Sample Based Estimation of Network Traffic Flow Characteristics

by
Lili Yang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Statistics)
in The University of Michigan
2009

Doctoral Committee:

Professor George Michailidis, Chair
Professor Vijayan N. Nair
Associate Professor Anna Catherine Gilbert
Assistant Professor Stilian Atanasov Stoev

To my parents

ACKNOWLEDGEMENTS

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I am deeply grateful to my PhD advisor, Professor George Michailidis. His broad knowledge and generous personal guidance have been of great value for me. Great thanks also give to my committee members, Professor Vijayan Nair, Professor Stilian Stoev and Professor Anna Gilbert for their invaluable guidance and support.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

Introduction

1.1 Background

Large scale computer networks are an integral part of today's information society. They are large, multi-layered distributed systems that accommodate a large number of users with diverse needs and quality-of-service requirements. The variability of the underlying traffic demands and the complexity of their interactions present constant challenges to service providers, in their effort to match available network resources to demands. A significant effort by network engineers is devoted to develop network measurement systems and techniques. The objective of network measurement is to provide the necessary data for characterizing the state of the network, the performance experienced by users and control actions required. The latter range from the deployment of new network infrastructure to improve performance, to dynamically rerouting traffic to alleviate congestion, to detecting and neutralizing network attacks. Further, the time scales involved in such actions range from seconds for rerouting, to minutes and hours for attack detection to months for infrastructure upgrades. Other network management and engineering tasks for which network measurements prove useful include usage based accounting, service level agreement verification, fault detection and quality of service provisioning.

Over the years two approaches have emerged for obtaining network measurements: an active one, where probe packets are sent between network nodes and their loss

and delay characteristics measured and passive measurements, where actual traffic packet information is collected. The first approach led to the development of the active network tomography field [34] and is most suitable for characterizing network-wide quality of service performance. The second approach is useful for capturing aggregate demand and for characterizing traffic patterns. However, as link capacity has increased at an extremely fast rate over the years, passive measurement techniques can lead to enormously large volumes of data. For example, on a 100 megabit/sec Ethernet link operating at 20% constant utilization rate, the number of packets traversing it per minute is approximately 100,000.

As pointed out in [17], the current need for measurement capabilities is because they were not incorporated in the original design of network protocols. The basic network TCP/IP protocol is based on *best effort* service model, that offered no hard performance guarantees. The goal of the protocol was to provide reliable delivery of packets from source to destination. However, the introduction of more time sensitive services -from Web browsing to Internet telephony and television- and the need by service providers to differentiate traffic in accordance to service level agreements requires more fined grained measurements. Traditionally, routers were capable of providing aggregate traffic statistics; e.g. total number of packets processed over a five minute period. Detailed measurements, both at a fine time scale and at the traffic flow level, became technically feasible only over the last few years with the introduction of new generation routers equipped with extremely fast memory. Nevertheless, as indicated above, collection of detailed measurements becomes practically infeasible for high-speed networks. A *scalable* alternative is provided by *sampling techniques*. Several recent types of routers implemented *systematic sampling*; i.e. one out of K packets processed was selected for detailed measurement. The latest versions enable *probabilistic sampling*.

We briefly discuss next some basic network components. Data are transmitted

between hosts across the network in *packets*, which are forwarded along a network path comprised of one or more links joined by network elements such as routers or switches. Packets are organized in traffic *flows*; a flow is a collection of packets with a common property, known as the flow key. The flow key is usually specified by the protocol header of each packet. For example, the transmission of the packets belonging to an e-mail message constitute a flow, since they have the same source and destination and belong to the same application. With this structure, passive measurements can be divided into *Packet Monitoring* and *Flow Monitoring*.

Packet monitoring entails copying a stream of packets from the internal fast memory, then selecting, storing, analyzing and exporting information on these packets. Hence, sampling a packet requires copying to an external memory. Flow Monitoring is the way to collect statistics at the flow level. The flow cache memory of a router is where the keys of active flows are maintained. Recent recommendations regarding packet and flow sampling have appeared in the Internet Engineering Task Force working groups [24] and its implementation in high-speed routers [5]. One thing to notice is that flow monitoring is different from packet monitoring due to the possible heavy-tailed nature of flows. As noticed in [53], a small proportion of the flows may contain most of the packets/bytes. Therefore, omission of a single flow can lead to biased results.

Many sampling schemes along with inferences have been proposed in the literature, which can be basically summarized in two main categories: packet sampling, and hash-based data streaming approaches.

The most popular packet sampling techniques in commercial networks are sFlow [25] and NetFlow [5], where Bernoulli or systematic samplings are applied on the packets, and flow records are inferred from the samples keeping track of information such as the number of packets/bytes in the flow, the timestamp of the first and last packets, et al. Statistical inference of NetFlow sampled data is further investigated by

Duffield et al.[13], where they addressed the problem of estimating the distribution of the flow length (in terms of number of packets per flow) from NetFlow sampled TCP flows data. Subsequently Bruno Ribeiro et al. [38] extended this work by providing the Fisher information matrix and the Cramer-Rao bound for the corresponding estimations. These two papers are the most closely related ones to our flow characteristics estimation from sampled data, but it is worth noting that both of them are limited to NetFlow selected TCP flows, which is not necessary in our approach. In order to address the heavy-tailed distribution issue in the computer network, several other approaches that target longer flows have been proposed in the literature. Estan and Varghese [21] proposed a *sample and hold* approach, as well as *multistage filters*, which takes a constant number of memory references per packet and uses a small amount of memory. A cache lookup is performed for the key of each incoming packet, updated for the existing key but with a certain probability for the new key. A similar idea appeared in [29] RATE, where sampling two-runs automatically biases the samples towards the larger flows, therefore making the estimation of these sources more accurate. These sampling schemes lead to significantly smaller memory requirement compared to random sampling schemes, but introduce bias; hence, there is an inherent trade-off between estimation accuracy and flow cache size. Physical hard resource limitations impose serious constraints in practice even with sampling, because large flows may produce too many samples thereby overflowing the cache memory. A number of papers [4, 19, 28] discussed specifically the problem of sampling under hard resource constraints. A recent progress on sampling under hard resource constraints is achieved in Cohen[8] and Duffield[9], where a number of new sampling designs extend the *Netflow* and *sample and hold* approaches by allowing changes in the sampling rate. Another development is that sampling may also be applied directly to collected flow records. Duffield, Lund and Thorup in a serial of papers [14, 15, 18] proposed a threshold sampling in this scenario where longer flows

are more heavily sampled, since they exceed a predetermined threshold. In [15] and Choi et al. [3], this idea is extended to the situation where the sampling rate can be controlled and altered.

Other interesting topics in the literature include hash-based sampling and its extension to trajectory sampling[20]. In hash-based techniques, the router computes a flow label on the packet header on each packet arrival, where the flow label is determined by a hash function. Each router maintains a table of the flows it is currently monitoring. Although the focus of our study is mainly on packet sampling for inference at the flow level, hash-based sampling is also of interest since it allows to directly sample flows. In this sense, our proposed Two-Stage Sampling in Section 2.7.2 can be justified in practice. A number of related papers have appeared in the literature. For example, [30] estimates the flow size distribution from hash-based sampled flows by inverting the hash collision process in the counter array using an EM algorithm. The motivation of this paper is very similar to ours in the sense of estimation of flow size distribution, but it focuses on recovering of hash collision in the data streaming procedure while ours is on packet sampling recovery. Recently [31, 45] extend this work to subpopulation distribution estimation, and a network-wide approach, respectively. Other important work related to hash-based sampling are [47, 43] to name a few.

Apart from sampling schemes to recover flow level information on a single router/link, it is also important to study the traffic across the entire network. However, traffic can be observed multiple times at different observation points. The resulting measurements are likely to contain duplicates. This issue is briefly considered in Duffield et al. [12] studies the combination of sampled traffic measurements at multiple observation points to achieve an unbiased estimation of the total traffic. However, this study was based on the assumption that flow selection is feasible without a one-to-one mapping hash technique, and this is not practical in an outline feature to the best

of our knowledge. In chapter 4, we discuss our findings on sampling packets across the network, a novel topic in the literature. Some related network-wide sampling design issues deal with selecting observation points on a network [46]. This problem is different from the one addressed in Chapter 4, where our focus is on allocating samples to different observation points.

1.2 A Brief Review of Some Sampling Techniques

In this section we provide a brief overview of some basic concepts in sampling theory.

Consider a finite population of size N, which is relevant in networking applications. A sample of size n is collected according to some sampling mechanism without replacement. Let $Y_i,\ i = 1, 2, ..., N$ denote the values of a variable of interest in the population and denote by $y_j,\ j = 1, 2, .., n$ the observed values in the sample. A quantity of interest is the population mean $\bar{Y} = \sum Y_i/N$, or the population total $Y = \sum_{i=1}^{N} Y_i$. These statistics are also of interest in a networking context, since they provide information about the total and/or average amount of traffic in a link over a fixed amount of time.

The objective is to estimate these quantities from the sampled measurements. Under many sampling schemes (discussed below), the sample mean is an unbiased estimate of the population mean. Further, an unbiased estimator of the population total is given by the Horvitz-Thompson estimator [7], also called the π estimator [42], given by

$$\hat{Y} = \sum_{j=1}^{n} \frac{y_j}{\pi_j}, \tag{1.1}$$

where π_j correspond to sampling weights. $\pi_j = p$ in Bernoulli sampling , and $\pi_j = 1/n$ in Systematic sampling, which will be discussed later in this chapter. An attractive property of this estimator is its unbiasness [6].

We discuss next some popular and fairly easy to implement sampling schemes. In *Bernoulli* (also known as binary) sampling, each unit in the population is selected with common probability p, independently of everything else. Hence, $\pi_j = p$ for all j and the Horvitz-Thompson estimator takes the form

$$\hat{Y} = \sum_{j=1}^{n} y_j/p. \tag{1.2}$$

The variance of the estimate of the population total is given by

$$\mathrm{Var}(\hat{Y}) = (1/p - 1)\sum_{i=1}^{N} y_i^2 \tag{1.3}$$

The main advantage of Bernoulli sampling is its easy implementation, which makes it particularly suitable for network traffic. However, it is difficult to control the final sample size in the sense of large variation on the sample size especially with small sampling rate.

Simple Random Sampling (SRS) overcomes the problem of the controlling the size of the samples collected. The mechanism is to select n units out of the N such that every one of the C_N^n distinct samples has an equal probability of being drawn. In practice, it can be carried out by a draw-sequential or list-sequential scheme as illustrated in [7] and [42]. The resulting inclusion probability is $\pi_i = \frac{n}{N}, i = 1, ..., N$, another constant sampling probability design. The estimator for the population total and mean under SRS have the same form as under Bernoulli sampling, the only difference being replacing p by the selection ratio $f = \frac{n}{N}$. Occasionally people call the process of dividing the numerator of the π estimator by p in Bernoulli sampling or f in SRS *normalization*.

Systematic Sampling is often used as a substitute for SRS, because of its simple implementation mechanism. As mentioned in the previous section, the first sampling scheme available in commercial routers was systematic sampling. In principle, a first element is selected at random among the first a elements in the population list with equal probability. The rest of the sample is determined by systematically taking

every a-th element thereafter, until the end of the list. With this mechanism, there are a possible samples, each with chance of $\frac{1}{a}$ being drawn. It can be seen that only a single random draw is required to determine the sample, which leads to a constant computational complexity. When the order of the elements in the population is randomized or can be regarded as random, i.e., the $Y_j's$ are independent and drawn from the same population, systematic sampling can be analyzed as SRS. It is well known that systematic sampling can lead to biased results, if the items being sampled exhibit periodic characteristics.

The three previously discussed sampling schemes have a constant selection probability for each item in the population and are therefore called uniform sampling schemes. However, these schemes can lead to biased results. For example, suppose that the items in the population correspond to packets, which are organized in flows. Further, suppose that one is interested in the characteristics of the flows, but only packets can be sampled. If the distribution of the flow lengths is heavy tailed, it is more likely under any of these schemes that packets from longer flows are more likely to be observed. Therefore, the collected sample would not be representative for studying flow characteristics.

Alternatives to uniform sampling that address this shortcoming are stratified and clustered sampling mechanisms. In *Stratified Sampling*, the population under study is first divided into L non-overlapping subpopulations, called *strata*. Items in the same strata are of certain similar characteristics, based on some external knowledge. A SRS is then taken in each stratum. *Cluster Sampling* also requires non-overlapping subpopulations, called *clusters* in this scenario. However, in cluster sampling, a two-stage mechanism is employed. In the first stage a random sample of clusters is obtained and at the second stage, a random sample of items from the selected samples is chosen. Both stratified and clustered sampling are useful for traffic measurements, although their implementation may add an extra layer of computational complexity.

In this study, clustered sampling is used for estimating network flow characteristics, where we are given that packets within a flow share same flow key (see Section 2.4.2).

1.3 Traffic measurement topics considered

The importance of sampling mechanisms for traffic measurement was discussed in the introductory section. In this study we primarily focus on the following problems: estimation of the flow length and byte size distribution on a single link. The importance of this problem in networking stems from the fact that simple averages are not adequate for scheduling-routing purposes, as well as capacity planning, since the right tail (upper quantiles) of the flow length and byte size distribution play a crucial role. New types of sampling designs will be addressed and compared with the uniform sampling procedures. Further, extensions of monitoring packet and byte totals across the entire (or part) of the network with given topology are discussed in the context of allocation of a fixed number of samples.

CHAPTER 2

Non-Parametric Estimation of Network Traffic Flow Characteristics on a Single Router

2.1 Introduction and Literature Review

Understanding the characteristics of traffic flows is crucial for allocating the necessary resources (bandwidth) to accommodate users' demand. The problem of using sampled flow statistics in order to estimate the number of active flows in a link and their packet length and to a lesser extent bytes distribution has attracted some attention in the literature. Duffield [13] addressed the problem of estimating the distribution of the flow length (in terms of number of packets per flow) from sampled data. He proposed a non-parametric maximum likelihood estimator for the flow length distribution of the actual traffic and discussed the use of external information obtained from SYN packets for TCP flows. This is the closest work to the problem under consideration. In [13], the flow length distribution and the number of active flows in a link are treated separately, leading to an inaccurate estimate for the latter quantity. Further, not particular attention was paid to longer flows. The problem of estimating the byte size distribution has not been addressed before in the networking literature.

An earlier work on this topic was by Hohn and Veitch [23]. It mainly focuses on sampling the flows themselves, rather than estimating their characteristics from their packets. Further, it is assumed that the number of active flows traversing the

link is known a priori.

2.2 Framework

We assume that the data have been generated by the following *hierarchical* mechanism. First, a random variable M is generated from a discrete uniform probability distribution on the integers $\{1, \cdots, \Delta\}$ to represent the number of total active flows, where Δ is a large number reflecting the capacity of the link under consideration.

These M flows are comprised of $N_m,\ m = 1, ..., M$ packets each. The number of packets in each flow is referred to as the *flow length*. Flow lengths N_m's are generated independently according to a discrete probability distribution $\phi = \{\phi_1, \phi_2, ..., \phi_N\}$, where ϕ_i is the probability that a flow of length i. It is assumed that the maximum length of a flow is bounded by a large integer N. Further, the payload of each packet consists of $Z_m^{(\ell)},\ \ell = 1, ..., N_m$ bytes, generated independently from a certain distribution H. Hence the size of the m-th flow in bytes is given by $B_m = \sum_{\ell=1}^{N_m} Z_m^{(\ell)}$, which is referred to as the *flow size*.

This mechanism generates a total of T packets given by $T = \sum_{m=1}^{M} N_m$. Subsequently, packets are sampled from the generated flows using a Bernoulli sampling scheme; i.e. each packet is selected with pre-defined probability p, independent of any other characteristic and its size $Z_m^{(\ell)}$ observed and recorded. We can see that the total number t of packets sampled follows a Binomial distribution with parameters T and p. Further, sampled packets can be assigned to a particular flow, by observing its flow key obtained from information available in the packet header [17]. Let $n_m \geq 0$ be the number of sampled packets from flow m. Note that we may not sample all M flows and we have no information about the number of n_m's that are zero. Therefore, our data is only the set of all positive n_m's, which are referred to as *sample lengths* of flow m. For the sampled packets, we also have their corresponding byte sizes, and thus we also have the *sample flow sizes* $b_m = \sum_{\ell=1}^{n_m} Z_m^{\ell}$. Obviously,

there is no information about the composition of active flows for which none of their packets are contained in our sample. Let

$$I_{m\ell} = \begin{cases} 1 & \text{if the } \ell\text{th packet on the } m\text{th flow is sampled} \\ 0 & \text{if the } \ell\text{th packet on the } m\text{th flow is not sampled} \end{cases}$$

where $\ell = 1, ..., N_m,\ m = 1, \cdots, M$. We can see that

$$n_m = \sum_{\ell} I_{m\ell};$$

and therefore, the conditional distribution of n_m given N_m is Binomial $(N_m = i, p)$. See (2.5) below for the unconditional distribution of n_m's. Further we will call 'sampled flows' the ones that result after sampling packets from the link and organize them into flows according to their key. The true flows traversing the network will be called the 'original' ones. These two populations in our hierarchical mechanism are illustrated in Figure 2.1

Our objective is twofold:

(i) estimate the discrete flow length distribution ϕ;

(ii) estimate the flow size (expressed in bytes) distribution H.

REMARK: Observe, that in our setup the numbers of packets within each flow $\{N_m\}_{m=1}^M$ are themselves random variables given M. Therefore, our framework differs from the classical setup in statistical sampling theory, where the flow-sizes $\{N_m\}_{m=1}^M$ would be considered the target population and the goal would then be to estimate the "population" distribution of flow lengths corresponding to the N_m's, detailed in Appendix.

In our application, it is not appropriate to consider the $\{N_m\}_{m=1}^M$ as a fixed population and then to try to estimate the resulting distribution. This is because this distribution changes for different traces and observation periods. Indeed, in practice, when observing the link at a different point of time will typically lead to different

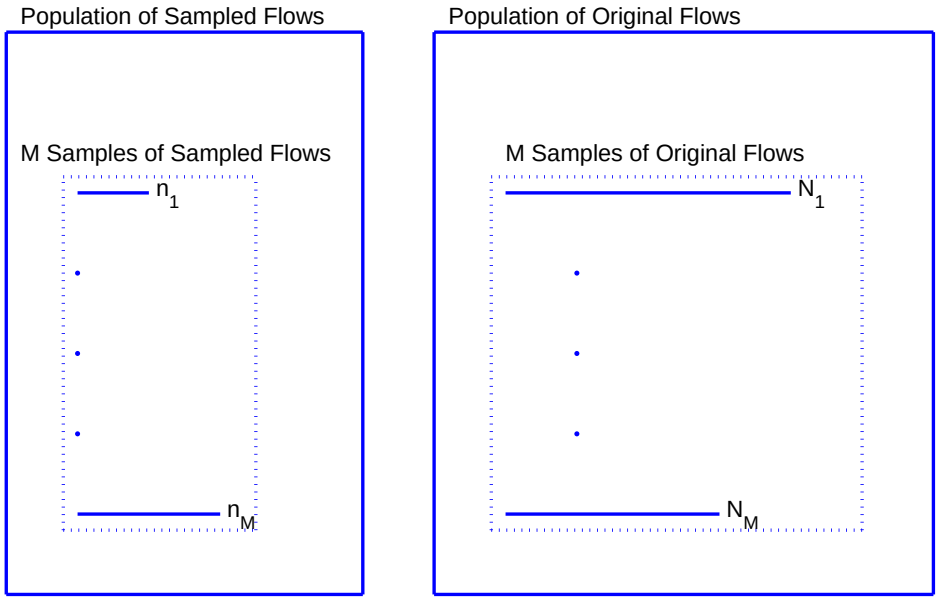


Figure 2.1: Two Populations in the Hierarchical Mechanism

collections of N_m's and often very different flow distributions. In our context, we assume that the N_m's are generated from a single non-random distribution ϕ, which we then try to estimate from sampled data.

Let $G = \{G_0, ..., G_N\}$, where G_j denotes the number of flows for which j packets have been sampled and $j = 1, ..., N$ and G_0 is the number of unobserved flows. We can write

$$G_j = \sum_{m=1}^{M} \mathcal{I}(n_m = j), \ j = 0, \cdots, N, \tag{2.1}$$

where $\mathcal{I}(\cdot)$ denotes the indicator function.

$$M = \sum_{j=0}^{N} G_j. \tag{2.2}$$

Notice that under this hierarchical mechanism, all flows have the *same* probability of belonging to the G_j-th group; namely, of having j of their packets sampled. This

probability is calculated as follows,

$$
\begin{aligned}
& \pi_{ij} \equiv \text{Prob(a flow of length } i \text{ and } j \text{ of its packets sampled, } i \geq j) \\
= \; & \text{Prob}(n_m = j \text{ and } N_m = i) = \text{Prob}(n_m = j | N_m = i)\text{Prob}(N_m = i) \\
= \; & \phi_i \begin{pmatrix} i \\ j \end{pmatrix} p^j (1-p)^{i-j} \equiv \phi_i \pi_{j|i}
\end{aligned} \tag{2.3}
$$

for any $m = 1, ..., M$; where $\pi_{j|i}$ is the probability that a flow has j packets sampled given original flow length i, which follows Binomial distribution, i.e.

$$
\pi_{j|i} = \begin{pmatrix} i \\ j \end{pmatrix} p^j (1-p)^{i-j}. \tag{2.4}
$$

Hence, the probability that a flow has j of its packets sampled is given by

$$
\begin{aligned}
& \text{Prob(flow has } j \text{ of its packets sampled)} \equiv \pi_j = \text{P}(n_m = j) \\
= \; & \textstyle\sum_{i=1}^{N} \text{Prob}(n_m = j \text{ and } N_m = i) = \sum_{i=1}^{N} \phi_i \pi_{j|i}
\end{aligned} \tag{2.5}
$$

for any $m = 1, ..., M$.

It can then be seen from equations (2.1), (2.2) and (2.5) that the conditional distribution of the G_j's, given the total number of active flows M, is a Multinomial distribution with parameters $(M, \{\pi_j\}_{j=0}^{N})$. We can then write the joint probability mass function of the G_j's as follows:

(2.6)

$$
\begin{aligned}
\text{Prob}(G_0 = g_0, ..., G_N = g_N) &= \textstyle\sum_{M=1}^{\Delta} \text{Prob}(G_0 = g_0, ..., G_N = g_N | M) f_M \\
&= \tfrac{1}{\Delta} \textstyle\sum_{M=1}^{\Delta} \text{Prob}(G_0 = g_0, ..., G_N = g_N | M) \\
&= \tfrac{1}{\Delta} \textstyle\sum_{M=1}^{\Delta} \begin{pmatrix} M \\ (M - \sum_{j=1}^{N} g_j), g_1, ...g_N \end{pmatrix} \prod_{j=0}^{N} \pi_j^{g_j} \\
&= \tfrac{1}{\Delta} \textstyle\sum_{M=1}^{\Delta} \begin{pmatrix} M \\ (M - \sum_{j=1}^{N} g_j), g_1, ...g_N \end{pmatrix} \prod_{j=0}^{N} (\sum_{i=1}^{N} \phi_i \pi_{j|i})^{g_j}
\end{aligned}
$$

where $g_0 = M - \sum_{j=1}^{N} g_j$.

From the above results of hierarchical mechanism, the likelihood of ϕ based on the observed frequencies $\{g_j\}_{j=1}^{N}$ of flows with $j > 0$ packets sampled is given by

$$L(\phi) = \frac{1}{\Delta} \sum_{M=1}^{\Delta} \binom{M}{(M - \sum_{j=1}^{N} g_j), g_1, ...g_N} \prod_{j=0}^{N} (\sum_{i=1}^{N} \phi_i \pi_{j|i})^{g_j}. \tag{2.7}$$

The objective becomes to maximize this likelihood function subject to the following constraints: $\sum_{i=1}^{N} \phi_i = 1$, and $\phi_i \geq 0$

Remark: Note that Duffield et al.[13] postulated a similar model, but treated the parameter M as given; specifically, it is estimated by $(1/p) \sum_{j=1} g_j^{SYN}$, where $\sum_{j=1} g_j^{SYN}$ is the total number of sampled SYN flows. In our work, we treat m as a random variable. Further, we provide a proper framework for justifying the above derived likelihood, and loose the restriction of TCP flows.

2.3 First EM Algorithm

We derive an Expectation-Maximization algorithm for maximizing the likelihood function given in (2.7). The main idea is that if the frequencies of all the original flows whose length is i packets were observed, then the estimation of ϕ would be rather trivial; hence, the EM algorithm [36] imputes such values (E-step) and then maximizes the likelihood over the parameters of interest (M-step). Its main steps are described next:

E-step:

We consider all the flows under study, and let

$$\{F_{ij} = \sum_{m=1}^{M} \mathcal{I}(n_m = j \text{ and } N_m = i),\ j = 0, \cdots, N; i = 1, \cdots, N\} \tag{2.8}$$

denote the number of flows that contain i packets out of which j have been sampled. We have that

$$M = \sum_{i=1}^{N} \sum_{j=0}^{N} F_{ij}. \tag{2.9}$$

Further, notice that the F_{ij} variables uniquely determine the observed variables G_j; specifically, we have $G_j = \sum_{i=1}^{N} F_{ij}$, $j = 1, ..., N$, and $G_0 = \sum_{i=1}^{N}\sum_{j=0}^{N} F_{ij} - \sum_{j=1}^{N} G_j$.

From equations (2.8), (2.9) and (2.3), we know that $\{F_{10}, ..., F_{NN}|M\}$ follows a Multinomial distribution with parameters $(M,\ \{\pi_{ij}\})$. Then, the joint probability distribution of the F_{ij} and M is given by

$$\begin{aligned} & \text{Prob}(F_{10} = f_{10}, ..., F_{ij} = f_{ij}, ..., F_{NN} = f_{NN}, M = m) \\ = \ & \text{Prob}(F_{10} = f_{10}, ..., F_{ij} = f_{ij}, ..., F_{NN} = f_{NN}|M) f_M(m) \\ = \ & \tfrac{1}{\Delta} \begin{pmatrix} M \\ \\ f_{10}...f_{NN} \end{pmatrix} \prod_{i=1}^{N}\prod_{j=0}^{N} \pi_{ij}^{f_{ij}} \end{aligned} \tag{2.10}$$

where π_{ij} is defined by equation(2.4). And hence the likelihood is given by

$$L(\phi) \propto \begin{pmatrix} M \\ \\ F_{10}...F_{NN} \end{pmatrix} \prod_{i=1}^{N}\prod_{j=0}^{N} (\phi_i \pi_{j|i})^{F_{ij}}.$$

Therefore, log-likelihood is given by

$$l_c(\phi) = log[\begin{pmatrix} M \\ \\ F_{10}...F_{NJ} \end{pmatrix}] + \sum_{i=1}^{N}\sum_{j=0}^{N} F_{ij} log(\phi_i \pi_{j|i}). \tag{2.11}$$

The expectation $Q(\phi, \phi^{(k)})$ of l_c conditional on variables $\{G_j\}_{j=1}^{N}$ and $\{\phi_i^{(k)}\}$, the estimated at the $k-$th step of the algorithm, is given by:

$$\begin{aligned} Q(\phi, \phi^{(k)}) \ &= \ E(log[\begin{pmatrix} M \\ \\ F_{10}...F_{NN} \end{pmatrix}] | \{G_j = g_j\}_{j=1}^{N}, \phi^{(k)}) \\ &+ \ \textstyle\sum_{i=1}^{N}\sum_{j=0}^{N} E(F_{ij}|\{G_j = g_j\}_{j=1}^{N}, \phi^{(k)}) \log(\phi_i \pi_{j|i}) \end{aligned} \tag{2.12}$$

The first term can be ignored, because it does not contain the parameters of interest $\{\phi\}$, which are to be estimated in the M-step. For the second term, notice that $F_{ij}|(\{G_j\}_{j=1}^{N}, \phi^{(k)})$ is determined uniquely by $F_{ij}|(\{G_j, \phi^{(k)}\})$ when $j \geq 1$, which follows a Multinomial distribution with parameters $(G_j, \pi_{i|j}^{(k)})$, where

$$\pi_{i|j}^{(k)} = \frac{\phi_i^{(k)} \pi_{j|i}}{\sum_{j=\ell}^{N} \phi_\ell^{(k)} \pi_{j|\ell}} \tag{2.13}$$

is the probability that given a flow with sample size j, it contains actually i packets in total, $j = 0, 1, ..., N$ and $i = j, ..., N$. Therefore,

$$E(F_{ij}|(\{G_j\}_{j=1}^N, \phi^{(k)})) = G_j \pi_{i|j}^{(k)}. \tag{2.14}$$

However, when $j = 0$ a specific form for $E(F_{i0}|(\{G_j\}_{j=1}^N, \phi^{(k)}))$ can not be obtained, since the variable G_0 is not observed. Further, there is no specific distributional assumption connecting G_0 and the G_j's, $j \geq 1$. In order to overcome this difficulty, the *nuisance parameter* $G_0^{(k)}$ is introduced and updated by

$$G_0^{(k)} = \frac{\sum_i \phi_i^{(k-1)} c_{i0}}{1 - \sum_i \phi_i^{(k-1)} c_{i0}} r,$$

weighting the updated odds ratio, defined by the probability of observing a sampled flow of length zero over the probability of observing non-zero flow lengths, on the number of observed flows $r = \sum_{j=1}^N G_j$. We then get that

$$E(F_{i0}|(\{G_j\}_{j=1}^N, \phi^{(k)})) = G_0^{(k)} \pi_{i|0}^{(k)}, \tag{2.15}$$

$$M^{(k)} = r + G_0^{(k)}.$$

M-step: Define $\phi^{(k+1)} = \arg\max Q(\phi, \phi^{(k)})$, such that

$$\sum_{i=1}^N \phi_i = 1, \text{ and } \phi_i \geq 0 \text{ for } i = 1, 2, ..., N. \tag{2.16}$$

The method of Lagrange multipliers gives

$$\phi_i^{(k+1)} = \frac{\sum_{i \geq j \geq 1} G_j \pi_{i|j} + G_0^{(k)} \pi_{i|0}}{\sum_{i=1}^N (\sum_{i \geq j \geq 1} G_j \pi_{i|j} + G_0^{(k)} \pi_{i|0})}, \tag{2.17}$$

where $\pi_{i|j}$ again corresponds to the probability that for a flow of length i given j of its packets have been sampled.

In the implementation of its algorithm, we initialize the $\phi_i^{(0)}$ by the corresponding observed frequency of $j = \llcorner ip \lrcorner$ as follows:

$$\phi_i^{(0)} = \frac{g_j}{\sum_{k=1}^{N} g_k + g_0^{(0)}}, \tag{2.18}$$

where $g_0^{(0)}$ is estimated by the odds ratio

$$g_0^{(0)} = \frac{\sum_{i=1}^{N} \frac{g_j}{\sum_{k=1}^{N} g_k} c_{i0}}{1 - \sum_{i=1}^{N} \frac{g_j}{\sum_{k=1}^{N} g_k} c_{i0}} r, \tag{2.19}$$

and r is the number of observed flows given by $r = \sum_{j=1}^{J} g_j$.

The algorithm iterates between the E- and the M-steps until a prespecified convergence criterion is met.

Remark: Theoretically, we would get an estimate of $\{\phi_i\}_{i=1}^{N}$. In practice, for speeding up the algorithm we focus on a limited set of i values, determined by the set $S_I = \{\llcorner \frac{j}{p} \lrcorner \bigcup \llcorner \frac{1}{2p} \lrcorner\}$, for $j \in S_J$ the collection of observed sample lengths. The intuition behind this selection is that given flow length i, the sampled packets j are drawn from a Binomial distribution with parameters i and p. We then have $E(j) = ip$, which guides our selection of S_I.

2.4 Estimation of Flow Lengths given Sampled Lengths

The next objective is to come up with an estimate of the length of sampled flows. Given the estimated flow length distribution ϕ, we can, through a straightforward application of Bayes formula, obtain the posterior probability distribution of a flow being of length L given that K of its packets have been sampled. Specifically, let $\mathcal{P}(L|K),\ K = 1, 2, ..., N,\ L = 1, 2, ..., N$ denote this probability distribution, $\mathcal{P}(L, K)$ the joint distribution of flow lengths and sampled flow lengths and $\mathcal{P}(K)$ the distribution of observing a sampled flow of length K. We then have that

$$\mathcal{P}(L|K) = \frac{\mathcal{P}(L, K)}{\mathcal{P}(K)} = \frac{\pi_{K|L}\phi_L}{\sum_{L=1}^{N} \pi_{K|L}\phi_L}. \tag{2.20}$$

For any given sampled flow of length $K = k$, we provide next two estimators of the original flow length $L(k)$.

(1) *Average.*

$$\hat{L}(k) = \mathrm{E}(L(k)) = \sum_{L=1}^{N} L\mathcal{P}(L|K = k). \tag{2.21}$$

This estimator (the posterior mean) is the weighted average of all possible flow lengths.

(2) *Maximum a posteriori estimator.*

$$\tilde{L}(k) = argmax_{L=1}^{N}\mathcal{P}(L|K = k). \tag{2.22}$$

The estimated length corresponds to the value that maximizes the probability of observing a sampled flow of length k.

The attraction of the second estimator ($\tilde{L}(k)$) is that it minimizes the total risk, where the risk function is given by

$$R(\hat{L}(k), L) = \mathrm{E}[LOSS(L, \hat{L}(k))||\text{True flow length is L}],$$

with the loss function $LOSS$ being

$$LOSS(A, B) = \begin{cases} 1 & \text{if } A \neq B \\ 0 & \text{if } A = B \end{cases}$$

On the other hand, the first estimator ($\hat{L(k)}$) gives smoother estimates, and our performance evaluation focuses on the first estimator.

2.5 Estimate on the Flow Size

The next task is to estimate both the flow size distribution H (in bytes) and the size of the active flows $B_m : n_m > 0, m = 1, 2, ..., M$ nonparametrically. The following observation motivates the proposed estimation procedure. One can view the sampling process of flow sizes as a one-stage cluster sampling scheme [7]; therefore,

the classical estimator is given by $\hat{B}_m = (\sum_{\ell=1}^{n_m} Z_m^{(\ell)})/p$. Subsequently, one can estimate non-parametrically the distribution H, using the estimates $\hat{B}_m$ as data. The above procedure works well for light tailed distributions, but proves problematic for heavy tailed ones, due to inherent length biasedness of the sampling scheme, i.e., long flows are more likely to be selected.

