分组采样技术研究*)

高文宇　陈松乔　王建新
(中南大学信息科学与工程学院　长沙410083)

摘　要　随着网络带宽的不断提高,分组采样技术作为网络测量的手段越来越受到重视。因为在高速网络中对所有的分组进行实时的统计分析代价太大,而通过分组采样可以大大减少测量的代价,从而具有更好的可扩展性。本文对近来提出的一些分组采样技术进行了系统的分析和研究,主要对它们的原理、精度和效果进行了详细分析,并对其中存在的问题提出了一些改进的措施。

关键词　分组采样,流量测量,可扩展性

Study on Packet Sampling

GAO Wen-Yu　CHEN Song-Qiao　WANG Jian-Xin
(College of Information Science and Engineering, Central South University, Changsha 410083)

Abstract　With the improvement of bandwidth, packet sampling, as an important technique of traffic measurement, is getting more and more attention. Because in high speed networks, statistics of all the arrived packets is very expensive. Through packet sampling, the measurement cost is largely decreased, which makes the measurement more scalable. In this paper, a number of packet sampling techniques are analyzed systemically, focusing on their basic idea, accuracy, and effectiveness. Furthermore, some advices are presented to improve existing problems.

Keywords　Packet sampling, Traffic measurement, Scalable

1　引言

随着 Internet 的飞速发展,并逐步深入到社会生活的各个角落,人们对 Internet 提出了更高的要求,包括 Internet 的可控和可管理性。对网络的控制和管理就涉及到对网络当前运行状态的获取,特别是网络的动态属性,即数据在网络中的传输特性。对于 Internet 这样的分组交换网络来说,获取网络动态属性最直接的办法就是对分组进行俘获(capture),并进行统计分析。从理论上来说,俘获分组并进行统计分析并不是很困难的事,然而,随着网络技术的进步,网络带宽以惊人的速度提高。因此在一些骨干网上,单位时间到达的分组数飞速提高,因此若对所有的分组进行俘获和统计分析,系统开销就非常大,面临严重的可扩展性问题。因此分组采样就成了对网络进行测量的最好选择。

分组采样实际上很早就引起了研究者的重视。J. Jedwag 等人早在1992就对分组采样进行了详细的误差分析,并建立了用于流量估计的采样测量系统[1]。K.C. Claffy 在文[2]中对当时的各种分组采样技术进行了比较研究。在工程实践领域,Cisco[3],Juniper[4],InMon[5]等公司都在其路由器或商用软件中将分组采样技术应用到相关的网络测量中。另外,在分组采样的研究和标准化方面,IETF 正在开展一些工作,为此专门成立了一个工作组 PSAMP(Packet Sampling),一些相关的文档正在起草和讨论中[6]。

事实上,分组采样在很多领域都大有用途。如在网络安全中,通过分组采样来估计网络负载变化,通过负载的异常变化来发现是否遭受网络攻击。另外,在 QoS 控制中,通过分组采样来估计当前网络资源的使用情况,从而做出接纳控制决策;以及根据采样结果来形成计费账单[5,9]。另外,在文[7]中通过分组采样来建立分组在网络中的流动轨迹,从而获得网络中流量的空间特性。

本文主要研究了近年来分组采样技术的一些新的进展,对它们的基本原理、用途和效果等进行了系统的分析和比较。

2　N 中取1("1 out of N")

"1 out of N"是最古老的采样方法,因此,当分组采样最早被用于网络测量和监控时采用的就是"1 out of N"。简单性是"1 out of N"的最大优势,同时它也是其它一些复杂的(如依赖于内容的)采样技术的基础。

2.1　基本原理

"1 out of N"目前在网络工程实践中广为应用。其原理非常简单,就是在路由器(或相关的采样设备)对接收到的分组进行计数,每接收到 N 个分组就随机取其中之一作为样本进行统计分析处理(N 是不变的)。"1 out of N"由于其实现机制的简单而在多种场合有着广泛应用,下面我们结合两个典型应用实例来说明。

对于负载估计(即在某个时间段内收到的数据的总量)而言,假设在时间 T 内共有 m 个分组到达,其大小分别为 X_1, $X_2,\cdots,X_m$,则在 T 时间段内,网络的实际负载为 $V=\sum_{j=1}^{m}X_j$。再假设我们在 T 时间段内共进行了 n 次分组采样,每次采样得到的分组的大小分别为 $\hat{X}_1,\hat{X}_2,\cdots,\hat{X}_n$,那么,根据采样估计的负载为 $\hat{V}=\frac{m}{n}\sum_{j=1}^{n}\hat{X}_j$。

