Flow Measurements - Counting, Sampling, etc.

Abhishek Kumar
Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
akumar@cc.gatech.edu

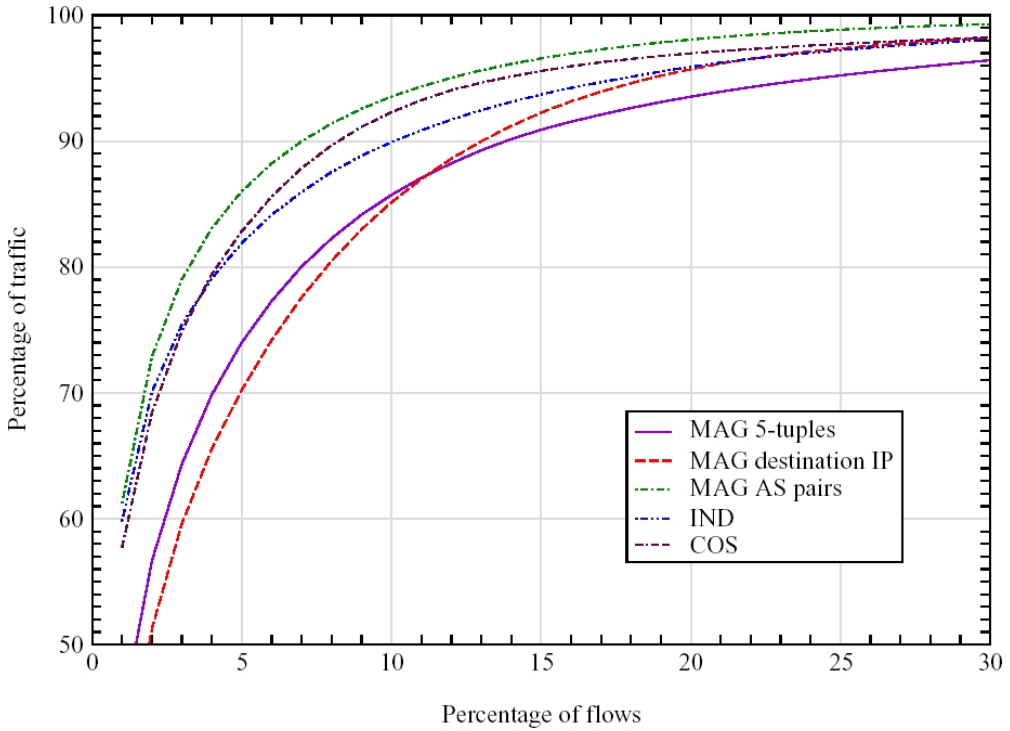
Overview

- **Why is per-flow measurement hard?**
- **Sampling - improvements and limitations.**
- **Lets filter out the elephants.**

Why is per-flow measurement hard?

- Keeping per flow state is not viable because of the high cost of maintaining data-structures.
- Majority of the packets belong to large flows, yet a majority of the flows are small.
- No clear definition of the "end" of a flow.
- Worst-case behavior of data-structures cannot be amortized due to the real time nature of the application.

The distribution of flow sizes



Cumulative distribution of flow sizes for various traces and flow definitions

New Directions in Traffic Measurement and Accounting. Cristian Estan and George Varghese, SIGCOMM '02

Paper 1

"Estimating Flow Distributions from Sampled Flow Statistics"

Nick Duffield, Carsten Lund and Mikkel Thorup
AT&T Labs–Research

SIGCOMM 2003

Sampling

Sample packets with a fixed probability p and trace headers of sampled packets. This is the approach used by Cisco Netflow.

Independent Sampling

Sample every packet independently with a probability $1/p$. Difficult to implement. Easy to analyze.

Periodic Sampling

Sample every $1/p$th packet with probability 1. Easy to implement. Difficult to analyze.

Distinguishing between independent and periodic sampling

Consider two sets of *sampled flow length frequencies*, $g = \{g_i : i = 1, \cdots, n\}$ **and** $g'\{g'_i : i = 1, \cdots, n\}$, **obtained through independent and periodic sampling respectively.**

Define chi-squared statistic as:

$$\chi = \sum_i \frac{(g'_i - g_i)^2}{g_i}$$

- **Null hypothesis (h_0)-** g **and** g' **are drawn from the same distribution.**
- **Alternate hypothesis(h_1) -** g **and** g' **are from different distributions.**
- **Fix significance level** $P_0 = \alpha$**=5%.** (P[**reject** $h_0|h_0$ **correct**])
- **Define** $P(\chi)$ **as the probability that a value of** χ **or greater is obtained, given** h_0**.**
- **Reject** h_0 **if** $P(\chi) < P_0$**.**

Results of Hypothesis testing

packets	Sampling Period N		
	10	100	1000
37M	2×10^{-5}	0.015	0.002
3.7M	0	0.044	0.16
0.37M	0	0.34	0.10

Table 1: Comparing Random and Periodic Sampling: Chi-square P-values. for sampling period $N = 10, 100$ and $1,000$, using subportions of trace COS

The two distributions are statistically distinguishable.

