

## 改进的资源限制流抽样算法

韦安明 王洪波 林宇 程时端

(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

**摘要:** 该文分析了 hardflow 算法抽样率与大流抽样损失率的关系, 提出两个改进算法以降低大流抽样损失率与抽样率的比值  $r_d$ , 并讨论了缓冲区大小、测量周期对相对误差的影响以及参数的选取。结果表明: 与 hardflow 相比, 在低抽样率(4.5%), 两种改进算法的  $r_d$  比 hardflow 分别降低 1.4 和 6.6, 当抽样率提高至一定水平时(>11.3%), 则 3 种算法的  $r_d$  均趋于 0。

**关键词:** 网络流测量; 非均匀抽样; 资源限制

**中图分类号:** TP393.06

**文献标识码:** A

**文章编号:** 1009-5896(2007)06-1343-05

## Improved Flow Sampling Algorithm on Resource Constraints

Wei An-ming Wang Hong-bo Lin Yu Cheng Shi-duan

(State Key Laboratory of Networking and Switching, Beijing University of Posts and Telecommunications, Beijing 100876, China)

**Abstract:** This paper analyzes the relationship between the Sampling Rate (SR) and Sampling Loss Rate of Elephant Flows (SLREF) of hardflow. Then two improved algorithms are proposed to reduce the ratio of SLREF to SR ( $r_d$ ). The relative error and the parameters setting are discussed with various sampling buffer sizes and various measurement periods. Compared to the hardflow, the results show that  $r_d$  of the two improved algorithms decrease 1.4 and 6.6 respectively when SR is low (e.g. 4.5%). Furthermore, all  $r_d$  incline to zero when SR is large (e.g. >11.3%).

**Key words:** Network flow measurement; Non-uniform sampling; Resource constraints

### 1 前言

流测量(flow measurement)在网络性能估计、接纳控制、故障定位、攻击检测防护、流量计费等应用中起重要作用, 是网络流量工程不可或缺的部分, 而抽样是进行高速链路流测量的重要方法<sup>[1]</sup>。网络行为本身的动态特性催生了测量系统资源限制(resource constraints)的思想。网络流量是一个时变参量, 存在突发现象, 或者受周期性流量变化、网络攻击、路由中断等事件的影响, NetFlow 生成的流记录数量会发生较大波动<sup>[2]</sup>, 进而影响测量系统内存、计算能力等资源的消耗量。为限制抽样过程中的资源消耗, 增强测量系统的健壮性, 文献[2]提出了资源限制的流抽样算法, 本文称之为 hardflow 算法。

在介绍本文的工作之前先回顾一下流测量的研究进展。文献[3]最早把抽样方法引入网络流量测量中以估计分组大小和分组到达间隔。Duffield 等人则将抽样引入流测量中<sup>[1,2,4,5]</sup>。文献[1]指出了均匀随机抽样在流抽样应用中的缺点, 将非均匀抽样思想引入流抽样领域。文献[4]基于文献[1]

的成本函数讨论流计费问题。文献[5]给出一个层次性的流抽样方法。文献[6]以保证大流的估计准确性为目标, 用自回归模型预测分组总数来动态调整分组抽样率。文献[7]指出了 NetFlow 中存在的问题, 并提出了自适应调整抽样率的方法。而文献[2]提出的 hardflow 资源限制的流抽样算法使用固定内存空间保存流记录样本, 使抽样率自适应于网络流量变动, 避免了欠抽样和过抽样, 同时保持资源消耗不变, 但 2.1 节中的分析表明低抽样率时其存在大流抽样损失率与抽样率之比过高的缺点。

本文在分析 hardflow 算法于低抽样率下存在大流抽样损失率与抽样率比值过高的缺点的基础上, 提出用大流抽样损失率与抽样率的比值来评价算法性能, 同时提出两个改进的流抽样算法(aflow1, aflow2), 目的是降低大流抽样损失率与抽样率的比值, 并通过实验对算法进行了评估。

本文组织: 首先讨论 hardflow 的不足, 然后介绍 aflow1, aflow2 原理及算法实现的关键问题, 接着对 aflow1 和 aflow2 进行实验验证, 并和 hardflow 算法做了比较, 最后是结束语。