另外,"1 out of N"采样还被用于每流流量的估计。假设在 T 时间内到达了 m 个分组,而其中对 n 个分组进行了采

*)基金项目:国家自然科学基金重大研究计划(90304010)。高文宇　博士研究生,主要研究方向为计算机网络;陈松乔　教授,博士生导师,主要研究方向为软件工程、计算机网络;王建新　博士,教授,主要研究方向为计算机网络。

样，在 n 个被采样的分组中，有 k 个分组是属于某个流，那么，可以推断在所有的 m 个分组中共有 $m\times\frac{k}{n}$ 个分组是来自该流。

2.2 分析和评价

"1 out of N"最主要的优点就是实现简单，用于处理的系统开销小，可以用于各种场合。

但是对于不同的应用，这种简单的"1 out of N"难以保证精度。而如果无限制地提高采样比(减小 N)则会带来处理开销的增大，造成浪费。

例如，用"1 out of N"进行每流(per-flow)流量的估计，由于某些流的所有分组可能一次都没有被采样到，因此对这些流的流量无法进行估计。而且在样本量比较小的时候，对流量的估计精度是难以保证的。

因此，"1 out of N"作为最基本的采样技术应根据不同的应用场合做适当的改进，结合一些别的策略以便对最后的估计精度提供一定程度的保证。

3 自适应随机采样(Adaptive Random Sampling)

针对"1 out of N"采样在负载估计中的不足，即由于 N 是固定的，难以保证采样精度。文[8]中提出了一种自适应的随机采样方法。该分组采样方法能根据网络负载的变化动态调整采样频率，从而使得根据采样计算出的最终的负载估计值的误差率保持在一个给定的范围内。

3.1 基本原理

假设在时间 T 内共有 m 个分组到达，其大小分别为 X_1，$X_2,\cdots,X_m$，则在 T 时间段内，网络的实际负载为 $V=\sum_{j=1}^{m}X_j$。再假设我们在 T 时间段内共进行了 n 次分组采样，每次采样得到的分组的大小分别为 $\hat{X}_1,\hat{X}_2,\cdots,\hat{X}_n$，那么，根据采样估计的负载为 $\hat{V}=\frac{m}{n}\sum_{j=1}^{n}\hat{X}_j$。如果要将相对误差 $|\frac{\hat{V}-V}{V}|$ 控制在一个给定的水平下，如$\{\eta,\varepsilon\}$，即：

$$Pr\{|\frac{\hat{V}-V}{V}|>\varepsilon\}\leqslant\eta$$

根据中心极限定理，有：

$$Pr\{|\frac{\hat{V}-V}{V}|>\varepsilon\}=Pr\{\frac{\sqrt{n}}{\sigma}|\frac{1}{n}\sum_{j=1}^{n}\hat{X}_j-u|>\frac{\varepsilon u\sqrt{n}}{\sigma}\}$$

$$\approx 2(1-\Phi(\frac{\varepsilon u\sqrt{n}}{\sigma}))\leqslant\eta$$

u 和 σ 分别是分组大小的均值和标准差。$\Phi(\cdot)$是标准正态分布的累积分布函数。

因此，为了满足给定的精度水平$\{\eta,\varepsilon\}$，采样次数 n 应该满足：

$$n\geqslant n^*=(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon}\times\frac{\sigma}{u})^2=z_p\times S$$

其中，$z_p=(\frac{\Phi^{-1}(1-\eta/2)}{\varepsilon})^2$，$S=(\sigma/u)^2$

得到了最小采样次数 n^*，就可以计算出最优的采样比：

$$p^*=\frac{n^*}{m}$$

然后在下一个时间段依据此最优采样比进行分组采样。

在前面的计算中，下一时间段的 $S=(\sigma/u)^2$和 m 是未知的，因此，还需对 S 和 m 进行预测。在文[8]中采用了 AR(Auto Regressive)模型对这两个参数进行了预测。