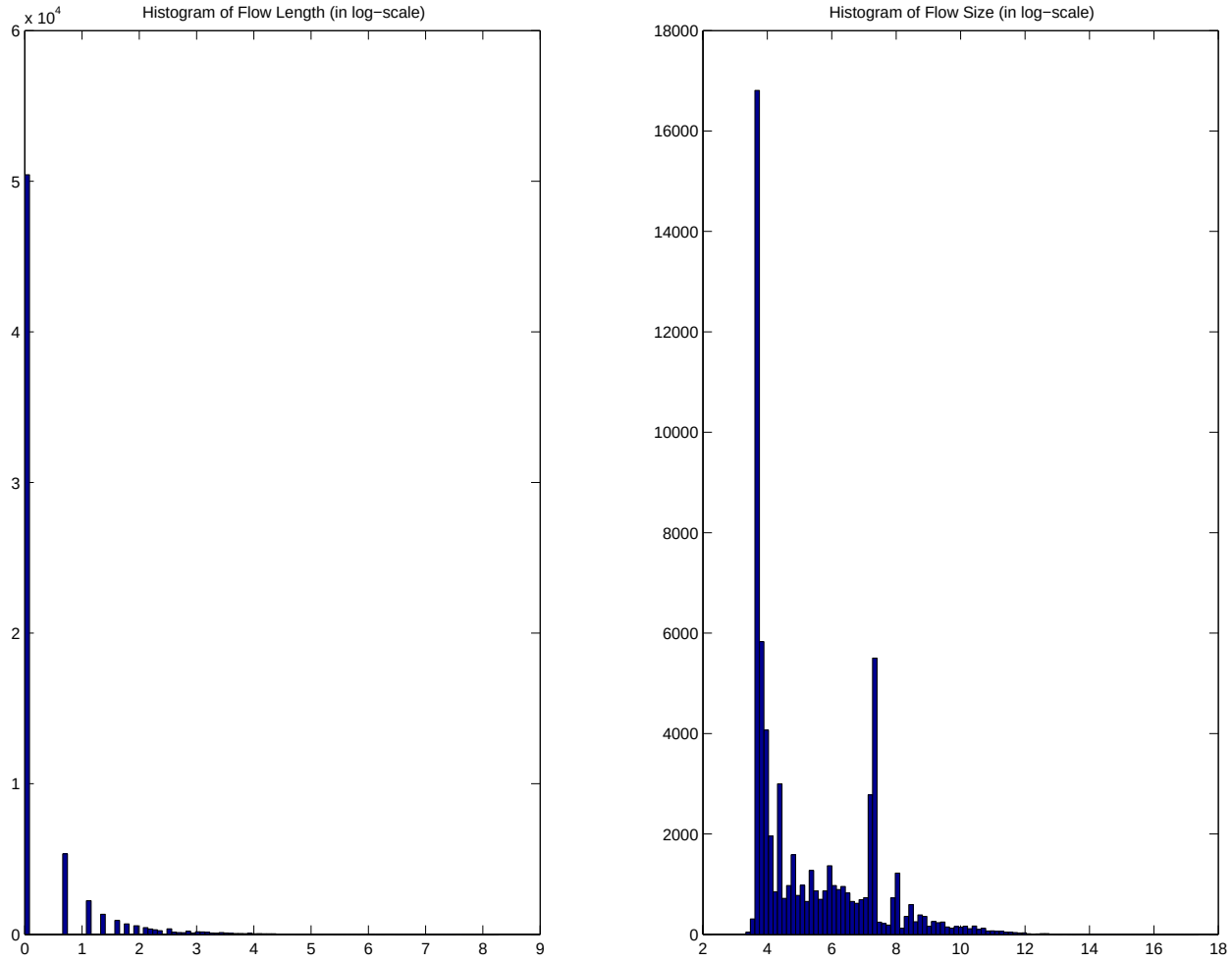


Figure 2.2: Histograms of Flow Lengths (left) and Flow Sizes (right) in log scale from NetFlow data

In order to overcome this deficiency, we propose a nonparametric model based on the general framework

$$H_B(B_0) = \sum_{i=1}^{N} \text{Prob}(\text{Flow size } B = B_0 | \text{Flow length } L = i)\phi_i. \tag{2.23}$$

We have already nonparametrically estimated flow length distribution ϕ, the next main issue to estimate flow size distribution H is to estimate conditional density

function $Q(B|L)$ according to equation(2.23). Shape of the distribution of flow size is roughly the same as the flow length distribution empirically as shown in the plot 2.2. We thereby postulate a linear relationship between flow length and flow size for any flow, given by

$$Q(B|L) \begin{cases} 1 & \text{if } B = \gamma_0 + \gamma_1 L \\ 0 & \text{o.w.} \end{cases}$$

i.e. $B(L) = \gamma_0 + \gamma_1 L$. The parameter γ_0 corresponds to an estimate in bytes of non-sampled very short flows, while γ_1 captures the correlation between flow length and size. They are estimated based on the following regression model:

$$b(j) = \gamma_0 + \gamma_1 j + \epsilon, \text{ for all } j;$$

i.e. the sampled size of the flow in bytes is a linear function of the number of observed packets from that flow with the same parameters γ_0 and γ_1. By obtaining the least squares estimates $\hat{\gamma}_0$ and $\hat{\gamma}_1$, we can subsequently estimate (predict) the flow sizes from

$$\hat{B}(L) = \hat{\gamma}_0 + \hat{\gamma}_1 L,$$

We proceed to estimate the distribution of the flow sizes H. From equation (2.23),

$$H(B) = \begin{cases} \phi_L & \text{if } B = \gamma_0 + \gamma_1 L \\ 0 & \text{o.w.} \end{cases}.$$

This yields an estimation

$$\hat{H}(B) = \begin{cases} \hat{\phi}_L & \text{if } B = \hat{\gamma}_0 + \hat{\gamma}_1 L \\ 0 & \text{o.w.} \end{cases},$$

for $L = 1, ..., N$. Therefore, estimates $\hat{B}(L)$ provides a collection of possible flow sizes with corresponding support S_B, whose mass function is determined by the corresponding flow length distribution ϕ.

To combine the shape of ϕ and the mass point of collection S_B, for implementation purpose, we propose an *Adaptive EM* algorithmic scheme. Since the sampling

scheme for the flow sizes is Bernoulli, the previously described EM algorithm is also suitable for the flow size distribution H. However, when we restricted attention to flow lengths only, the collection of possible flow lengths S_I (i.e. the support of the distribution ϕ) remained fixed throughout the iterations. In the implementation of the original EM algorithm, ϕ was estimated based on S_I, which is not the same as $\hat{L}(S_J)$; however, the support S_B of the flow size distribution depends only on $\hat{L}(S_J)$. Hence, the difference in the supports of the two distributions make it difficult to utilize flow length information in the estimation of the flow size distribution. However, by updating each L in S_I by $\hat{L}(S_J)$ as in equation (2.21), where $L^{(0)}(S_J)$ is initialized by $\{\llcorner j/p \lrcorner \bigcup \llcorner 1/2p \lrcorner\}$ as before, the two supports are made comparable. Consequently, by updating the support S_I within the iterations, the support S_B of the flow size distribution coincides with the updated one for the flow length S_I. It can then be seen that $\hat{\phi}$ is the maximum likelihood estimate of the flow byte distribution as well.

In Figure 2.3, the quantile-quantile plot of the original flow size vs the estimated one (sampling rate $p = 0.01$) for 1,000 Poisson flows of mean length 5,000 and uniform size distribution with support $[1200, 1500]$ using the adaptive EM algorithm is shown. It can be seen that with the exception of the tails of the distribution, the estimate is highly accurate.

2.6 Statistical Inference

We briefly discuss next some of the asymptotic properties of the derived maximum likelihood estimator. Theorem 11.3.3 in [27] and its remark gives that 'If Ω is compact, $\mathrm{E}_\theta \sup_\Omega W < \infty$, and $P_\omega \neq P_\theta$ for all $\omega \neq \theta$, then under P_θ, $\hat{\theta} \rightarrow^p \theta$'. In our context, first, notice that the parameter space $\Omega = \{\phi : \sum_{i=1}^N \phi = 1, \phi \geq 0\}$ of the posited multinomial model is closed and compact. In addition, the model satisfies the following: for all $\omega \neq \theta$ in the parameter space Ω, the distribution

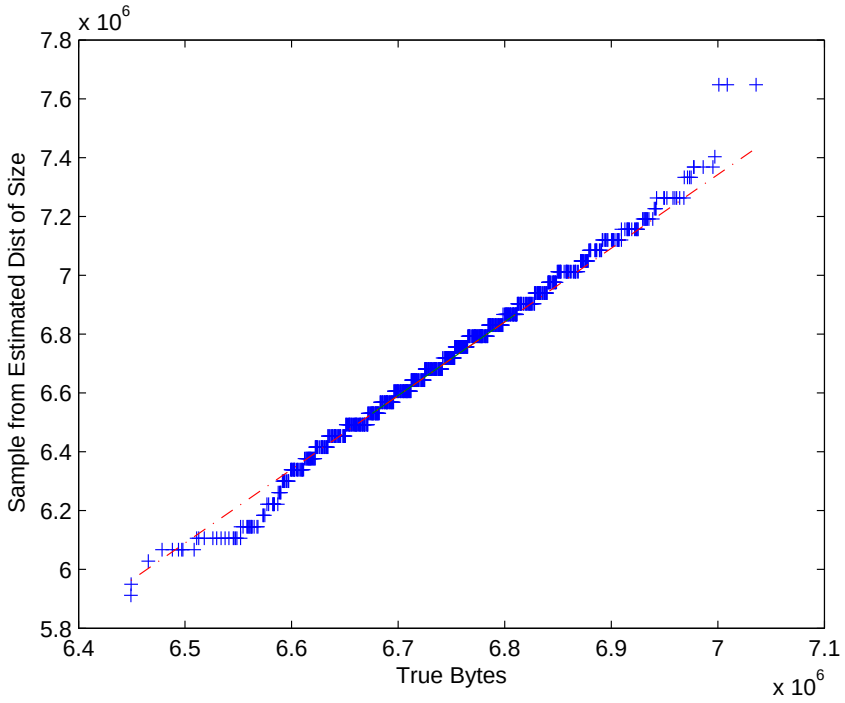


Figure 2.3: Quantile-quantile plot of the true vs the estimated flow size for 1,000 Poisson flows with uniformly distributed byte sizes

$P_\omega(G_0, ..., G_N) \neq P_\theta(G_0, ..., G_N)$. This is because $0 \leq \pi_{j|i} \leq 1$ and $\sum_{i=1}^{N} \phi_i = 1$ for all $\phi \in \Omega$, which implies

$$\{\sum_{i=1}^{N} \theta_i \pi_{j|i}\}_{j=0}^{N} \neq \{\sum_{i=1}^{N} \omega_i \pi_{j|i}\}_{j=0}^{N}$$

if $\omega \neq \theta$. This can easily be seen for $j = 0$, where $\pi_{j|i} > 0$ and then

$$\sum_{i=1}^{N} (\omega_i - \theta_i)\pi_{j|i} = 0 \text{ if and only if } \omega = \theta.$$

Further, from the previous discussion

$$P_\phi(G_0 = g_0, ..., G_N = g_N) = \frac{1}{\Delta} \sum_{M=1}^{\Delta} \begin{pmatrix} M \\ (M - \sum_{j=1}^{N} g_j), g_1, ...g_N \end{pmatrix} \prod_{j=0}^{N} (\sum_{i=1}^{N} \phi_i \pi_{j|i})^{g_j}.$$

Hence $P_\omega(G_0, ..., G_N) \neq P_\theta(G_0, ..., G_N)$ if $\omega \neq \theta$.

In order to establish consistency, the following condition also needs to be satisfied.

$$\mathrm{E}_\theta \mathrm{sup}_\Omega W < \infty, \tag{2.24}$$

where $W(\omega) = log[\frac{L(\omega)}{L(\theta)}]$. $\omega = \tilde{\phi} \in \Omega = \{\phi : \sum_{i=1}^{N} \phi = 1, \phi \geq 0\}$ is another set of values of these parameters, and $\theta = \phi$ in this context. We show next the above

condition holds for the model.

$$\mathrm{E}_\theta \mathrm{sup}_\Omega W = \mathrm{E}_\theta \mathrm{sup}_\Omega [l(\omega) - l(\theta)] = \mathrm{sup}_\Omega l(\omega) - \mathrm{E}_\theta l(\theta)$$

$$l(\omega) = log[\sum_{M=1}^{\Delta} \binom{M}{(M - \sum_{j=1}^{N} g_j), g_1, ...g_N} \prod_{j=0}^{N} (\sum_{i=1}^{N} \tilde{\phi}_i \pi_{j|i})^{g_j}] \leq log(\Delta). \tag{2.25}$$

$$\begin{aligned} \mathrm{E}_\theta l(\theta) \quad &= \textstyle\sum_x log[\sum_{M=1}^{\Delta} \binom{M}{(M - \sum_{j=1}^{N} g_j), 1, ...g_N} \prod_{j=0}^{N} (\sum_{i=1}^{N} \phi_i \pi_{j|i})^{g_j}] \\ &\times [\textstyle\sum_{M=1}^{\Delta} \binom{M}{(M - \sum_{j=1}^{N} g_j), g_1, ...g_N} \prod_{j=0}^{N} (\sum_{i=1}^{N} \phi_i \pi_{j|i})^{g_j}] \end{aligned} .$$

Since $\sum \phi_i = 1$, and $\pi_{j|i} > 0$ for $i \geq j$, $\sum_i \phi_i \pi_{j|i} > 0$, and for $M > \sum g_j$,

$$\binom{M}{(M - \sum_{j=1}^{N} g_j), 1, ...g_N} > 0.$$

Hence,

$$\sum_{M=1}^{\Delta} \binom{M}{(M - \sum_{j=1}^{N} g_j), 1, ...g_N} \prod_{j=0}^{N} (\sum_{i=1}^{N} \phi_i \pi_{j|i})^{g_j} > 0,$$

which implies

$$\mathrm{E}_\theta l(\theta) > -\infty,$$

and

$$\mathrm{E}_\theta \mathrm{sup}_\Omega W < \infty.$$

We can then conclude that the maximum likelihood estimator converges to the true parameter vector ϕ; i.e.

$$(\hat{\phi}) \rightarrow^p \phi \text{ as } r \rightarrow \infty, \tag{2.26}$$

where $r = \sum_{j=1}^{N} G_j$.

Given the consistency result, if Fisher Information Matrix for ϕ exists, and second derivative of the likelihood function with respect to ϕ exists and continuous in ϕ, we can obtain an asymptotic distribution for the estimated flow length distribution $\hat{\phi}$. However, our $\hat{\phi}$ is calculated from the related EM algorithm instead of directly from the likelihood function. Here we implemented Louis' [35] procedure for obtaining the observed information matrix when using the EM Algorithm. First, gradient vectors of the complete log-likelihood take the following form:

$$S = (\frac{F_1}{\phi_1}, ..., \frac{F_N}{\phi_N})'. \tag{2.27}$$

Then, the observed Information Matrix is given by

$$I_{obs} = \sum_{i=1}^{N} S(\hat{F}_i, \hat{\phi}) S(\hat{F}_i, \hat{\phi})', \tag{2.28}$$

where $\hat{F}_i$ is calculated from $E_{\hat{\phi}}[\sum_j f_{ij} | f_{ij} \in R]$ with all the notations stay the same as discussed before. f_{ij} is the frequency matrix stemming from the multinomial model as in equation (2.10). And R is defined as all the sets of complete data which yield the same result for the incomplete data, i.e. $R(j) = \{f_{ij} : \sum_i (f_{ij}) = e_j\}$, where e_j is a N dimensional indicator vector with all the elements, except the jth one g_j, being 0.

The resulting observed Information Matrix is symmetric with the lower triangular component given by

$$\begin{bmatrix} \frac{\sum_{j=1}^{N} \pi_{1|j}^2 g_j}{\hat{\phi}_1^2} & & \\ \frac{\sum_{j=1}^{N} \pi_{1|j}\pi_{2|j} g_j}{\hat{\phi}_1 \hat{\phi}_2} & \frac{\sum_{j=1}^{N} \pi_{2|j}^2 g_j}{\hat{\phi}_2^2} & \\ \cdots & \cdots & \\ \frac{\sum_{j=1}^{N} \pi_{1|j}\pi_{N|j} g_j}{\hat{\phi}_1 \hat{\phi}_N} & ..... & \frac{\sum_{j=1}^{N} \pi_{N|j}^2 g_j}{\hat{\phi}_N^2} \end{bmatrix} \tag{2.29}$$

Assuming the regularity conditions ([35], [27],[58]) on existence of fisher information matrix and second derivative existence of the original likelihood with respect to ϕ hold, the above calculated observed information matrix can be inverted to find the covariance matrix of $\hat{\phi}$.

Recall that the posterior mean estimator for the an original flow length given a sampled one of length k is defined to be

$$\mathcal{L}(k) \equiv \hat{L}(k) \equiv \mathrm{E}(L(k)) = \frac{\sum_L L\pi_{k|L}\hat{\phi}_L}{\sum_L \pi_{k|L}\hat{\phi}_L},$$

and is continuous in ϕ. By an application of the Delta method [58], we then get that

$$\hat{\mathrm{Var}}(\hat{L(k)}) = \nabla\mathcal{L}_{\hat{\phi}} I_{obs}^{-1} \nabla\mathcal{L}'_{\hat{\phi}}. \tag{2.30}$$

The continuity of the above defined functional in ϕ gives that $\sqrt{r}(\hat{L}(k) - L(k))$ is approximately Normal with variance $\mathrm{Var}(\hat{L}(k))$, where $\mathrm{Var}(\hat{L}(k))$ is estimated by $\hat{\mathrm{Var}}(\hat{L}(k))$. The significance of this result is, that simultaneous confidence intervals for the original flow lengths given sampled ones, can be constructed, thus providing a measure of uncertainty about the obtained estimates.

Remark: For this study, our contribution is mainly on the well formulated derivation of likelihood function. The derived EM Algorithm is for practical implementation, which shows convergency numerically. We have not theoretically researched on the convergence of this EM algorithm. The above discussion on the statistical inference is based on the fact this convergency holds. Otherwise, observed information matrix and the succeeding confidence interval for the original flow lengths would not be valid; however, the consistency of MLE holds regardless the convergency of the EM algorithm.

2.7 Estimating Mixture Distributions from Sampled Data

Experimental evidence suggests that the proposed maximum likelihood estimator does not perform well with several real network traces, since both the packet length and consequently the byte size distribution are mixtures of two components; the first, representing short flows, and the second representing considerably longer flows. An example of such a scenario with data obtained from the link that connects the campus of the University of North Carolina at Chapel Hill to the Internet is shown

in Figure 2.4. The estimation using the proposed Adaptive EM is shown in Figure 2.23, where we see the gaps from the empirical distribution; both the support and the overall distribution are shifted to the right somehow. In particular, the estimate of the number of active flows proves highly problematic, due to the severe nature of the biased length sampling issue in such a setting; namely, the first component of the distribution is heavily under-sampled. We propose next two procedures that deal with the problem of estimating mixture distributions. The first approach is based on the original Bernoulli sampling scheme, but employs a two-stage EM algorithm suitable for mixture distributions. The second approach looks at the problem from a different point of view and relies on an alternative sampling scheme.

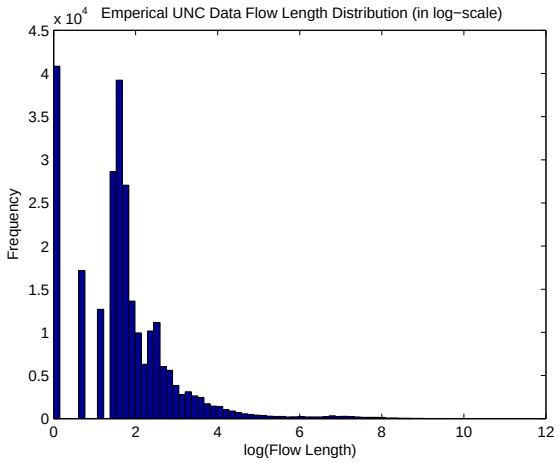


Figure 2.4: Empirical UNC flow length distribution

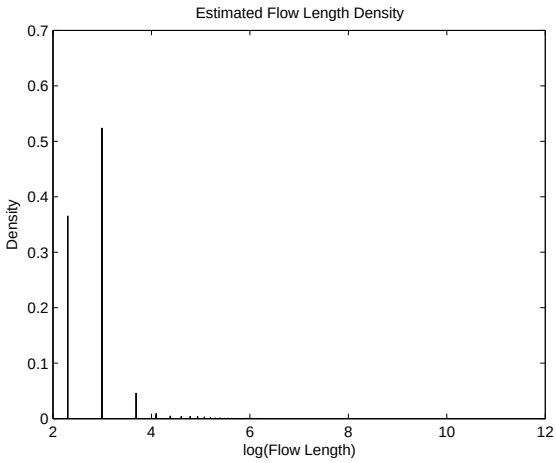


Figure 2.5: Estimated UNC flow length distribution

2.7.1 Two Stage EM Algorithm

It is assumed that the original flow length distribution $\phi \equiv F$ (and consequently the flow size one H) is a mixture of two components; i.e.

$$F = \alpha F_1 + (1 - \alpha) F_2,$$

with mixing coefficient $\alpha \in (0, 1)$. In the remainder of the section, we will focus on the case where the first component is a point mass at 1; i.e. $F_1 \equiv \delta_1$, since with a

small sampling rate p at most one packet would be sampled from short flows. This assumption simplifies somewhat the derivations, but nevertheless the proposed two-stage EM algorithm works for the general case provided that the two components are adequately separated [44].

The main idea behind the proposed algorithm is as follows: in the first stage, the adaptive EM algorithm previously described is used to estimate the relative frequencies ϕ, the support of the distribution S_I and the number of active flows. In the second stage, another EM algorithm is employed that based on the current estimates of these parameters, estimates the mixing coefficient α. Therefore, the two-stage algorithm splits the parameters of interest into two subsets (blocks) and in each iteration alternates between the blocks by fixing the parameters of the other block in their current values. The theory of block relaxation algorithms [10] guarantees convergence to a local maximum. However, there are some subtle issues that need to be carefully considered. Notice that in the first stage, the sampled information on the point mass component has been included, since it is not possible to directly separate whether a sampled flow of length one comes from the first or the second component. By conditioning, we can separate $g_j(1) = g_j^1(1) + g_j^2(1)$ with proportion of $P(\text{Flow length } 1|j = 1)$ and $P(\text{Flow length } i \in S_I^2|j = 1)$ respectively, where S_I^2 denotes the support of the second component $S_I - \{1\}$. Then, $g_j^2(1)$ is used in the first stage EM algorithm, together with the remaining g_j's. On the other hand, all the g_j's (including g_j^1) are used in the second stage to estimate α. We discuss next the second stage of the EM algorithm that estimates the mixing coefficient α.

EM Algorithm for estimating the mixing coefficient α

Now we fix M, ϕ and S_I at the estimations achieved from the Adaptive EM as in section 2.5, and target on estimating mixing coefficient α. Given the hierarchical mechanism, we still have all the flows have the same probability of sample length j, denoted by π_j. A similar derivation as (2.7) follows. The so-called *profile* likelihood

function for α is then given by the probability function of $\{G_0, ..., G_N\}$ as follows.

$$\begin{aligned} L(\alpha) &= \sum_{M=1}^{\Delta} \binom{M}{g_0, g_1, ..., g_N} \prod_{j=0}^{N} (\pi_j)^{g_j} \\ &= \sum_{M=1}^{\Delta} \binom{M}{g_0, g_1, ..., g_N} \prod_{j=0}^{N} [\alpha \pi_{j|1} + (1-\alpha)\pi_{j|S_I^2}]^{g_j}. \end{aligned}$$

It is expressed as a mixture of two conditional densities for the two mixture components, where $\pi_{j|S_I^2} = \sum_{i \in S_I^2} \pi_{j|i}\phi_i$ is the estimated probability that the length of a sampled flow from the second component will be j packets; $\pi_{j|1}$ is the estimated probability that the length of a sampled flow from the first component will be j packets. Note that $\pi_{j|S_I^2}$ stays identical for all the flows assigned to the second component. In fact, one could interpret this model as an extension of the hierarchical mechanism, in the sense that adding an additional layer assigning flows to either first component or the second one based on probability given by mixing coefficient; then for the ones in the second component, flow lengths are again independently determined by a discrete probability function $\{\phi_i\}_{i \in S_I}$ which is estimated in the Adaptive EM step.

Correspondingly, the likelihood function for α based on the complete data $\{g_j, y_{jk}\}_{j=0,k=1}^{N,2}$ treats y_{jk}, an indicator variable that identifies the mixture component ($k = 1, 2$) that a sampled flow of length j belongs to, as missing data and estimates them as conditional expectations. Number of unobserved flows g_0 is estimated in the Adaptive EM step as $\hat{g}_0 = \hat{M} - r$. The likelihood function for the complete data can be written as

$$L(\alpha) = \prod_{j \geq 0} (\alpha^{y_{j1}} \pi_{j|1}^{y_{j1}} (1-\alpha)^{y_{j2}} \pi_{j|S_I^2}^{y_{j2}})^{g_j}.$$

Notice that $\mathrm{E}(y_{j1}^{(t)} | g_j, \alpha^{(t)}) = \frac{\alpha^{(t)} \pi_{j|1}^{(t)}}{\alpha^{(t)} \pi_{j|1}^{(t)} + (1-\alpha^{(t)}) \pi_{j|S_I^2}^{(t)}}$. The steps of the EM algorithm are given next:

Initialize $\alpha^{(0)}$ by the empirical estimate g_1/r.

E-step

$$\begin{aligned} \mathrm{E}_{\alpha^{(t)}}(\log L(\alpha)|g_j) &= (\textstyle\sum_{j\geq 0} \mathrm{E}_{\alpha^{(t)}}(y_{j1}|g_j)g_j)\log\alpha + \\ & \quad (\textstyle\sum_{j\geq 0} \mathrm{E}_{\alpha^{(t)}}(y_{j2}|g_j)g_j)\log(1-\alpha) \end{aligned}$$

M-step

By taking the derivative of $\mathrm{E}_{\alpha^{(t)}}(\log L(\alpha)|g_j)$ with respect to α and setting it to 0 we obtain

$$\hat{\alpha}^{(t+1)} = \frac{\sum_{j\geq 0}(\mathrm{E}_{\alpha^{(t)}}(y_{j1}|g_j)g_j)}{\hat{M}}$$

Iterate E-step and M-step until a pre-specified convergence criterion is satisfied.

2.7.2 Two-Stage Sampling Scheme

We discuss next an alternative procedure that relies on a different sampling scheme. It still follows the hierarchical mechanism, where the number of flows M selected based on the discrete Uniform distribution $(1, \Delta)$; and flow lengths $\{N_m\}_{m=1}^{M}$ are then determined independently according to the discrete probability function $\{\phi\}$; while now the third layer of the hierarchical mechanism is applied in a two-stage sampling way after all the flows and packets are collected and to be sampled. In the first stage flows are sampled uniformly with probability p_f irrespective of their lengths; while in the second stage, packets are sampled uniformly with probability p_p from the flows selected from the first stage. The resulting outcome observation is $\{G_j\},\ j = 00, 1, ..., N$, where G_{00} is the number of flows selected at the first stage but no packets selected on the second stage and thus observed. Such a sampling procedure is feasible if different flows go through different queues on a router. It can be seen that this two stage sampling scheme overcomes the difficulty posed by length biased sampling, since each flow has an equal probability of being selected. Notice that another difference compared to the original Bernoulli sampling scheme is that the two tasks of estimating the total number of active flow M, and the flow length/bytes distributions are well separated. A trivial unbiased estimator for the number of active flows is given by $\hat{M} = \frac{r}{p_f}$ [7]. Also note that under the hierarchical

mechanism $\{G_0, G_1, ..., G_N\}|M \sim$ Multinomial$(M, \check{\pi}_j)$ still holds, where $\check{\pi}_j$ now is given by $\sum_{i=1}^{N} \phi_i p_f \check{\pi}_{j|i}$ for $j > 1$, and $\check{\pi}_{j|i}$ updates p in $\pi_{j|i}$ by p_p; the main difference is that G_0 now is comprised of two components G_{0ns} representing the number of unobserved flows, and G_{00} representing the number of flows selected at the first stage but no packets selected on the second stage and thus observed. Therefore, $\check{\pi}_0 = \check{\pi}_{0ns} + \check{\pi}_{00} = \sum_{i=1}^{N} \phi_i(1 - p_f) + \sum_{i=1}^{N} \phi_i p_f \check{\pi}_{0|i}$.

The likelihood function for ϕ given observed data under this sampling scheme is given by

$$
\begin{aligned}
L(\phi) &= \sum_{M=1}^{\Delta} \binom{M}{g_0, g_1, \ldots g_N} \prod_{j=0}^{N} (\check{\pi}_j)^{g_j} \\
&= \sum_{M=1}^{\Delta} \binom{M}{g_{0ns}, g_{00}, g_1, \cdots g_N} \prod_{j=0ns,00,1,\ldots,N} (\check{\pi}_j)^{g_j}.
\end{aligned} \tag{2.31}
$$

Once again, direct maximization of the likelihood function is challenging and hence we resort to using the EM algorithm, whose steps are described next:

(1) *Initialize* $\phi_i^{(0)}$ by the corresponding observed frequency of $j = \llcorner ip \lrcorner$, i.e,

$$
\phi_i^{(0)} = \frac{g_j}{\sum_{k=00}^{N} g_k}, \tag{2.32}
$$

(2) *E-step:* We are given the complete set of data $(\{F_{ij}\}_{i=1,\ldots,N;j=0,\ldots,N}, G_{0ns})$, where F_{ij} are the observed flows with i original packets and j of them sampled. From the hierarchical mechanism, $(\{F_{ij}\}_{i=1,\ldots,N;j=0,\ldots,N}, G_{0ns}|M)$ follows a multinomial distribution with parameters $M = \sum_{i=1,j=0}^{N,N} F_{ij} + G_{0ns}$ and $(\{\check{\pi}_{ij}\}, \check{\pi}_{0ns})$, where $\check{\pi}_{ij} = \phi_i p_f \check{\pi}_{j|i}$, and $\check{\pi}_{0ns} = \sum_{i=1}^{N} \phi_i(1 - p_f)$. The likelihood based on the complete data is then given by

$$
L_c(\phi) = \prod_{i=1,j=0}^{N,N} (\check{\pi}_{ij})^{F_{ij}} \times (\check{\pi}_{0ns})^{G_{0ns}}, \tag{2.33}
$$

The expectation $Q(\phi, \phi^{(k)})$ of log-likelihood conditional on the known frequencies

g_j is given by:

$$
\begin{aligned}
Q(\phi, \phi^{(k)}) &= \sum_{i \geq j \geq 0} E_{\phi^{(k)}}(F_{ij}|G_j) \log(\phi_i \breve{\pi}_{j|i} p_f) \\
&\quad + E_{\phi^{(k)}}(G_{0ns}|G_j) \log[\textstyle\sum_{i=1}^{N} \phi_i (1 - p_f)].
\end{aligned} \tag{2.34}
$$

Again, notice that

$$E_{\phi^{(k)}}(F_{ij}|\{G_{00}, ..., G_N\}) = G_j \breve{\pi}_{i|j}^{(k)}.$$

Meanwhile, $E_{\phi^{(k)}}(G_{0ns}|\{G_0 0, ..., G_N\}) = M^{(k)} - \sum_{j=00}^{N} G_j$.

(3) *M-step:* $\phi^{(k+1)} = \arg\max Q(\phi, \phi^{(k)})$, such that

$$\sum_{i \in S_I} \phi_i = 1, \text{ and } \phi_i(j) \geq 0 \text{ for } i \in S_I. \tag{2.35}$$

Iterate steps (2)-(3) until a pre-specified convergence criterion is satisfied.

In order to obtain estimates of the flow sizes (in bytes) we can apply the various regression models discussed in Section 2.5.

2.8 Experimental Evaluation

In this section, we provide empirical evidence of the performance of the derived estimators for a variety of simulated and real network traffic traces.

We start by examining both the original EM Algorithm proposed in Section 2.3, as well as the Adaptive EM algorithm developed for mixture distributions and the one based on the Two-stage sampling scheme, using synthetic and real data sets.

2.8.1 Simple Flow Length Distribution

The first set of experiments is to evaluate the performance of estimation on flow lengths distribution ϕ and the number of active flows M. To be specific, for different setting of M, we first generate independent M variables $\tilde{N_m}$ from the following distributions: (i) uniform with domain $[0, 10000]$, (ii) Poisson with mean 5,000 and (iii) Pareto with shape parameter 100/99, scale parameter 50. And we then take flow lengths $N_m = \llcorner \tilde{N_m} \lrcorner$ as the close discrete integers below the generated $\tilde{N_m}$. The parameters for these three distributions were set so as to match their expected values.

Theoretically, for the shape parameter of Pareto distribution $k > 1$, the smaller the value of k, the skewer the Pareto distribution would be. And hence, our setting of $k = 100/99$ represents a very heavy right tail distribution.

We first look at the estimates ($\hat{M}$) of the true number of flows, plotted against the true ones (M), where the flow length distribution is Poisson with mean 5,000 for varying values of M with a sampling rate of $p = .01$ (see Figure 2.6). It can be seen that the estimated number of flows is extremely accurate, a result also obtained for the uniform distribution. However, this estimate becomes problematic for very heavy-tailed distribution, as shown with real NetFlow traces (Section 2.2).

