# Estimating Cardinality Distributions in Network Traffic

## [Extended Abstract]

Aiyou Chen, Li Li, and Jin Cao
Bell Labs, Alcatel-Lucent Technologies
600 Mountain Ave, Murray Hill, NJ, USA
{aychen,erranlli,cao}@research.bell-labs.com

## ABSTRACT

Information on network host connectivity patterns are important for network monitoring and traffic engineering. In this paper, an efficient streaming algorithm is proposed to estimate cardinality distributions including connectivity distributions, e.g. percent of hosts with any given number of distinct communicating peers or flows.

## Categories and Subject Descriptors

G.4 [**Mathematical Software**]: Algorithm Design and Analysis

## General Terms

Algorithm

## Keywords

Cardinality distribution, streaming algorithm

## 1. INTRODUCTION

Understanding the communication connectivity patterns between network hosts, such as the number of distinct destinations or flows for each host, is important for many network management functions. Often, changes in the connectivity patterns will be reflected through changes in the *cardinality distributions* defined over these distinct counts (a distinct count is also called cardinality). Such distributions are very useful in network traffic monitoring and diagnostics. If the number of peers for many hosts increases over time as a result of increasing number of peer-to-peer (P2P) hosts, a new mode of the cardinality distribution may appear for the common number of peers that the P2P hosts are communicating with. On the anomaly detection side, if the number of peers for many hosts has a sudden increase as a result of attack activities such as port scans, the cardinality distribution may have a shift in its mode. Such distributional changes cannot be easily detected using marginal aspects such as entropy.

In this paper, we present a novel statistical approach for efficient online estimation of the afore-mentioned cardinality distributions in network traffic: given a number $n$, how many IP addresses communicate with $n$ different destinations or has $n$ number of flows as observed in a network. A naive solution would be estimating the cardinality for each host, which is memory expensive for a high speed large network. Instead, our approach maintains only a Flajolet-Martin sketch (at most 32 bits) [4] per host and then obtains accurate online estimation using likelihood inference that aggregates the information from individual hosts efficiently. We demonstrate its excellent performance using both corporate and university network traces for detecting anomalies and P2P connectivity pattern discovery.

Prior work on traffic feature distributions has focused primarily on volume [2, 5]. Marginal aspects such as cardinality counts (e.g. counting active flows) [3] and entropy mentioned above (e.g. entropy of packets over various ports) have been a subject of great interest.


*SIGMETRICS'08,* June 2–6, 2008, Annapolis, Maryland, USA.
ACM 978-1-60558-005-0/08/06.

## 2. METHODOLOGY

For simplicity, we demonstrate the approach under the scenario of estimating the host-peer counting distribution in a network. Let $m$ be the number of hosts to be monitored. If there are too many hosts, we apply a uniform sampling procedure to obtain $m$ sampled hosts.

Our approach consists of two modules: 1) online streaming using continuous Flajolet-Martin sketches [1]; 2) distribution estimation at the end of each measurement epoch using an EM algorithm. The online streaming module maintains a record (at most 32 bits) for each host. Let $\mathbf{Y}$ be a hash table of size $m$ initialized with values 1. We are interested in hosts with attribute $t$ (e.g. internal hosts) from a packet stream $\mathcal{T}$. Let $g$ be a universal hash function that maps an IP pair to a uniform random number in [0,1]. Let $h$ be a universal hash function that maps an IP to a number in $\{1, \cdots, m\}$. Given the attribute filter function $t$, universal hash functions $g, h$, and the number of hosts $m$, the online streaming module is summarized in Algorithm 1.

**Algorithm 1** Algorithm for updating sketches

1: Initialize a hash table $\mathbf{Y}$ of size $m$ with values 1.
2: **for** each new packet with IP pair ($s$,$d$) of $\mathcal{T}$ **do**
3: If $t(s) == 1$, hash $s$ to a bucket $i = h(s)$, and update $\mathbf{Y}[i]$ by $\min(\mathbf{Y}[i], g(s, d))$
4: If $t(d) == 1$, hash $d$ to a bucket $i = h(d)$, and update $\mathbf{Y}[i]$ by $\min(\mathbf{Y}[i], g(d, s))$
5: Return $\mathbf{Y}$ at the end of a measurement epoch.

Let $N$ be the number of peers that a random host communicates with. Since each host-peer corresponds to a uniform random number, the record $Y$ for that host is the minimum of $N$ independent uniform random numbers. Let $Z = -\log(1 - Y)$, then $Z$ is distributed as $\frac{\epsilon}{N}$, where $\epsilon$ is a