Another test - Weighted Mean Relative Difference (WMRD)

- **For a given length of sampled flow i, absolute difference =** $|g_i - g_i'|$ **.**
- **Relative difference =** $\frac{|g_i - g_i'|}{(g_i + g_i')/2}$**.**
- **To obtain the typical relative difference over all i, assign weight $(g_i + g_i')/2$ to the relative difference at sampled flow length i.**
- **Take the mean of this** *weighted relative difference* **to get:**

$$WMRD = \frac{\sum_i |g_i - g_i'|}{\sum_i (g_i + g_i')/2}$$

Results using WMRD

packets	Sampling Period N		
	10	100	1000
37M	0.0069	0.0063	0.0015
3.7M	0.023	0.022	0.032
0.37M	0.032	0.039	0.13

Table 2: Comparing Random and Periodic Sampling: WMRD. for sampling period $N = 10, 100$ and $1,000$, using subportions of trace COS

The accuracy is within 1%, which is acceptable for many applications.

Impact of Sampling

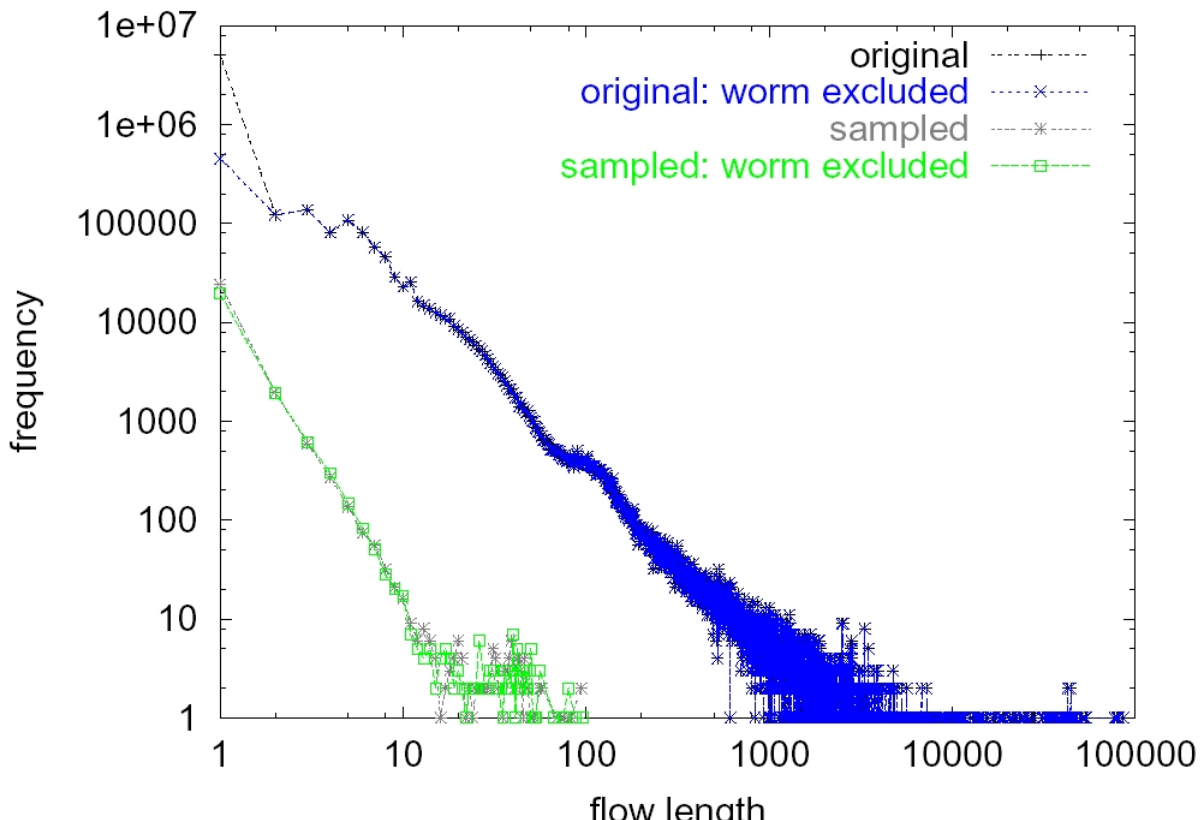


Figure 1: Original and Sampled flow length distributions. Sampling rate, $1/p = N = 1000$

TCP specific details in sampling

- **Sampling with probability p. Also, $1/p = N$.**
- **g_i – freq. of sampled flows of length i.**
- **If a flow has at least one SYN packet, it is called a SYN flow.**
- **g_i^{SYN} – freq. of sampled flows containing at least one SYN packet, and of length i.**

Two ways of scaling to obtain the total number of TCP flows

- $M^{(1)} = N \sum_{i \geq 1} g_i^{SYN}$ **is an unbiased estimator of the total number of SYN Flows.**

- $g_0 = (N-1) g_1^{SYN}$ **is an unbiased estimator of the total number of** *unsampled* **SYN flows.**

- $M^{(2)} = \sum_{i \geq 0} g_i$ **is an unbiased estimator of the total number of SYN flows.**

Two Scaling Estimators for TCP flows

Estimator 1 ($\hat{f}^{(1)}$):

- **Since SYN packets are sampled with probability $p = 1/N$, assign a weight of N to each g_j^{SYN}.**
- **g_j^{SYN} corresponds to flows of average size $1 + N(j-1)$.**
- **Distribute this weight evenly in a region of size N with its center at the above average.**

Estimator 2 ($\hat{f}^{(2)}$):

- **Each g_j corresponds to one flow of average size Nj.**
- **Distribute the weight g_j evenly in a region of size N with its center at the above average.**