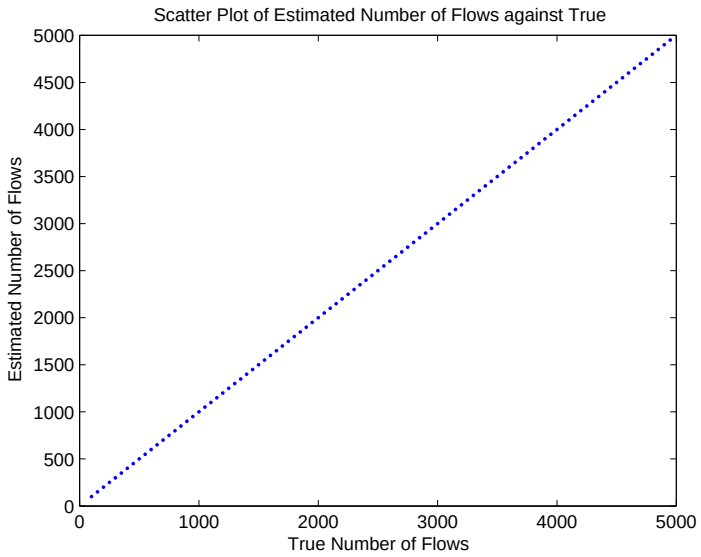
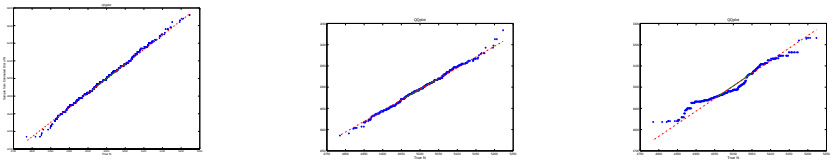


Figure 2.6: Scatter plot of true vs estimated number of active flows, where flow length distribution is Poisson with mean 5,000 for varying values of M with a sampling rate of $p = .01$

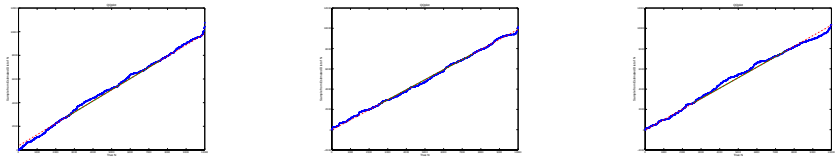
We examine next quantile-quantile plots, in Figures 2.7-2.9, of the true flow lengths, vs the non-parametrically estimated flow length distribution for $M = 1,000$ flows, whose lengths follow a mean 5,000 Poisson, Uniform and Pareto distributions respectively with 0.05 sampling rate. For the two-stage sampling scheme, $p = 0.05$ is implemented by choosing $p_f = 0.5$, $p_p = 0.1$ to make it comparable with other methods. However, the optimal choice of p_f and p_p is still not clear so far, and requires further study. It can be seen the high degree of agreement between the true



(a) First EM (b) Adaptive EM (c) Two-stage Sampling

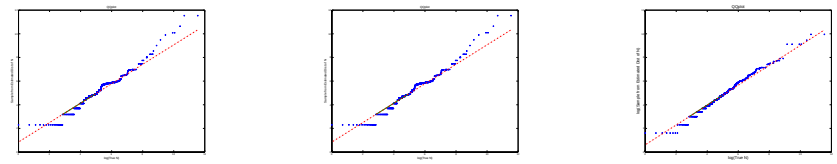
Figure 2.7: Quantile-quantile plot of the true vs the estimated flow length distribution for 1,000 Poisson flows with .05 sampling rate ($p_f = 0.5,\ p_p = 0.1$ in Two-Stage Sampling)

and estimated distributions for all these scenarios. Moreover, the two-stage scheme provides a better estimation than the algorithms based on Bernoulli sampling in the more realistic Pareto scenario. Similar conclusions are reached by examining the histograms (bar plots) (Figures 2.10-2.12) of the true and estimated distributions. We further provide plots (Figures 2.13-2.15) of their cumulative density function (CDF) that reveal some interesting features. Specifically, solid lines with a star (*) mark show the 'true' distribution, while the dashed lines correspond to estimated CDFs obtained by sampling a number of times from the true data. It is interesting to see that the adaptive EM and the two-stage sampling scheme improve substantially over the original EM algorithm in the Poisson case but less obvious in the Uniform or Pareto case. Further, the upper quantiles of the Pareto and uniform distributions are estimated extremely accurately, whereas the lower quantiles prove problematic for the Pareto distribution. This is mainly due to the heavy tailed nature of the distribution that leads to undersampling of short flows. A further remark on the Poisson distribution is that its original variance is quite small with most of its mass centered around the mean. All the proposed methods estimate the mean very accurately, but not particularly well the upper and lower quantiles. The accurate estimation of the right tail is the most important feature in a networking context and hence the adaptive EM and the two-stage scheme prove particularly useful. However, in other applications where the entire body of the distribution is of interest, one has to use either some other sampling mechanism or increase the sampling rate (e.g. $p = 0.10$).



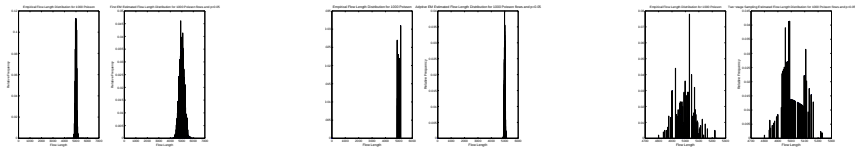
(a) First EM (b) Adaptive EM (c) Two-stage Sampling

Figure 2.8: Quantile-quantile plot of the true vs the estimated flow length distribution for 1,000 Uniform flows with .05 sampling rate ($p_f = 0.5,\ p_p = 0.1$ in Two-Stage Sampling)



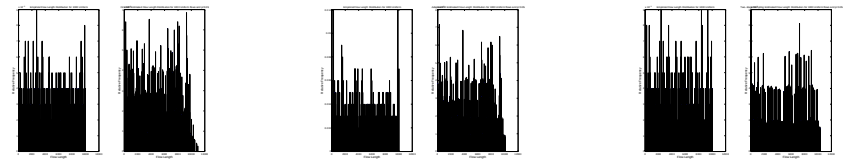
(a) First EM (b) Adaptive EM (c) Two-stage Sampling

Figure 2.9: Quantile-quantile plot of the true vs the estimated flow length distribution for 1,000 Pareto flows with .05 sampling rate ($p_f = 0.5,\ p_p = 0.1$ in Two-Stage Sampling)



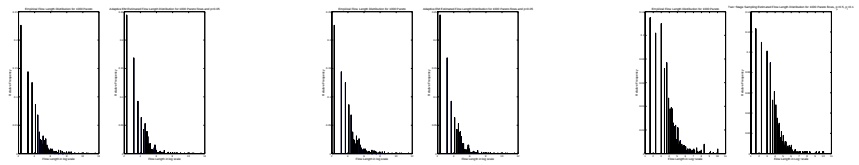
(a) First EM (b) Adaptive EM (c) Two-stage Sampling

Figure 2.10: Bar plot of the estimated flow length distribution for 1,000 Poisson flows with .05 sampling rate ($p_f = 0.5,\ p_p = 0.1$ in Two-Stage Sampling)



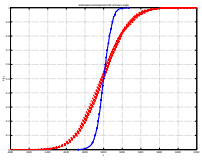
(a) First EM (b) Adaptive EM (c) Two-stage Sampling

Figure 2.11: Bar plot of the estimated flow length distribution for 1,000 Uniform flows with .05 sampling rate ($p_f = 0.5,\ p_p = 0.1$ in Two-Stage Sampling)

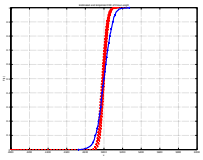


(a) First EM (b) Adaptive EM (c) Two-stage Sampling

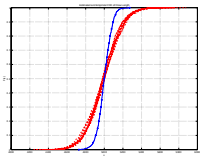
Figure 2.12: Bar plot of the estimated flow length distribution for 1,000 Pareto flows with .05 sampling rate ($p_f = 0.5,\ p_p = 0.1$ in Two-Stage Sampling)



(a) First EM

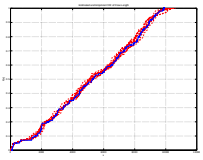


(b) Adaptive EM

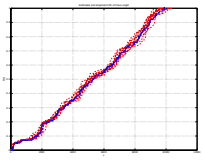


(c) Two-stage Sampling

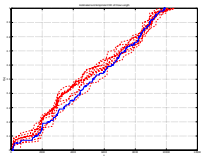
Figure 2.13: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for 1,000 Poisson flows with .05 sampling rate ($p_f = 0.5$, $p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution



(a) First EM

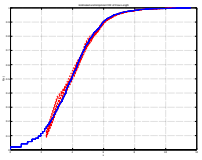


(b) Adaptive EM

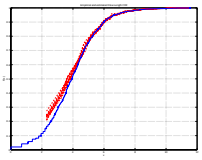


(c) Two-stage Sampling

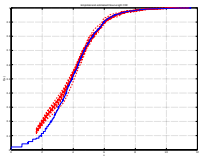
Figure 2.14: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for 100 Uniform flows with .05 sampling rate ($p_f = 0.5$, $p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution



(a) First EM



(b) Adaptive EM



(c) Two-stage Sampling

Figure 2.15: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for 1,000 Pareto flows with .05 sampling rate ($p_f = 0.5$, $p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution

In order to obtain a better perspective about the effect of the sampling rate, the estimated flow length distribution for 1,000 Poisson flows of mean length 5,000 are given in Figures 2.16 and 2.17, corresponding to sampling rates of $p = .01$ and .05. It can be seen that with a higher sampling rate the number of possible sampled flow lengths (S_J in our notation) increases significantly, thus allowing a more accurate estimate of the underlying distribution.

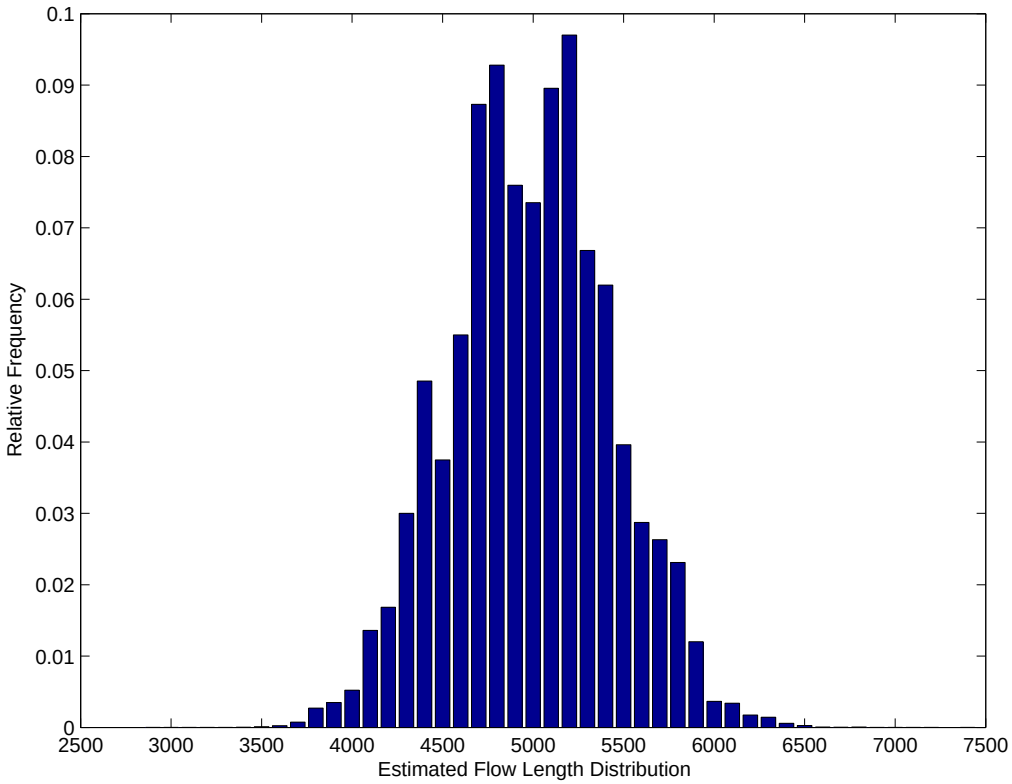


Figure 2.16: Estimated flow length distribution of 1,000 Poisson flows with .01 sampling rate

Moreover, the confidence intervals obtained for the original flow length estimates $\hat{L(k)}$, when a sampled flow of length k was observed for 100 Poisson flows of mean length 5,000 are shown in Figure 2.18. It can be seen that the confidence intervals are fairly tight except at the two ends, which reflects the higher uncertainty of our estimates for the corresponding ϕ_i parameters.

In Table 2.1 we give χ^2-distances between the true and estimated flow length distributions, together with mean squared errors (MSE) of the $\hat{L}(k)$ estimates for a variety of models, where $\chi^2 = \sum_{i \in S_I} \frac{(\hat{\phi}_i - \phi_i)^2}{\phi_i}$ as a measurement of how close the two

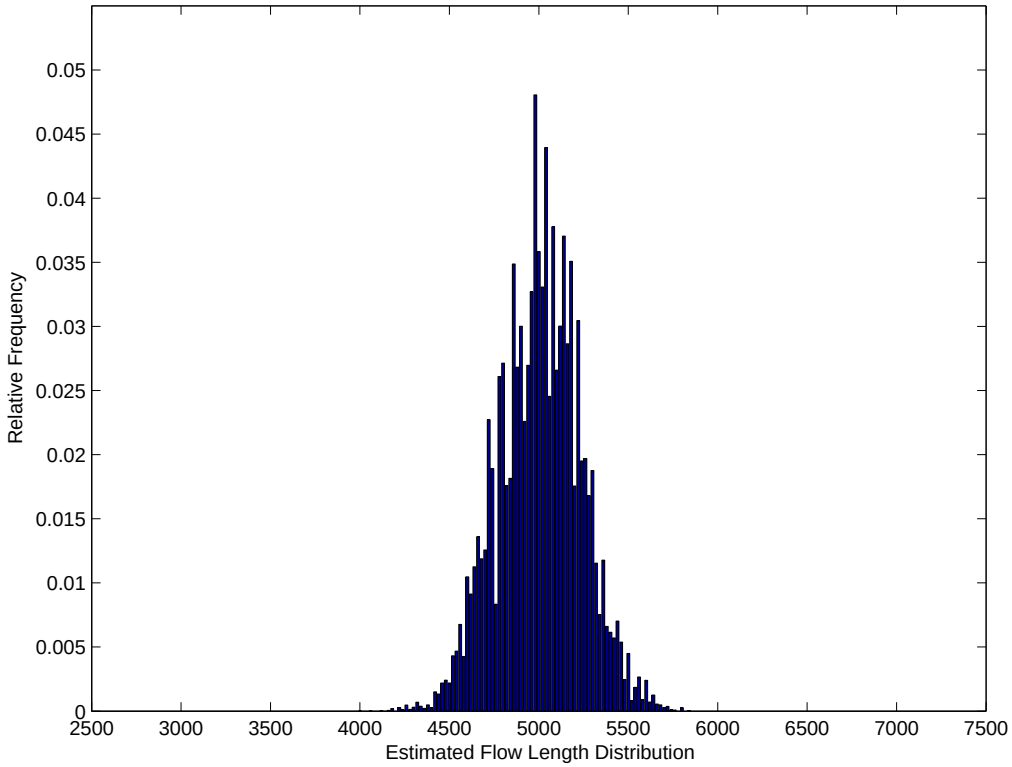


Figure 2.17: Estimated flow length distribution of 1,000 Poisson flows with .05 sampling rate

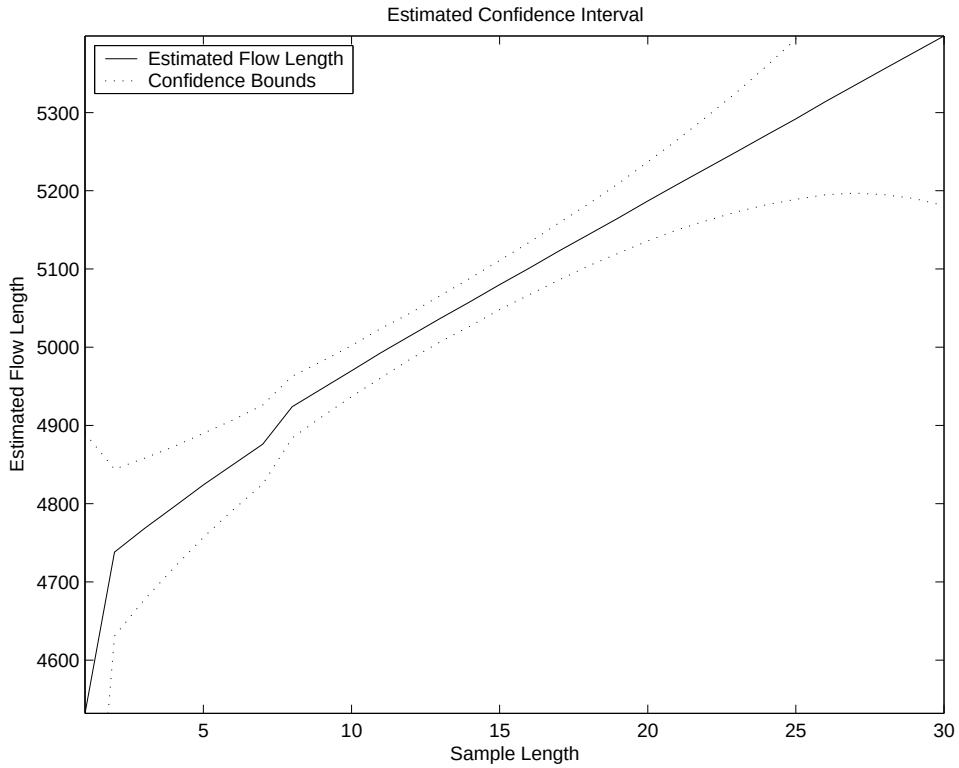


Figure 2.18: Confidence intervals for 100 Poisson Flows with sampling rate 0.01

Table 2.1: Statistics from Simulated Data(I)

χ^2 Statistics				
		Uniform	Poisson	Pareto
n=100	p=0.01	225.6667	437.2905	1.24E+03
	p=0.05	121.8333	285.0726	88.1264
n=1000	p=0.01	4.7457e+04	1.2491e+05	2.92E+06
	p=0.05	4.9931e+03	3.8215e+04	4.45E+05
n=2500	p=0.01	3.2981e+05	2.3523e+05	6.23E+07
	p=0.05	3.5027e+04	2.2789e+05	6.61E+06
MSE				
		Uniform	Poisson	Pareto
n=100	p=0.01	3.9394e+005	2.9761e+004	3.73E+04
	p=0.05	7.9572e+004	1.0548e+004	3.48E+03
n=1000	p=0.01	4.0625e+05	3.7223e+04	8.30E+04
	p=0.05	8.8508e+04	1.3122e+04	1.88E+04
n=2500	p=0.01	4.5259e+05	1.7896e+04	2.45E+05
	p=0.05	9.0463e+04	1.6553e+04	1.51E+04

distributions are; and $\mathrm{MSE}(\hat{L}) = \sum_{j \in S_J} (\hat{L}(k) - L(k))^2 + \mathrm{Var}(\hat{L}(k))$ as a combined measure of variation and biasedness of estimated $L(k)$. Therefore, the smaller the χ^2 statistics and MSE the better the estimations are. We only list χ^2 statistics in the table instead of p-value, because the corresponding p-values are very small considering the small sampling rate, the difficulty of this nonparametric estimation and the roughness of the nonparametric support. The corresponding degrees of freedom are the number of nonparametric point mass in the support S_I, which increases as sampling rate increases. From this table, it can be seen that as the sampling rate increases both performance measures decrease, which presents a better estimation. Further, MSE shows rather comparable performance for the relatively large amount of data with 2500 flows among all the three simulated flow lengths distributions, this makes the estimation on heavy-tailed Pareto distribution remarkable.

Table 2.2 shows the comparison between the true 25, 50, 75, 90 and 95-percentiles and the estimated one with the proposed algorithms. It can be seen that the estimated ones match the true ones remarkably well.

Table 2.2: Statistics from Simulated Data(II)-Percentiles

Percentiles	25	50	75	90	95
Poisson Distribution					
True	4953	5000	5052	5094.5	5120.5
First EM	4840	5000	5160	5320	5380
Adaptive EM	4980	5005	5033	5056	5073
Two-Stage Sampling	4870	4990	5110	5230	5290
Uniform Distribution					
True	2576.5	5167	7588.5	9056.5	9423.5
First EM	2640	5180	7600	9020	9420
Adaptive EM	2599	5148	7597	9169	9332
Two-Stage Sampling	2700	5490	7750	8980	9320
Pareto Distribution					
True	2489.5	5096.5	7431	8996	9494
First EM	2500	5140	7420	9020	9420
Adaptive EM	2520	5100	7440	8960	9400
Two-Stage Sampling	2540	4820	7540	8920	9360

Table 2.3: Statistics from NS2 Simulator

	χ-Statistic		MSE	
	UDP/CBR	TCP	UDP/CBR	TCP
p=0.01	122.5285	765.3391	5.2948e+003	7.4811e+05
p=0.05	17.0740	242.1123	1.5526e+003	7.8458e+04

2.8.2 Application to NS and Real Data

Another set of simulated data were obtained from the ns2 network simulation package [51]. Two networking scenarios were considered. In the first scenario lasting 2 minutes, 100 constant bit rate sources generated traffic, whose duration followed a Pareto distribution with shape parameter 1.5 and scale parameter 100/3. Packet sizes were identical within the same flow (source), but different across flows, following a normal distribution with mean 800 and standard deviation 100. In the second scenario 100 ftp transmissions were generated on a link, whose duration follows the previously defined Pareto distribution. Table 2.3 evaluates the original EM Algorithm by statistics, sharing the similar interpretation as in the simulated data above. Whereas Figure 2.19 of QQplot for the generated UDP data reveals more inherent problems of the estimation in the case of heavy-tailed distribution.

Two real data sets were also considered to examine the first EM Algorithm. The

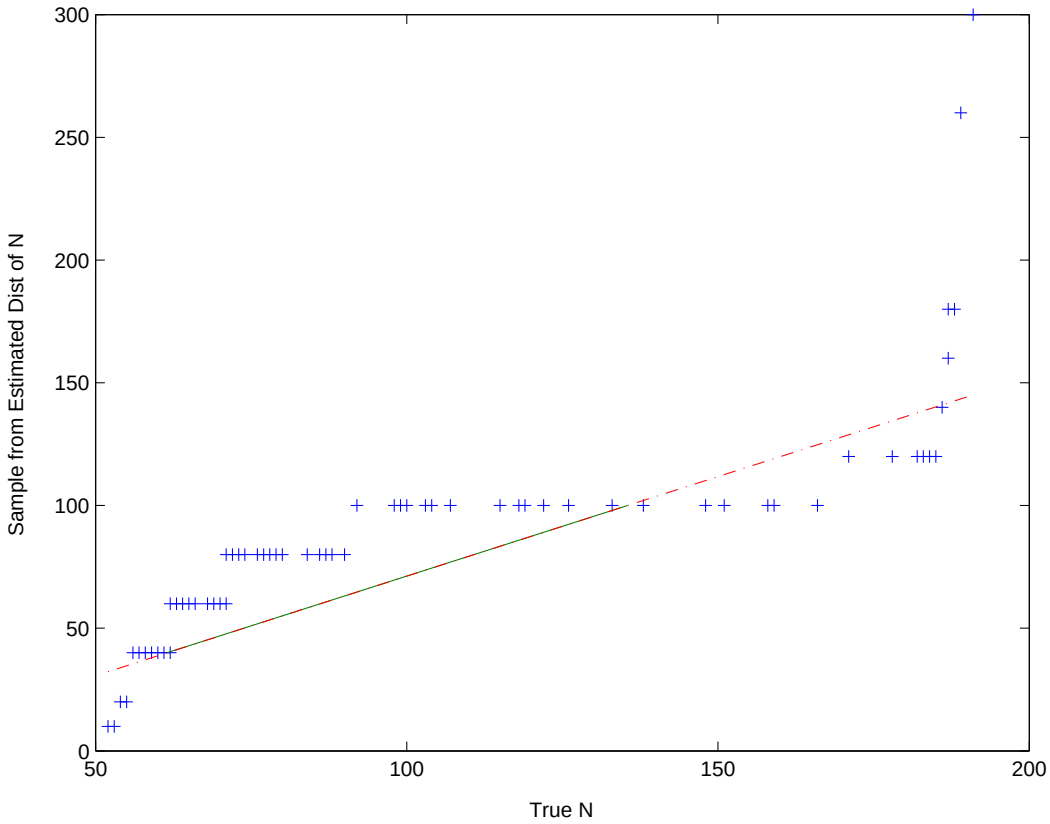


Figure 2.19: Quantile-quantile plot of the true vs the estimated flow length distribution for simulated UDP flows, sampling rate 0.05

first one contains 34,514 network flows collected over a 2 hour period at the router of a small local area network. The average flow length consists of 29 packets, but the variance is 4.5×10^5. The second data set contains 256,835 flows connecting at the gateway link of the UNC campus network. The average flow length consists of 39 packets with a variance of 3.22×10^5. The distribution of the true flow lengths (in log-scale) for these data sets are shown in Figures 2.20 and 2.21, respectively. It can easily be seen that both data sets have heavy right-tailed flow length distributions.

Figures 2.22 and 2.23 show the estimated flow length distribution for LAN and UNC flow data, respectively. We applied rather low sampling rate, 1% for LAN flow data, and 0.1% for UNC flow data because of the large volume of data. Compare these two with the original flow length distribution in Figures 2.20 and 2.21, the estimated nonparametric distributions illustrates some discrepancy from the original distributions, although they both capture some patterns of the distributions. We

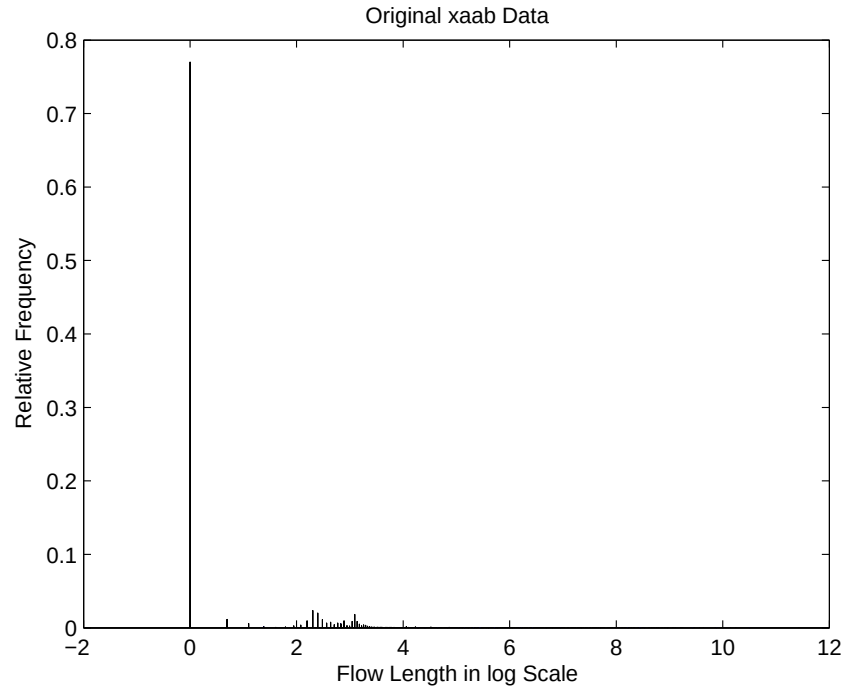


Figure 2.20: Empirical distribution (in log-scale) of the LAN flow data

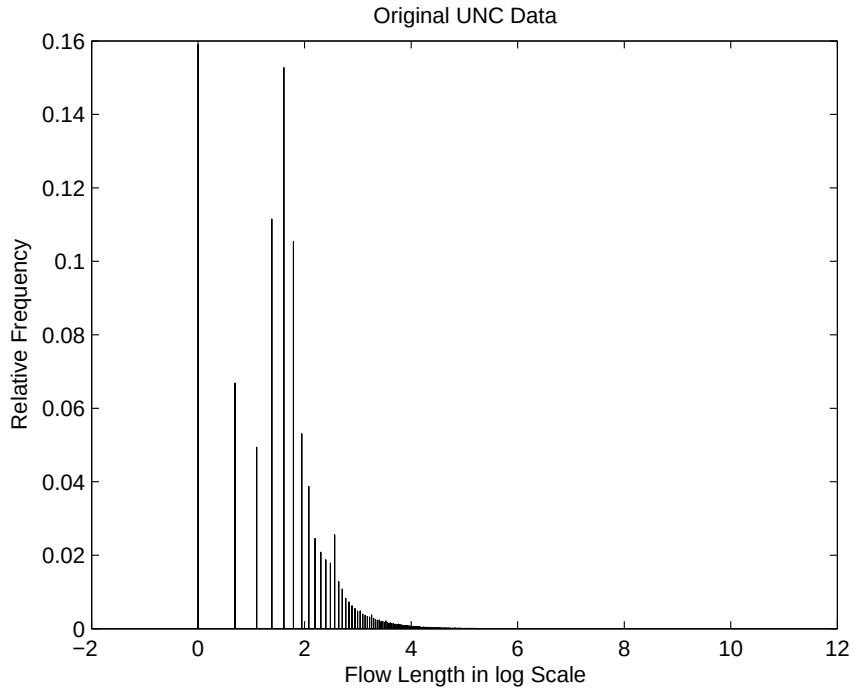


Figure 2.21: Empirical distribution (in log-scale) of the UNC flow data

have to admit that this estimation is not as good as in the regular distributions in section 2.8.1. Figure 2.24 is the multi-cdf curves we made based on LAN flow data. It reveals the similar problem.