### 2 改进的资源限制流抽样算法

#### 2.1 hardflow 算法及不足

hardflow 算法<sup>[2]</sup>是 Smart Sampling 算法<sup>[4]</sup>的改进, 它引

2006-03-20 收到, 2007-03-12 改回

国家自然科学基金(90604019, 60502037, 60603060, 60472067), 国家 863 计划项目(2006AA01Z235), 国家 973 规划项目(2006CB7 01306, 2003CB314806), 和中国下一代互联网示范工程(CNGI-04-8-1D)资助课题

进了资源限制的思想,使测量系统能适应更高速链路和流量大幅波动,但在随后的分析中将发现在较低抽样率情况下 hardflow 算法会产生较大的大流抽样损失率与抽样率比值,这意味着在较低抽样率下,存在较严重的大流抽样损失。文献[1,4]已经证明流抽样过程中丢失的大流会给估计误差带来大的影响,同时在流量工程中,过多的大流抽样损失会影响计费的准确性和流量调控的有效性。

为简便起见,在分析之前先定义抽样损失率与抽样率比值为评价目标。相同抽样率下比值越小表明抽样过程大流损失越少,同时降低该比值也是算法改进的目标。

**定义 1** 首先令  $m$  为每个测量周期的样本数,  $n$  为总流记录数,则抽样率  $r_s = m/n$ ; 其次令  $k$  为大流数,  $l$  为抽样过程损失的大流数,则大流抽样损失率  $r_l = l/k$ ; 最后令抽样损失率与抽样率比值为  $r_d = r_l/r_s$ 。

我们用 hardflow 对表 1 中数据 II 的前 5s 的流进行抽样以分析大流抽样损失率与抽样率之间的关系。数据中总流个数  $n=22096$ ,将最大的前 1191 个流视作有重要影响的大流,第 1191 个流大小为 15,625byte,即图 1 中的直线 A。如果要保证这 1191 个流被抽中,须设定抽样阈值  $z'_i < 16349$ ,见图 1 中的直线 B,其上方的点为被抽取的样本,此时样本数为 2402,抽样率  $r_s = 10.9\%$ ,  $r_l = 0\%$ ,故  $r_d = 0$ 。如果需要进一步降低抽样率,则须提高抽样阈值  $z'_i$ :例如将抽样率降低到  $r_s = 8\%$ ,见直线 C 标定的阈值( $z'_i = 81134$ ),A, C 两直线将平面划分为 4 个区域,落入 I, IV 区中流记录被全部抽中, II, III 中流没被抽中,而 III 中未被抽中 432 个流属于指定的前 1191 个大流,  $r_l = 36.3\%$ ,而  $r_d = 4.5$ 。

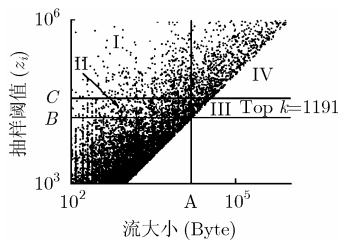


图 1 大流抽样损失现象

可见 hardflow 算法的抽样率低于一定值(如上分析的 10%)时,大流抽样损失出现并且增加较快。hardflow 想进一步降低抽样率并保持低的大流抽样损失率比较困难,须以较大的抽样缓冲区为代价。鉴于流大小的重尾分布特性及大流的重要性,可以通过算法改进增加大流在样本中的比例,同时减少小流比例,以达到在  $r_l$  和  $r_d$  指标不变的前提下降低  $r_s$  之目的,从而进一步降低存储和计算等资源消耗量。

表 1 实验数据描述

编号	起始时间	链路描述	时长(s)	bits/s	pkts/s	flows/s
I	2002.08.14-09:00:00	IPLS->CLEV, 2.5Gbps	60	403.4M	72.6K	4.3K
II	2002.08.27-05:40:05	IPLS->CLEV, 2.5Gbps	60	357.0M	72.0K	1.7K
III	2004.06.01-19:31:21	IPLS->KSCY, 10Gbps	60	683.3M	109.2K	6.9K