Results of estimation through Scaling

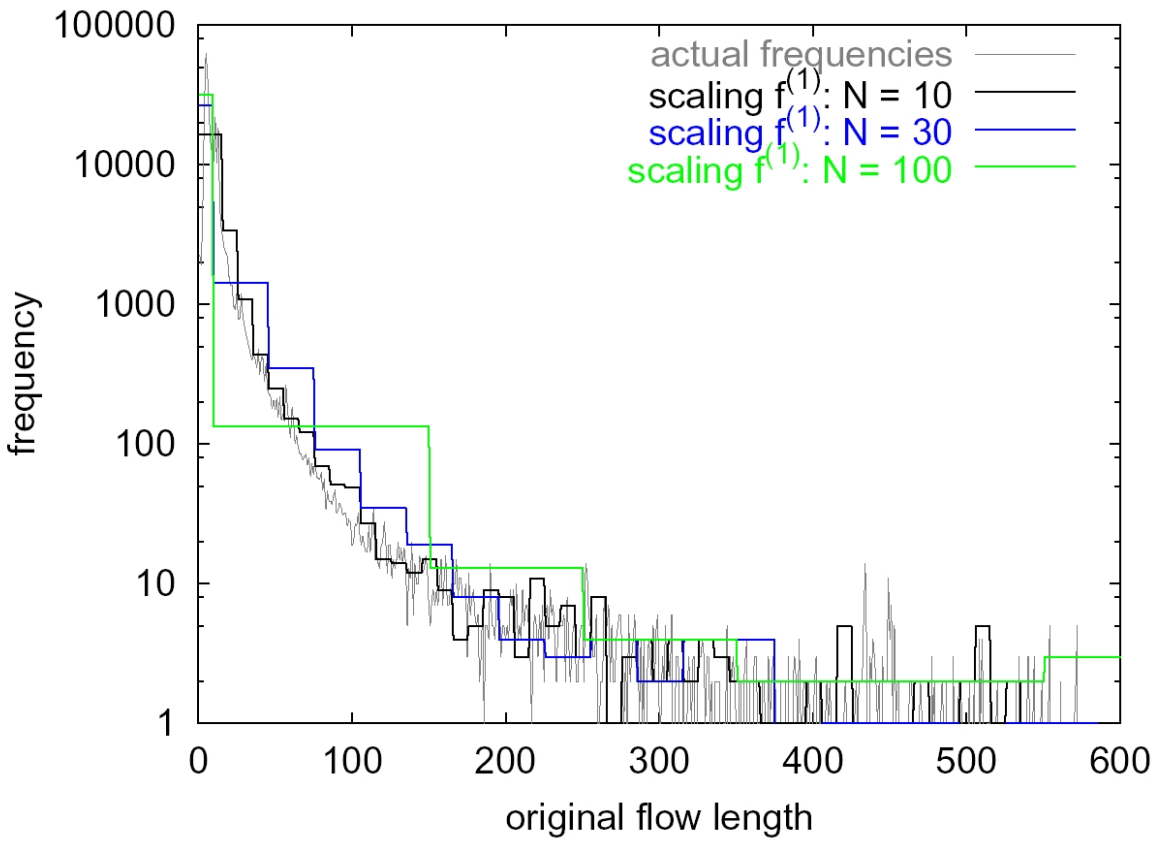


Figure 2: Original TCP flow length distributions and estimations using $\hat{f}^{(1)}$. Sampling periods, $N = 10, 30$ and 100.

Results of estimation through Scaling (contd.)