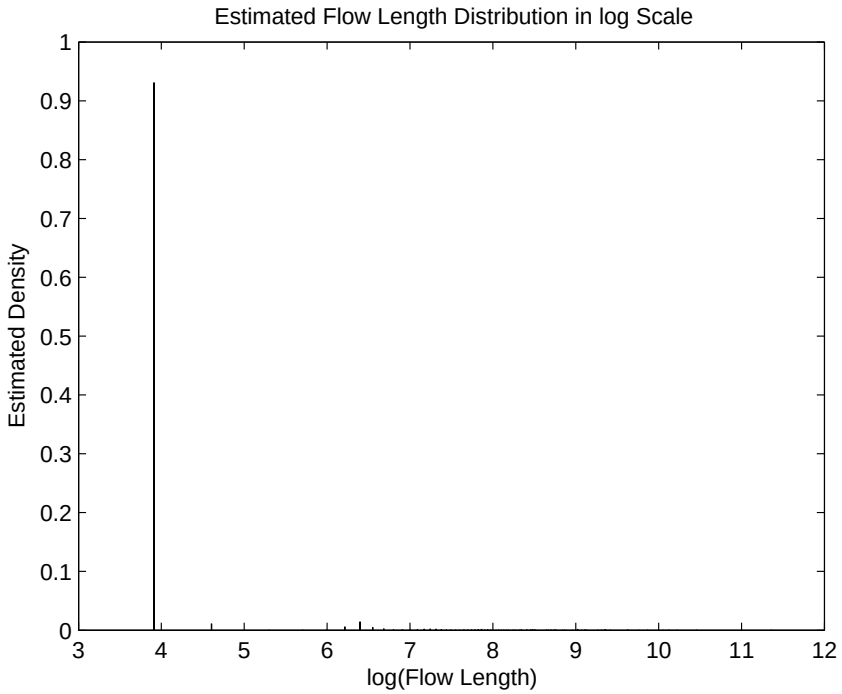


Figure 2.22: Estimated LAN flow length distribution

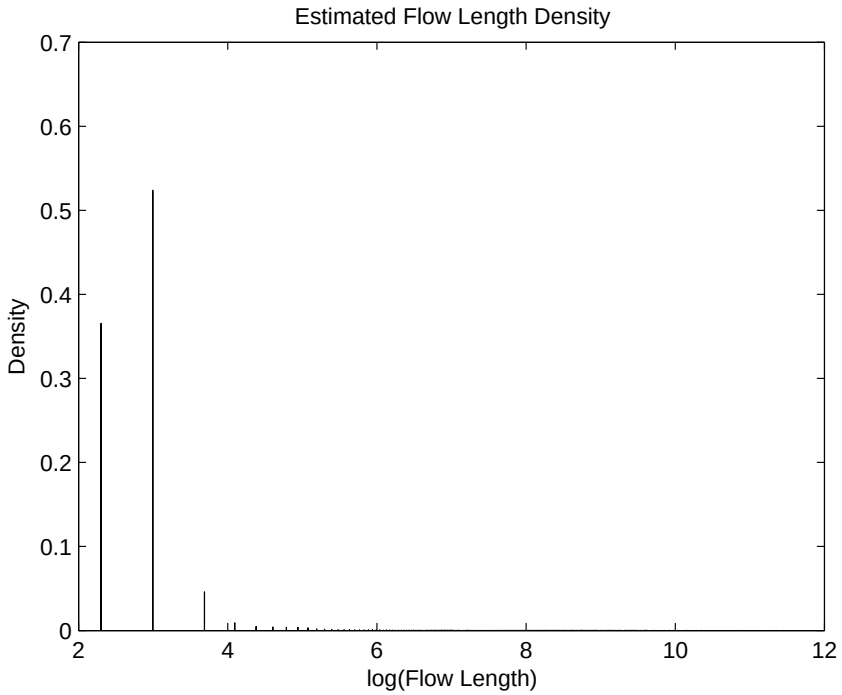
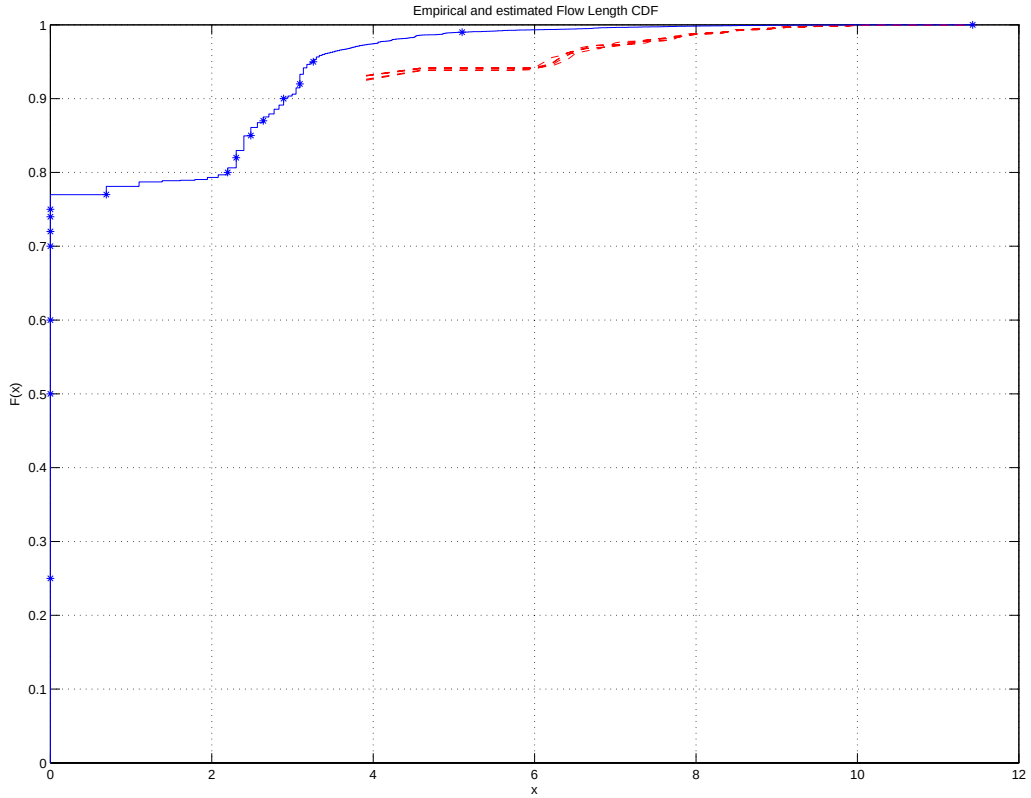
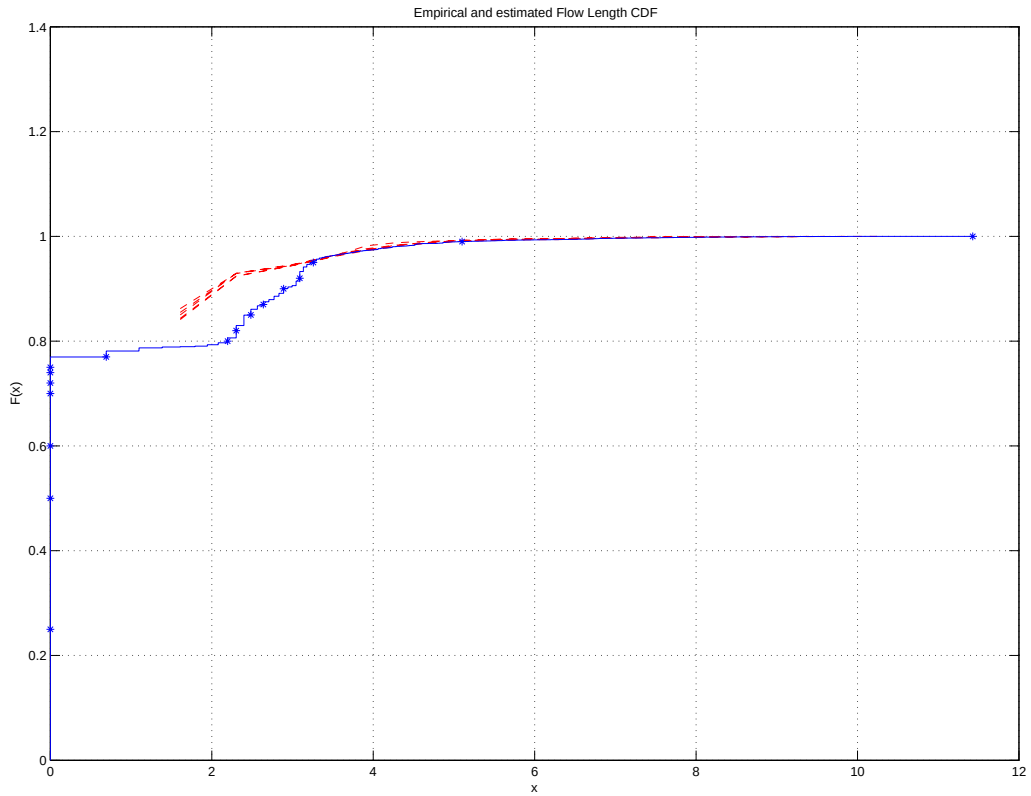


Figure 2.23: Estimated UNC flow length distribution

A very sever problem exists in the estimate of the number of active flows $\hat{M}$ for these heavy-tailed distributions. In the uniform and poisson case, it is estimated almost perfectly. Whereas with pareto distribution, it starts underestimate by re-



(a) First EM



(b) Two-Stage Sampling

Figure 2.24: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for LAN data with .01 sampling rate ($p_f = 0.1,\ p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution

porting around 70% of the true flows with 5% sampling rate. Although the observed number of flow is even much less than what we recovered, we still do not like our estimator for such downside. It becomes worse with the real data of much heavier tails.

2.8.3 Mixture Flow Length and Byte Size Distributions

We turn our attention to mixture distributions (both flow length and byte size) and evaluate the three proposed algorithms. Specifically, we consider simulated flow length data from $M = 1,000$ active flows obtained from a mixture distribution with point mass at 1, while the second component follows a: (i) uniform with domain $[3000, 7000]$, (ii) Poisson with mean 5,000 and (iii) Generalized Pareto with shape parameter 100/90, location and scale parameter 50. The parameters for these three distributions in the second component were set so as to match their expected values. The flow sizes were generated for all cases from a uniform distribution with domain $[100, 500]$ bytes. The mixing coefficient α was set to $0.3, 0.5$ and 0.7, while the sampling rate to $p = 0.01$ and 0.05.

An example of one realization of such a mixture distribution ($\alpha = 0.7$ and sampling rate $p = .05$) and its estimate, where the second component is Poisson distributed, is shown in Figures 2.25 and 2.26, respectively. It can be seen that the two-stage EM estimate captures very well the support of the original distribution and the mixing coefficient α, as well as the second component. It should be noted that the apparent visual discrepancy between the original and the estimated distributions is mostly to the somewhat large bin size used in the histogram of the original one.

In Figure 2.27, boxplots for the two-stage EM estimates of $\alpha = .7$, the mean flow length (5,000 packets) and the mean flow size (464,000 bytes) of the second component are shown. It can be seen that the estimates are very good, with a fairly

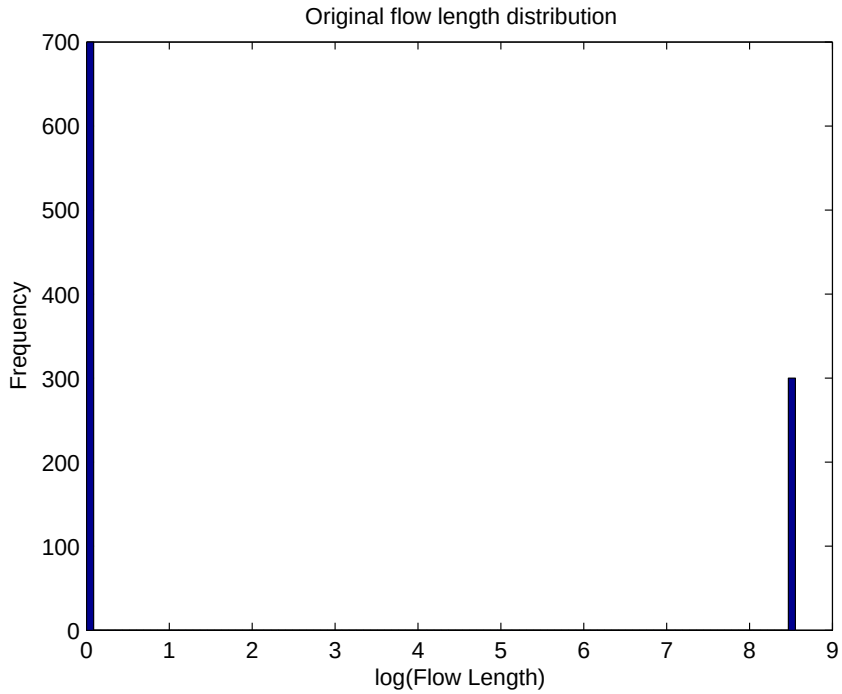


Figure 2.25: Original flow length distribution

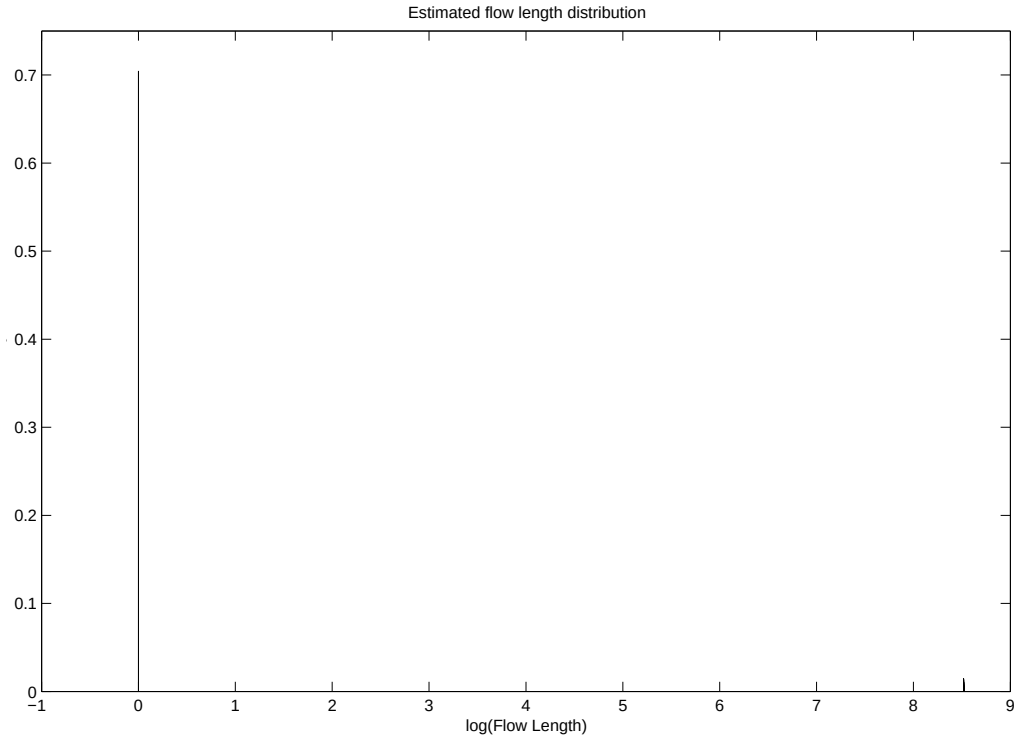


Figure 2.26: Estimated flow length distribution, through a two stage EM algorithm

narrow range of values.

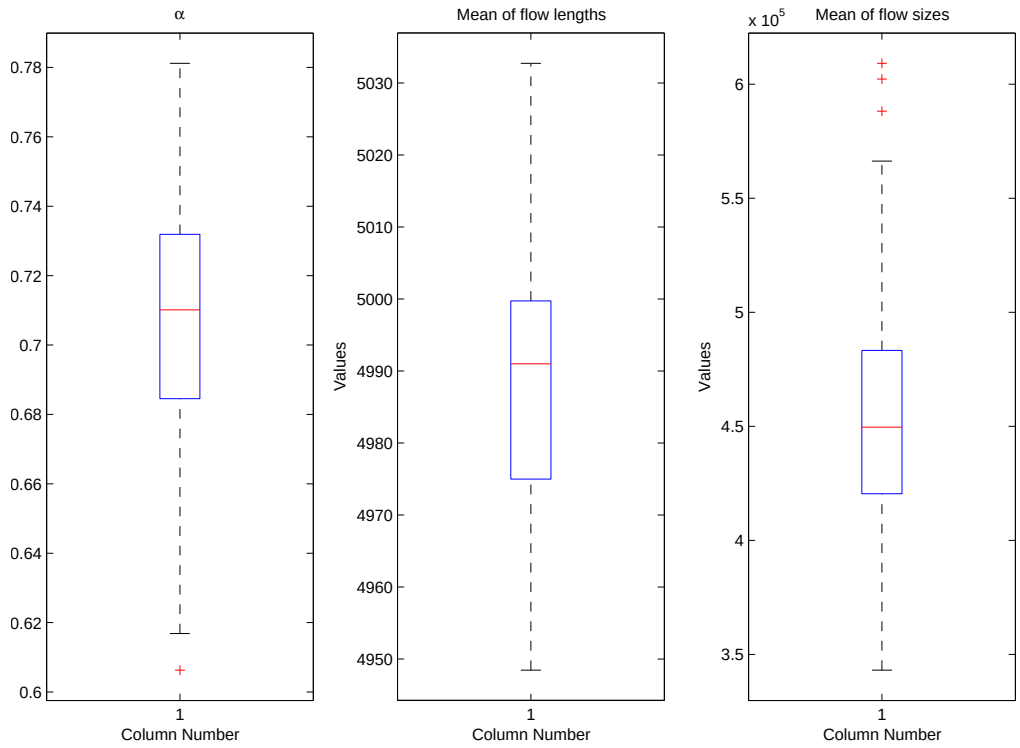


Figure 2.27: Boxplots of various parameters of a mixture distribution

In Table 2.4, the mean squared errors (MSE) obtained from 100 replications, for the estimates of the number of active flows M, the mean and variance of the flow length (in packets) and size (in bytes) are shown for different mixing coefficients α for the three distributions under consideration.

It can be seen that the MSE for the estimated weight is extremely small, indicating a very precise estimate for all values of α. Further, as the contribution of the second component increases (smaller α) the quality of the estimates for both the mean number of packets improves for all distributions. This is because less information on the second component is available when α is larger. On the other hand, the MSE for the mean number of bytes exhibits the opposite behavior, while the MSE for the corresponding variances are of the similar order for all values of α. This phenomenon is due to the fact that longer flows are of more variability. These re-

Table 2.4: Mean squared errors for various parameter estimates obtained through the 2-stage EM algorithm for different distributions, with sampling rate $p = 0.01$.

α	$\hat{\alpha}$	Mean Packets	Var Packets	Mean Bytes	Var Bytes
Poisson Distribution					
0.7	0.0053	1.23E+07	5.26E+06	1.69E+10	9.48E+09
0.5	0.0162	1.23E+07	5.26E+06	1.69E+10	9.48E+09
0.3	0.0043	2.26E+06	5.26E+06	3.97E+10	7.77E+10
Uniform Distribution					
0.7	4.35E-06	1.19E+07	3.42E+06	2.19E+10	1.28E+10
0.5	6.19E-06	6.06E+06	5.06E+06	2.53E+10	3.01E+10
0.3	4.38E-06	2.10E+06	3.74E+06	3.90E+10	7.38E+10
Pareto Distribution					
0.7	0.0573	2.13E+07	4.13E+08	1.37E+11	1.22E+13
0.5	0.1801	1.04E+07	2.75E+08	3.63E+11	2.64E+13
0.3	0.4086	5.46E+06	3.47E+07	7.10E+11	1.57E+13

sults indicate the difficulty of estimating non-parametrically mixture distributions, especially in the presence of a dominant point mass, and also demonstrate that procedures that ignore the mixture structure would not fare well. Similar conclusions are reached when the sampling rate increases to $p = 0.05$. The main difference is that due to the higher sampling rate the performance of the estimators improves considerably.

We examine next the performance of the maximum likelihood estimator based on the two stage sampling mechanism for the same set of mixture distributions. However, in this case we consider a range of sampling rates for the flows (p_f parameter) while fixing the sampling rate for the packets.

An example of one realization of such a mixture distribution ($\alpha = 0.7$ and sampling rates $p_f = 0.5$ and $p_p = 0.2$, respectively) and its estimate, where the second component is uniformly distributed is shown in Figures 2.28 and 2.29, respectively. It can be seen that the estimate captures very well the support of the original distribution and to a large extent the mixing coefficient α.

In Figure 2.30, boxplots for the estimates of M, the mean flow length (1,500 packets) and the mean flow size (464,000 bytes) of the entire distribution are shown. It can be seen that once again the estimates are very good, exhibiting a fairly narrow

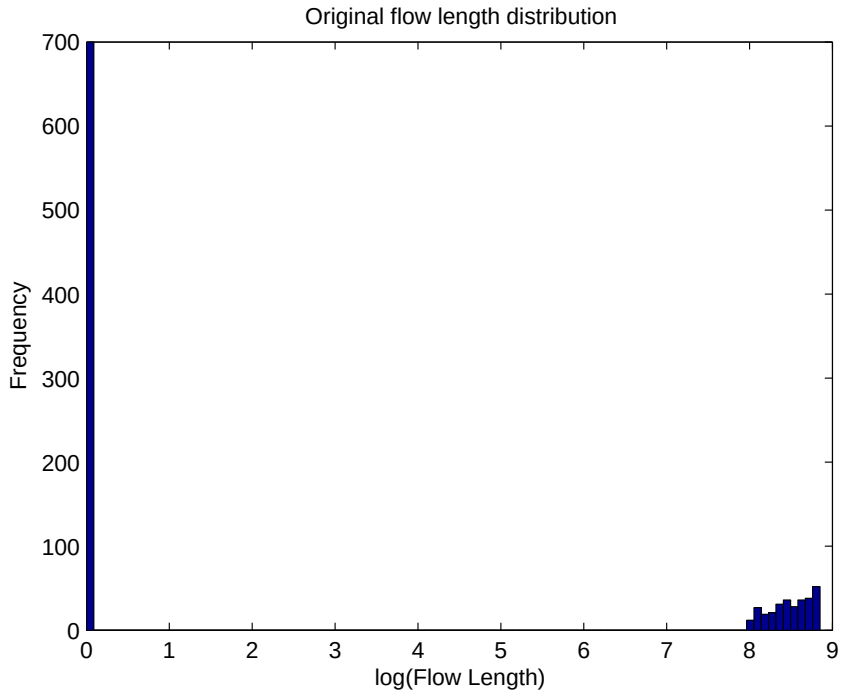


Figure 2.28: Original flow length distribution

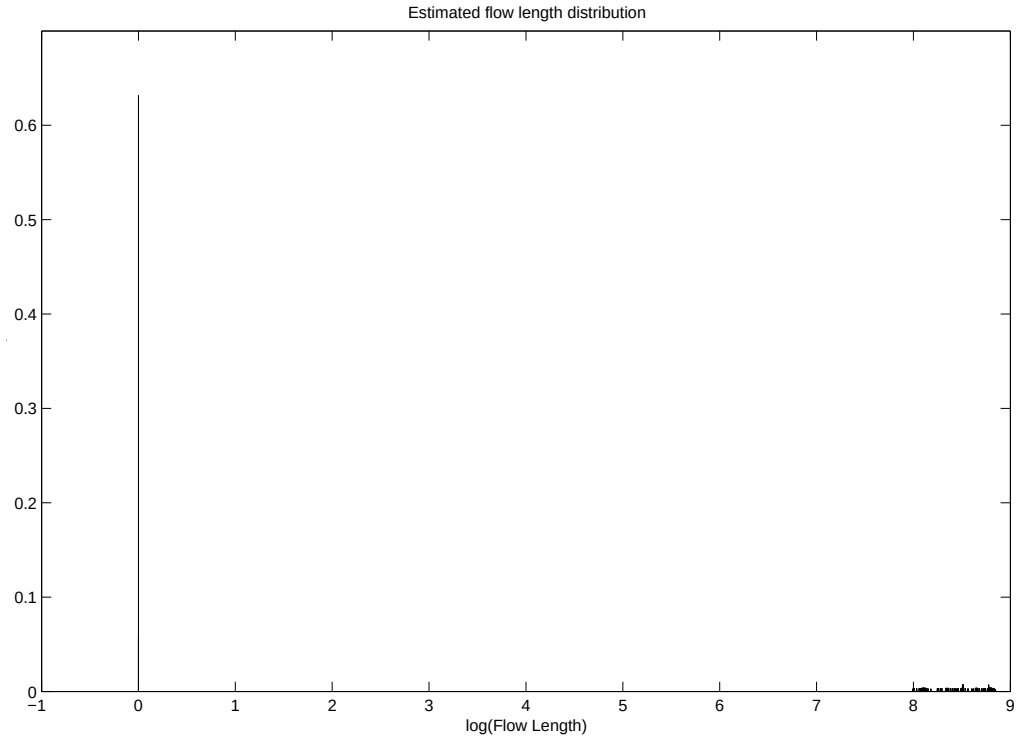


Figure 2.29: Estimated flow length distribution, through a two stage sampling scheme

range of values.

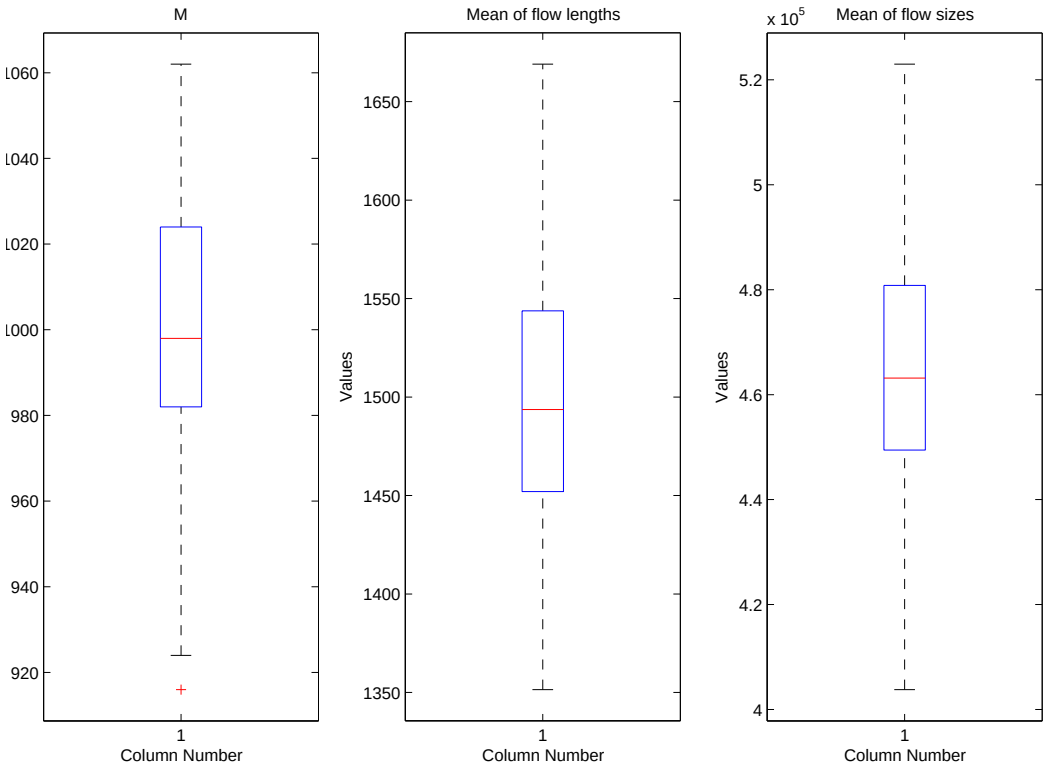


Figure 2.30: Boxplots for various parameters of a mixture distribution

In Table 2.5, the mean squared errors (MSE) obtained from 100 replications, for the estimates of the number of active flows M, the mean and variance of the flow length (in packets) and size (in bytes) are shown for different flow sampling rates p_f for the three distributions under consideration.

It can be seen that as the sampling rates of the flows increases, the quality of the estimates improves, as expected. The reason is that by sampling more flows, we are able to better capture their characteristics, even when the packet sampling rate is very small. Further, the results are comparable for the estimates of M for all three distributions, but vary for the other parameters. For example, the quality of estimates for the flow lengths and sizes deteriorates for the heavy-tailed Pareto distribution. There is an interesting interplay between the two sampling rates for

Table 2.5: Mean squared errors for various parameter estimates obtained through the 2-stage sampling scheme for different distributions, with sampling rate $p = 0.01$.

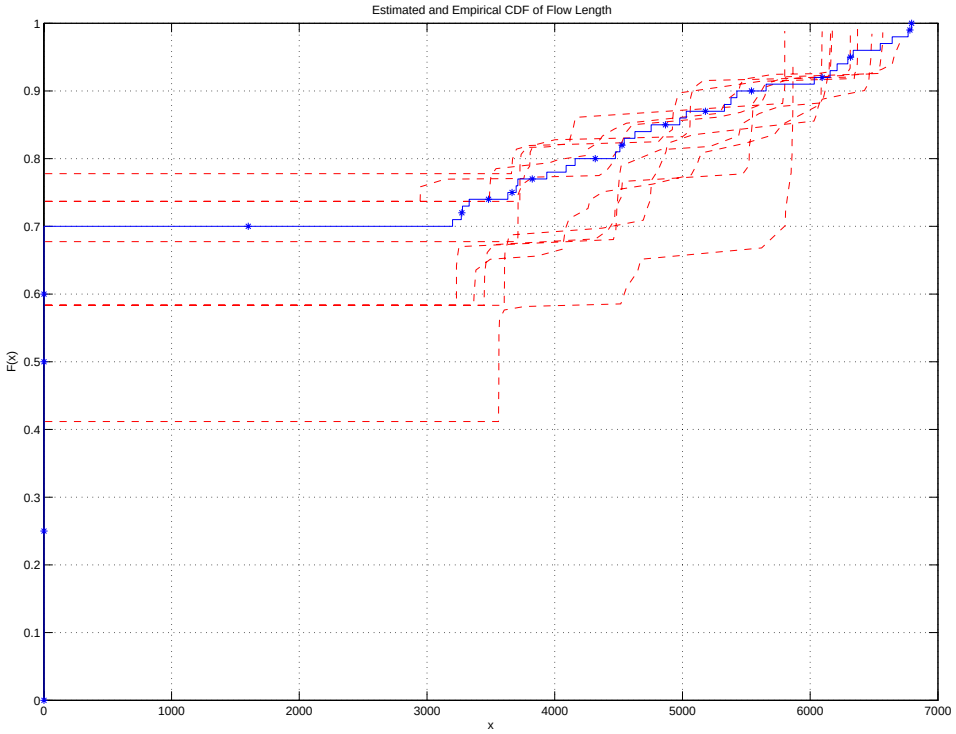
p_f	M	Mean Packets	Var Packets	Mean Bytes	Var Bytes
Poisson Distribution					
0.05	17232	1.01E+05	2.67E+04	1.05E+10	1.24E+10
0.1	8871	5.12E+04	9.89E+03	6.19E+09	8.04E+09
0.3	1900	9.32E+03	2.07E+03	1.18E+09	5.17E+09
0.5	923.24	5.00E+03	1.33E+03	6.92E+08	4.44E+09
Uniform Distribution					
0.05	23212	8.96E+04	3.63E+04	1.09E+10	1.75E+10
0.1	8383	4.40E+04	2.31E+04	6.28E+09	1.28E+10
0.3	2110	1.70E+04	6.29E+03	1.59E+09	6.47E+09
0.5	982.32	6.18E+03	2.83E+03	6.55E+08	6.12E+09
Pareto Distribution					
0.05	18548	7.28E+06	4.88E+08	8.16E+11	5.57E+13
0.1	7491	4.02E+06	4.13E+08	4.54E+11	4.70E+13
0.3	2660	8.97E+05	2.44E+08	1.05E+11	2.85E+13
0.5	865.64	4.15E+05	1.33E+08	5.51E+10	1.63E+13

flows and packets within flow, but at present its full understanding is not available and is a topic of further study. Experience suggests that a flow sampling rate $p_f = 0.3$ performs well, even when coupled with a very small packet sampling rate, so that the overall rate $p = p_f \times p_p$ remains small.

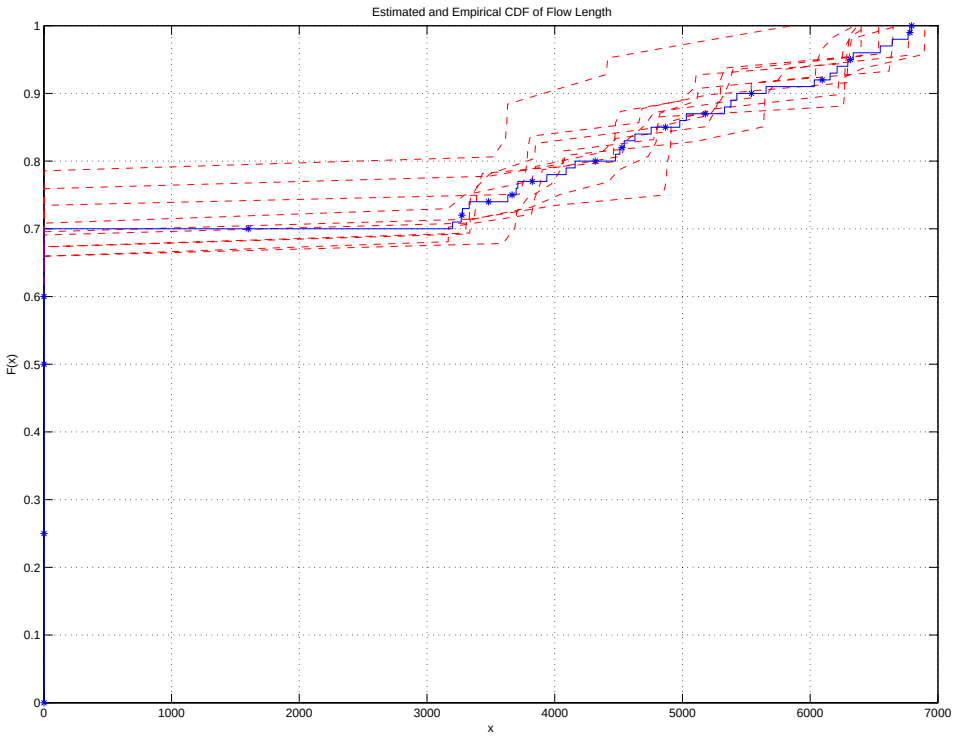
Similar qualitatively results are obtained for a larger packet sampling rate of 0.05, although the accuracy of the estimates naturally improves. However, the improvements are not usually large enough to compensate for the increased computational complexity both in the data collection and processing.

The following Figures 2.31-2.33 show a number of estimated CDF curves from the various distributions obtained from sampling a number of times from the true data and the true CDF curve, for mixtures of uniform, poisson and pareto respectively. The sampling methods used here are Bernoulli sampling with the adaptive EM algorithm and Two-Stage Sampling. Once again we see good agreement between the true and estimated CDF, although a large degree of variability is exhibited regarding the mixture coefficient α. It can further been seen that the Two-stage sampling scheme clearly outperforms Bernoulli sampling.