3.2 分析和评价

该采样方法主要是针对网络负载估计，并在此情况下得到了估计精度与采样次数的定量关系。因此该采样方法理论上对于负载估计的精度是有保证的。但是注意到，在实践中，该方法需对下一时间段到达的分组的大小的分布特性及分组到达数做出预测，一方面，预测带来了较为复杂的计算；另一方面，预测也引入了更多的误差源。预测误差和采样产生的估计误差相互叠加后，整体误差更难以分析和控制。因此，该方法值得进一步改进。首先是对于下一个时段的分组到达数的预测事实上可以避免。因为在确定了在某个时间段 T 内的最小采样次数后，可采用时间触发的方式进行采样，从而无需对总的分组到达数进行预测。另外，对于分组大小的分布特性($(\sigma/u)^2$)的预测，也可以进一步优化，甚至避免。因为在网络上传输的分组的大小是有一定限制的，分组的大小不能是任意大的，也不可能小到仅有一个字节，所以$(\sigma/u)^2$的变化也被限定在一定的范围内。基于这个特点，可以采用更简单的机制得到较为准确的预测结果；甚至，采用最保守的方法，$(\sigma/u)^2$值取可能的最大值。通过这两个改进措施，可以极大地减少计算量，并提高预测精度，更重要的是，实施的复杂度降到与"1 out of N"类似，从而更具有工程的可行性。

4 用于识别"大"流的分组采样技术

在文[9]中，提出了两种用于识别网络流量中的"大"流的分组采样技术。所谓"大"流指其在单位时间所发送的数据量在该时间内网络总数据流量中占据一个较大的比例；或该流的带宽资源占用率较大(如超过总带宽的1%)。

根据以往的研究表明，在 Internet 上，少数的"大"流所发送的数据量占到了网络总数据量的绝大部分[10,11]。因此识别出"大"流既可以做近似的负载估计，也可针对"大"流实施网络优化。而在文[9]中，识别"大"流的主要目的是用于对"大"流的流量进行精确的统计，并以此形成计费账单。在该文中，提出了两种用于识别"大"流的技术。分别是"Sample and Hold"(采样并维持)和"Multistage Filter"(多重过滤器)。

从理论上讲，识别大流也非常简单，只需在路由器为每个流经的流建立一条记录，对流经的分组将其大小累加到该分组所属的流的记录，这样在测量间隔结束的时候，根据每条记录的累计值就可以知道哪些是大流。

同样，问题是对每个分组进行统计分析，以及在路由器区分每流进行统计对于高速网络来说是不现实的，存在严重的可扩展性问题。因此为了识别"大"流，只能选择性地为"大"流建立相应的记录，而不是对所有的流都建立相应的记录。

4.1 Sample and Hold

"Sample and Hold"基于一个非常直观的原理，即"大"流由于在单位时间发送的数据量更多，因此属于"大"流的数据被采样的概率也越大。基本过程如下：

当属于某个流的分组第一次被路由器采样到后，则该路由器为该流建立一条记录，用于统计所有的属于该流的数据量(即对到达的属于该流的分组的大小进行累加)，以后每当属于该流的分组到达时，就将分组大小在相应的记录进行累加。若属于某个流的分组被第2次或第3次采样到时，由于此时内存中已有该流的记录，因此只是将被采样到的分组的大小在相应的记录进行累加，而不再创建新的记录。这样，当测量时间间隔结束时，路由器的存储器中就保存了许多的流的大小的记录，凡是在该测量间隔内被采样过的流(即至少有一个属于该流的分组被采样到)在内存中都有一条记录，而其中累计值大的(超过某一阈值)就是识别出的"大"流。

对于该方法的效果，即识别“大”流的精确程度，我们可以结合下面的实例来讨论。

假设每个字节被采样的概率为 p，那么一个大小为 s 字节的分组被采样的概率就是 $p_s=1-(1-p)^s$（即1减去该分组的每一个字节都没有被采样的概率）。假设我们定义“大”流的标准是，在每个测量间隔，该流占用链路带宽的1%，因此最多有100个“大”流。那么，保守起见，我们在内存中开辟10000个用于区分流的记录空间。我们希望对每个字节采样的概率为 p，使得平均的采样次数是10000（因为即使10000次采样得到的都是属于不同的流的分组，则最多也只会生成10000条记录），如果在该测量间隔共有 C 字节能够被传送，则 $p=10000/C$。