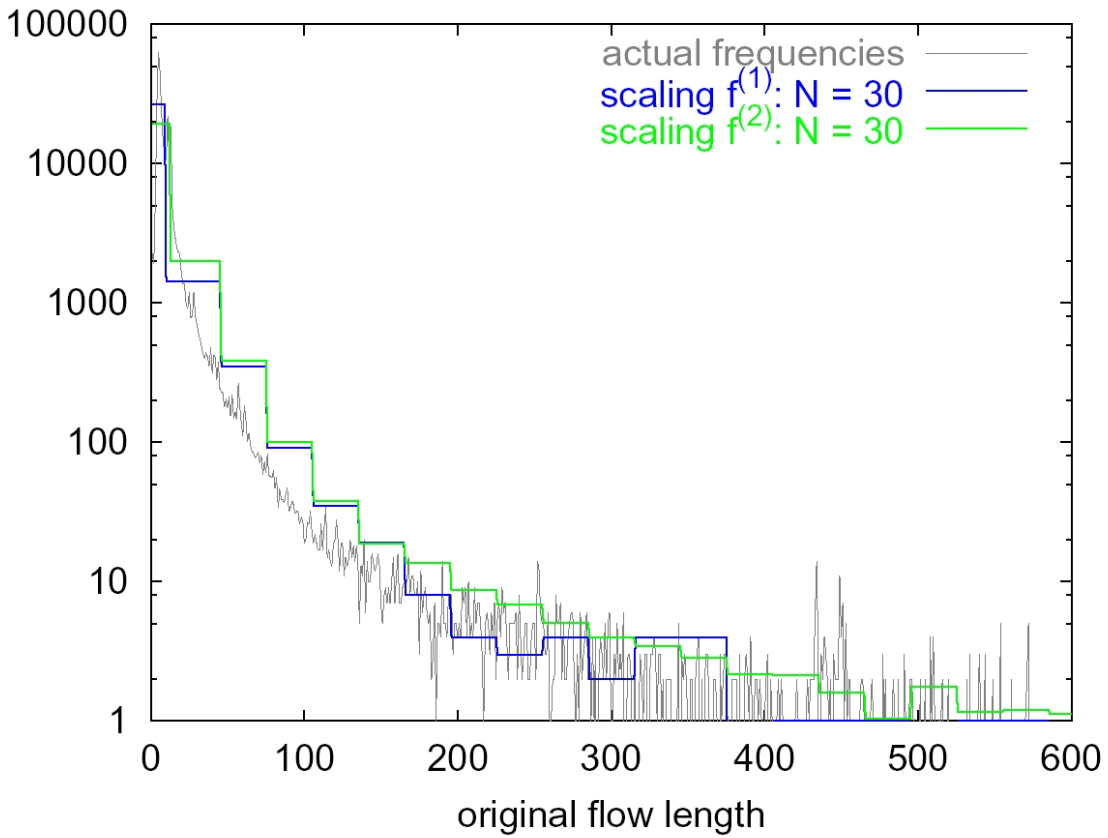


Figure 3: Original TCP flow length distributions and estimations using both $\hat{f}^{(1)}$ and $\hat{f}^{(2)}$, with sampling period $N = 30$. $\hat{f}^{(1)}$ is more accurate at low lengths while $\hat{f}^{(2)}$ has lower variability and hence is better for higher flow lengths.

Maximum Likelihood Estimation of TCP flow-length distribution

Let ϕ_i be the probability that an original TCP flow has i packets.

Overall objective:

From the sampled flow length distribution g^{SYN}, estimate the total number of original flows n, and their distribution ϕ.

Objective of ML Estimation

Find the distribution ϕ^* that maximizes the likelihood of observing the sampled flow length distribution g^{SYN}.

Maximum Likelihood Estimation (TCP flows) – $\hat{f}^{(3)}$

- **The probability that a flow of size i gives rise to a sampled SYN flow of length j is $c_{ij} = \binom{i-1}{j-1} p^{j-1}(1-p)^{i-j} * p$.**

- **Given ϕ, the probability of observing one sampled SYN flow of length j is given by : $\sum_{i \geq j} \phi_i c_i j$.**

- **The probability of g_j^{SYN} sampled SYN flows is $\left[\sum_{i \geq j} \phi_i c_i j\right]^{g_j^{SYN}}$.**

- **Taking log of the above term, we get:**

$$\mathcal{L}(\phi) = \sum_{j \geq 1} g_j^{SYN} \log \sum_{i \geq j} \phi_i c_{ij}$$

- **Maximize $\mathcal{L}(\phi)$ to obtain $\hat{\phi} = \phi^*$.**

ML Estimation – Extension to General (non-TCP) flows

- For general flows, we cannot directly estimate the number of original flows from the number of sampled flows.
- Two stage approach - First estimate the distribution ϕ' of flows that had at least one packet sampled.
- Recover the unconditional distribution ϕ from this.
- The mechanism to obtain ϕ' is similar to $\hat{f}^{(3)}$ with adjustments for using the sampled distribution g instead of g^{SYN}.
- This is the fourth estimation mechanism $\hat{f}^{(4)}$.

Performance of ML Estimation

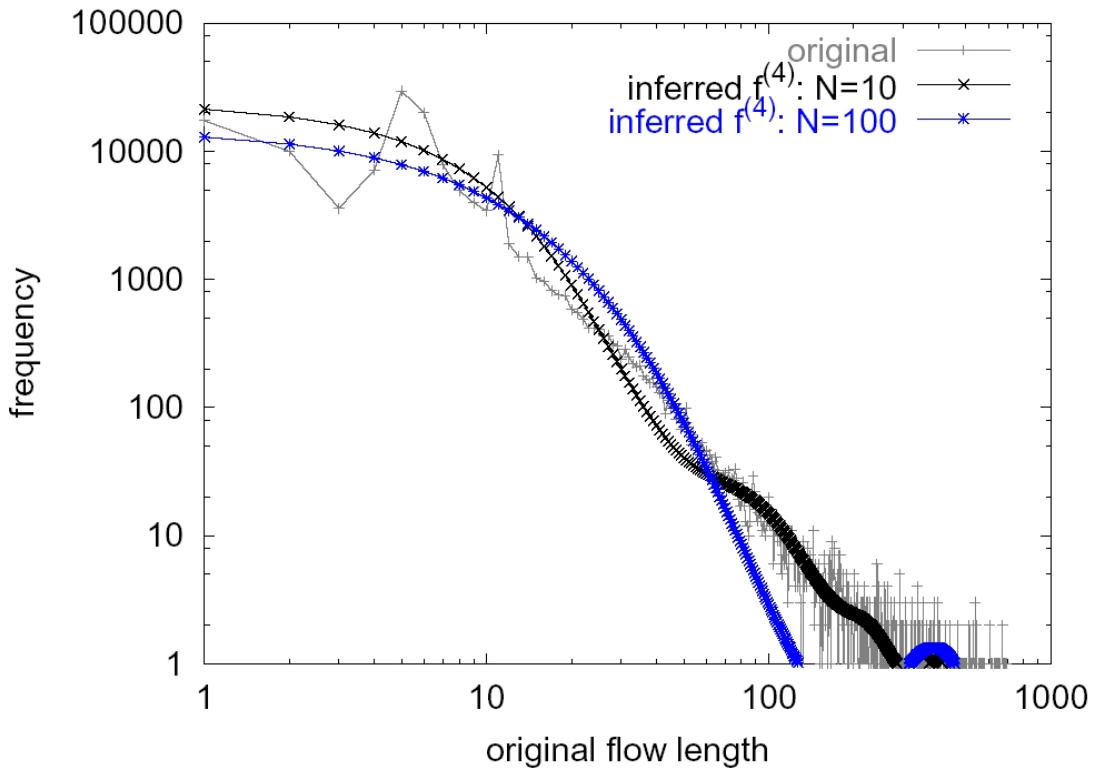


Figure 4: ML estimations using $\hat{f}^{(4)}$, with sampling period $N = 10$ and 100 on web traffic.