Remark: As indicated in Chapter 1, it is hard to implement online a 2-stage sam-

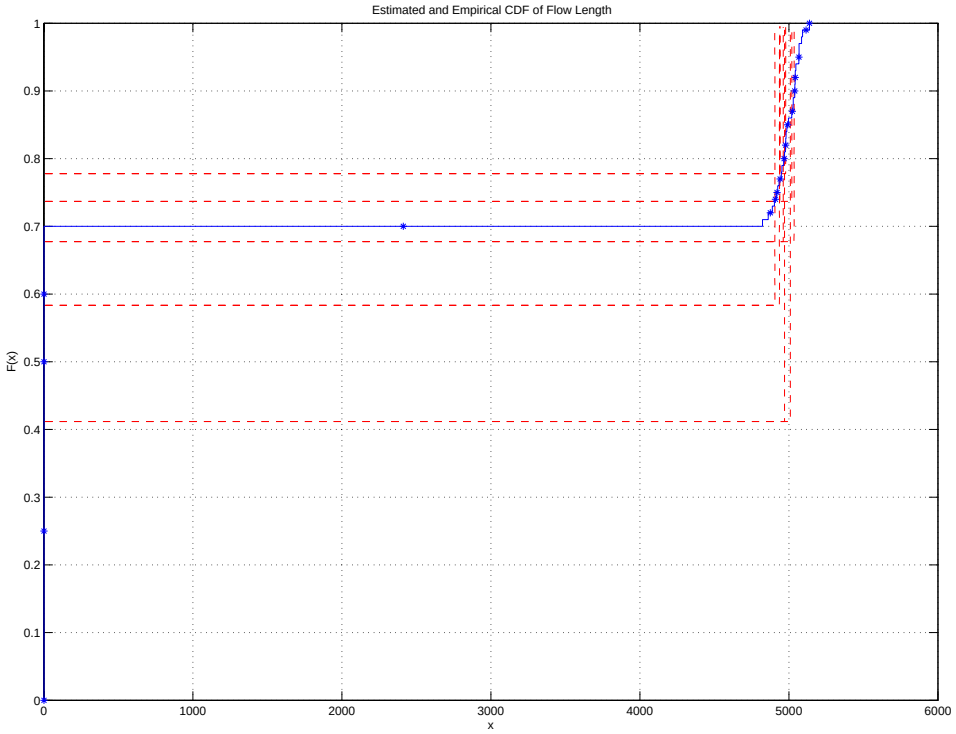


(a) Two-Stage EM

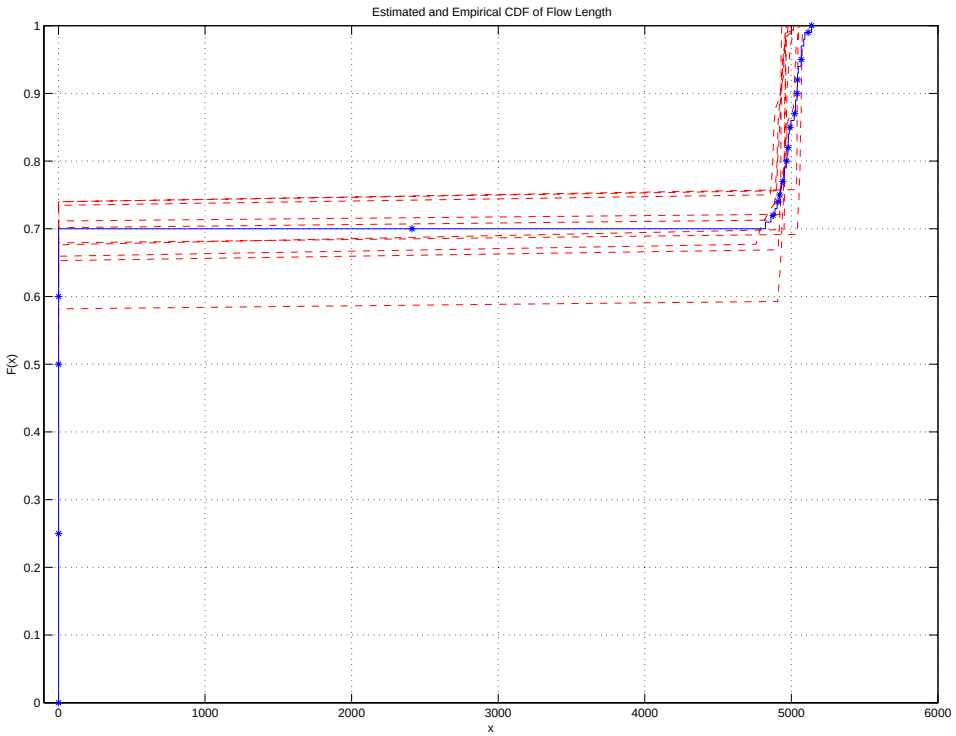


(b) Two-Stage Sampling

Figure 2.31: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for Mixture of spike at 1 and 100 Uniform flows, sampling rate is 0.05 ($p_f = 0.1,\ p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution

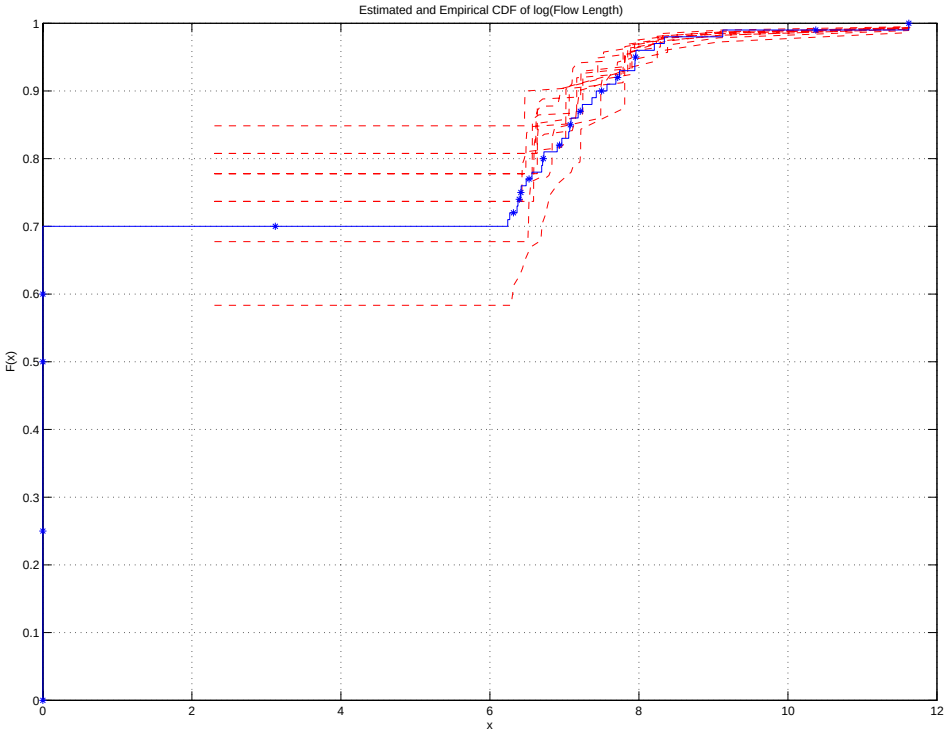


(a) Two-Stage EM

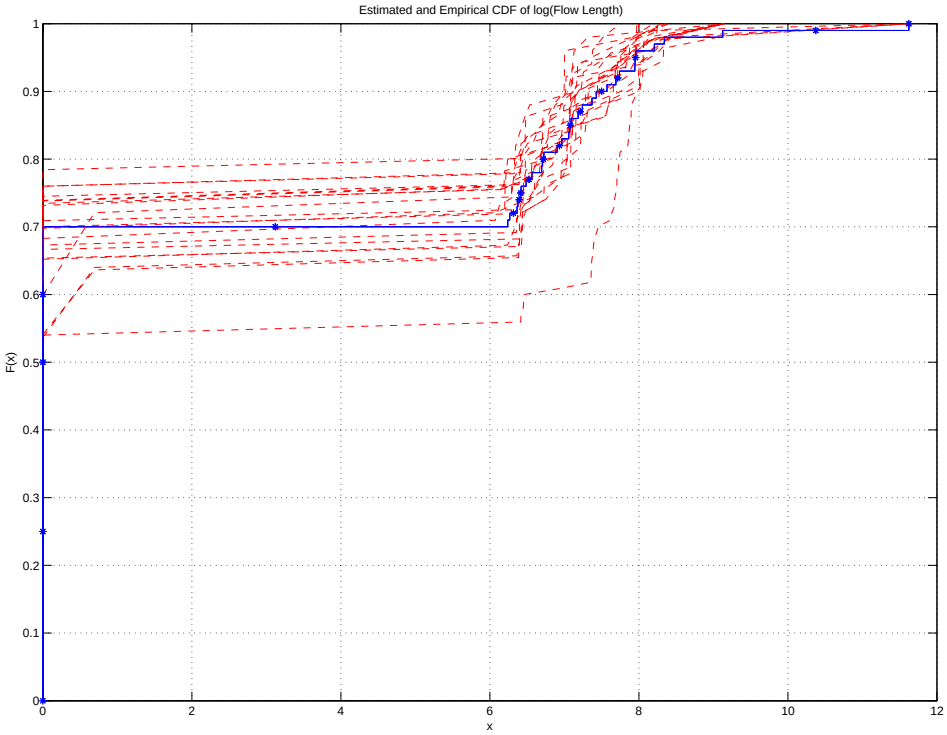


(b) Two-Stage Sampling

Figure 2.32: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for Mixture of spike at 1 and 100 Poisson flows, sampling rate is 0.05 ($p_f = 0.1,\ p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution



(a) Two-Stage EM



(b) Two-Stage Sampling

Figure 2.33: Dash lines are CDF Curves of the estimated flow length distribution from different sampled data for Mixture of spike at 1 and 100 Pareto flows, sampling rate is 0.05 ($p_f = 0.1,\ p_p = 0.1$ in Two-Stage Sampling); solid line with '*' is CDF Curve of the true flow length distribution

pling scheme at the flow level. What is feasible, is to implement it at the protocol/application level. For example, present day routers are capable of distinguishing between packets belonging to flows that are given different priorities either because of the underlying protocol (Real Time Protocol vs TCP) or because of terms in the service level agreement.

The following plots (Figures 2.34–2.36) correspond to QQ-plots for flow byte size distributions. The true data are simulated from Pareto, Poisson and uniform distributed flows with bytes per packet following a Normal(1350,100). The adaptive EM algorithm with sampling rate p=0.05 is applied. From these three QQ-plots we see good agreement for the flow bytes distribution, since all the dots are along the 45° line fairly tight.

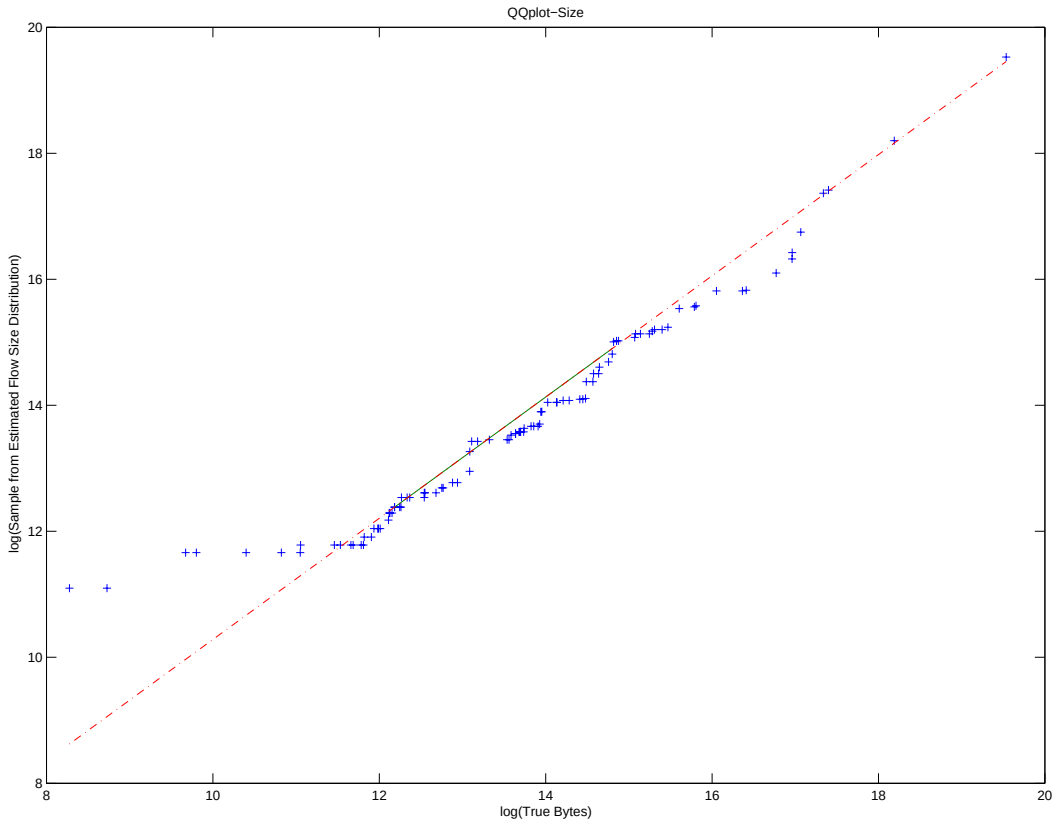


Figure 2.34: QQplot for true vs estimated flow size distribution from 100 pareto flows with bytes per packet following normal(1350,100); Adaptive EM with sampling rate is p=0.05

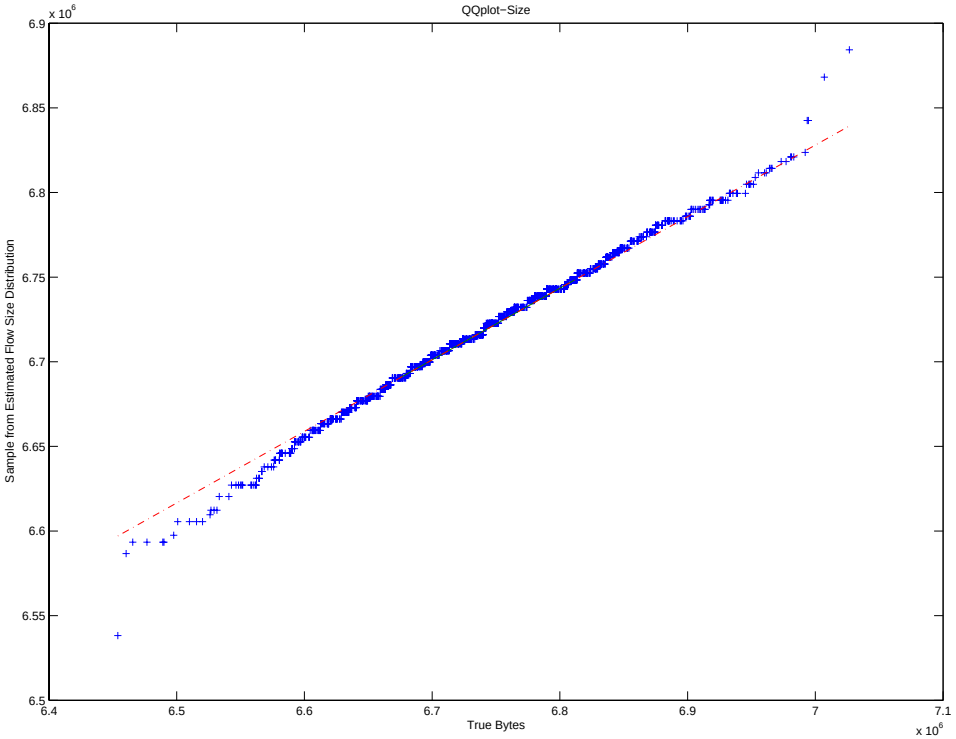


Figure 2.35: QQplot for true vs estimated flow size distribution from 1000 poisson flows with bytes per packet following normal(1350,100); Adaptive EM with sampling rate is p=0.05

2.8.4 Application to Abilene Data

We consider an application of the proposed methods to a real network trace obtained from the router of the Abilene network at Denver in June of 2005. The trace covers a 5-minute period and contains 65,535 active flows. The average flow length consists of 3 packets, but the variance takes a value of 430. Similarly, the average flow size is 2.7141×10^3 bytes, while the variance is 1.9377×10^{10}. The distributions of the true flow lengths and bytes (in log-scale) for this data set are shown in Figures 2.2. It can easily be seen that both distributions are heavy tailed, and to a large extent comprised of two separate components.

The data were sampled by both Bernoulli and two-stage sampling mechanisms, and the flow length and size distributions estimated by the proposed algorithms of two-stage EM and EM for two-stage sampling. The rate for single stage Bernoulli

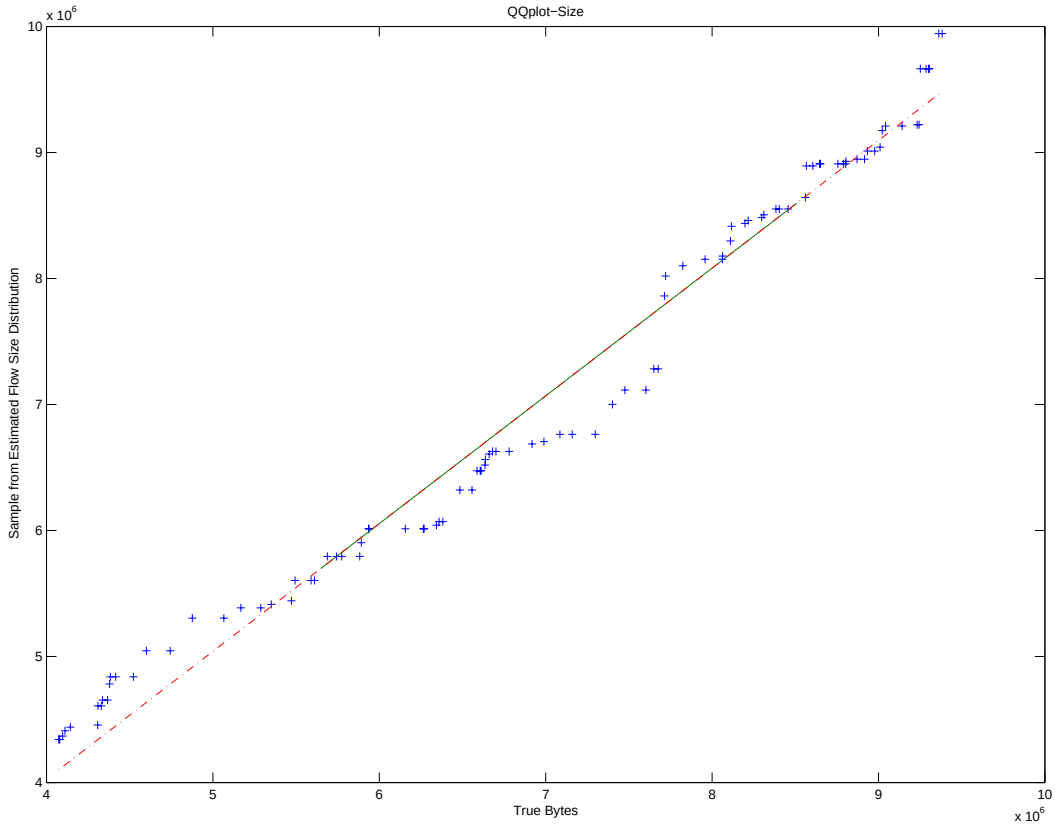


Figure 2.36: QQplot for true vs estimated flow size distribution from 100 uniform flows with bytes per packet following normal(1350,100); Adaptive EM with sampling rate is p=0.05

sampling was $p = 0.01$, while the rates for the two stage mechanism were $p_f = 0.05$ and $p_p = 0.2$. The estimate for the number of active flows under Bernoulli sampling was about 15,000, an underestimation of the true value. On the other hand, the estimate under the 2-stage sampling mechanism is almost perfect, $\hat{M} = 66,208$.

In Figures 2.37-2.40, the estimates of the flow length and size distributions are shown.

It can be seen that the 2-stage EM algorithm captures well the first component and the support of the flow length distribution, while it focuses on the second spike in the original distribution for the flow sizes. This is mainly due to the fact that due to the severe underestimation of the number of active flows, the algorithm mainly focuses on the second component comprised of larger flows. The estimates produced by the two

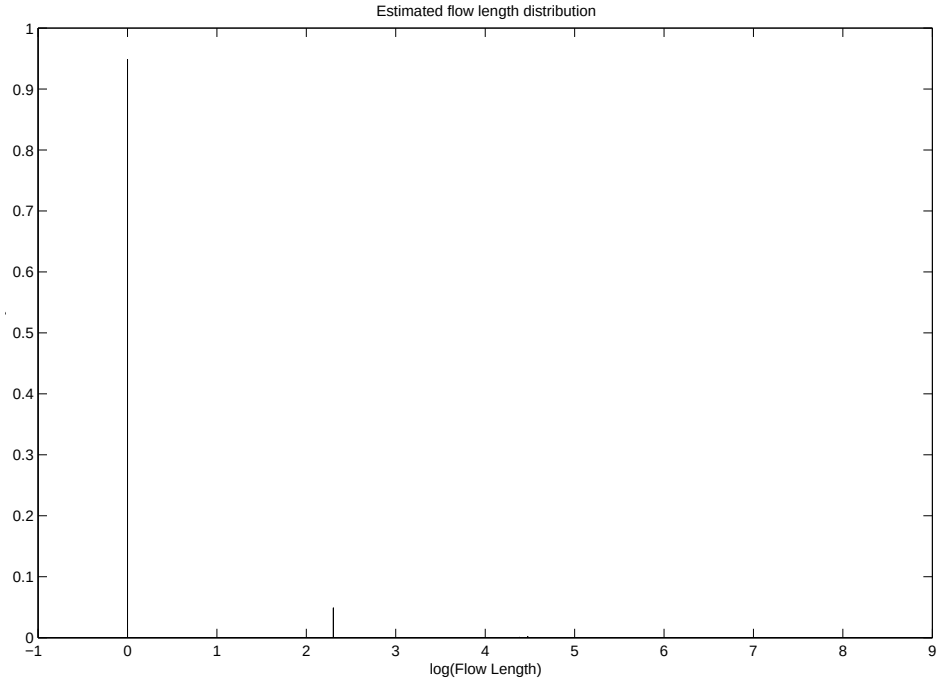


Figure 2.37: Estimated flow length distribution (in log-scale) of Abilene trace using a 2-stage EM algorithm

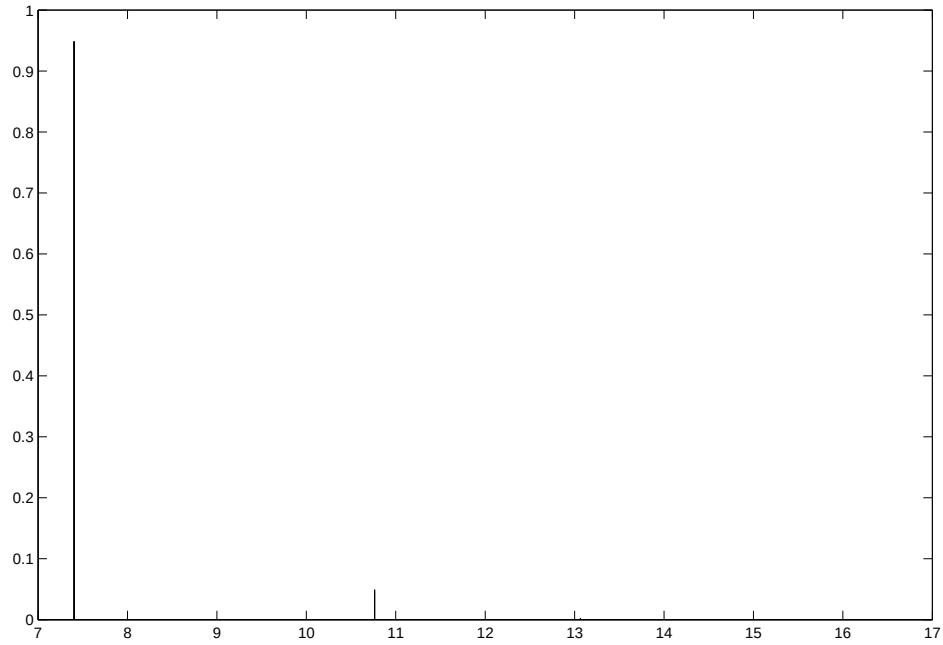


Figure 2.38: Estimated flow size distribution (in log-scale) of the Abilene trace using a 2-stage EM algorithm

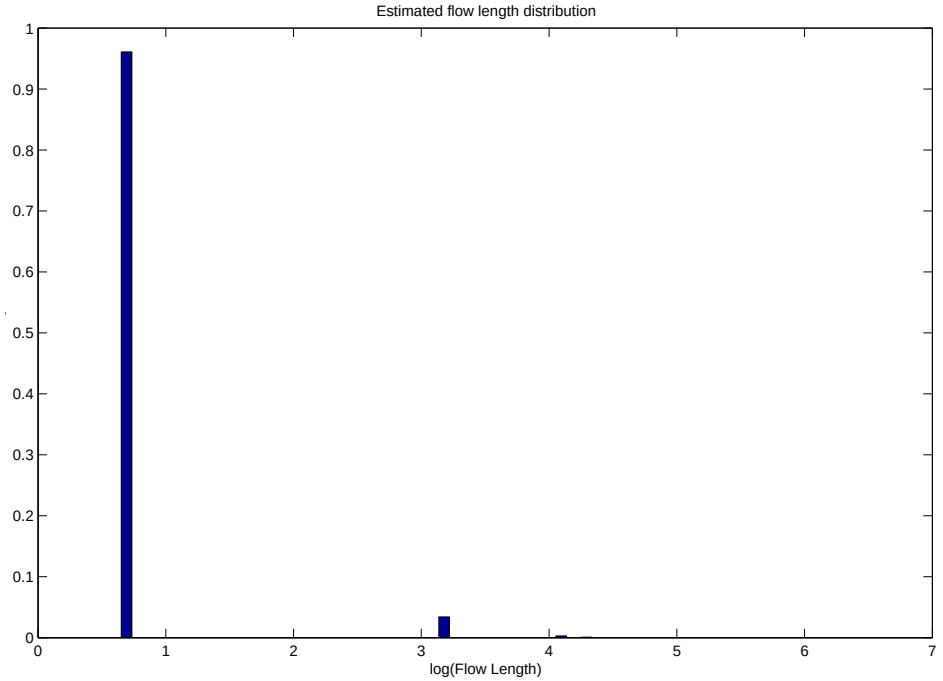


Figure 2.39: Estimated flow length distribution (in log-scale) of the Abilene trace using a 2-stage sampling mechanism

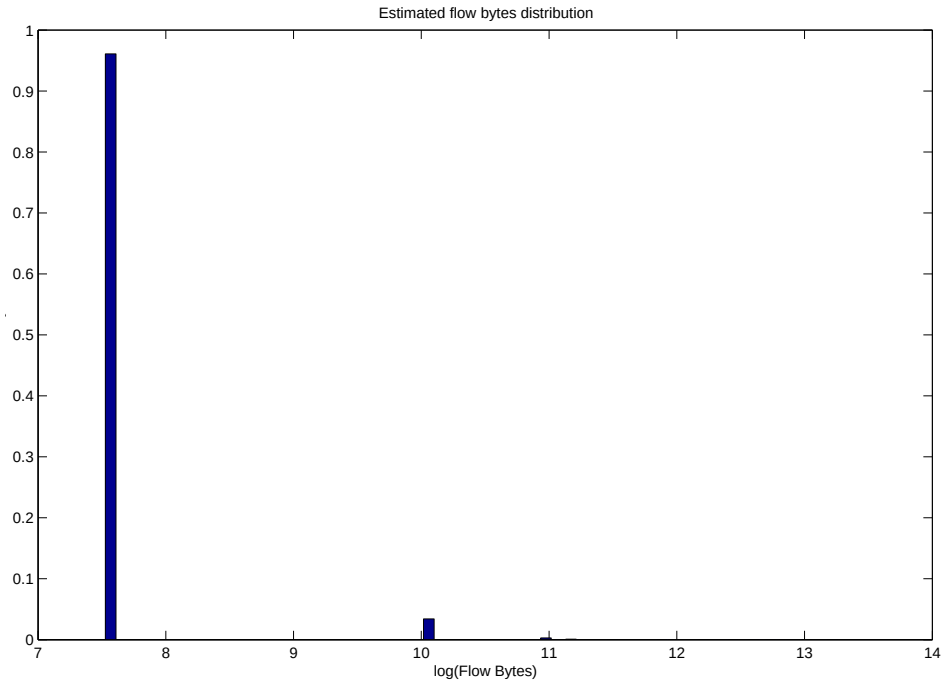


Figure 2.40: Estimated flow size distribution (in log-scale) of the Abilene trace using a 2-stage sampling mechanism

stage sampling scheme exhibit similar characteristics, although the spike in the flow length distribution is shifted slightly to the right. In this case, the difficulty comes from the fact that the flow sampling rate was set to a fairly small value ($p_f = 0.05$), which is as noted above does not produce particularly good estimates. This data example shows the challenging nature of the problem at hand, and its sensitivity both to the shape of the underlying distribution and the need to balance accuracy with computational efficiency.

Table 2.6: Mean squared errors for various parameter estimates obtained through the 2-Stage Sampling algorithm for different distributions, with sampling rate $p = 0.01$, and M=1000.

pf	$\hat{M}$	Mean Packets	Var Packets	Mean Bytes	Var Bytes
Pareto Distribution					
0.05	16616	3.23E+07	1.74E+09	2.06E+13	1.11E+15
0.1	7710	1.57E+07	1.26E+09	1.00E+13	8.00E+14
0.5	1104	1.69E+06	3.08E+08	1.05E+12	1.95E+14
Uniform Distribution					
0.05	18384	1.59E+05	3.09E+04	1.02E+11	2.00E+10
0.1	8330	7.26E+04	1.05E+04	4.73E+10	6.93E+09
0.5	940.92	7.82E+03	4.05E+03	5.02E+09	2.67E+09
Poisson Distribution					
0.05	15452	449.75	2.05E+03	3.77E+10	2.93E+09
0.1	9674	501.74	1.69E+03	1.05E+10	2.49E+09
0.5	915.28	494.04	382.51	7.57E+08	1.03E+09

CHAPTER 3

Moment-Based Methods to Estimate Flow Distributions on a Single Router

In the previous Chapter, estimation of flow characteristics based on nonparametric methods were discussed. However, for large for data sets in terms of number of flows (M), convergence of the proposed algorithms is rather slow [36]. Therefore, we introduce next a faster alternative that concentrates on estimating the first two moments of flow characteristics.

3.1 Framework

As in previous developments, we are interested in estimating characteristics of flow lengths and flow sizes (in bytes). We still call 'sampled flows' the ones that result after sampling packets from the link and organize them into flows according to their key. The true flows traversing the network will be called the 'original' ones. As before, some original flows are not going to be represented in the collection of sampled ones.

Since we are interested in estimating moments, no specific assumptions about the distributions of flow lengths and sizes are made, beyond that of their existence in the population of original flows. We denote by μ_N and σ_N^2, the mean and variance of the flow lengths, and by μ_B and σ_B^2 the mean and variance of the flow sizes. The hierarchical mechanism introduced in Chapter 2, is still employed. Specifically, M original flows are generated from a uniform distribution. Subsequently, from an

unknown distribution of flow lengths, the length of each flow $N_m,\ m = 1, 2, \cdots, M$ is generated. Similarly, the byte sizes of each packet Z_m^ℓ are generated from some other distribution whose mean and variance are denoted by μ_Z and σ_Z^2, respectively. Hence, we can obtain the size of each original flow m, from the expression $B_m = \sum_{\ell=1}^{N_m} Z_m^\ell$. To recap, we have that N_m are independent and identically distributed samples from a distribution ϕ, while packet sizes are independent and identically distributed according to some distribution H. As noted before, existence of the first two moments of ϕ and H is assumed.

Now, let μ_B denote the mean of the flow size distribution and σ_B^2 its variance. We can related these moments to those of flow lengths and packet sizes as follows:

$$\mu_B = \mathrm{E}(B_m) = \mathrm{E}(\sum_{\ell=1}^{N_m} Z_m^\ell) = \mu_Z \mu_N, \tag{3.1}$$

$$\sigma_B^2 = \mathrm{Var}(\sum_{\ell=1}^{N_m} Z_m^\ell) = \mu_N \sigma_Z^2 + \mu_Z^2 \sigma_N^2, \tag{3.2}$$

Further, the covariance between flows lengths and flow sizes is given by

$$\begin{aligned} \mathrm{COV}(N_m, B_m) &= \mathrm{COV}(N_m, \textstyle\sum_{\ell=1}^{N_m} Z_m^\ell) \\ &= \mathrm{E}(N_m \textstyle\sum_{\ell=1}^{N_m} Z_m^\ell) - \mathrm{E}(N_m)\mathrm{E}(\textstyle\sum_{\ell=1}^{N_m} Z_m^\ell) \\ &= \mu_Z(\mathrm{E}(N_m^2)) - \mu_N(\mu_N \mu_Z) \\ &= \mu_Z(\sigma_N^2 + \mu_N^2) - \mu_N(\mu_N \mu_Z) \\ &= \mu_Z \sigma_N^2. \end{aligned} \tag{3.3}$$

The last equalities of the Equations (3.1) and (3.2) follow according to the discussion in Examples 3.10 and 3.17 in the book [40], by first conditioning on N_m and then taking expectation.

The sampled flows are obtained by sampling according to a Bernoulli scheme of packets from the original flows. Hence, every packet is sampled with probability p. Let $\{n_m\}_{m=1}^M$ denote the number of packets of the m_{th} sampled flow and $\{b_m\}_{m=1}^M$ its corresponding flow size. Obviously, $n_m = 0$ corresponds to original flows *not*

observed. By using an indicator variable $I_{m\ell}$ to denote whether the ℓ-th packet of the m-the original flow is sampled or not, we can write b_m as $b_m = \sum_{\ell=1}^{N_m} I_{m\ell}$. This gives M random samples from a second population.

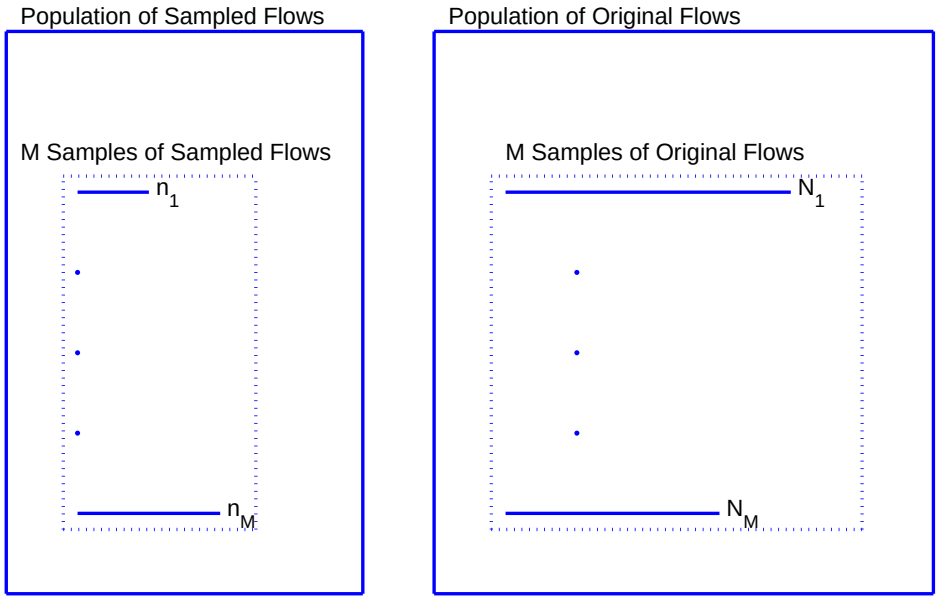


Figure 3.1: Two Populations in the Hierarchical Mechanism

We elaborate on this population next. The second population under consideration is a collection of sampled flows, whose elements are all the sampled flows containing packets selected according to Bernoulli sampling procedure with sampling rate p from packets from the population of original flows. This population is characterized by sample flow lengths $\{\mathcal{N}\}$ and sample flow sizes $\{\mathcal{B}\}$. A sample set of M elements from this population are selected according to the hierarchical mechanism described above. We now explain the claimed randomness of these M samples. From the hierarchical mechanism, we know that elements in this population space have sample lengths distributed according to a distribution $\{\pi_j\}_{j=0}^{N}$, the probability of a sampled flow having sample length j regardless of the original flow length. Given sample size M,

these sampled flows are selected according to this population distribution realized by the hierarchical mechanism, and thereby the M sampled flows are random samples of this second population.

To illustrate these concepts, we use the following toy example. Suppose the population of original flows contains flows of length $N = 1,\ N = 2$, and corresponding flow lengths distribution $\phi_1 = 0.5,\ \phi_2 = 0.5$. We can see that $\mu_N = 1.5, \sigma_N^2 = 0.25$. Suppose a Bernoulli sampling procedure of rate $p = 0.1$ is applied at the packet level. The corresponding sampled flow level population then has three possible sample lengths $\{0, 1, 2\}$. We then know that flows in the population of sampled flows are distributed following $\pi_0 = 0.5 \times 0.9 + 0.5 \times 0.9^2 = 0.855$, $\pi_1 = 0.5 \times 0.1 + 0.5 \times 2 \times 0.9 \times 0.1 = 0.14$ and $\pi_2 = 0.5 \times 0.1^2 = 0.005$ resulting from the hierarchical mechanism. Sampled flows are generated from this distribution, having 2 packets with probability 0.05, 1 packet with probability 0.14, and $\emptyset$ with probability 0.855. Given a sample size $M = 10$, suppose that we observe 2 sample flows with sample length 1, while the remaining 8 are missing. We can say that this set of samples with size 10 is a good representative of the population of sampled flows, referred as random samples above.

