 2001: 1-2.

[30] Duffield N, Lund C. Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure[C]. Proc of the 3rd ACM Sigcomm Conf on Int Measurement. Miami Beach, 2003: 27-29.

[31] Duffield N, Lund C, Thorup M. et al. Sample less: Control of volume and variance in network measurement[J]. IEEE Trans on Information Theory, 2005, 51(5): 1756-1775.

[32] Tatsuya Mori, Masato Uchida, Ryoichi Kawahara. Identifying elephant flows through Periodically Sampled Packets[C]. ACM Sigcomm/Internet Measurement Conf. Italiy, 2004: 115-120.

[33] Mondal Amit, Kuzmanovic Aleksandar. Upgrading mice to elephants: Effects and end-point solutions[J]. IEEE-ACM Trans on Networking, 2010, 18(2): 367-378.

[34] Boyer R S, Moore J S. MJRTY-a fast majority vote algorithm[C]. Automated Reasoning: Essays in Honor of Woody Bledsoe, Automated Reasoning Series. Kluwer Academic Publishers, 1991: 105-117.

[35] Manku G S, Motwani R. Approximate frequency counts over data streams[C]. Proc of the 28th Int Conf on Very Large Date Bases. Endowment, 2002: 346-357.

[36] Dimitropoulos X, Hurley P, Kind A. Probabilistic lossy counting an efficient algorithm for finding heavy hitters[J]. ACM Sigcomm Compter Communication Review, 2008, 38(1): 7-16.

[37] Rong Q, Zhang G X, Xie G G. Mnemonic lossy counting: An efficient and accurate heavy-hitters identification algorithm[C]. Performance Computing and Communications Conf. IEEE, 2010: 255-262.

[38] 张玉, 方滨兴, 张永铮. 高速网络监控中大流量对象的识别[J]. 中国科学, 2010, 40(2): 340-355.
(Zhang Y, Fang B X, Zhang Y Z. Identifying heavy hitters in high-Speed network monitoring[J]. Science China, 2010, 40(2): 340-355.)

[39] 李臻, 杨雅辉, 谢高岗, 等. 一种基于数据流计数的概率衰落大业务流识别方法[J]. 计算机研究与发展, 2011, 48(6): 1010-1017.
(Li Z, Yang Y H, Xie G G, et al. An identification method combining data streaming counting with probabilistic fading for heavy-hitter flows[J]. J of Computer Research and Development, 2011, 48(6): 1010-1017.)

[40] Smitha Kim I, Reddy A. Identifying long term high rate flows at a router[C]. High Performance Computing, Hyderabad. India, 2001: 361-371.

[41] 王洪波, 裴育杰, 林宇, 等. 基于LRU的大流检测算法[J]. 电子与信息学报, 2007, 29(10): 2487-2492.
(Wang H B, Pei Y J, Lin Y, et al. A LRU based algorithm for identifying and measuring large flows[J]. J of Electronics and Information Technology, 2007, 29(10): 2487-2492.)

[42] Martin Z, Marco C, Andrew W, et al. Tracking elephant flows in internet backbone traffic with an FPGA-based cache[C]. Proc of the 19th Int Conf on Field Programmable Logic and Applications. IEEE, 2009: 640-644.

[43] Wang F Y, Gong B, Guo S Q, et al. Monitoring heavy-hitter flows in high-speed network goncurrently[C]. The 4th Int Conf on Network and System Security. Melbourne: IEEE, 2010: 160-167.

[44] Wang F Y, Guo S Q , Hu Y, et al. REFI: Extracting out heavy-hitter flows accurately and rapidly[C]. The 3rd Int Conf on Advanced Computer Theory and Engineering. IEEE, 2010: 274-279.

[45] Zhang N B, Wang F Y, Gong B, et al. Identifying heavy-hitter flows fast and accurately[C]. The 2nd Int Conf on Future Computer and Communication. IEEE, 2010: 326-330.

[46] 金澈清, 钱卫宁, 周傲英. 流数据分析与管理综述[J]. 软件学报, 2004, 15(8): 1172-1181.
(Jin C Q, Qian W N, Zhou A Y. Analysis and management of streaming date: A survey[J]. J of Software, 2004, 15(8): 1172-1181.)

[47] Burton H B. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM, 1970, 13(7): 422-426.

[48] Sun Y, Zhang Z B, Guo L, et al. An effective algorithm for counting active flows based on loop filter[C]. Proc Int Conf on Networking, Architecture and Storage. Chongqing 2008: 104-109.

[49] 张震, 汪斌强, 陈庶樵, 等. 基于多维计数型布鲁姆过滤器的大流检测机制[J]. 电子与信息学报, 2010, 32(7): 1608-1613.
(Zhang Z, Wang B Q, Chen S Q, et al. A mechanism of identifying heavy hitters based on multi-dimensional counting bloom filter[J]. J of Electronics and Information Technology, 2010, 32(7): 1608-1613.)

[50] Ozkasap O, Aglar M, Alagoz A. Principles and performance analysis of second: A system for epidemic peer-to-peer content distribution[J]. J of Network and Computer Applications, 2009, 32(3): 666-683.

[51] Xu J B, Wang X, Zhao J, et al. I-swifter: Improving chunked network coding for peer-to-peer content distribution[J]. Peer-to-peer Networking and Applications, 2011, 5(1): 30-39.

[52] McKeown N, Anderson T, Balakrishnan H. OpenFlow: Enabling innovation in campus networks[J]. Computer Communication Review, 2008, 38(2): 69-74.

[53] Greenberg A, Hamilton J R, VL2: A scalable and flexible data center network[J]. Communications of the ACM, 2011, 54(3): 95-104.

[54] Lan Kun-chan, John Heidemann. On the correlation of internet flow characteristics[R]. Technical Report ISI-TR-574, USC/ISI, 2003.