Performance of ML Estimation

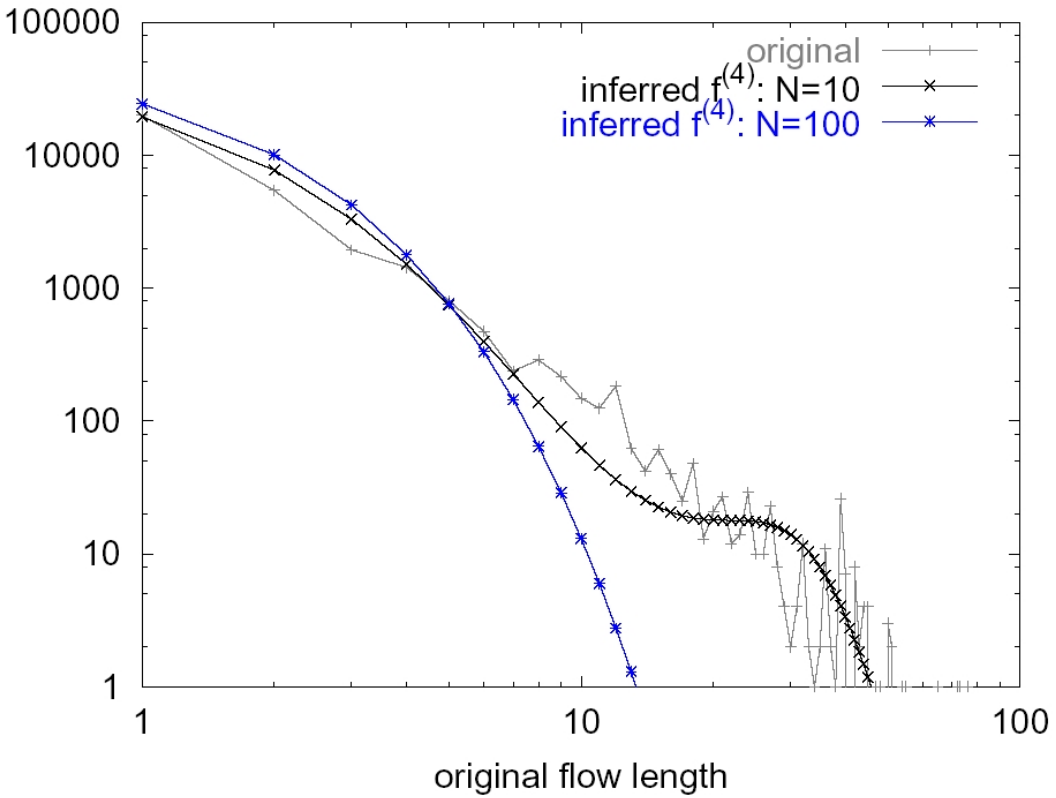


Figure 5: ML estimations using $\hat{f}^{(4)}$, with sampling period $N = 10$ and 100 on DNS traffic.

Paper 2

"New Directions in Traffic Measurement and Accounting"
Cristian Estan and George Varghese
UCSD
SIGCOMM 2002

Problem - Keep track of elephants

If we're keeping per-flow state, we have a scaling problem, and we'll be tracking millions of ants to track a few elephants.

– Van Jacobson, End-to-end Research meeting, June 2000.

- **Fast algorithm to identify (filter) packets from large flows.**
- **Maintain counters for large flows only.**
- **Success in tracking the largest few flows with limited memory.**

Motivation

- **Scalable Threshold Accounting: Measure all aggregates that utilize more than $z\%$ of the link. For small z, this accounts for most of the traffic.**

- **Real-time Traffic Monitoring: Allows rerouting of a small number of flows to reduce congestion. Can be used for attack detection.**

- **Queue managementPer-flow state for large flows only facilitates AQM mechanisms with a small memory.**

Sample and hold

Sample packets with probability p, and create a counter for them if not yet created. Once a counter for a flow has been created, count ALL packets in that flow.