Remark: It should be noted that the sampling procedure of selecting M sampled flows employed does not correspond to an unequal probability sampling mechanism without replacement as discussed in Cochran [7]. Although different sample lengths have different sampling rates, these rates are determined by the population distribution. Unequal probability sampling without replacement for a finite population is defined as a sampling procedure, where each element in the population has different selection rate. On the other hand, in this case, all the elements face an equal sampling probability; however, the resulting larger sampling rate on some of the sampled flows stems from the larger number of elements in the population. We use the following example to illustrate the differences between the two. Suppose a finite population has elements 1,1,1,2,3. If we are to perform a sampling procedure with

replacement give a certain sample size predefined, we can generate random samples from this population from a distribution who has value 2 with probability 0.2, value 3 with probability 0.2, and value 1 with probability 0.6. The resulting random sample should represent the population because every single elements have the same probability of sampling. However, if we define sampling rate for the item of value $x = 1, 2, 3$ proportional to x, then the previous equal sampling rate balance will not hold. This latter sampling procedure is then unequal probability sampling.

Given the above sampling mechanism, the observed data correspond to the packets and their sizes in bytes of the sampled flows. Hence, we can construct the flow lengths and size of the sampled flows given by the collections $\{n_m : n_m > 0\}_{m=1}^{M}$ and sample sizes $\{b_m : n_m > 0\}_{m=1}^{M}$, respectively.

The goal of this study is to estimate the first and second moments of the original flow level quantities, namely $(\mu_N, \sigma_N^2, \mu_B, \sigma_B^2)$.

Notice that we can easily estimate the first two moments of the packet size (μ_Z, σ_Z^2), using the observed quantities, i.e.

$$\hat{\mu}_Z = \sum_m \sum_\ell Z_m^\ell I_{m\ell} / \sum_m \sum_\ell I_{m\ell} = \sum_m b_m / \sum_m n_m, \tag{3.4}$$

and

$$\hat{\sigma}_Z^2 = \sum_\ell \sum_m (Z_m^\ell I_{m\ell} - \sum_\ell \sum_m Z_m^\ell I_{m\ell} / \sum_\ell \sum_m I_{m\ell})^2 / (\sum_\ell \sum_m I_{m\ell} - 1). \tag{3.5}$$

Next, we derive relationships connecting moments of the sampled flows to those of the original flows. Subsequently, we discuss two methods to estimate the quantities of interest utilizing the system of equations (3.6 – 3.10).

$$\begin{aligned} \mu_n &= \mathrm{E}(n_m) = \mathrm{E}(\textstyle\sum_{\ell=1}^{N_m} I_{m\ell}) \\ &= \mathrm{E}_{N_m} \mathrm{E}(\textstyle\sum_{\ell=1}^{N_m} I_{m\ell} | N_m) \\ &= \mathrm{E}_{N_m}(N_m \mathrm{E}(I_{m\ell})) = p\mu_N. \end{aligned} \tag{3.6}$$

$$
\begin{aligned}
\sigma_n^2 &= \mathrm{Var}(n_m) = \mathrm{Var}(\textstyle\sum_{\ell=1}^{N_m} I_{m\ell}) \\
&= \mathrm{E}(N_m)\mathrm{Var}(I_{m\ell}) + \mathrm{Var}(N_m)(\mathrm{E}(I_{m\ell}))^2 \\
&= p(1-p)\mu_N + p^2\sigma_N^2.
\end{aligned} \tag{3.7}
$$

$$
\begin{aligned}
\mu_b &= \mathrm{E}(b_m) = \mathrm{E}(\textstyle\sum_{\ell=1}^{N_m} Z_m^\ell I_{m\ell}) \\
&= \mu_N \mathrm{E}(Z_m^\ell I_{m\ell}) \\
&= \mu_N \mu_Z p = \mu_n \mu_Z.
\end{aligned} \tag{3.8}
$$

$$
\begin{aligned}
\sigma_b^2 &= \mathrm{Var}(b_m) = \mathrm{Var}(\textstyle\sum_{\ell=1}^{N_m} Z_m^\ell I_{m\ell}) \\
&= \mathrm{E}(N_m)\mathrm{Var}(Z_m^\ell I_{m\ell}) + \mathrm{Var}(N_m)(\mathrm{E}(Z_m^\ell I_{m\ell}))^2 \\
&= \sigma_Z^2 \mu_n + \mu_Z^2 \sigma_n^2.
\end{aligned} \tag{3.9}
$$

$$
\begin{aligned}
\mathrm{cov}(b,n) &= \mathrm{cov}(b_m, n_m) = \mathrm{COV}(n_m, \textstyle\sum_{\ell=1}^{n_m} z_m^\ell) \\
&= \mathrm{E}(n_m \textstyle\sum_{\ell=1}^{n_m} z_m^\ell) - \mathrm{E}(n_m)\mathrm{E}(\textstyle\sum_{\ell=1}^{n_m} z_m^\ell) \\
&= \mu_Z(\mathrm{E}(n_m^2)) - \mu_n(\mu_n \mu_Z) \\
&= \mu_Z(\sigma_n^2 + \mu_n^2) - \mu_n(\mu_n \mu_Z) \\
&= \mu_Z \sigma_n^2.
\end{aligned} \tag{3.10}
$$

The independence between sampling procedure and packet size distribution is assumed. Equations (3.6) and (3.7) use the fact that each packet is selected independently with probability p, and thus $I_{m\ell}$ follows a Bernoulli distribution with parameter p. In Equation (3.10), z_m^ℓ is the size (in bytes) of the ℓth observed packets on flow m. Equation (3.8) and (3.9) use the results of Equation (3.6).

3.2 Method-of-Moment Like Method (MM)

We borrow the idea of Method of Moments, where the desired population moments can be estimated by matching the empirical sample estimates to the theoretical

population ones[39]. Notice that the moments of the sampled flows can be estimated from the observed data. By matching the moments of the sampled flows to those of the original flows, we obtain estimates of the latter. The first two equations (3.6) and (3.7) give such estimates of the mean and variance of original flow lengths

$$\hat{\mu}_N^{(MM)} = \hat{\mu}_n/p \tag{3.11}$$

$$\hat{\sigma}_N^{2(MM)} = \frac{\hat{\sigma}_n^2 - (1-p)\hat{\mu}_n}{p^2}. \tag{3.12}$$

Combining the above two sets of estimates (3.11-3.12) and (3.4-3.5), we further provide the estimate of moments of flow sizes based on Equations (3.1-3.2).

$$\hat{\mu}_B^{(MM)} = \hat{\mu}_Z \hat{\mu}_N^{(MM)},$$

and

$$\hat{\sigma}_B^{2(MM)} = \hat{\mu}_N^{(MM)} \hat{\sigma}_Z^{2(MM)} + \hat{\mu}_Z^{2(MM)} \hat{\sigma}_N^{2(MM)}$$

where the independence of flow lengths and packet bytes is assumed.

3.3 Moment Least Square Method (MLS)

The above method-of-moment-like estimates uses only two of the equations. In this study we propose a Moment Least Squares Method to utilize all the information contained in the system of equations. The idea of the proposed MLS is to estimate $(\mu_N, \sigma_N^2, \mu_Z, \sigma_Z^2)$, which is equivalent to estimate $(\mu_N, \sigma_N^2, \mu_B, \sigma_B^2)$ from the above discussion, by minimizing a given target function, an approach that uses all the five equations in the system. We define the target function as the sum of squares of the differences of the empirical estimates and the theoretical results, as the following equations.

$$\begin{aligned} L(\mu_N, \sigma_N^2, \mu_Z, \sigma_Z^2) &= [\hat{\mu}_n - p\mu_N]^2 + [\hat{\sigma}_n^2 - p(1-p)\mu_N + p^2\sigma_N^2]^2 \\ &\quad + [\hat{\mu}_b - \hat{\mu}_n\mu_Z]^2 + [\hat{\sigma}_b^2 - \sigma_Z^2\hat{\mu}_n + \mu_Z^2\hat{\sigma}_n^2]^2 \\ &\quad + [\hat{\mathrm{cov}}(b,n) - \mu_Z\hat{\sigma}_n^2]^2 \end{aligned} \tag{3.13}$$

This optimization can be achieved by the Newton-Raphson algorithm [48]. Further exploration reveals that the target function can be broken down into two subsystems. The first system consists of the first two components of the target function $L_1(\mu_N, \sigma_N^2) = [\hat{\mu}_n - p\mu_N]^2 + [\hat{\sigma}_n^2 - p(1-p)\mu_N + p^2\sigma_N^2]^2$; and it can be used alone to estimate the moments of flow lengths. The second subsystem consists of the remaining components $L_2(\mu_Z, \sigma_Z^2) = [\hat{\mu}_b - \hat{\mu}_n\mu_Z]^2 + [\hat{\sigma}_b^2 - \sigma_Z^2\hat{\mu}_n + \mu_Z^2\hat{\sigma}_n^2]^2 + [\hat{\mathrm{cov}}(b,n) - \mu_Z\hat{\sigma}_n^2]^2$ to estimate the bytes moments.

3.4 Bias Correction

In both MM and MLS, we establish the estimates of original flow level moments from the empirical estimate of sampled flow level moments. In this section, we focus on the empirical estimate of sampled flow level moments $\{\mu_n,\ \sigma_n^2\}$. The population under consideration now is sampled flows with lengths distributed according to a distribution $\{\pi_j\}_{j=0}^N$.

Because of randomness of the M samples from population of sampled flows, sample average would be a good estimator for the population mean[7], i.e., $\hat{\mu}_n = \sum_m n_m/M$ and $\hat{\sigma}_n^2 = \sum_{m=1}^N (n_m - \hat{\mu}_n)^2/(M-1)$. Initially, one would easily initiate an estimation of average sample flow length μ_n by the observed sample lengths $\hat{\mu}_n^{(1)} = \sum_m n_m/r$, which is essentially an unbiased estimator of $\mathrm{E}(n|n>0) = p\mathrm{E}_N(\frac{N}{1-(1-p)^N})$; similarly, the sample variance of all positive sample lengths flows to estimate σ_n^2, $\hat{\sigma}_n^{2(1)} = \sum_{m:n_m>0}(n_m - \hat{\mu}_n^{(1)})^2/r$.

However, given the sampling procedure of selecting packets uniformly and the heavy-tail nature of flow lengths/bytes distribution, a lot of short flows may well be unobserved with $n_m = 0$, especially for the presence of small sampling rate p. Therefore, $\hat{\mu}_n^{(1)}$ would overestimate the average sample lengths by ignoring the 0 sample lengths flows.

We correct this biasedness by estimating the total number of active flows under

study M. We propose an estimate $\hat{M} = \frac{r}{1-c_{\hat{\mu}_N 0}}$. The intuition behind this procedure is that the expected number of observed flows is the total number of flows multiplied by the probability of a flow being observed. The denominator is an estimator of this probability, where $\hat{\mu}_N$ is initialized by the estimated average flow lengths applying $\hat{\mu}_n^{(1)}$ in either MM or MLS. Give the estimated number of active flows $\hat{M}$, $\hat{\mu}_n^{(2)}$ and $\hat{\sigma}_n^{2(2)}$ are updated by

$$\hat{\mu}_n^{(2)} = \sum_m n_m / \hat{M}$$

and

$$\hat{\sigma}_n^{2(2)} = (\sum_{m:n_m>0} (n_m - \hat{\mu}_n^{(2)})^2 + (\hat{M} - r)(0 - \hat{\mu}_n^{(2)})^2)/\hat{M}$$

to accommodate the unobserved flows. Subsequently, MM or MLS will be applied again with $\mu_n^{(2)}$, $\sigma_n^{2(2)}$ to get a new set of estimation of $(\mu_N, \sigma_N^2, \mu_B, \sigma_B^2)$. We will provide the performance assessment of these two sets of estimations in experiment demonstration (Section 3.7).

3.5 Subsampling

There are some variations in the estimation, especially for the parameter σ_N, and σ_B. This variability is enlarged by the coefficient of p^2 in the denominator of the systems of equations containing these parameters. Subsampling can be applied to reduce variation of the estimation, as long as the number of distinct sample lengths is large enough. The procedure is a two stage one. Firstly, we repeat the procedure of sample packets with new sampling rate pp several times, and perform the above analysis on each subsample set. Then next averaging the estimations of subsamples gives the overall estimate for the original data. This whole procedure is essentially the same as bernoulli sampling of sampling rate $p_n = p * pp$, but with reduced variability on the estimation.

3.6 Estimation with Constraint

In some cases, for example flows with Gamma, Poisson length distribution, there is additional information on the relationship between $\sigma_N^2 = \gamma\mu_N$, where γ is an coefficient needs estimated. With this additional information, MLS method is designed to utilize the most of the systems. This leads the first component of the target function $L1$ being optimized more efficiently with the constraint. The initial point for the optimization might be selected from MM estimator, i.e, $\hat{\mu}_N = \hat{\mu}_n/p$ and $\hat{\gamma} = \frac{\sigma_n^2 - (1-p)\hat{\mu}_n}{p\mu_n}$.

3.7 Experiment Demonstration

In this section, we provide empirical evidence of the performance of the derived moment-based estimators for a variety of simulated and real network traffic traces. Additionally, a simulated dynamic system is studied to demonstrate the performance of anomaly detection applying our methods.

The first set of experiments uses simulated flow length data from the following distributions: (i) Gamma , (ii) Poisson and (iii) Log-Normal. The parameters for these three distributions were set so as to match their mean value of 50, 500 and 5000. Packet sizes are simulated from a Uniform distribution with range $(1200, 1500)$ to mimic the real network traces. Bernoulli Sampling is applied to each dataset 100 times with expected sampling rate 0.01 on the packets. Method-of-Moment-Like(MM) estimators as well as ones with bias correction are calculated. This set of experiments is designed to assess bias correction method by comparing the average estimators and the corresponding empirical MSE. Table 3.1 shows the numerical results. Columns from left to right correspond to the bias of estimated average flow lengths, the bias of estimated variance flow lengths, and empirical MSEs associated with these two quantities, followed by the same quantities for the flow bytes. We first look at the average estimates of flow length in the third column of the table. The

method of estimating M reduces the bias and outperform the original MM method in all the three scenarios, especially in the presence of short flows. This also holds for the estimated variance of flow lengths and the statistics of flow bytes. It can also be seen that the MSEs of the estimated average flow lengths/sizes also are smaller when M is estimated. It worth noting that as the real average flow length increases the bias itself decreases, and this is because more flows can be observed and the number of missing flows are essentially reduced.

Table 3.1: Semiparametric Estimation on Multismpl Simulated Data, where packet size is from Unif(1200,1500).

		Bias(N_m)	B(V(N_m))	MSE(N_m)	MSE(V(N_m))	Bias(B_m)	Bias(V (B_m))	MSE(B_m)	MSE(V(B_m))
Gamma(50)	MM	80.38	-380.9	6.47E+05	2.45E+07	109929	-553710000	1.21E+12	8.08E+19
	Bias Correction	44.919	-500	2.02E+05	2.50E+07	62169	-832812000	3.87E+11	8.08E+19
Gamma(500)	MM	16.26	12080	3.84E+04	2.48E+11	34790	28084000000	1.43E+11	7.65E+23
	Bias Correction	13.42	-4721	3.04E+04	2.48E+11	30960	-2667000000	1.19E+11	7.65E+23
Gamma(5000)	MM	10.1	12400	2.51E+07	5.01E+06	70400	4.521E+11	4.57E+13	9.35E+12
	Bias Correction	10.1	12200	2.51E+07	5.01E+06	70400	4.516E+11	4.57E+13	9.35E+12
LogN(50)	MM	80.31	-327.84	6.46E+05	2.43E+07	59928	1601430000	3.60E+11	4.67E+19
	Bias Correction	44.676	-500	2.00E+05	2.50E+07	32381	361730000	1.05E+11	4.68E+19
LogN(500)	MM	13.47	12187	2.84E+04	2.48E+11	5360	7459000000	1.25E+10	4.77E+23
	Bias Correction	10.57	-3905	2.19E+04	2.48E+11	3190	-4001000000	1.09E+10	4.77E+23
LogN(5000)	MM	9	11100	5.59E+05	2.50E+15	-85300	-6.066E+11	1.35E+12	5.26E+27
	Bias Correction	9	11100	5.59E+05	2.50E+15	-85300	-6.066E+11	1.35E+12	5.26E+27
Pois(50)	MM	76.93	-29.72	5.93E+05	2.43E+05	104067	69601000	1.08E+12	7.75E+17
	Bias Correction	37.827	-50	1.43E+05	2.50E+05	51237	-30140000	2.63E+11	7.75E+17
Pois(500)	MM	3.04	9122.7	5.51E+03	1.62E+07	3520	18635340000	9.57E+09	7.32E+19
	Bias Correction	-0.22	529.3	4.82E+03	2.28E+07	-890	2896140000	8.83E+09	7.32E+19
Pois(5000)	MM	0.5	4194.9	4.28E+04	2.44E+09	-1500	1.95346E+11	7.84E+10	8.11E+21
	Bias Correction	0.5	4194.9	4.28E+04	2.44E+09	-1500	1.95346E+11	7.84E+10	8.11E+21

We also evaluate the proposed methods on a real traffic trace from Abilene backbone network. The results are given in table 3.2. We compare all the proposed methods in terms of the estimated average and variance of flow lengths, and the empirical MSE. Among these methods, the method of subsampling MLS that incorporates a constraint outperforms all the others in all the quantities considered. The gains in efficiency are almost 100%. This is not surprising, because this method utilizes extra information. Furthermore, the bias correction method exhibits the best performance.

Next, a NS simulated experiment is designed to assess the performance of the methods in an anomaly detection scenario. In this experiment, there are 3 phases for traffic on one single link. At each phase of the experiment, flows are sent across

Table 3.2: Semiparametric Estimation on Abilene Data.

	Mean Packets	Var Packets	MSE(Mean Packets)	MSE(Var Packets)
True	28.974	4.46E+05		
Method of Moment	511.47	7.60E+06	2.62E+05	7.60E+06
Subsampling MM	601.8	9.03E+06	3.62E+05	9.03E+06
Subsampling MLS	601.8	9.03E+06	3.62E+05	9.03E+06
Subsampling MLS Restriction	385.97	5.79E+06	1.49E+05	5.79E+06
Bias correction $\hat{M}$	510.14	7.58E+06	2.60E+05	7.58E+06

the links but their number changes. We keep the packet size distribution, but increase the number of flows from 75, to 100, to 150. Meanwhile, in the last phase, there is a sudden change of the average flow length from 47.03 to 104.33. Estimations of all the related moments are listed in Table 3.3. It can be seen that the change on the average flow length in the last phase is captured by all methods. However, the bias correction based on estimated $\hat{M}$ outperforms the others, and additionally, capture the change of number of active flows.

Table 3.3: Semiparametric Estimation on NS-Simulated 3-phase UDP Data.

	Phase 1			Phase 2			Phase 3		
	True	MM	Bias Corr	True	MM	Bias Corr	True	MM	Bias Corr
Avg packet sizes	797.85	798.11	798.11	799.41	799.54	799.54	797.94	798.66	798.66
Var packet sizes	77.746	63.321	63.321	95.518	149.32	149.32	79.349	71.761	71.761
Avg flow lengths	36.173	127.59	90.244	47.03	133.33	97.959	104.33	170.71	139.67
Var flow lengths	244.33	0.00E+00	0.00E+00	186.25	0	0	5474.5	0	0
Avg flow sizes	28861	1.02E+05	7.20E+04	37596	1.07E+05	78322	83250	1.36E+05	1.12E+05
Var flow sizes	1.56E+08	1.03E+06	5.16E+05	1.19E+08	2.65E+06	1.43E+06	3.42E+09	2.09E+06	1.40E+06
Number of flows	75		41	100		49	150		121

To sum up, we understand through these numerical result that the proposed Bias Corrected MLS performs better on estimate the moments given the fact of heavy-tailed distribution. And all the estimations perform better as the sampling rate increases. These algorithms all run fairly fast and can be implemented online, especially compared with the non-parametric estimation. On the other hand, Non-parametric estimates are more accurate and detailed to help understand the traffic structure.

CHAPTER 4

Estimation of traffic characteristics across the network

4.1 Introduction and Literature Review

The work so far has focused on estimating the flow length and byte size distributions on a single link. One can use the proposed methods on a number of links, as in Duffield [12] to obtain a network-wide estimate of traffic characteristics. However, such an approach ignores the fact that routers are linked together and hence a certain number of packets go through a number of routers, resulting in correlated measurements. Further, on a large network comprised of hundreds of routers and thousands of links between them, such a naive approach would not scale well. So, it would be interesting to consider the problem of selecting routers and links to sample traffic from. This would require developing models of how traffic aggregates, such as total number of packets and bytes, are related between neighboring links.

In the literature, Duffield et al[12] study the combination of sampled traffic measurement at multiple observation points to achieve an unbiased estimation of the total traffic. This study is related to our study in the sense of analyzing the entire network, but differs by not allowing for optimizing the sampling design. Moreover, this study was based on the assumption that flow selection is feasible without a one-to-one mapping hash technique, and this is not practical to the best of our knowledge. Meanwhile, [45] addresses the network-wide hash-based flow monitoring problem.

We are going to utilize same ideas from experimental design, briefly discussed

below. In the design of experiment, the factor setting is selected according to some criterion after the model is identified. Two commonly used criterion are the A and D ones.

The *D-optimality* criterion is defined to minimize the volume of the confidence ellipsoid of the estimated coefficients, i.e,

$$\min_d |(X_d^T X_d)^{-1}|,$$

which is also equivalent to

$$\max_d |X_d^T X_d|,$$

where X_d is the corresponding model design.

The *A-optimality* criterion is to minimize the trace of the inverse of the information matrix $(X_d^T X_d)^{-1}$. This results in minimizing the average variance of the coefficients estimates based on a pre-specified model.

4.2 Simple problem

Before we address the sampling design problem for a whole network, we first introduce three simple network operations, which are the basic building blocks for any complex network.

1) **The Two Links Operation**: Consider a network comprised of successive three routers, R_1, R_2 and R_3 connected by two links a and b, as shown in Figure 4.1(a). Packets go from router R_1 through R_2 to R_3, where a certain number of packets are 'lost' on router R_2 with probability p.

2) **The Merging of Flows Operation**: Consider a network comprised of four routers connected by three links as shown in Figure 4.1(c). Packets go from router R_1 and R_2 through R_3 to R_4, where a certain number of packets are 'lost' on router R_3 with probability p.

3) **The Splitting of Flows Operation**: This corresponds to the opposite direction of merging, as shown in Figure 4.1(b), where a certain number of packets are

'lost' on router R_2 with probability p.

The reason of these losses is that some flows follow a different path on the transition router (R_2 in case 1 and 3 and R_3 case 2) to some other destination which is not monitored; or some of the flows terminate at the transition router. Hence, from the perspective of the source the packets of these flows are considered 'lost'. Correspondingly, we use $q \equiv 1 - p$ to denote the independent traversing rate for all the packets. We first assume that q is known, and then briefly discuss the case where q is estimated from sampled measurements.

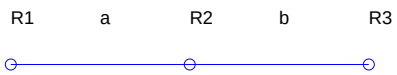
In the rest of section, we will discuss how to allocate samples in these three simple operations, which will be further generalized to the whole network in section 4.4. Furthermore, the following discussion will describe the above simple operations in two systems, a **closed system**, i.e, the topology we observe corresponds to the entire network; and a **partial open system**, where the topology we observe is part of a bigger network. The main difference is whether additional flows from unobserved sources can be present.

4.2.1 Closed System

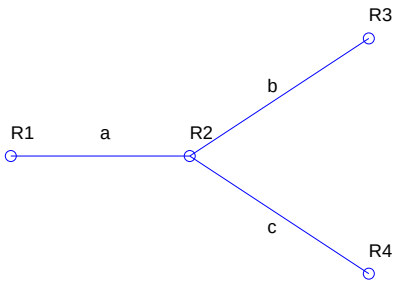
In this system, there is no presence of flows from unknown sources joining the system. The following discussion will focus on the above described three simple network operations.

Case 1: Two Link Network

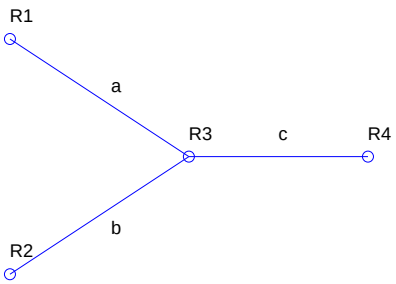
In this case, there are three routers connected by two links, and delivering traffic in one direction from R_1 to R_3. A Simple Random Sampling (SRS) [7] is applied on all the packets in the system in a certain observation period controlling the total sample size of K. Our objective is to allocate samples K_a and K_b on link a and b, respectively, to estimate the total load on each connections, subject to the constraint $K = K_a + K_b$.



(a) Two-Links

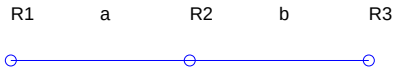


(b) Splitting



(c) Merging

Figure 4.1: Three Simple Operations



We start by defining the quantities of interest. Let

$$X_a = \sum_{i=1}^{N} x_i$$

, $X_b = \sum_{i=1}^{N} x_i I(i \in b) be the load of total bytes on the link a and b, respectively, during a certain measu$ is the size (in byte) of the ith packet entering the system. We assume that packet sizes x_i are i.i.d. from a distribution with mean μ and variance σ^2. Additionally, we assume μ and σ are the same across different flows. We use $I(.)$ to denote an indicator function taking values in $\{0, 1\}$ depending on whether the condition is satisfied or not, e.g., $I(i \in b) = 1$ when the ith packet goes through link b, and 0 otherwise.

Given the SRS mechanism, let $Z = (Z_a, Z_b)'$ be the vector of estimators of $X = (X_a, X_b)'$:

$$Z_a = \frac{\sum_{i \in S_a} x_i}{K_a} N,$$
$$Z_b = \frac{\sum_{i \in S_b} x_i}{K_b} N_b,$$

where S_j denotes the set of sampled packets from link j, $j \in \{a, b\}$. Notice that μ and σ can be estimated unbiasedly from sampled packets as follows

$$\hat{\mu} = \sum x_i I[i \in (S_a \bigcup S_b)]/K \tag{4.1}$$

$$\hat{\sigma} = \sum (x_i I[i \in (S_a \bigcup S_b)] - \hat{\mu})^2/(K-1). \tag{4.2}$$

For now, we do not allow samples on link b to be used to estimate link a, even though in the case of a closed system such aggregation proves useful as discussed later in this section.

This allocation problem is aiming to control both the precision and accuracy of estimation of the total traffic bytes load on each of the observed links. It has already been established by the classical theory on simple random sampling that our estimators Z are unbiased regardless of the selection of $K_a,\ K_b$. On the other hand, we also want the estimators Z to be precise. For example, in the two-link case we can derive the covariance matrix of Z as follows.

$$\begin{aligned}
\mathrm{COV}(Z_a, Z_b) &= \mathrm{COV}[\textstyle\sum_{i=1}^{N} x_i I(i \in S_a), \sum_{j=1}^{N} x_j I(j \in S_b)] \frac{NN_b}{K_a K_b} \\
&= \textstyle\sum_{i=1}^{N} \mathrm{COV}[x_i I(i \in S_a), x_i I(i \in S_b)] \frac{NN_b}{K_a K_b} \\
&= N[\Pr(i \in S_a \bigcup S_b)\mathrm{E}x_i^2 - \Pr(i \in S_a)\Pr(i \in S_b)(\mathrm{E}x_i)^2] \frac{NN_b}{K_a K_b} \\
&= Nq\sigma^2,
\end{aligned}$$

where the second equality comes from the assumption of independent packet bytes distribution. The last equation comes from the fact that

(4.3)

$$\Pr(i \in S_a \bigcup S_b) = \Pr(i \in S_b | S_a)\Pr(i \in S_a) = (q\frac{K_b}{N_b})\frac{K_a}{N} = \frac{K_a K_b}{N_b N} q = \Pr(i \in S_a)\Pr(i \in S_b),$$

where N_b is the total number of packets on link b;

(4.4)
$$\sigma^2 = \mathrm{E}x_i^2 - (\mathrm{E}x_i)^2.$$

Meanwhile, from SRS theory we get

$$\mathrm{Var}(Z_a) = \frac{N(N - K_a)}{K_a}\sigma^2,$$

$$\mathrm{Var}(Z_b) = \frac{N_b(N_b - K_b)}{K_b}\sigma^2.$$

Given all packets pass router R_2 independently with traversing rate q, N_b follows a binomial distribution with parameters the total number of packets N and traversing rate q, and thus can be unbiasedly estimated by $\hat{N}_b = Nq$.

As a result, we can approximate the covariance matrix of Z as

$$\mathrm{COV}(Z) = \begin{bmatrix} \frac{N(N-K_a)}{K_a}\sigma^2 & Nq\sigma^2 \\ Nq\sigma^2 & \frac{Nq(Nq-K_b)}{K_b}\sigma^2 \end{bmatrix}$$

The goal of precision can be achieved by minimizing $\mathrm{COV}(Z)$ as a function of K_a and K_b.

To perform this minimization we borrow two optimization criteria from experimental design, **A-Optimality** and **D-Optimality** as introduced in section 4.1. The modified definition in our case is then as follows:

A-Optimality is to optimize a matrix based on the trace, i.e., sum of the diagonal elements of the matrix.

In our problem, it is to minimize the sum of the variances of the estimates on each link. It ends up suggesting to allocate K_a and K_b in the ratio of $1 : q$, i.e,

$$K_a = \frac{1}{1+q}K, \text{ and } K_b = \frac{q}{1+q}K$$

D-Optimality aims at minimizing the determinant of the matrix. In our problem, this includes the covariance of the measurements on neighboring links, as well as the variances of each measurements. It suggests to allocate

$$K_a = \frac{K - q - \sqrt{((K-q)^2 - (1-q)(K^2 - qK))}}{1-q},$$
$$\text{and } K_b = K - K_a.$$

Both of these two optimal allocations show that the total number of distinct packets N is essentially not necessary, although we assume a prior knowledge on it. More details of the derivation are provided in the Appendix.

The above derived allocations ignore the potential role of samples of link b on the estimation of link a. If we are given that there are no other sources of flows except that of router R_1 (as in this closed system), then we may also use the samples collected on link b in addition to the samples of link a to estimate the total bytes on link a. This procedure of *sample aggregation* includes more samples for analysis, and thus gives more precise results. In the considered closed two-link system, we now have a different estimator of the total loads on link a, combining all the sampled packets on both links a and b, and is given by:

$$\tilde{Z}_a = \frac{\frac{Z_a}{N}K_a + \frac{Z_b}{N_b}K_b}{K_a + K_b}N = \frac{\sum_{i\in S_a} x_i + \sum_{i\in S_b} x_i}{K_a + K_b}N.$$

Updating the estimate of Z_a, we further have

$$\mathrm{VAR}\tilde{Z}_a = \frac{\sigma^2 N^2}{K^2}[K_a(N-K_a)/N + K_b(N_b-K_b)/N_b + 2K_aK_b/N],$$

$$\text{and } \mathrm{COV}(\tilde{Z}_a, Z_b) = \frac{N\sigma^2}{K}[K_a q + N_b - K_b].$$

The resulting covariance matrix is thus updated by

$$\mathrm{COV}(Z) = \begin{bmatrix} \mathrm{Var}\tilde{Z}_a & \mathrm{COV}(\tilde{Z}_a, Z_b) \\ \mathrm{COV}(\tilde{Z}_a, Z_b) & \mathrm{Var}Z_b \end{bmatrix}$$

Detailed derivations of the above equations are given in the Appendix. Next one can move on to optimize updated $\mathrm{COV}(Z)$ by either A- or D-Optimal criterion as discussed in the no sampling aggregation situation.

Case 2: The Splitting Operation

We now consider a closed system, whose flows split on router R_2 (Figure 4.2); i.e., traffic goes in one direction from routers R_1, R_2 to R_3 or R_4, without any unknown traffic joining the system from the outside world. Therefore, we may use samples

collected on links b and c in addition to the samples of link a to estimate the total bytes on link a. This gives

$$\tilde{Z}_a = \frac{\frac{Z_a}{N}K_a + \frac{Z_b}{N_b}K_b + \frac{Z_c}{N_c}K_c}{K_a + K_b + K_c}N,$$

where N_b is estimated by $\hat{N}_b = Nq_b$ and similarly $\hat{N}_c = Nq_c$; and q_b, q_c are the traversing rate from router R_2 to router R_3 and R_4 respectively. Again, we allow possible loss or earlier drop at router 2, and hence $0 < q_b + q_c \leq 1$.

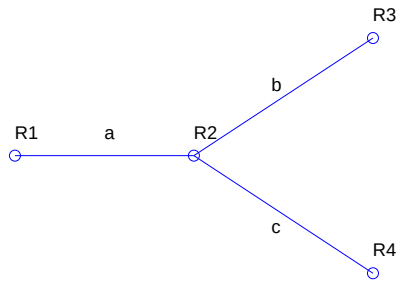


Figure 4.2: The Splitting Operation

The covariance matrix of Z is then derived in the same way as in the two-link operation and given by

$$\mathrm{COV}(Z) = \begin{bmatrix} \mathrm{Var}\tilde{Z}_a & \mathrm{COV}(\tilde{Z}_a, Z_b) & \mathrm{COV}(\tilde{Z}_a, Z_c) \\ \mathrm{COV}(\tilde{Z}_a, Z_b) & \mathrm{Var}Z_b & \mathrm{COV}(Z_b, Z_c) \\ \mathrm{COV}(\tilde{Z}_a, Z_c) & \mathrm{COV}(Z_b, Z_c) & \mathrm{Var}Z_c \end{bmatrix},$$

where

(4.5)

$$\begin{aligned}
\mathrm{Var}(Z_j) &= \tfrac{N_j(N_j-K_j)}{K_j}\sigma^2,\ j=\{b,c\},\\
\hat{\mathrm{VAR}}\tilde{Z}_a &= \tfrac{N}{K^2}[K_a(N-K_a)\sigma^2 + K_b(Nq_b-K_b)/q_b\sigma^2 + K_c(Nq_c-K_c)/q_c\sigma^2\\
&\quad +2K_aK_b\sigma^2 + 2K_aK_c\sigma^2 - 2K_aK_c\mu^2],\\
\hat{\mathrm{COV}}(\tilde{Z}_a, Z_b) &= \tfrac{N}{K}[K_aq_b\sigma^2 + (Nq_b-K_b)\sigma^2 - K_cq_b\mu^2],\\
\hat{\mathrm{COV}}(\tilde{Z}_a, Z_c) &= \tfrac{N}{K}[K_aq_c\sigma^2 + (Nq_c-K_c)\sigma^2 - K_bq_b\mu^2],\\
\mathrm{COV}(Z_b, Z_c) &= -Nq_bq_c\mu^2.
\end{aligned}$$

We then apply the A-Optimality or D-optimality Criteria on the derived matrix to obtain the optimal allocations. A closed form expression of the optimal allocation for K_a,K_b and K_c is not possible to derive. Therefore, numerical results of these two criteria will be given in section 4.3 along with some comments.