将测量过程划分为长度为  $T$  的连续时间窗口。在每个  $T$  周期内,有  $n$  个大小为  $\{x_i : i = 1, \dots, n\}$  的流记录到达,且每个流记录携带一个区别于其他流记录的属性值  $c_i$  (属性值可以是 IP 地址、协议号、端口号等)。算法需要从每个测量周期中抽取定常个样本  $m$  (抽样缓冲区)。本文提出的改进资源限制流抽样算法可简单描述如下:对于每一个属于集合  $C$  的属性  $c \in C = \{c_i : i = 1, \dots, m, \text{且 } m \leq n\}$ ,通过抽取  $m$  个样本,然后重正化,形成关于所有具有  $c$  属性的流大小(size, 可以指流记录分组数、字节数等参量)的无偏估计  $X(c) = \sum_{i:c_i=c} x_i$ 。

**2.2 aflow1 算法**

基于流大小服从重尾分布的特性和大流的重要性,本文在 hardflow 算法中引入比例因子,以克服 hardflow 算法采用线性不等概率抽样造成抽样损失率与抽样率比值过高的不足。基本思想是引入形如  $\alpha(x) = e^{\lambda x}$  指数率的比例因子对流记录  $x_i$  进行放大,然后进行抽样。之所以采用  $\alpha(x) = e^{\lambda x}$  指数率比例因子是基于这样的考虑:(1)在系数  $\lambda x \approx 0$  时  $\alpha$  取值接近于 1,通过合理选择  $\lambda$  系数,使  $\lambda x_{\max} \approx 1$ ,则可控制放大倍数在合理的范围内。(2)从图 2 部分 II 可以看到符合指数率特性的曲线尾部迅速升起,起始阶段则很平缓,接近于 1。由于流大小服从重尾分布,多数流  $x_i$  较小,对应到比例因子曲线上就是对应于曲线平直的开始部分,而对于少部分大流则对应增长迅速的曲线尾部。这样就达到增加大流抽样概率的目的。

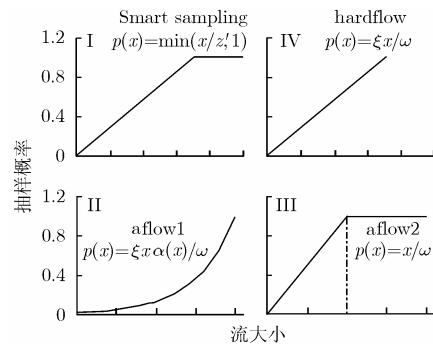


图 2 各种流抽样方法原理

分 4 步描述 aflow1 算法原理,且将测量周期划分为长度为  $T$  的连续时间窗,并假设  $T$  间隔内到达的流记录数  $n$  是一个与网络流量构成特征相关的随机变量:

- (1) 为每个流记录生成一个 (0,1] 区间的均匀随机数  $\omega_i, i = 1, \dots, n$ ;

(2)生成流记录  $z_i$  值, 令  $z_i = \alpha(x_i) \cdot x_i \cdot (1/\omega_i)$ ,  $i = 1, \dots, n$ ;

(3)取前  $m$  个最大  $z_i$  对应的流记录为样本, 并记第  $m+1$  大的  $z_i$  为抽样阈值  $z_{[m+1]}$ ;

(4)对  $z_i > z_{[m+1]}$  的  $m$  个流对应的  $x_i$  进行重正化, 即令  $x_i = \max(x_i, z_{[m+1]}/\alpha(x_i))$ 。

aflow1 算法的物理实现采用优先级队列模型<sup>[2]</sup>, 见图 3 示意, 对到达的流记录用  $\langle x_i, z_i, c_i \rangle$  顺序标识, 见步骤①, 如果缓冲区未满则将该项插入, 如果已满, 则遍历缓冲区, 找出  $z$  最小的项  $\langle x_j, z_j, c_j \rangle$ , 见步骤②, 如果  $z_n > z_j$  则用  $\langle x_i, z_i, c_i \rangle$  置换  $\langle x_j, z_j, c_j \rangle$ , 见步骤③,  $T$  周期结束后找到最小的  $z$  标为  $z_{[m+1]}$ , 用  $x_i = \max(x_i, z_{[m+1]}/\alpha(x_i))$  重正化缓冲区中所有的  $x_i$ , 见步骤④。