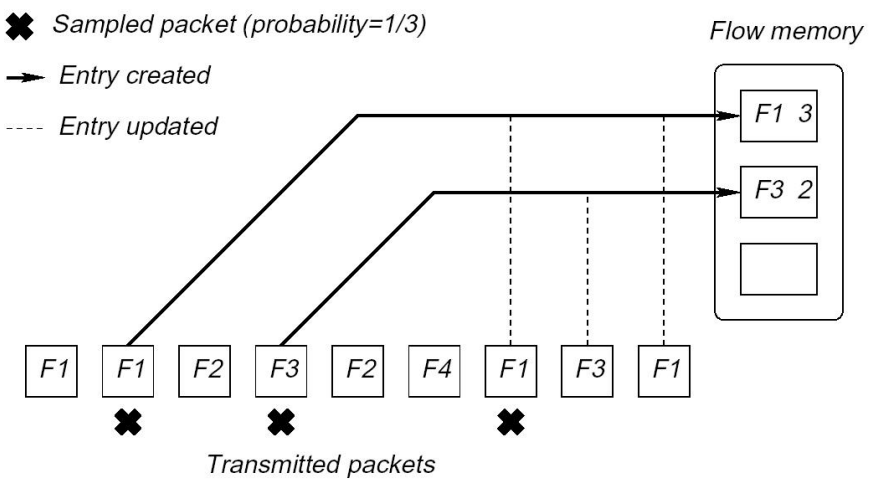


Fig. 1. The leftmost packet with flow label $F1$ arrives first at the router. After an entry is created for a flow (solid line) the counter is updated for all its packets (dotted lines)

Sample and hold vs. Netflow sampling

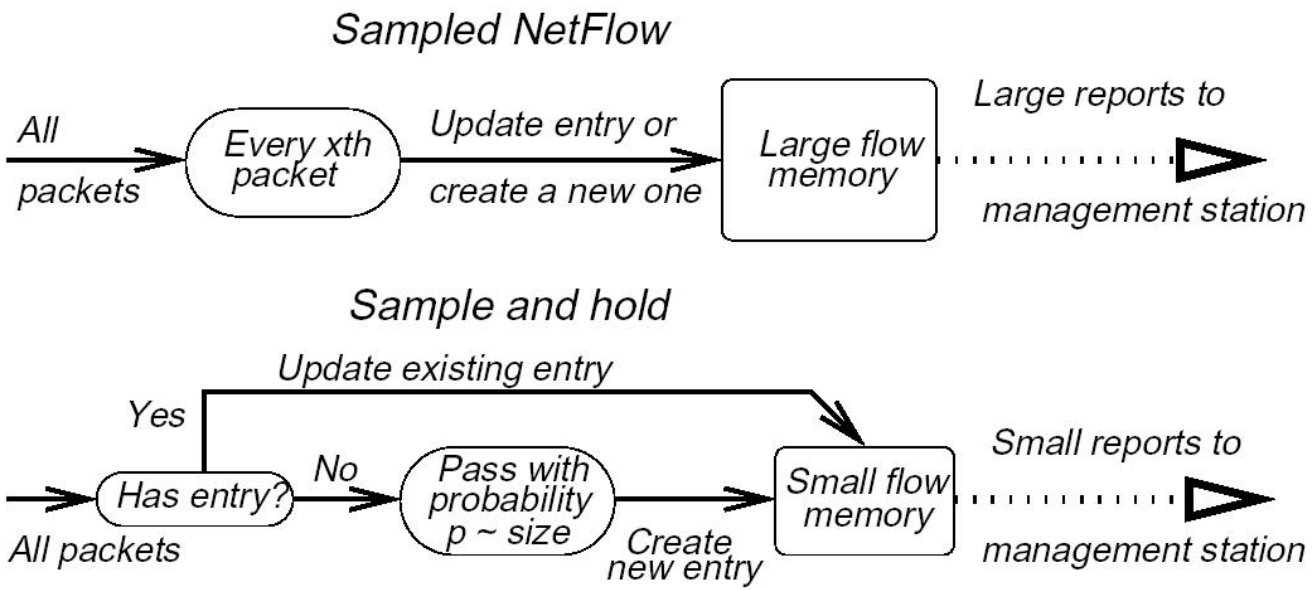


Fig. 2. Sampled NetFlow counts only sampled packets, sample and hold counts all after entry created

Multistage filters

- A stage is a table of counters, indexed by a hash function computed on packet flow ID.
- Counters are initialized to zero at start of measurement interval.
- Each packet arrival causes the incrementing of the corresponding counter.
- Packets whose corresponding counters are large "pass" through the filter.
- There are false positives but no false negatives.
- False positives occur due to collision of small flows with large ones, or collision of multiple small flows.

Multistage filters

To reduce false positives, use many such filters in parallel.