Case 3: The Merging Operation

The merging of flows in a closed system is very similar to that of splitting, and differs only by the zero covariance between two adjacent links a and b instead of negative values $-Nq_aq_b$ in splitting. This is because of the independence between the two sources of incoming traffic from R_1 and R_2 (Figure 4.3); while in the splitting operation, the adjacent links share the same traffic source from R_2 (Figure 4.2). Sampling aggregation is still feasible in this Merging Operation by aggregating samples from link a and b on c to estimate total loads on link c. Consequently, covariance matrix for Z is given next.

$$\mathrm{COV}(Z) = \begin{bmatrix} \mathrm{Var}Z_a & 0 & \mathrm{COV}(Z_a, \tilde{Z}_c)\\ 0 & \mathrm{Var}Z_b & \mathrm{COV}(Z_b, \tilde{Z}_c)\\ \mathrm{COV}(Z_a, \tilde{Z}_c) & \mathrm{COV}(Z_b, \tilde{Z}_c) & \mathrm{Var}\tilde{Z}_c \end{bmatrix},$$

where

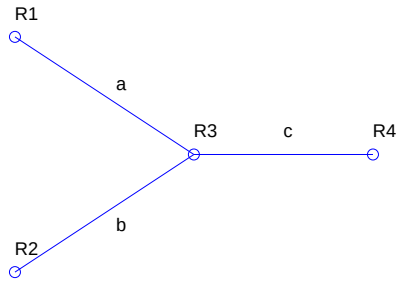


Figure 4.3: The Merging Operation

(4.6)

$$\begin{aligned}
\mathrm{Var}(Z_j) &= \tfrac{N_j(N_j-K_j)}{K_j}\sigma^2,\ j=\{a,b\},\\
\hat{\mathrm{VAR}}\tilde{Z}_c &= \tfrac{N}{K^2}[K_c(N-K_c)\sigma^2 + K_b(Nq_b-K_b)/q_b\sigma^2 + K_a(Nq_a-K_a)/q_a\sigma^2\\
&\quad +2K_cK_b\sigma^2 + 2K_aK_c\sigma^2],\\
\hat{\mathrm{COV}}(\tilde{Z}_c, Z_b) &= \tfrac{N}{K}[K_cq_b\sigma^2 + (Nq_b-K_b)\sigma^2],\\
\hat{\mathrm{COV}}(\tilde{Z}_c, Z_a) &= \tfrac{N}{K}[K_cq_a\sigma^2 + (Nq_a-K_a)\sigma^2].
\end{aligned}$$

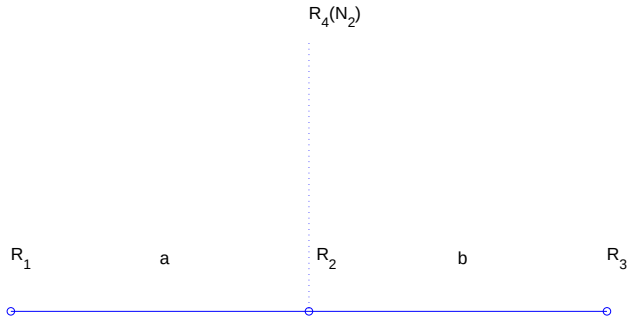
Similarly, closed form expressions for the optimal allocation are not possible to service and numerical results will be given in Section 4.3.

4.2.2 Partially Open System

In a partially open system, there is a presence of flows from an unknown source. We further assume that there is prior knowledge on the new traffic volume relative to the traffic from the observed sources. The following analysis is parallel to that in the closed system, and focuses on the three basic network operations.

Case 1: The Two-Link Network

We have traffic flows from router R_1 through R_2 to R_3. However, there are packets from outside unknown sources, named R_4, coming into the system through



R_2 and R_3. It is assumed that N_2 packets are injected from source R_4 to the system. Packets from both sources are sampled according to Simple Random Sampling with total sample size K. However, the sampling procedure only takes place on links a and b. Again, a certain number of packets are 'lost' on router R_2 with probability p, i.e., traversing rate $q = 1 - p$. The reason of this loss is again that some flows follow a different path from router R_2 to some other destination which is not monitored.

Our objective is again to allocate samples K_a and K_b on links a and b to estimate the total load of each connections, subject to the cost constraint of $K = K_a + K_b$. One should also notice that there should be no samples allocated on the unobserved link connecting R_4 to R_2.

Let $X_a = \sum_{i=1}^{N} x_i I(i \in a)$, $X_b = \sum_{i=1}^{N} x_i I(i \in b)$ be the load of total bytes on the links a and b, respectively, in a certain measured period. Notice that now $N = N_1 + N_2$ is the total number of packets during this period including the ones from router R_1 N_1, which is given, and the ones from unknown sources N_2. Again, let $Z = (Z_a, Z_b)'$ be the vector of estimators of $X = (X_a, X_b)'$ according to the classical theory on simple random sampling, where $Z_a = \frac{\sum_{i \in S_a} x_i}{K_a} N_1$, and $Z_b = \frac{\sum_{i \in S_b} x_i}{K_b} N_b$.

The covariance matrix of Z is then given by

$$\text{COV}(Z) = \begin{bmatrix} \frac{N_1(N_1-K_a)}{K_a} & N_1 q \\ N_1 q & \frac{N_b(N_b-K_b)}{K_b} \end{bmatrix} \sigma^2, \tag{4.7}$$

among which N_b can be estimated by $(N_1 + N_1\rho)q$. This ρ is to capture the ratio of N_2 over N_1, and is either given or can be estimated by the ratio of two partitions of K_b, $\hat{\rho} = K_{b2}/K_{b1}$, where K_{b1} is the number of samples on link b and originated from R_1 and K_{b2} is the number of samples on link b and originated from R_4

Given the above covariance matrix, this allocation problem can be modeled as an optimization problem assuming q is given. Under the A-Optimality criterion for COV(Z), we get the target function in the form of

$$\frac{N_1(N_1 - K_a)}{K_a} + \frac{N_1(1+\hat{\rho})q[N_1 q(1+\hat{\rho}) - K_b]}{K_b}.$$

The minimizer is then calculated as

$$K_a = \frac{1}{1+q(1+\hat{\rho})}K, \;\; K_b = \frac{q(1+\hat{\rho})}{1+q(1+\hat{\rho})}K.$$

Under the criteria D-Optimality, the target function is given by

$$\frac{N_1^2(1+\hat{\rho})q(N_1 - K_a)[N_1(1+\hat{\rho})q - K_b]}{K_a K_b} - N_1^2 q^2,$$

Then essentially we need to minimize

$$\frac{(N_1 - K_a)[N_1(1+\hat{\rho})q - K_b]}{K_a K_b}.$$

The optimal solution is given by

$$\hat{K_a} = \frac{(K - q(1+\hat{\rho})) + \sqrt{(K - q(1+\hat{\rho}))^2 - (1 - q(1+\hat{\rho})) \times (K^2 - Kq(1+\hat{\rho}))}}{1 - q(1+\hat{\rho})},$$

$$\text{and } \hat{K_b} = K - \hat{K_a}.$$

From this we can see that the closed system without sample aggregation is essentially one special case of this partial system with $\rho = 0$.

Now let us consider the case where traversing rate q is unknown and needs to be estimated. It is further assumed a constant traversing rate q for all sources. We propose a two-stage-sampling procedure to handle this allocation problem. At the first stage, K_1 samples are allocated to link a, among which K_{1b} samples will be observed on link b. Then, we use these observations to estimate the traversing rate on router R_2 by $\hat{q} = \frac{K_{1b}}{K_1}$. At the second stage, K_a and K_b samples are allocated to links a and b respectively, together with the observation at the first stage, to estimate the total number of bytes. The whole procedure is under the cost restriction of $K_1 + K_a + K_b = K$. In this context, the marginal covariance matrix of Z is estimated as the expectation of the conditional covariance matrix given q; i.e.,

$$\mathrm{COV}(Z) = \begin{bmatrix} \frac{N_1(N_1-K_a)}{K_a}\sigma^2 & (N_1-1)\hat{q}\sigma^2 \\ (N_1-1)\hat{q}\sigma^2 & \frac{N_1(1+\hat{\rho})\hat{q}(N_1(1+\hat{\rho})\hat{q}-K_b)}{K_b}\sigma^2 + \frac{N_1^2(1+\hat{\rho})^2\hat{q}(1-\hat{q})}{K_1}\mu^2 \end{bmatrix}.$$

Detailed derivation is in the Appendix.

The A-Optimality and D-Optimality criteria can then be used to obtain the optimal allocations.

Case 2: The Splitting Operation in a Partially Open Network

Now we consider the splitting operation in a partially open network, where packets from router R_1 go through R_2 to R_3 and R_4. Further, there are packets from outside unknown sources, named R_5, that go through R_2 to R_3 and R_4. Packets from both sources are sampled by Simple Random Sampling with sample size K on links a, b and c only.

Our objective is to allocate samples K_a, K_b and K_c on link a, b and c, respectively, to estimate the total load of each connections on this partially open network, subject to the cost constraint of $K = K_a + K_b + K_c$. Again, there should be no samples allocated on the unobserved link.

As before, let $X_a = \sum_{i=1}^{N} x_i I(i \in a)$, $X_b = \sum_{i=1}^{N} x_i I(i \in b)$, $X_c = \sum_{i=1}^{N} x_i I(i \in c)$

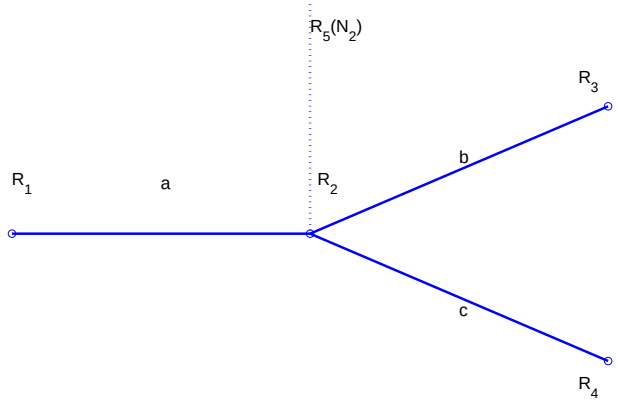


Figure 4.4: Splitting Operation in an Open System

be the load of total bytes on the link a, b and c, respectively, in a certain measured period with N being the total number of packets coming from both sources, N_1 from R_1, which is given, and N_2 from unobserved source during this period. Let $Z = (Z_a, Z_b, Z_c)'$, where $Z_j = \frac{\sum_{i \in S_j} x_i}{K_j} N_j$, $j \in \{a, b, c\}$. We assume that the proportions of the packets going through R_2 to link b and c, q_b and q_c are given. Otherwise, they can be estimated by another stage of sampling as in the two-link case. Note that $q_b + q_c \leq 1$, thus allowing for loss on router R_2.

The COV(Z) is symmetric with upper diagonal elements given next.

$$\begin{bmatrix} \frac{N_1(N_1-K_a)}{K_a}\sigma^2 & N_1 q_b \sigma^2 & N_1 q_c \sigma^2 \\ & \frac{N_b(N_b-K_b)}{K_b}\sigma^2 & -N_1(1+\rho) q_b q_c \mu^2 \\ & & \frac{N_c(N_c-K_c)}{K_c}\sigma^2 \end{bmatrix},$$

among which N_b and N_c are the total number of packets on link b and c respectively and can be estimated by $(N_1 + N_1\rho)q_b$ and $(N_1 + N_1\rho)q_c$. And ρ is to capture the ratio of the total loads N_2 from unknown source R_5 over observed traffic N_1.

The corresponding optimization problem again uses A-Optimality and D-Optimality criterion on the covariance matrix of Z as before. The optimal sample allocations

under A-Optimality are given by $K_a : K_b : K_c = 1 : (1+\rho)q_b : (1+\rho)q_c$.

For the D-Optimality criterion, the target function is the determinant of the derived covariance matrix. Numerical results show that the optimal allocation is approximately uniform; distribute even number of samples on all the links. This can be explained by the flat shape of the target functions shown for a particular setting in Figure 4.5.

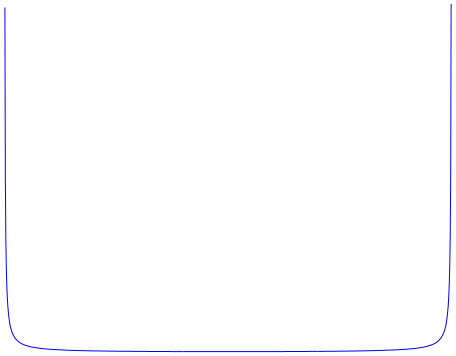


Figure 4.5: Convexity of D-Optimality in a two-link Open System with horizontal dimension as allocation on one link increasing from 0 to K, and vertical dimension as the target function value. The approximately flat bottom shows that optimal allocation is approximately the same as uniform allocation, which corresponds to the horizontal center

Case 3: The Merging Operation in a Partially Open Network

The only difference between merging and splitting in an open system is that the correlation of the two adjacent splitting branches a and b (Figure 4.3) is 0 in merging, which is not true for b and c in splitting (Figure 4.4). In an open network, this does not impact the A-Optimal allocation but only the D-Optimal allocation, because the A-Optimality criterion only considers the variance terms. The impact on the D-Optimal allocation is even quite small because of the small covariance between Z_b and Z_c compared with other terms in the covariance matrix in the splitting case, and it gives approximately the same optima as in the splitting case. Thereby, in practice we can treat merging and splitting identically in a partial open system especially for large networks.

The above analysis on a partially open system assumes that there is prior knowledge on the new traffic volume, relative to the traffic from the observed sources. If this is not the case, a two stage sampling can always be performed to set aside certain number of samples in order to estimate this quantity, continued by a similar analysis as we performed here.

4.3 Experimental Results

We next evaluate the optimal allocation obtained for all these basic network operations under both A-Optimality and D-Optimality criterion using simulated data. Packet sizes $x_i,\ i = 1, ..., 50000$ are simulated following a Normal distribution with mean 800, and standard deviation 10. The total sample size is controlled to be 500 which is equivalent to a 1% sampling rate.

For comparison purposes, we define an *Efficiency* metric as the ratio of the target function values under the optimal allocation over the naive allocation, where naive allocation is taken to be the uniform allocation. For example, in the two-link network given q, the naive allocation is defined as $K_a = K_b = K/2$; while in the two-link one without knowledge of q, it is taken to be $K_1 = K_a = K_b = K/3$.

In the two-link network case, we generate traffic comprised of 50000 packets from R_1, and the traversing rate q is set to 0.8. The resulting allocation in both closed and open systems (with $\rho = 1$) are listed in Table 4.1. There are some differences in the allocation when the ratio of unknown traffic to the given traffic ρ varies, but the impact is rather small and negligible in this example. One may also observe that the D-Optimal allocation is approximately equivalent to the naive allocation in the partially open system.

Table 4.1: Allocation for 2link case.

	CLOSE		OPEN	
	A-Optim	D-Optim	A-Optim	D-Optim
K_a	495	1	192	253
K_b	5	499	308	247

For the splitting case, traffic of 50000 packets is generated at R_1, and the traversing rates to R_3 and R_4 are set to 0.8 and 0.2, respectively. As indicated in table 4.2, both criteria place the majority of the samples on link b in the closed system, which supports the intuition of sampling more on the busy link. However, in the open system, the naive allocation is optimal under the D-Optimality criterion. The A-Optimality criterion in the open system supports allocating more samples on b but not as many as in the closed system. Table 4.3 contains the allocation results for the merging operation. It exhibits a similar pattern as in the split case, as we expected.

Table 4.2: Allocation for Split Operation.

	CLOSE		OPEN	
	A-Optim	D-Optim	A-Optim	D-Optim
K_a	1	1	167	166
K_b	352	490	266	167
K_c	147	4	67	167

Table 4.3: Allocation for Merge Operation.

	CLOSE		OPEN	
	A-Optim	D-Optim	A-Optim	D-Optim
K_a	1	1	167	166
K_b	399	474	266	167
K_c	100	25	67	167

4.4 Extension to Larger Networks

Previously we have discussed the allocation design of samples in all the basic operations of a network. In this section, we will generalize this result to a more realistic scenario; i.e., to larger networks whose operations can be decomposed to the basic ones. In this section, discussion will be again focused on closed and open systems. We further assume that the traversing rate is given, otherwise, one can always apply a two-stage analysis as discussed in Section 4.2.2. We also assume the packet bytes distribution is constant across all the flows, and for simplicity purposes that flow traffic is always in one given direction.

We adopt the following terminologies to describe the network topology. We will

use *root link* to describe the link whose connected end traffic router splits following the traffic direction, and *branch links* to describe the succeeding split links; further this succeeding relation will be called *father-son* relationship, and the two parallel neighboring links connected by the same root router will be called *brother links*. Finally, *end links* connect the terminal destination router on one side. The traffic matrix is defined as a $n \times n$ matrix with entries $(i, j) = 0$ meaning no traffic goes from node i to j, and 1 indicating the existence of traffic from node i to j, where n is the total number of routers in the system. We assume the number of packets sourcing from the observed routers are given but may not be the case for the ones from unobserved sources. In this study we assume one-direction traffic only, which simplifies the problem and guarantees the one-to-one mapping between the traffic matrix and traffic in the given topology network.

4.4.1 Closed System

In a closed system, we assume again there is no outside traffic present, and therefore samples from branch links can be used as part of the information for the root links. We generalize the result from the simple network operations, and obtain covariance measurements for any network of given topology and traffic direction.

For a given closed network with n links, unbiased estimator of load on each link $X = (X_1, ..., X_n)'$ from SRS $Z = (Z_1, ..., Z_n)'$ takes value

$$Z_j = \frac{\sum_{i \in S_j} x_i}{K_j} N_j$$

for the end links. Considering sample aggregation, for the father link 0 with m succeeding son links, where $m \geq 1$,

$$\tilde{Z}_0 = \frac{\sum_{i \in S_0} x_i + \sum_{j=1}^{m} \sum_{i \in S_j} x_i}{K_0 + \sum_{j=1}^{m} K_j} N.$$

N is the total number of packets under study and is given, and S_j is the sample collection on link j. Covariance matrix of estimator Z is a $n \times n$ matrix, where the

diagonal entries are variances and takes values generalized from the simple operations as follows.

(i) Any given end branch link j has a variance in the form of classical simple random sampling theory, i.e,

$$v_j = \frac{N_j(N_j - K_j)}{K_j}\sigma^2.$$

(ii) The variance of any non-ending branch link 0 with m splitting son links, namely $i = 1, ..., m$, can be approximated by

$$\sum_{i=0}^{m} \frac{K_i^2}{(\sum_{j=0}^{m} K_j)^2 q_i^2} v_i,\ m \geq 1.$$

The off-diagonal entries in the covariance matrix of Z is given next.

(i) Any two brother links i, j, sharing the same root router 0 in a splitting case, have covariance in the form of

$$C_{ij} = -N_0 q_i q_j \mu^2.$$

(ii) And for any father-link 0 with m son-links has the covariances

$$C_{0j} = \sum_{i=0}^{m} \frac{K_i}{(\sum_{\ell=0}^{m} K_\ell) q_i} C_{ij},\ j = 1, ..., m,\ m \geq 1.$$

We consider only correlation of the directly connected father-son or brother links, i.e, one-degree correlation. Numerical result supports the negligibility of high-degree correlations. This is because the correlation between two links decay fast as the number of intermediate routers increases, and this holds especially for large networks.

In the following example (see Figure 4.6), we consider a tree topology network comprised of 5 links. The source generates $N = 50000$ packets and the traversing rates are $q_2 = 0.8$ from link 0 to 2 and $q_3 = 0.8$ from link 2 to 3, and correspondingly $q_1 = 0.2$ from link 0 to 1 and $q_4 = 0.2$ from link 2 to 4. Suppose that $K = 500$ packets used to be allocated. Our analysis gives an A-Optimal Allocation: $K_0 = 10, K_1 = 190, K_2 = 90, K_3 = 170, K_4 = 40$, while for D-Optimal Allocation we get:

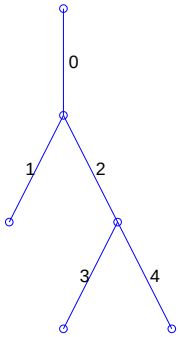


Figure 4.6: Tree Topology-Close System

$K_0 = 430, K_1 = 40, K_2 = 10, K_3 = 10, K_4 = 10$. The different allocations of these two criterion are due to the fact that D-Optimal counts on the correlation of the estimations on the links in addition to the variances of each estimation.

4.4.2 Open Systems

In an open system, there is a presence of outside flows joining the system. This results in a different covariance matrix because samples from branch links can not be aggregated to be used for the root links as in the closed system. We again assume the total number of packets sourcing from observed routers is given, while the ones from outside sources may or may not be given.

Unbiased estimator of total load on each link Z takes values

$$Z_j = \frac{\sum_{i \in S_j} x_i}{K_j} N_j.$$

The covariance matrix of Z for any open network of given topology has diagonal entries

$$v_j = \frac{N_j(N_j - K_j)}{K_j} \sigma^2,$$

and off-diagonal entries

$$C_{ij} = N_i q_j \sigma^2$$

Table 4.4: A-Optimal Allocation Effeciency

	ρ_a	
ρ_b	5	0.2
5	0.4367	0.5176
0.2	0.6788	0.7812

for the father-son links and

$$-N_0 q_i q_j \mu^2$$

for the brother links, where N_0 reflects the number of packets on the connected root router, and N_j's are estimated by traffic volume on the observed source link and the ratio of the total loads from the unknown source over observed traffic as well as traversing rates on each router.

In this system, we have that A-Optimality allocates samples as follows: $K_i : K_j = N_i : N_j$. This is easily interpreted as the heavier traffic needs more samples. Meanwhile, D-Optimality follows the naive uniform allocation due to the lower correlation and a few number of zero appearing in the covariance matrix.

In the following example, we show how our A-Optimal allocation gains efficiency over the naive uniform allocation. This system has $N = 50000$ packets entering the system from the top of the tree passing through the end. Let the traversing rate from link 0 to 2 be $q_2 = 0.8$ and 2 to 3 with passing rate $q_3 = 0.8$. Correspondingly $q_1 = 0.2$, $q_4 = 0.2$, although we allow for loss on the connecting routers in general. The unknown coming inflows have traffic volume given by a ratio $\rho_a = N_2/N$, and $\rho_b = N_3/N$. Remember that we have defined the efficiency as the ratio of the trace of covariance matrix under A-Optimal allocation and the trace of covariance matrix under uniform allocation (naive baseline). Obviously the smaller the value of efficiency the better our allocation performs. Table 4.4 gives the efficiency of A-Optimal allocations under various setting of parameters ρ_a and ρ_b.

From this exercise, we see that when $\rho_a = \rho_b = 5$, we gain 100% in efficiency using the A-Optimal allocation compared with a naive uniform allocation; and when

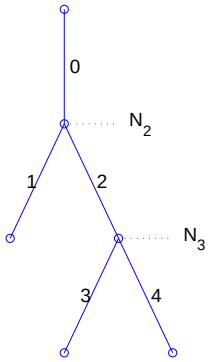


Figure 4.7: Tree Topology-Open System

$\rho_a = \rho_b = 0.2$, A-Optimal allocation is about one and half times efficient. In addition, we see that the ratio of parameters q_3 and q_4 does not change the efficiency or the remaining allocation, but only the allocation on the ending links of 3 and 4. Further, as ρ_a increases, A-Optimality allocates less samples on link 0; and similarly, more samples should be allocated on link 3 and 4 when ρ_b increases.

We also examine this method on Netflow data collected on the GEANT backbone network [50, 59]. GEANT is a European research network across 30 countries and 26 national networks, and it contains 23 nodes and 38 links. The traffic matrix we used covers a 15-minute period starting from 13:00 January 2, 2005 and traffic from the TOTEM project. This origin-destination traffic matrix skips the information of all the transition routers. In order to recover all of the traffic paths, we apply Floyd's shortest path algorithm [33], and assume that they correspond to the real paths. Traffic is then aggregated on each router for all the passing traffic. Additionally, the original data is built upon bi-directional flows, and thus needs to be adjusted to one-direction to test for our allocation method. This adjustment is done by allowing only lower indexed routers to transmit to a higher indexed router, i.e, keep traffic from router i to j if and only if (1) there is traffic in between and (2) $i < j$. Another

assumption we make is that bytes per packet follow the same distribution across the entire network, and more strictly, the ratio of flow size and flow length is a constant. This assumption is built to overcome the problem that packet level information in the original data is not available.

After all these adjustments, we have a 23-node connected one-direction open system. The allocation obtained from A-optimality results in a 0.131387 efficiency parameter, i.e, more than 6 times more efficient than uniform allocation.

4.5 Sensitivity Analysis

In this section we examine in more detail the efficiency issue. We defined the efficiency as the ratio of the target function values given an optimal allocation over the naive allocation, where naive allocation is taken to be the uniform one. From this definition, the smaller value of the efficiency parameter, the more we gain from the optimal allocation. We would also want to study how sensitive this efficiency is by tuning other parameters, i.e, sample size K; traffic size measured by the total number of packets N; and topology which will be discussed in two scenarios, the total number of links n, and traffic distribution controlled by ρ and q. The specific form of Efficiency on a closed system depends on the topology because of sample aggregation. The remaining efficiency sensitivity analysis will focus on the open system.

For the A-Optimal allocation, the target function is the summation of variances of estimated total bytes on each link. Hence, the efficiency is given by

$$Eff_A = \frac{\sum \frac{N_i(N_i - K_i^A)}{K_i^A}}{\sum \frac{N_i(N_i - K/n)}{K/n}},$$

where N_i is decided by the topology and traffic volume from both internal and external sources. For D-optimal allocation, efficiency takes the general form of

$$Eff_D = \frac{\det(\text{covariance matrix applying D-Optimal allocation})}{\det(\text{covariance matrix applying Uniform allocation})},$$

and the detailed input varies upon the topology as previously discussed. We have further seen that D-optimal allocation is approximately the same as uniform allocation in the open system, which then yields an efficiency of approximately 1. The remaining discussion will focus on A-Optimal allocation for open systems.

We first study the sensitivity of efficiency with respect to sample size. This is necessary when there is a budget change; for example, one has to reduce the number of samples overall when facing a budget shrinkage. This action will cause an increase in the efficiency parameter, which is in fact a sign of decrease in the efficiency of A-Optimal allocation.

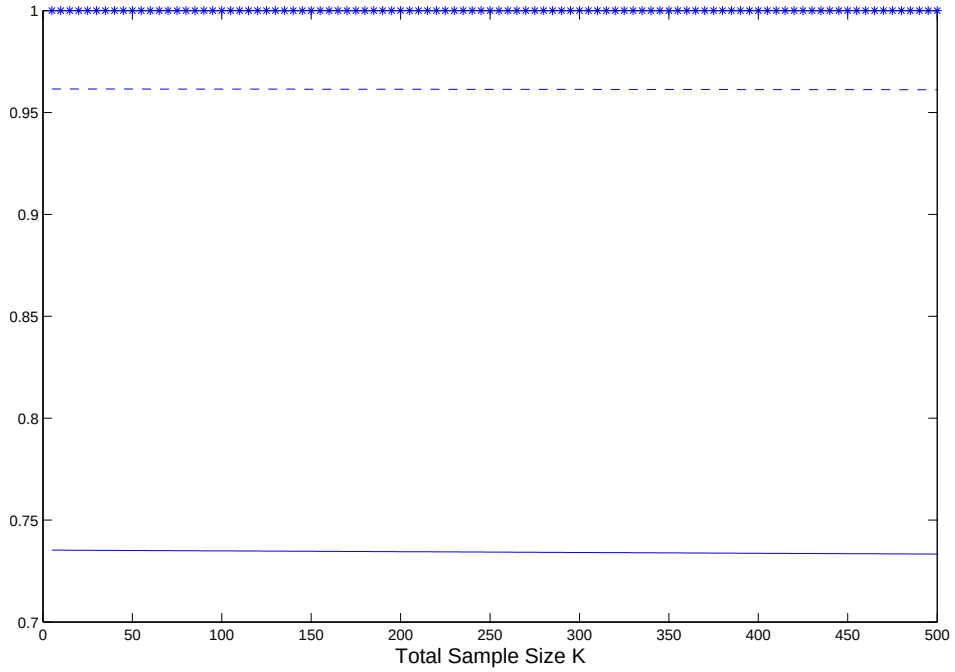
For A-Optimal allocation in an open system, we have $K_i^A : K_j^A = N_i : N_j$ where i, j denotes index for links in the system, and therefore we have the expression that $K_i = A_i K$, where $A_i = N_i / \sum_i N_i$. Then,

$$log(Eff_A) = log(\sum \frac{N_i(N_i - A_i K)}{A_i K}) - log(\sum \frac{N_i(N_i - K/n)}{K/n})$$

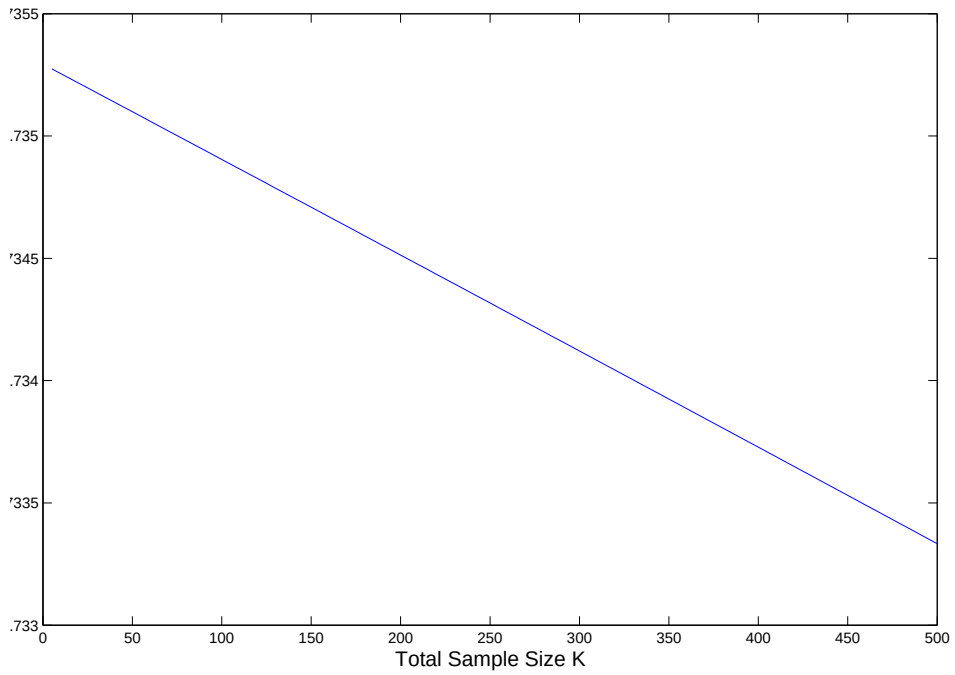
$$\frac{d(log(Eff_A))}{dK} = \frac{-\frac{\sum N_i^2}{A_i}\frac{1}{K^2}}{\sum \frac{N_i(N_i - A_i K)}{A_i K}} - \frac{-n\sum N_i^2 \frac{1}{K^2}}{\sum \frac{N_i(N_i - K/n)}{K/n}}$$

We can see that this first order derivative is always negative, and also because of the 1-1 mapping of log transform, Eff_A monotonically decreases as K increases. In Figure 4.8(a), efficiency changes under three different traversing rates are illustrated, while the total traffic is fixed. It shows better efficiency for A-Optimal allocation over the uniform one with small traversing rate in this particular example, while more detailed analysis on the efficiency sensitivity as the traversing rate changes will be studied later in this section. Figure 4.8(b) is a close look for the 25 % percent traversing rate scenario, showing a clear decreasing trend following the above theoretical result. The horizontal star line in the first plot is an extreme case with $q = 1$ in an open system. Not surprisingly, it is 1 because of the equivalence of A-Optimal allocation to Uniform allocation with $q = 1$ in an open system.

From the above sensitivity analysis of efficiency on K, we come to a similar



(a) Three different traversing rate: solid line for 25% , dash line for 2/3, and star line for 100%



(b) Zoom-In efficiency plot for 25% passing rate

Figure 4.8: Sensitivity of Efficiency as Sample Size Changes

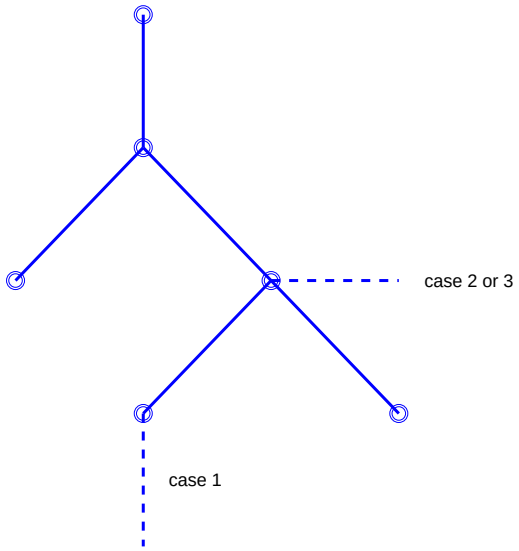


Figure 4.9: Three scenarios when introducing a new link

conclusion on N. An increase in N while holding other parameters constant is equivalent to a decrease in K while holding other parameters constant, and vice versa.