### 2.3 aflow2 算法

实验中还发现引入指数比例因子可以有效降低  $r_d$ , 但参数  $\lambda$  过小则结果不明显, 参数  $\lambda$  过大则歧视小流过于严重。因此本文结合 Smart sampling 与 hardflow 算法得到 aflow2 算法, 思路如下: 给定样本空间  $m$ , 总流记录数  $n$ ,  $k$  为大流数, 例如  $k=1191$  表示 1191 个大流被抽取并保存于抽样缓冲区中, 取定  $k \leq m$ , 剩下的  $(m-k)$  个样本空间用于容纳 hardflow 算法从  $(n-k)$  个个体里抽出的样本。之所以采用 hardflow 而 aflow1 作为剩余  $(n-k)$  个个体的抽样算法是因为最重要的大流已被抽取, 而 hardflow 采用与流大小成线性关系的非均匀抽样方法, 要比 aflow1 少一个指数运算。依定义 1, 该方法的  $r_d$  总为 0。aflow2 算法可分 5 步描述:

(1)标定最大的  $k$  个流为样本;

(2)为剩下的  $(n-k)$  个流各生成一个  $(0,1]$  区间的均匀随机数  $\omega_i, i = 1, \dots, (n-k)$ ;

(3)为对应的  $(n-k)$  个流生成  $z_i$  值,  $z_i = x_i \cdot (1/\omega_i)$ ,  $i = 1, \dots, (n-k)$ ;

(4)取最大的前  $(m-k)$  个  $z_i$  所对应流为样本, 并记第  $(m-k+1)$  大的  $z_i$  为抽样阈值  $z_{[m-k+1]}$ ;

(5)对  $z_i > z_{[m-k+1]}$  的  $(m-k)$  个流对应的  $x_i$  进行重正化, 即  $x_i = \max(x_i, z_{[m-k+1]}/\alpha(x_i))$ 。

aflow2 最后也得到  $m$  个样本, 它的物理实现是在 aflow1 前端加一个容量为  $k$  的缓冲区前端, 如图 4。令到达的流记录大小为  $x_i$ , 缓冲区 I 中的最小  $x_i$  为  $x_{[k].\min}$ , 如果  $x_i >$

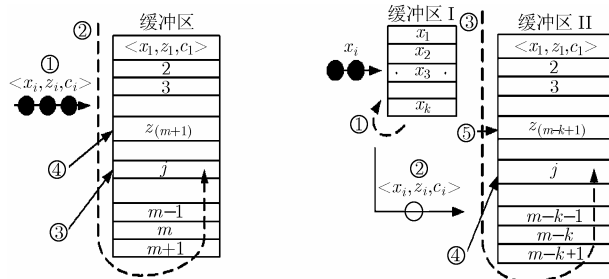


图 3 aflow1 实现

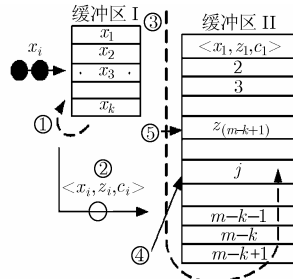


图 4 aflow2 实现

$x_{[k].\min}$  则置换  $k$  中对应项, 见步骤①, 被置换出缓冲区的流记录进入缓冲区 II, 否则  $x_i$  进入缓冲区 II, 见步骤②。缓冲区 II 的大小为  $(m-k+1)$ , 其步骤和 aflow1 的一致。

### 3 算法实现关键问题

算法实现关键问题包括重正化过程和参数设置。依样本对总体进行估计的过程称为重正化, 例如以概率  $p$  对  $n$  个流记录进行抽样, 抽到具有属性  $c$  的样本数为  $y$ , 则通过式  $y/p$  可估算出总体中具有  $c$  属性的流记录数。

#### 3.1 aflow1 重正化公式

令  $i \in I, m < n = \#I$  且  $C = \bigcup_{i \in I} \{c_i\}, I = \{x_i : i = 1, 2, \dots, n\}$  为流记录集合,  $m$  为抽样缓冲区大小,  $\#I$  为集合大小。如果大小为  $x_i$  的流以概率  $p(x_i)$  被抽中, 则通过式  $\hat{x}_i = x_i / p(x_i)$  可得到  $x_i$  的估计, 公式来源解释如下: 当  $x$  取  $x_i$  时,  $z_i > z_{[m+1]}$  的概率即  $x_i$  被抽中的概率  $p(x_i)$ , 用式(1)表示

$$p(x_i) = p[z_i \geq z_{[m+1]}] = p\left[\frac{x_i \cdot \alpha(x_i)}{\omega_i} \geq z_{[m+1]}\right] = p\left[\omega_i \leq \frac{x_i \cdot \alpha(x_i)}{z_{[m+1]}}\right] = \begin{cases} 1, & x_i \cdot \alpha(x_i) \geq z_{[m+1]} \\ x_i \cdot \alpha(x_i) / z_{[m+1]}, & x_i \cdot \alpha(x_i) < z_{[m+1]} \end{cases} \quad (1)$$