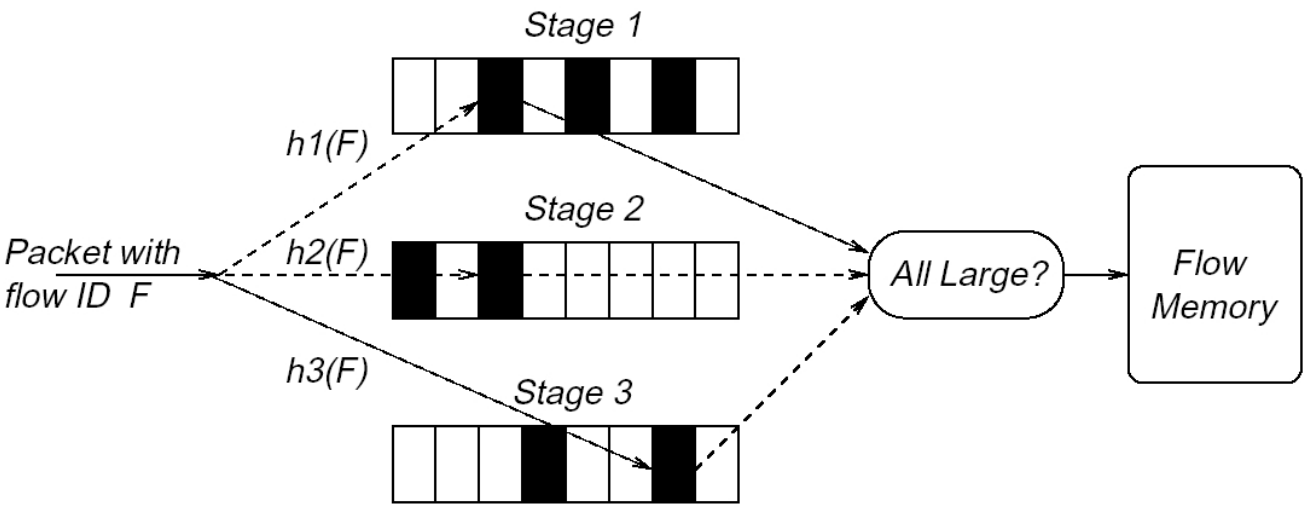


Fig. 3. In a parallel multistage filter, a packet with a flow ID F is hashed using hash function $h1$ into a Stage 1 table, $h2$ into a Stage 2 table, etc. Each table entry contains a counter that is incremented by the packet size. If *all* the hashed counters are above the threshold (shown bolded), F is passed to the flow memory for individual observation.

Conservative Update to Counting Bloom Filters

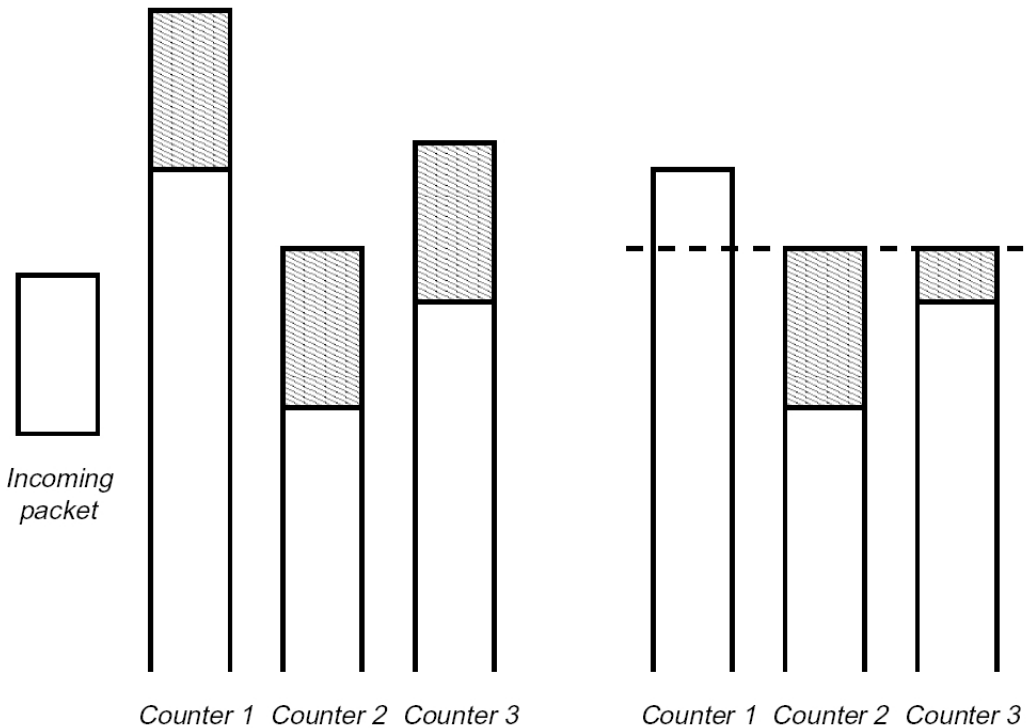


Fig. 5. Conservative update: without conservative update (left) all counters are increased by the size of the incoming packet, with conservative update (right) no counter is increased to more than the size of the smallest counter plus the size of the packet