根据前面的假设，考虑一个流 F 占用了1%的带宽。因此它在该测量间隔内至少发送了 $C/100$ 字节，由于我们以概率 $p=10000/C$ 对流过的每个字节进行随机采样，因此在该测量间隔，该流没有被登记在内存中的概率为：$(1-10000/C)^{C/100}$，该值近似于 e^{-100}。另外，流 F 在发送了它的5%的数据后就被登记在内存中的概率为：$1-e^{-5}>99\%$。因此该方法能以较高的精确度识别“大”流，并对“大”流的总流量作出较为精确的估计。

4.2 Multistage Filters

“Multistage Filter”采用了一种更巧妙的方法识别“大”流，而且对“大”流的数据量的统计更为精确（以非常高的概率保证最后的统计值就是“大”流的实际数据发送量）。在“Multistage Filter”中，采用了所谓的“过滤器”（Filter），形象地说，若一个流能成功地通过“过滤器”，就认为是一个“大”流。每个过滤器实际上是一个由一个 Hash 函数索引的表，表的每一格内容为一个流的大小的计数。在测量间隔开始的时候，表的内容都初始化为0。

参见图1，当一个分组到达时，将分组头部的用于标识流的部分（如协议类型、源地址、目的地址、端口号）输入到一个 Hash 函数中，得到的输出即为该流在表中对应的位置，从而将该分组的大小累加到表的相应位置。由于来自相同的流的所有分组计算出的 Hash 值是一样的，因此，来自同一个流的分组的大小在表中的同一个位置被累加，所以，表中的每一个位置所记录的值就是某个流的大小（累计发送数据量）。这样，如果某个分组到达时，它的大小在 Hash 表中对应的位置累加后，该位置的值超过了阈值 T，则这个位置所对应的流就被认为是一个“大”流，同时该位置的值被复制到另一个地方进行保存（这个地方专门用于保存识别出来的“大”流），以后再有分组能通过 Hash 表构成的“过滤器”时，就将该分组的大小直接累加到“大”流所对应的记录。

理想情况下，若 Hash 表的长度是无限的，而且 Hash 函数的值是不重复的，那么我们将在 Hash 表中记录所有流经的流的大小计数，并将识别出的“大”流挑选出来。

事实上，我们的 Hash 表的长度应该是很有限的，而且 Hash 函数的值也是会重复的，即 Hash 函数会把不同的输入值映射到同一个输出值。那么，根据前面的描述，属于不同的流的分组可能被映射到 Hash 表中的同一个位置。这会导致两类错误，一方面，两个或多个流的分组的大小在同一个位置进行累加，从而多个“小”流的数据量被累积后会超过阈值 T，而被错误地当作一个“大”流识别出来。另一方面，一些“小”流的数据量被累加到“大”流的数据量上，从而造成“大”流数据量计数的不准确。

为了解决上述问题，可以通过采用多个过滤器的方法。即使用多个不同的 Hash 函数处理到达的分组，并在多处进行累加。当一个分组到达时，只有当它经过多个 Hash 对应的地址的累积值都超过阈值 T 时，它才被认为是一个属于“大”流的分组，并最终被记录下来。如图1，假设当流 F 的某个分组到达时，根据其头部的信息由 Hash 函数 $h1$，$h2$，$h3$ 计算出的值在表中对应的位置分别为3，1，7；而这三个位置的累计值都超过了阈值 T（图1中 Hash 表的黑色部分表示该处的累计值超过了 T），这该分组所属的流 F 被认为是一个“大”流，从而该分组的大小被累加到内存中相应的位置。而若某个流的分组被 Hash 函数 $h1$，$h2$，$h3$ 计算出的值在表中对应的位置分别为3，1，2，由于 Hash 表3的第2个位置的累计值没超过 T，因此该分组所属的流不被认为是一个“大”流。

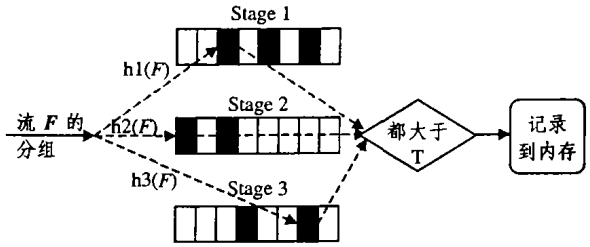

图1 Multistage Filter 示意图

对于该方法识别“大”流的效果，我们也结合下面的实例来分析。

假设一条100Mbytes/s 的链路（为计算简便，约定1M=1000k，1k=1000bytes），在某个时段有100，000个流通过，如果我们想在1s 的测量间隔内识别出占用带宽超过1%的流，再假设每个过滤器有1000条记录的位置，阈值 $T=1$M。