则有

$$\hat{x}_i = \frac{x_i}{p(x_i)} = \max(x_i, z_{[m+1]}/\alpha(x_i)) \quad (2)$$

式(2)即为 aflow1 重正化公式。对于属性集合  $C$  中的某属性  $c \in C$ , 用具有该属性的流记录大小估值之和  $\hat{X}(c)$  估计  $X(c)$ , 有

$$\hat{X}(c) = \sum_{i \in I(c)} \hat{x}_i = \sum_{i=1}^m \chi^I \cdot \{c_i = c\} \cdot \max\{x_i, z_{[m+1]}\} \quad (3)$$

式(3)中  $\chi^I \cdot \{c_i = c\}$  为指示函数, 当某流记录具有属性  $c$  且被抽样为样本这样的事件发生时, 其取值为 1, 否则为 0。

#### 3.2 aflow2 重正化公式

将 aflow2 看作为两个子抽样缓冲区的结合, 容量为  $k$  缓冲区中的样本由于以概率 1 抽取样本, 故不需要重正化, 容量为  $(m-k)$  的缓冲区中的样本采用以下方法进行重正化<sup>[2]</sup>: 当  $x$  取  $x_i$  时,  $z_i > z_{[m-k+1]}$  的概率即  $x_i$  被抽中的概率  $p(x_i)$ , 用式(4)表示

$$p(x_i) = p[z_i \geq z_{[m-k+1]}] = p\left[\frac{x_i}{\omega_i} \geq z_{[m-k+1]}\right] = p\left[\omega_i \leq \frac{x_i}{z_{[m-k+1]}}\right] = \begin{cases} 1, & x_i \geq z_{[m-k+1]} \\ x_i / z_{[m-k+1]}, & x_i < z_{[m-k+1]} \end{cases} \quad (4)$$

则有

$$\hat{x}_i = x_i / p(x_i) = \max(x_i, z_{[m-k+1]}) \quad (5)$$

式(5)即为 aflow2 重正化公式。用  $\hat{X}(c)$  估计  $X(c)$ , 有

$$\hat{X}(c) = \sum_{i \in I(c)} \hat{x}_i = \sum_{i=1}^{m-k} \chi^I \cdot \{c_i = c\} \cdot \max\{x_i, z_{[m-k+1]}\} \quad (6)$$

3.3 λ 参数的选取

根据大流重要性来决定参数 λ，由图 2 子图 II 可见调整 λ 可控制抽样曲线的弯曲程度，λ 越大，曲线尾部越接近垂直，大流得到的抽样概率越高。但如果 λ 取值过大则有可能造成运算量过大或运算结果溢出的后果，具体的参数数值可参考网络流量情况调整：也就是说如果服从重尾分布的占流个数比重很小的大流对网络运行状况评估、计费等业务起非常重要作用的话，λ 应该取较大的值；反之可以取较小的值。实验结果表明比较合理的取值是使 λ = 1/x<sub>max</sub>，这时 α<sub>max</sub> = e<sup>1</sup>，最大的那个流放大 e(自然对数底数)倍。实验中取 λ < 1/x<sub>max</sub> (即 1 < α(x) < e<sup>1</sup>) 时已经取得很好的效果，由于 α(x) 是指数增长的，随着 x 的增大，流抽样概率亦呈指数率增长。但实际测量中 x<sub>max</sub> 是个动态值，无法预知其为何数值，因此 x<sub>max</sub> 必须从测量历史结果中寻找，所以只能是一个估计值。这个估值在网络流量突发的情况下可能会产生大的误差，比如在流量突发时可能会发生网络中最大流的 x<sub>max</sub> 极大于历史最大值的情况，此时 α<sub>max</sub> ≫ e<sup>1</sup>，结果是该大流被抽为样本的概率极大增加了，而这正是本文想对大流采取的措施，对于其他没有发生突变的大流其被抽样概率还是和其大小相关，所以 λ 是可以根据历史测量结果用公式 λ = 1/x<sub>max</sub> 确定的。