Multistage filters

Serial filters

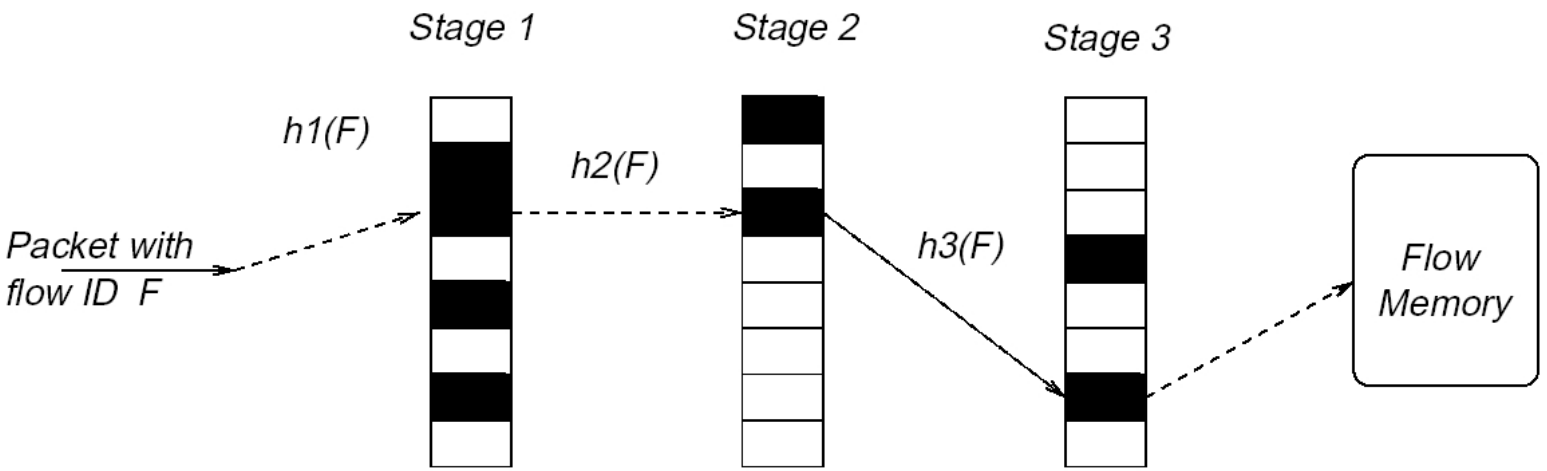
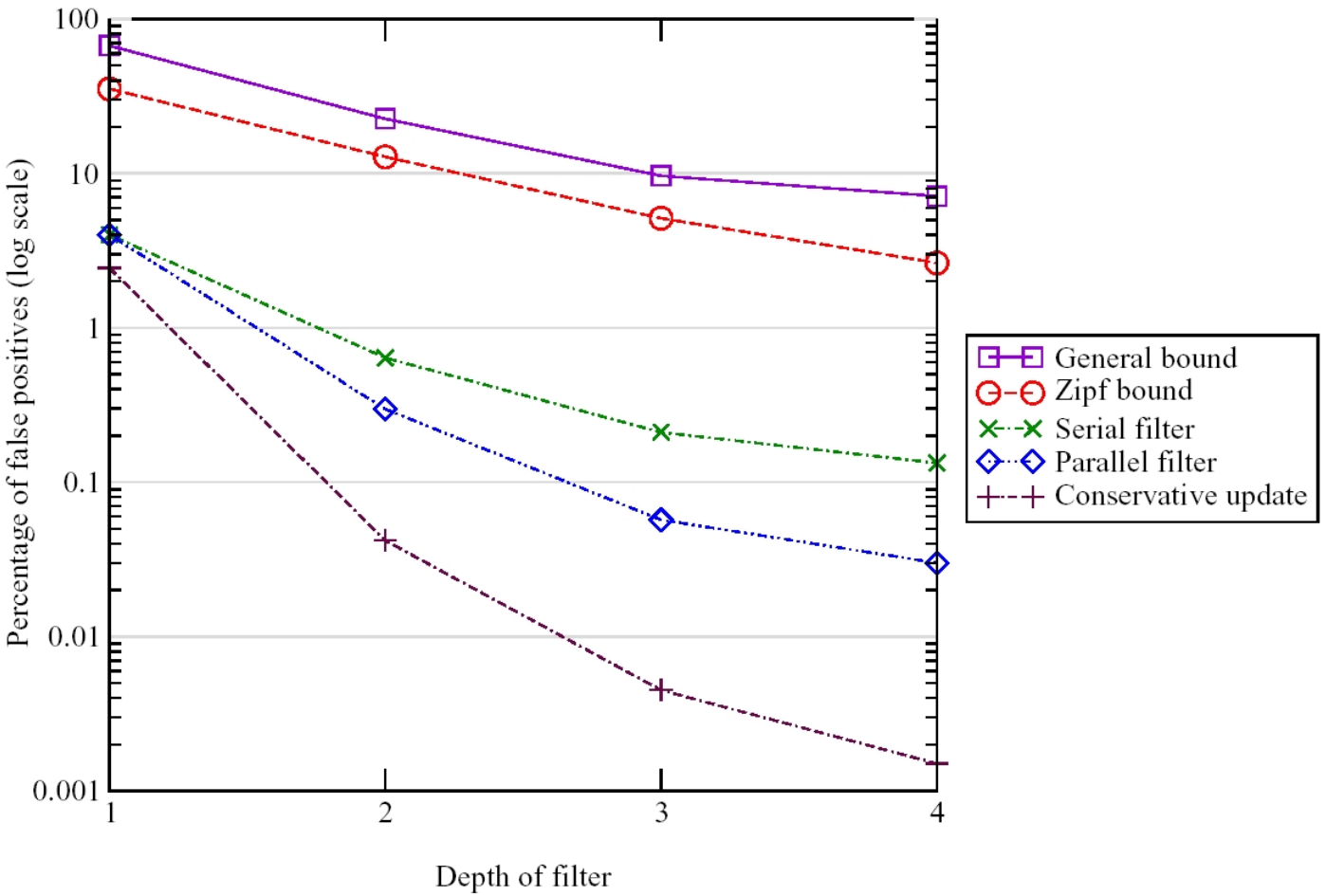


Fig. 4. In a serial multistage filter, a packet with a flow ID F is hashed using hash function $h1$ into a Stage 1 table. If the counter is below the stage threshold T/d, it is incremented. If the counter reaches the stage threshold the packet is hashed using function $h2$ to a Stage 2 counter, etc. If the packet passes all stages, an entry is created for F in the flow memory.

Performance of multistage Filters



Conclusions

- Sampling can be used to talk about the aggregate traffic distribution.
- Statistical techniques allow us to guess the flow distribution much better than naive scaling.
- Large flows can be identified using filtering mechanisms.
- Maintaining per-flow counters for large flows is possible with a small amount of fast memory.
- New techniques are coming out everyday !!