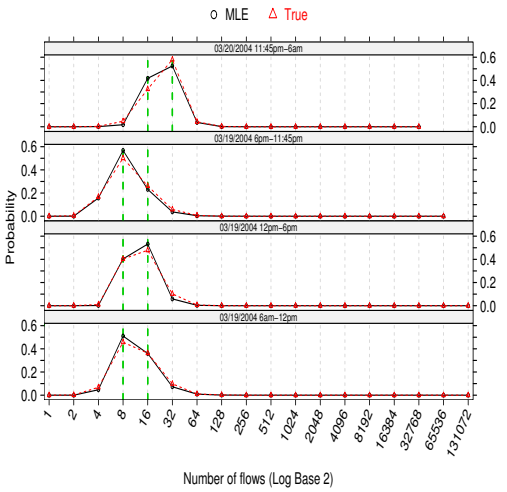


**Figure 1: Mode shift in the host-flow counting distributions during Witty worm outbreak.**

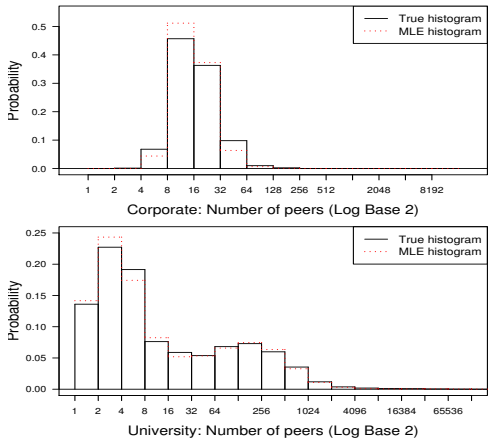


**Figure 2: Difference of host-peer counting distributions between a large corporate network and a large university network.**

unit exponential random number independent from $N$. The target is to estimate the distribution of $N$ from the vector $\mathbf{Y}$, whose components are independent. We parameterize the distribution as follows: let $\mathbf{p} = (p_0, \cdots, p_K)^T$ be the probability vector such that $P(N \in \{2^k, \cdots, 2^{k+1} - 1\}) = p_k$, $k = 0, \cdots, K$, where $K$ is a specified upper bound for $\log_2(N)$ and $\sum_{k=0}^{K} p_k = 1$. The estimation module for $\mathbf{p}$ is carried out using an EM algorithm and summarized in Theorem 1. Let $f_k(z) = q^{2^k}(1-q^{2^k})/2^k(1-q)$ with $q = e^{-z}$. Let $\mathbf{a}(z)$ be the column vector of $-\{\frac{d}{dz}(f_k(z)) : k = 0 \cdots, K\}$.

THEOREM 1. *Given a hash table $\mathbf{Y}$ of size $m$ from the online stream module, the maximum likelihood estimate (MLE) of $\mathbf{p}$ is unique unless the number of distinct values in $\mathbf{Y}$ is less than $K+1$. The MLE can be obtained from the iteration below no matter what starting point on the simplex is used:*

$$\mathbf{p} \leftarrow \frac{1}{m} \sum_{i=1}^{m} \frac{\mathbf{p} \cdot \mathbf{a}_i}{\mathbf{p}^T \mathbf{a}_i}, \qquad (1)$$

*where $\mathbf{a}_i = \mathbf{a}(-\log(1 - \mathbf{Y}[i]))$ and $\mathbf{p} \cdot \mathbf{a}_i$ is a column vector defined by component-wise products of $\mathbf{p}$ and $\mathbf{a}_i$.*

## 3. EXPERIMENTAL STUDIES

We now evaluate our algorithm on two real network traces. The first one is a trace collected at a large corporation's gateway router on March 19 and 20 of 2004, during which the network was hit by the Witty worm. The corporation's IP address space consists of two /16 address blocks, most of which are not used. The second one is one-hour trace collected at a large university's gateway router in the daytime of a weekday in Feb 2006. Its IP address consists of one /16 address block, about half of which is used. The corporate trace has about 131,000 internal hosts, and the university one has about 46,000 internal hosts. The space requirements of implementing our algorithm for these two networks are about 0.5M and 0.2M bytes respectively.

1). *Distributional change during unusual events.* Figure 1 shows the flow cardinality distribution (histogram) of internal hosts from trace 1 in 4 periods, each lasting for about 6 hours. Before the worm outbreak starting at about 11:45PM March 19, 2004, the mode is around 8-16. However, after that, the mode shifts significantly from 8-16 to 16-32, caused by external worm scanning. A closer look at the trace tells that the 8, 16 modes before worm outbreak are due to about 10 external hosts that scan at least half of the internal addresses. The long tail is due to the corporate servers such as VPN, DNS, etc. The estimation error is within 2%.

2). *Connectivity patterns suggest P2P existence.* We now compare the host-peer distributions for internal hosts between the corporate and university networks. The estimation error is within 2%. For trace 2 (university network), the bottom panel of Figure 2 tells that the distribution has two modes around 4 and 256 respectively. There are around 30% internal hosts for which the number of communicating peers is around the mode 256. A closer look at the trace reveals that a large amount of bi-directional traffic is exchanged among these hosts, which indicates that these hosts are very likely running P2P applications. In contrast, for the corporate network, the upper panel of Figure 2 shows that more than 98% of internal hosts have a single mode, with 4 to 64 communicating peers. This demonstrates that the host-peer connectivity patterns can vary dramatically among different organizations, depending on the applications.

## 4. REFERENCES


[1] A. Chen, J. Cao, and T. Bu. A simple and efficient estimation method for stream expression cardinalities. In *VLDB'07*, pages 171–182.

[2] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. In *SIGCOMM'03*, pages 325–336.

[3] C. Estan, G. Varghese, and M. Fisk. Bitmap algorithms for counting active flows on high speed links. *IEEE Trans. Networking*, 14(5):925–937, 2006.

[4] P. Flajolet and G. N. Martin. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences*, 31(2):182–209, 1985.

[5] A. Kumar, M. Sung, J. J. Xu, and J. Wang. Data streaming algorithms for efficient and accurate estimation of flow size distribution. In *SIGMETRICS/Performance '04*, pages 177–188.