3.4 k 参数的选取

aflow2 中参数 k 的设置是以测量应用需求为依据的，例如用户需要了解网络的流构成情况，但是又特别关心流速率大于链路带宽 p% 的流，即不希望他们在抽样过程中有损失，此时设定 k = (1/p)%，同时根据 2.4 小节“k ≤ m”的约束条件，设定 m 大于 k。实际上链路中几乎不会出现有 k 个速率大于链路带宽 p% 流的情况(此时链路效率 100%)，进入大流保留缓冲区的将是链路中最大的那 k 个流，但这并不影响用户分析流构成。

4 实验

本节用来自 NLNR<sup>[8]</sup> 的互联网数据对 aflow1, aflow2 进行检验，共使用来自 3 条不同高速链路的 3 组数据，每组数据取长 60s，再拆分为长度为 T 的测量段。采用五元组区分流，即<源 IP, 源端口, 目的 IP, 目的端口, 协议类型>。在不同的测量缓存区大小(m 取 200~20000 间的某些离散量)及 λ 因子(0.00001~0.000001)的条件下，分别以 5s, 10s, 15s, 20s 为测量周期，采用本文的方法对 3 组数据进行流抽样，每种参数取值条件下独立运行 100 次并取均值。表 1 给出了实验所用数据特征。

4.1 实验设计

针对网络中 8 种主要的应用流，以端口号为属性计算每种应用的字节数的相对误差(绝对误差与真值的比值)。首先考察不同的抽样缓冲区大小对相对误差的影响。分别取 m

等于 1000, 5000, 10000, 15000(保证 n>m 的前提下)，对数据 II 的前 5s 数据进行 100 次抽样，取 100 次相对误差的均值绝对值作为最后的相对误差值，实验中 λ = 0.000001，T = 5s, k = 200。其次考察测量周期 T 分别取值 5s, 10s, 15s, 20s、λ=0.000001，m = 5000，k = 200 时相对误差的变化情况。最后在分析了对相对误差产生影响的几个要素以后，于 4.3 节中将 aflow1, aflow2 和 hardflow 算法的大流抽样损失率与抽样率比值进行了比较。

4.2 结果分析

图 5、图 6 分别是设置不同缓冲区大小时用 aflow1、aflow2 进行抽样的相对误差结果，可观察到随着缓冲区的增大，流的抽样相对误差呈减小趋势。实际上增大缓冲区相当于提高抽样率，相对误差减小符合抽样理论分析结果。图 7 是 aflow1 在取不同抽样周期 T 时得出的结果，随着 T 的增大，误差有增大的趋势，原因是在抽样缓冲区不变的条件下，T 增大相当于降低抽样率，故误差增大。但在图 8 中情况略有不同，随着 T 的增大，误差并没有表现出单调增大趋势，这个现象可以解释为 aflow2 算法以概率 1 保存大流后，剩下的小流对统计结果产生的影响很小，不足以决定误差变动走势。

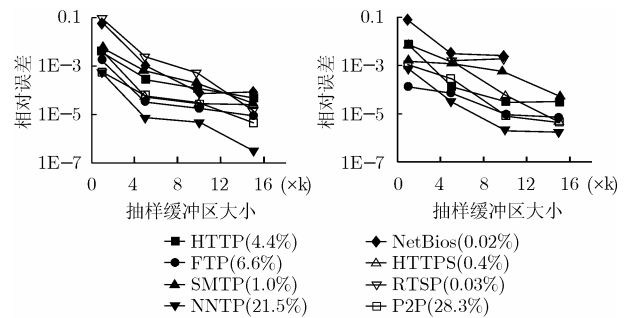


图 5 缓冲与相对误差关系(aflow1)

图 6 缓冲与相对误差关系(aflow2)