那么，我们来看一看，若某个流在这个测量间隔（1s）内发送了100k 数据，那么，若这个流要想在内存中被记录下来，它必须能通过每一个过滤器。它若要通过一个过滤器，则它还需别的流在它的记录中累加1M－100k＝900k 数据。假设 Hash 函数是均匀的，那么在这1s 内通过的所有数据为100M，减去该流的100k 后剩下100M－100k＝99900k，这99900k 数据若均匀地被分到每条记录，并且记录的最少值为900k，那么，最多占据999000k/900k＝111条记录，而每个过滤器有1000条记录位置，因此该流通过该过滤器的概率最大为111/1000＝11.1%。若同时使用4个不同的过滤器，则该流通过4个过滤器并在内存中留下记录的概率为 $(0.111)^4\approx1.52\times10^{-4}$，因此使用“Multistage Filter”可以有效地识别大流。

4.3 分析和评价

相当于前面的“Sample and Hold”来说，“Multistage Filter”能以更高的精度识别“大”流，并对“大”流的流量作出更精确的估计。但是，很显然，“Multistage Filter”需要对每个到达的分组进行多个 Hash 函数计算，而且在采样节点还要进行多个 Hash 表的管理，因此该方法在实现上较“Sample and Hold”要复杂。另外，该方法中 Hash 函数的选择对于方法最后的效果至关重要。一方面，在一个过滤器中，Hash 函数应能将分组的流标识信息进行均匀的映射，这需要根据分组头部的信息熵分析和对实际网络数据进行详细的分析来确定；另一方面，不同过滤器的 Hash 函数应相互独立，若不同的 Hash 函数之间存在相关性，就会使得多重过滤器的效果降低，从而使得精度下降。

识别“大”流实际上是一种面向“流”的采样。根据分组的

内容的不同(分组属于不同的流)决定是否对其进行采样。这种区分"流"的采样比纯面向分组的采样能获得更多的信息,即它可以获得每个"单流"的统计信息;而前面的两种采样主要目的是获得关于"聚集流"的统计信息(当然也可获得相对不太精确的"单流"信息)。当然,这种对流进行区分的采样付出的处理代价也更大,其中最大的挑战就是要在节点进行每流的区分和每流状态的管理(per-flow state management),在高速网络中,对每流的区分和管理会带来严重的可扩展性问题。所以,在识别"大"流的采样技术中,只选择性地对"大"流进行采样和统计处理,这在一定程度上解决了可扩展性问题,在实践中,识别"大"流的采样技术对于很多应用场合是有效的,因为,根据已有的对网络流量的分析,正常情况下,"大"流在很大程度上决定了网络流量的整体特性。

5 轨迹采样(Trajectory Sampling)

文[7]中提出了一种分组采样技术用于确定分组在网络中的流动轨迹。主要是采用基于 Hash 函数的方法使得某个分组要么不被采样,要么在其流经的路径上的所有网络节点都被采样。因此通过对来自全网各个节点的采样数据的收集就可以建立分组(或流)在网络中运动的轨迹。

5.1 基本原理

考虑在一个时间间隔 T 内,N 个分组($P_1, P_2, \cdots, P_N$)进入网络,分组 P_i 的轨迹就是该分组经过的节点(或链路)。在单播的情况下,轨迹就是一条从入口到出口的路径;在组播的情况下,轨迹就是一棵以入口节点为根的树。

轨迹采样的基本思想是,在每个网络节点,将到达的分组的头部的不变部分输入到一个 Hash 函数,根据 Hash 函数的输出是否满足预定的条件来决定是否对该分组进行采样。

分组头部的不变部分指,当分组在网络中传输时,分组头部的各个数据域中内容保持不变的数据域。如源地址、目的地址等就是不变部分,而 TTL(Time To Live)域、Header Checksum 域等就是在分组传输过程中会发生变化的部分。

因此,如果在网络中的所有节点都部署同样的 Hash 函数,那么将某个分组在该网络的任意节点进行 Hash 计算后,所得到的输出都是一样的(因为 Hash 的输入是分组头部的不变部分,并且都有节点的 Hash 函数是一样的)。因此,根据 Hash 函数的输出做出的采样决策在整个网络具有一致性,即某个分组要么不被采样,要么在其流经的路径上的所有网络节点都被采样。