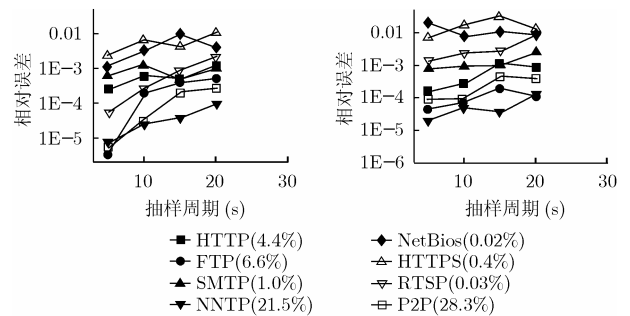
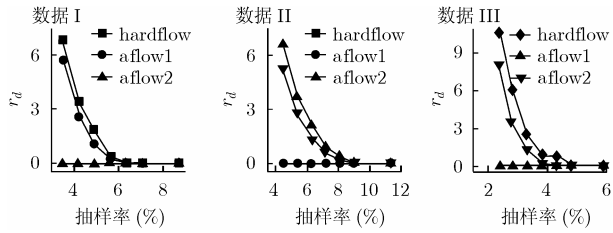


图 7 T 对相对误差的影响(aflow1)

图 8 T 对相对误差的影响(aflow2)

4.3 算法对比

在分析了算法的误差特性后，本节将它们和 hardflow 算法做了对比，主要考察大流抽样损失率与抽样率比值 r<sub>d</sub>。设定参数 λ = 0.00001, k = 1000，为每个抽样率独立进行 100 次实验，取得到的最大值为结果，见图 9。在 aflow1 中，它的 r<sub>d</sub> 值相对于 hardflow 的有明显减小，例如数据 I 中，在

图 9  $r_d$  比较

$m = 1000, k = 1000$  情况下,  $r_d$  降低了 1.1, 在  $m = 1800, k = 1000$  情况下,  $r_d$  降低了 0.015, 而此时 aflow1 的  $r_d$  已经降到 0, 其他的两组数据有相同的情况。由于 aflow2 总是以概率 1 保存  $k$  个大流, 故其  $r_d$  总是为 0。

## 5 结束语

本文深入分析了文献[2]提出的资源限制流抽样算法存在的大流抽样损失与抽样率比值过高的不足及其成因, 并根据网络流大小服从重尾分布的重要特征, 提出了改进算法 aflow1 和 aflow2。采用互联网数据进行的实验结果表明 aflow1, aflow2 算法在精度与 hardflow 算法保持相当的情况下可大幅降低大流抽样损失与抽样率比值。与之前的流抽样算法对比, aflow1 和 aflow2 在资源可控性和大流抽样损失控制方面表现优越。

## 参考文献

- [1] Duffield N G, Lund C, and Thorup M. Learn more, sample less: Control of volume and variance in network measurement. *IEEE Trans. on Information Theory*, 2005, 5(51): 1756-1775.
- [2] Duffield N G, Lund C, and Thorup M. Flow sampling under hard resource constraints. Proceedings of the joint international conference on Measurement and modeling of computer systems, New York, ACM Press, 2004: 85-96.
- [3] Claffy K, Polyzos G C, and Braun H W. Application of sampling methodologies to network traffic characterization. ACM SIGCOMM'93, San Francisco, 1993: 13-17.
- [4] Duffield N G, Lund C, and Thorup M. Charging from sampled network usage. Internet Measurement Conference, San Francisco, ACM Press, 2001: 245-256.
- [5] Duffield N G and Lund C. Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure. Internet Measurement Conference, Miami, ACM Press, 2003: 179-191.
- [6] Choi B Y, Park J, and Zhang Z L. Adaptive packet sampling for accurate and scalable flow measurement. IEEE Globecom'04, Dallas, 2004, Vol.3: 1448-1452.
- [7] Estan C, Keys K, Moore D, and Varghese G. Building a better NetFlow. SIGCOMM'04, Portland, ACM Press, 2004: 245-256.
- [8] NLANR. Available from: <http://pma.nlanr.net>, 2006.

韦安明: 男, 1975 年生, 博士生, 研究领域为 IP 网络测量及网络安全。

王洪波: 男, 1975 年生, 博士, 讲师, 主要研究领域为 IP 网络测量与管理。

林宇: 男, 1976 年生, 博士, 副教授, 主要研究领域为互联网服务质量管理与测量、P2P 计算。

程时端: 女, 1940 年生, 教授, 博士生导师, 主要研究领域为 QoS 分析与控制、P2P 计算、传感器网络等。