当分组在某个网络节点被采样后,分组的相关信息(如源地址、目标地址、分组大小等)被提取处理,然后将这些相关信息发送到一个集中的测量收集系统。在测量收集系统对来自网络中各个节点的采样数据进行统一的分析处理,这样就可以建立分组在网络中的运动轨迹。

5.2 分析和评价

轨迹采样也是一种基于分组的内容的采样方法,它将分组内容作为 Hash 函数的输入,然后根据 Hash 函数的输出决定是否对其进行采样。该方法的目标很特殊,它并不像前面提到的分组采样方法一样关注分组或流的统计特性,而是关注于分组在网络中的流动轨迹。

该方法中采用 Hash 函数的计算结果决定是否对分组进行采样的方法非常巧妙。使得通过采样来确定分组的轨迹甚至网络拓扑成为可能,并且根据对在各个节点的采样结果的集中收集和统计分析,还可以建立整个网络的流量矩阵(即网络中各个节点之间的通信量),这对于网络优化和流量工程是非常有益的。但是这种方法需要在全网的每个节点部署同样的 Hash 函数计算装置,这对于实施和更新带来一定的困难。

结束语 分组采样技术在网络测量和监控中有着广泛的应用。而且,随着网络传输带宽的飞速提高,分组采样越来越成为网络分析和测量必不可少的手段。根据前面的分析,我们可以得到以下一些结论。

不同的采样方法适用于不同的网络状态的测量和估计。很难找到一种通用的采样方法,能满足所有的网络测量和分析目的。因此,在实施网络测量前,应确定所需获取的网络状态信息类型,根据不同的网络应用需求采取不同的分组采样方法。

由于采样毕竟只获得部分数据,因此根据部分数据推断整体的各种特性必然会带来一些误差,因此对于误差的估计和控制是非常重要的。文[8]中提供了一种很好的思路,即通过对采样策略的动态调整来控制误差。

随着采样在网络工程实践中的大量应用,在实施采样前,还需考虑采样带来的代价,包括计算代价(如根据分组内容进行 Hash 运算),采样数据的存储代价,采样数据的传输代价(如将分散的采样数据集中收集)。综合考虑上述因素,尽量降低采样给网络带来的额外负担。

随着测量需求的提高,不仅需要得到粗粒度的信息(如聚集流的信息),还需要获得细粒度的信息,如每流信息。因此,对分组采样而言,简单的随机采样难以达到所需的效果,而依赖于分组内容的采样则提供了更好的选择,可针对特定的流、特定的分组进行采样,从而能获得更精确的信息。而目前,对于简单随机采样的研究和应用已较完善,而依赖于内容的采样的研究则大有可为。

在今后的研究中,我们将结合经典的统计技术和网络的特有属性,根据网络应用的需求,寻找更为有效,并且能保证精度的分组采样方法,以满足网络测量对于精度和可扩展性的要求。

参 考 文 献

1 Jedwab J, Phaal P, Pinna B. Traffic estimation fro the largest sources on a network, using packet sampling with limited storage: [HP technical report]. HP Laboratories, Bristol, Mar. 1992

2 Claffy K C, Polyzos G C, Braun H W. Application of Sampling Methodologies to Network Traffic Characterization. In: Proc. ACM SIGCOMM'93, Sept. 1993

3 Cisco. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/netflow_techdocs.shtml

4 Juniper Packet sampling. http://www.juniper.net

5 InMon. sFlow accuracy and billing. http://www.inmon.com/PDF/sFlowBilling.pdf

6 PSAMP (Packet Sampling). IETF Working Group. https://ops.ietf.org/lists/psamp/

7 Duffield N G, Grossglauser M. Trajectory Sampling for Direct Traffic Observation. IEEE/ACM Transactions on Networking, 2001, 9(3): 280~292

8 Choi B Y, Park J, Zhang Z. Adaptive Random Sampling for Load Change Detection. ACM SIGMETRICS Performance Evaluation Review, 2002

9 Estan C, Varghese G. New Directions in Traffic Measurement and Accounting. In: Proc. ACM SIGCOMM'02, Aug. 2002

10 Fang W, Peterson L. Inter-AS traffic patterns and their implications. In: Proc. IEEE GLOBECOM, Dec. 1999

11 Feldmann A. Deriving traffic demands from operational IP networks: Methodology and experience. In: Proc. ACM SIGCOMM'00, Aug. 2000