The following discussion will show a positive impact of the total number of links n on efficiency, which means the more complex the system, the more the gains in efficiency. For simplicity, we only demonstrate the topology change from n to $n+1$, i.e. introducing a new link. There are generally three scenarios when introducing a new link (Figure 4.9): (i) one additional link at the end node; (ii) one additional link on a father node; and (iii) one additional link on a father/end node.

(i) In this case, the new link brought in is located at an end node with some of the traffic going to the new added destination router. The impact of this change is that the total traffic in the observed period becomes $\sum_{i=1}^{n+1} N_i = \sum_{i=1}^{n} N_i + N_{n+1}$. We will keep K_i's fixed by increasing K proportionally to the increase in the total

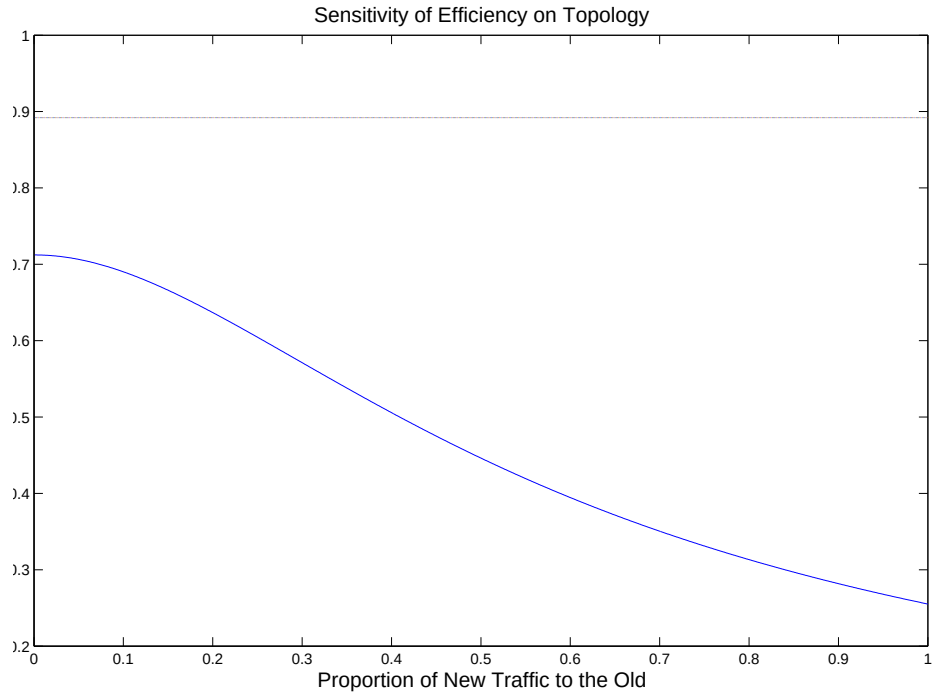


Figure 4.10: First scenario of introducing a new link: at an end node. The upper horizonal straight line demonstrates Eff_A^n as a reference, and the lower curve is Eff_A^{n+1}.

traffic, i.e,

$$K' = \frac{\sum_{i=1}^{n+1} N_i}{\sum_{i=1}^{n} N_i} K.$$

Now $K_{n+1}^A : K_i^A = N_{n+1} : N_i$. The updated efficiency of the A-Optimal allocation is then given by

$$Eff_A^{n+1} = \frac{\sum_{i=1}^{n} \frac{N_i(N_i - K_i^A)}{K_i^A} + \frac{N_{n+1}(N_{n+1} - K_{n+1}^A)}{K_{n+1}^A}}{\sum_{i=1}^{n} \frac{N_i(N_i - K/n+1)}{K/n+1} + \frac{N_{n+1}(N_{n+1} - K/n+1)}{K/n+1}},$$

which is less than Eff_A^n as shown in Figure 4.10, where the upper horizonal straight line demonstrates Eff_A^n as a reference, and the lower curve is Eff_A^{n+1}. One extreme case is $N_{n+1} = 0$ or close to 0; then, this can be easily interpreted because of the dilution of traffic in uniform allocation, while A-Optimal allocation stays the same as that of the n links case. From this plot, we can also tell that as the proportion of N_{n+1} to $\sum_{i=1}^{n} N_i$ increases, the efficiency parameter decreases, showing an increasing efficiencies under A-Optimality.

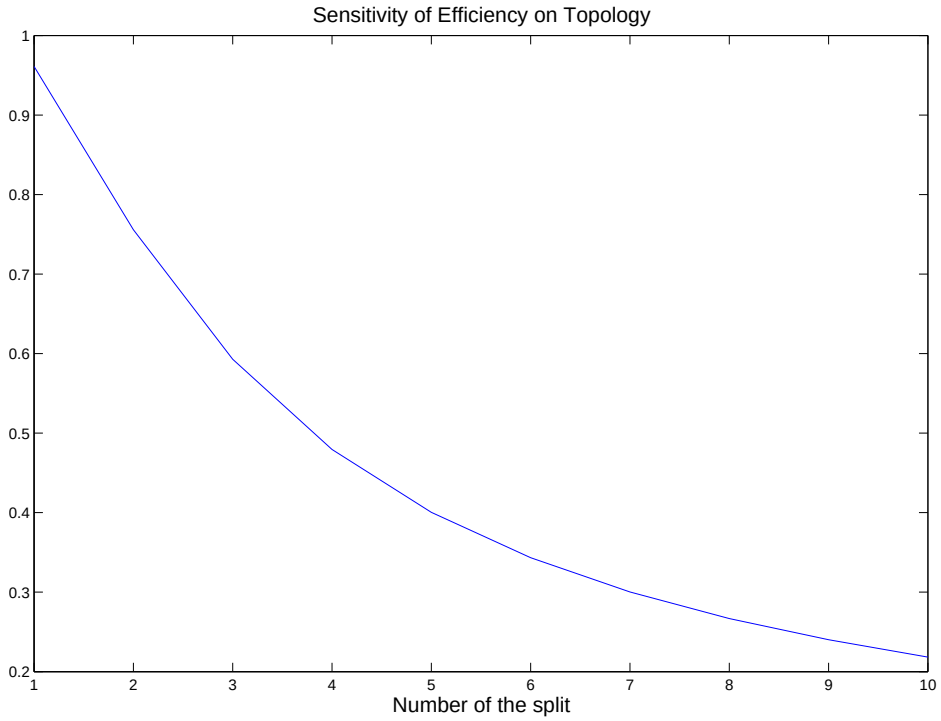


Figure 4.11: Second scenario of introducing a new link: at a father node.

(ii) In the case where the new link is located at a father node, all the other links stay the same except for its brothers. The total traffic volume is unaffected, but the traffic on the brother links is reduced by the amount added to the new link i.e, N_{n+1}. Correspondingly, we keep the total sample size unchanged. All the K_i's will remain unaffected but $K'_j < K_j$ for $j \in \{$S: index for the affected brother links$\}$. The updated efficiency of A-Optimal allocation is then given by

$$Eff_A^{n+1} = \frac{\sum_{i \notin S \cup \{n+1\}} \frac{N_i(N_i - K_i^A)}{K_i^A} + \sum_{j \in S} \frac{N'_j(N'_j - K'^A_j)}{K'^A_j} + \frac{N_{n+1}(N_{n+1} - K^A_{n+1})}{K^A_{n+1}}}{\sum_{i \notin S \cup \{n+1\}} \frac{N_i(N_i - K/n+1)}{K/n+1} + \sum_{j \in S \cup \{n+1\}} \frac{N'_j(N'_j - K/n+1)}{K/n+1}},$$

and this is smaller than Eff_A^n with sufficiently large n. Further, more split (increasing n all on one father node) gives better efficiency from an A-Optimal allocation, as indicated in Figure 4.11.

(iii) The last scenario of increasing n is by adding a new source of traffic, which may occur anywhere in the system. If it introduces traffic directly to an end node, then essentially it is equivalent to case (i). We now discuss the situation when it

brings new traffic N_{n+1} to a father of an end node, which can be easily generalized to any other nodes. The impact of this new traffic is in the succeeding links, denoted by S, and all the others stay the same. For all $i \in S$,

$$N_i' = N_i + \frac{N_i}{\sum_{i\in S} N_i} N_{n+1}.$$

We will increase the sample size K proportionally to the increase on the total traffic volume, in order to keep $K_i' = K_i$ unaffected for $i \notin S$. And then for $i \in S$, $K_i' = K_i + \frac{N_i}{\sum_{i\in S} N_i} K_{n+1}$. The updated efficiency of A-Optimal allocation is then given by

$$Eff_A^{n+1} = \frac{\sum_{i\notin S\cup\{n+1\}} \frac{N_i(N_i - K_i^A)}{K_i^A} + \sum_{j\in S} \frac{N_j'(N_j' - K_j'^A)}{K_j'^A} + \frac{N_{n+1}(N_{n+1} - K_{n+1}^A)}{K_{n+1}^A}}{\sum_{i\notin S\cup\{n+1\}} \frac{N_i(N_i - K/n+1)}{K/n+1} + \sum_{j\in S\cup\{n+1\}} \frac{N_j'(N_j' - K/n+1)}{K/n+1}},$$

and this is smaller than Eff_A^n as indicated in Figure 4.12. Again, the upper horizontal line is the reference of efficiency when n links and the lower curve is for efficiency when $n+1$ links. From this plot, we can also see that as the proportion of N_{n+1} to $\sum_{i=1}^n N_i$ increases, the efficiency parameter increases a little because of the slower increase of uniform allocation on total variance magnitude compared with the A-Optimal one as traffic increases; it is followed by a continuous decrease, because of the increasing speed of the growing variance under a uniform allocation as a function of traffic volume.

We examine next the sensitivity analysis of efficiency with respect to traversing rates and the unknown source traffic volume ratios. Since they all depend on the specific topology of a system, we will explore it by showing an experimental study, whose topology is shown in Figure 4.13. This system has $N = 50000$ packets entering the system from the top of the tree passing through the end, among which we select $K = 500$. We let traversing rate from node 0 to 2 q_2, 2 to 3 q_3 and the unknown source traffic volume ratios ρ_a and ρ_b to vary. The resulting efficiency under various parameters settings is shown in the following table 4.5.

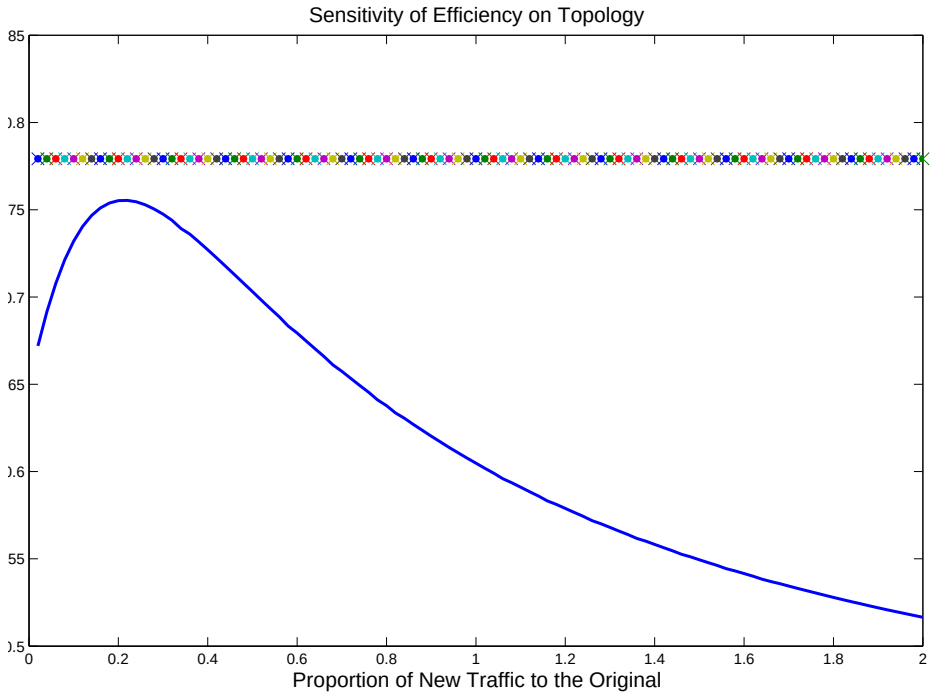


Figure 4.12: Third scenario of introducing a new link: adding a new source of traffic: the upper horizontal line is the reference of efficiency when n links and the lower curve is for efficiency when $n+1$ links.

Table 4.5: A-Optimal Allocation Efficiency Sensitivity Study on Topology

ρ_a	ρ_b	q_2	q_3	q_2	q_3	q_2	q_3	q_2	q_3
		0.8	0.8	0.8	0.2	0.2	0.8	0.2	0.2
5	5	0.4367		0.4367		0.6649		0.6649	
0.2	5	0.5176		0.5176		0.7820		0.7820	
5	0.2	0.6788		0.6788		0.5301		0.5301	
0.2	0.2	0.7812		0.7812		0.6080		0.6080	

From this example, we see that as ρ_a increases, A-Optimality allocates less samples on link 0 and achieves more efficiency; and similarly, more samples should be selected on link 3 and 4 when ρ_b increases, however the efficiency does not show a clear trend with ρ_b nor q_2. In addition, we see that the parameters q_3 does not change the efficiency or the rest allocation but only the allocation on the connected two end links.

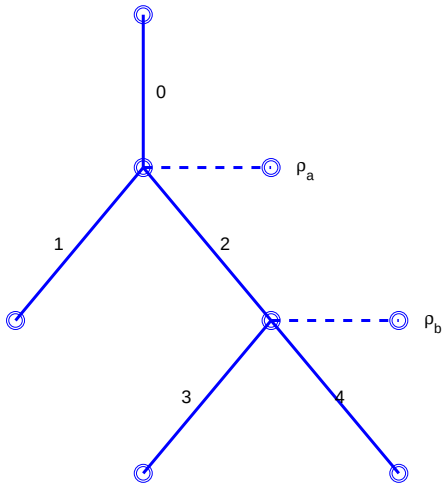


Figure 4.13: Example of Efficiency sensitivity on the network topology.

CHAPTER 5

Future Work

5.1 h-Run Sampling

As the sampling capabilities of routers improve, it may be useful to consider alternative sampling schemes. In the literature, several approaches that target longer flows have been proposed. Estan and Varghese [21] proposed a *sample and hold* approach, as well as *multistage filters*, which takes a constant number of memory references per packet and uses a small amount of memory. A cache lookup is performed for the key of each incoming packet, updated for the existing key but with a certain probability for the new key. A similar idea appeared in [29] RATE, where sampling two-runs automatically biases the samples towards the larger flows. A recent progress on this sampling scheme design issue under hard resource constraint is achieved by Cohen et al paper [8] and [9], where a number of new sampling designs extend the *Netflow* and *sample and hold* approach by allowing sampling rate changes. Sampling may also be applied to completed flow records. Duffield, Lund and Thorup [14, 15, 18] proposed a threshold sampling in this scenario where longer flows are more heavily sampled, since they exceed a predetermined threshold. In [15] and Baek-Young Choi et al 2004 paper [3] extend this idea to the situation where sampling rate needs to be controlled.

All the above studies in the literature focus on detecting long flows, yet nothing has been done to capture short flows. However, the recovery on the short flows is

crucial for estimating the entire flow length/size distribution and for estimating the total number of flows. In this section, we briefly discuss one possibility that is to combine Bernoulli and systematic sampling.

It works as follows: packets are sampled using Bernoulli sampling with probability p^* independent of any other characteristics. A flow information look-up is performed on each selected packet. If it is selected, we then also select the next succeeding $h-1$ packets, thus forming an h-block of packets. The block parameter h is selected so that on average the desired sampling rate p is achieved. This means that there is a tradeoff between p^* and h, briefly illustrated below. This scheme is designed to relieve the length bias issues associated with heavy tailed flow length distributions, if the packets from all the flows are 'properly' mixed.

Some Preliminary Results

We start from a general scenario, where all the packets are mixed well, ignoring which flow they are from. We use $I_k(i) = 1$ to indicate that the ith packet is the kth one in an h-block selected, where $k \in [1, h]$ and i takes value from 1,...,N, and N is the total number of packets. Q will be used as an indicator in the following way, $Q(i) = 1$ if the ith packet is in the sample and 0 if not. The probability that the ith packet is selected is then given by

(5.1)

$$\begin{aligned} p = P[Q(i)=1] &= P[I_1(i)=1] + P[I_2(i)=1] + ... + P[I_h(i)=1] \\ &- \textstyle\sum_{k\neq\ell} P[I_k(i) = I_\ell(i) = 1] + \sum_{k_1\neq k_2\neq k_3} P[I_{k_1}(i) = I_{k_2}(i) = I_{k_3}(i) = 1] \\ &... \quad +(-1)^{h-1} P[I_1(i) = I_2(i) = ... = I_h(i) = 1] \\ &= hp^* - \tfrac{h(h-1)}{2} p^{*2} + \tfrac{h(h-1)(h-2)}{3!} p^{*3} + ... + (-1)^{h-1} p^{*h} \\ &= \textstyle\sum_{k=1}^{h} (-1) C_h^k p^{*k} \end{aligned}$$

The reason for the overlapping in this equation comes from the fact that the selection procedure is stage by stage. When a packet is selected on the second stage,

Table 5.1: h-run sampling rate vs first stage selecting rate

h	2	3	5	10	20
$p = \alpha p^*$	1.99	2.97	4.9	9.56	18.21

it has a slight chance of already being selected in the first stage, and this will also impact the succeeding $h-1$ packets as well. Table 5.1 shows how the sampling rate p is related to the fist stage selecting rate p^*. We see the coefficient α is close to, but always less than h.

Our h-run sampling does not outperform Bernoulli sampling for capturing shorter flows in this well mixed situation. This is because packets are sampled unbiasedly from different flows as in Bernoulli sampling. Packets from shorter flows do not draw additional attention as we suppose to. The next few modification would make it particularly suited for the case where packets cluster according to their type. The updated version of this sampling design is that after the first stage selection, if the selected packet is from a short flow we then select the next succeeding $h-1$ packets, thus forming an h-block of packets; otherwise, skip this second step for the long flows. This updated design turns more attention to the short flows. The ideal scenario for this sampling design is when we know there are two types of flows, for example, short UDP flows and long TCP flows; in addition, flow of the same type tends to be highly clustered. In such a situation, more packets from smaller flows would be captured, thus improving the estimate of the number of active flows M, plus the estimate of the left tail of the distribution.

Suppose all of the $h-1$ second stage samples are from the short flows, indicating no ending packets from any short-flow clusters is selected at the first stage. In this case, the probability of a packet from any long flows being sampled is obviously p^*, and h^*p^* for that from short flows. The corresponding probability for a short flow containing N_s packets being observed is $1-(1-h^*p^*)^{N_S}$; and $1-(1-p^*)^{N_L}$ for a long flow containing N_L packets. Now this procedure balances the sampling rates

for long and short flows, and thus reduce the bias towards longer flows.

Now let us generalize it by breaking the assumption that the entire $h-1$ second stage samples are from the short flows. In this case, the probability of any packet from long flows selected at the second stage increases to $p_{L2} = \sum_{k=1}^{h-1} p^*(1-p^*)^{k-1}\frac{kM_L}{\sum_{i=1}^{M_L} N_{Li}}$, where M_L is the total number of long flows, and N_{Li} is ith long flow's length. When $h=2$, this reduces to $p^*\frac{M_L}{\sum_{i=1}^{M_L} N_{Li}}$, which comes from the possibility that every first packet in the long flow clusters is selected at the second stage, and is essentially the possibility that the last packet in the short flow arrays is sampled in the first stage. Correspondingly, the probability of a packet from short flows selected at the second stage is reduced by $p_{S2} = \sum_{k=1}^{h-1} p^*(1-p^*)^{k-1}\frac{kM_L}{\sum_{i=1}^{M_S} N_{Si} - \text{No. of packets from short flows selected at 1st stage}}$, which can be approximated by its expectation $p^*(1-p^*)^{h-1}\sum_{i=1}^{M_S} N_{Si}$. Now packet sampling rate for long flows is $p_L = p^* + p_{L2}$, and $p_S = h^*p^* - p_{S2}$ for short flows. The corresponding probability for a short flow containing N_s packets being observed is $1-(1-p_S)^{N_S}$; and $1-(1-p_L)^{N_L}$ for a long flow containing N_L packets. Since $p_S > p_L$, this design still successfully balances the sampling rate and reduce the biasness towards long flows. It is also worth noting that $p_S \to h^*p^*$ as $h \to \infty$, converging to the best scenario we discussed above.

Future work will focus on the numerical study of this analysis, as well as finding the tradeoff between mixture structure and the unbiasness gained from this sampling design.

APPENDIX

APPENDIX A

Appendix

Another look at the problem in Chapter 2:

Let $F_i = \sum \mathcal{I}_{m=1}^{M}(N_m = i)$ be the number of flows of length i. From our hierarchical mechanism $\{F_i\}_{i=1}^{N}$ is a random vector. However, one may want to treat it as parameters. Then the generated M flows with lengths $\{F_i\}_{i=1}^{N}$ are the entire population, and parameters M and $\{F_i\}_{i=1}^{N}$ are the ones to be estimated. Notice that the total number of active flows is still given by $M = \sum_{i=1}^{N} F_i$. Further, we still have $T = \sum_{i=1}^{N} iF_i$ and $t = \sum_{j=1}^{N} jg_j$ as before, with t following a Binomial distribution with parameters T and p.

However, in this setting, each flow does not have the same probability of being of length i and having j of its packets sampled, which was the case under the hierarchical mechanism (recall that $\pi_{ij} = \phi_i \pi_{j|i}$). Instead, each flow of length i has a different probability of having j of its packets sampled, i.e., sample length j given flow length i follows a Binomial distribution with parameter (i, p).

In this case, the joint mass function of $G = \{G_0, ..., G_N\}'$ is not multinomial given parameter $M = \sum F_i$, but rather a convolution of N different multinomial vectors, i.e.,

$$G = \sum_{i=1}^{N} \xi_i, \text{ such that } \sum_{i=1}^{N} F_{ij} = G_j, \text{ for all } j = 0, 1, ..., N, \tag{A.1}$$

where $\xi_i = (F_{i0}, ..., F_{iN})'$ follows a multinomial distribution with parameters $(F_i,$

$\{\pi_{j|i}\}_{j=0}^{N}$), i.e.,

$$\text{Prob}(\xi_i = (f_{i0}, ..., f_{iN})') = \begin{pmatrix} F_i \\ f_{i0}...f_{iN} \end{pmatrix} \prod_{j=0}^{N} \pi_{j|i}^{f_{ij}}$$

For example, there are $M = 3$ flows under study with true flow lengths $F_1 = 2,\ F_2 = 1$. Now $i = 1, 2,\ j = 0, 1, 2$. After Bernoulli sampling, observations are $\{G_1 = 1, G_2 = 1\}$. Further $G_0 = \sum F_i - G_1 - G_2 = \sum F_i - 2$. Then the joint mass

$$\begin{aligned} \text{P}\{G_0, ..., G_N\} &= \textstyle\sum_{(f_{10}+f_{20}=\sum F_i-2)} (\text{P}(\xi_1 = (f_{10}, 0, 0)')\text{P}(\xi_2 = (f_{20}, 1, 1)')) \\ &+ \textstyle\sum_{(f_{10}+f_{20}=\sum F_i-2)} (\text{P}(\xi_1 = (f_{10}, 1, 0)')\text{P}(\xi_2 = (f_{20}, 0, 1)')). \end{aligned}$$

Further, the likelihood for the complete set of data in this situation is given by a product of multinomials below:

$$\begin{aligned} &\text{Prob}(F_{10} = f_{10}, ..., F_{ij} = f_{ij}, ..., F_{NN} = f_{NN}) \\ &= \begin{pmatrix} F_1 \\ f_{10}...f_{1N} \end{pmatrix} \textstyle\prod_{j=0}^{N} \pi_{j|1}^{f_{1j}} ... \begin{pmatrix} F_N \\ f_{N0}...f_{NN} \end{pmatrix} \prod_{j=0}^{N} \pi_{j|N}^{f_{Nj}} \\ &= \textstyle\prod_{i=1}^{N} \begin{pmatrix} F_i \\ f_{i0}...f_{iN} \end{pmatrix} \prod_{j=0}^{N} \pi_{j|i}^{f_{ij}}. \end{aligned} \tag{A.2}$$

Now let us connect this to our previous result derived directly from hierarchical mechanism by free out parameters $\{F_i\}_{i=1}^{N}$ and let it be a random vector. When $\{F_i\}_{i=1}^{N}$ becomes a random vector, $M = \sum F_i$ is also a random variable. Then this procedure is essentially the same as our hierarchical mechanism. The two probability function we derived in the latter situation becomes conditional probability given $(\{F_i\}_{i=1}^{N}, M)$, i.e.,

$$G|(\{F_i\}_{i=1}^{N}, M) = \sum_{i=1}^{N} \xi_i, \text{ such that } \sum_i F_{ij} = G_j \text{ for all } j = 0, 1, ...N, \tag{A.3}$$

$$\begin{aligned} &\text{Prob}(F_{10} = f_{10}, ..., F_{ij} = f_{ij}, ..., F_{NN} = f_{NN}|(\{F_i\}_{i=1}^{N}, M)) \\ &= \textstyle\prod_{i=1}^{N} \begin{pmatrix} F_i \\ f_{i0}...f_{iN} \end{pmatrix} \prod_{j=0}^{N} \pi_{j|i}^{f_{ij}}. \end{aligned} \tag{A.4}$$

We show next that this approach in fact reaches the same conclusion with the one we discussed earlier.

Likelihood from the complete data:

$$
\begin{aligned}
& \text{Prob}(F_{01}, ..., F_{NN}|M) \\
= \; & \textstyle\sum_{(F_i:\sum F_i=M)} \text{Prob}(F_{01}, ..., F_{NN}|(\{F_i\}_{i=1}^N, M))\text{Prob}(\{F_i\}_{i=1}^N|M) \\
= \; & \text{Prob}(F_{01}, ..., F_{NN}|(\{F_i\}_{i=1}^N, M))\text{Prob}(\{F_i\}_{i=1}^N|M) \\
= \; & [\textstyle\prod_{i=1}^N \begin{pmatrix} F_i \\ f_{i0}...f_{iN} \end{pmatrix} \prod_{j=0}^N \pi_{j|i}^{f_{ij}}] \begin{pmatrix} M \\ F_1, ..., F_N \end{pmatrix} \prod_{k=1}^N \phi_k^{F_k} \\
= \; & \begin{pmatrix} M \\ f_{01}, ..., F_{NN} \end{pmatrix} \textstyle\prod_{i=1}^N \prod_{j=0}^N \pi_{j|i}^{f_{ij}} \prod_{k=1}^N \phi_k^{F_k} \\
= \; & \begin{pmatrix} M \\ f_{01}, ..., F_{NN} \end{pmatrix} \textstyle\prod_{i=1}^N \prod_{j=0}^N (\phi_i \pi_{j|i})^{f_{ij}}
\end{aligned}
\tag{A.5}
$$

which ends up the same as the result from direct derivation (2.10). The second equality comes from the inherent constraints $\sum_{j=0}^N F_{ij} = F_i$ and $\sum_{i=1}^N \sum_{j=0}^N F_{ij} = M$. And the third equality is from the fact that $F_{i1}, ..., F_{iN}|F_i \sim$ Multinomial distribution; and so is $\{F_i\}_{i=1}^N|M$.

Similarly, likelihood from the observed data:

(A.6)

$$
\begin{aligned}
& \text{Prob}(G_0 = g_0, ..., G_j = g_j, ..., G_N = g_N | M) \\
= & \sum_{(F_i:\sum F_i = M)} \text{Prob}(G_0 = g_0, ..., G_j = g_j, ..., G_N = g_N | (\{F_i\}_{i=1}^N, M)) \text{Prob}(\{F_i\}_{i=1}^N | M) \\
= & \sum_{(F_i:\sum F_i = M)} \text{Prob}(G_0 = g_0, ..., G_j = g_j, ..., G_N = g_N | (\{F_i\}_{i=1}^N, M)) \text{Prob}(\{F_i\}_{i=1}^N | M) \\
= & \sum_{(F_i:\sum F_i = M)} [\sum_{(\sum_i f_{ij} = G_j)} \prod_{i=1}^N \left(\begin{array}{c} F_i \\ f_{i0} ... f_{iN} \end{array} \right) \prod_{j=0}^N \pi_{j|i}^{f_{ij}}] \left(\begin{array}{c} M \\ F_1, ..., F_N \end{array} \right) \prod_{k=1}^N \phi_k^{F_k} \\
= & \sum_{\{(f_{ij}:\sum_i f_{ij} = G_j, \sum_j f_{ij} = F_i) \text{and} (F_i:\sum F_i = M)\}} \\
& [\prod_{i=1}^N \left(\begin{array}{c} F_i \\ f_{i0} ... f_{iN} \end{array} \right) \prod_{j=0}^N \pi_{j|i}^{f_{ij}} \left(\begin{array}{c} M \\ F_1, ..., F_N \end{array} \right) \prod_{k=1}^N \phi_k^{F_k}] \\
= & \sum \prod_{i=1}^N \prod_{j=0}^N \pi_{j|i}^{f_{ij}} \prod_{k=1}^N \phi_k^{F_k} \left(\begin{array}{c} M \\ f_{01}, ..., f_{NN} \end{array} \right) \\
= & \sum \left(\begin{array}{c} M \\ f_{01}, ..., f_{NN} \end{array} \right) \prod_{i=1}^N \prod_{j=0}^N (\pi_{j|i} \phi_i)^{f_{ij}} \\
= & \left(\begin{array}{c} M \\ G_0, ..., G_N \end{array} \right) \sum \prod_{j=0}^N \left(\begin{array}{c} G_j \\ f_{0j}, ..., f_{Nj} \end{array} \right) \prod_{i=1}^N \prod_{j=0}^N (\pi_{j|i} \phi_i)^{f_{ij}} \\
= & \left(\begin{array}{c} M \\ G_0, ..., G_N \end{array} \right) \sum \prod_{j=0}^N [\left(\begin{array}{c} G_j \\ f_{0j}, ..., f_{Nj} \end{array} \right) \prod_{i=1}^N (\pi_{j|i} \phi_i)^{f_{ij}}] \\
= & \left(\begin{array}{c} M \\ G_0, ..., G_N \end{array} \right) \sum \prod_{j=0}^N [\left(\begin{array}{c} G_j \\ f_{0j}, ..., f_{Nj} \end{array} \right) \prod_{i=1}^N (\frac{\pi_{j|i} \phi_i}{\sum_{i=1}^N \pi_{j|i} \phi_i})^{f_{ij}}] \\
& \times [\prod_{j=0}^N \prod_{i=1}^N (\sum_{i=1}^N \pi_{j|i} \phi_i)^{f_{ij}}] \\
= & \left(\begin{array}{c} M \\ G_0, ..., G_N \end{array} \right) [\prod_{j=0}^N \prod_{i=1}^N (\sum_{i=1}^N \pi_{j|i} \phi_i)^{f_{ij}}] \\
= & \left(\begin{array}{c} M \\ G_0, ..., G_N \end{array} \right) [\prod_{j=0}^N (\sum_{i=1}^N \pi_{j|i} \phi_i)^{G_j}]
\end{aligned}
$$

where the last three equalities come from the fact

$$\frac{\pi_{j|i} \phi_i}{\sum_{i=1}^N \pi_{j|i} \phi_i} = \pi_{i|j}$$

and

$$\sum_{\{(f_{ij}:\sum_i f_{ij}=G_j,\sum_j f_{ij}=F_i)\text{and}(F_i:\sum F_i=M)\}}\prod_{j=0}^{N}[\left(\begin{array}{c} G_j \\ f_{0j},...,f_{Nj} \end{array}\right)\prod_{i=1}^{N}(\frac{\pi_{j|i}\phi_i}{\sum_{i=1}^{N}\pi_{j|i}\phi_i})^{f_{ij}}]=1.$$

This also gives the same result as from direct derivation (2.7).

Section 4.2 formulates the sample allocation into optimization problems. For the two-link open system, the target function to minimize under A-Optimal Criterion is

$$\frac{N_1(N_1 - K_a)}{K_a} + \frac{N_1(1+\rho)q(N_1q(1+\rho) - K_b)}{K_b}.$$

This is equivalent to minimize

$$N_1/K_a + N_1q^2(1+\rho)^2/K_b,$$

and more generally equivalent to minimize

$$N_a/K_a + N_b/K_b,$$

subject to the constraint $K_a + K_b = K$. By method of Lagrange multipliers, this optimization with constraint can be solved by minimize $L = (N_a/K_a + N_b/K_b) - \lambda(K - K_a - K_b)$. Take derivative of L with respect to K_a, K_b and λ and set to 0. We then solve a system of equations comprised of these three equations, and achieve

$$K_a = \frac{N_a}{N_a + N_b}K = \frac{1}{1 + (1+\rho)q}K, \text{ and } K_b = \frac{N_b}{N_a + N_b}K = \frac{(1+\rho)q}{1 + (1+\rho)q}K.$$

We can further generalize this result to any open system with m links has A-Optimal allocation

$$K_1 : K_2 : ... : K_m = N_1 : N_2 : ... : N_m,$$

and the distribution of N_i's are controlled by the network topology.

Now let us consider the same two-link open network, then D-Optimal criterion set the target function as

$$\frac{N_1^2(1+\rho)q(N_1 - K_a)[N_1q(1+\rho) - K_b]}{K_aK_b} - N_1^2q,$$

which is equivalent to minimizing

$$\frac{(N_1 - K_a)[N_1q(1+\rho) - K_b]}{K_aK_b},$$

subject to $K_a + K_b = K$. Applying method of Lagrange multipliers, we get

$$K_a = \frac{K - q - \sqrt{((K-q)^2 - (1-q)(K^2 - qK))}}{1-q}.$$

$$\text{and } K_b = K - K_a.$$

In a close system of two links, $\tilde{Z}_a = \frac{\frac{Z_a}{N}K_a + \frac{Z_b}{N_b}K_b}{K_a+K_b}N$. The followings are detailed derivation of the corresponding covariance matrix.

(A.7)

$$\begin{aligned}
\mathrm{VAR}\tilde{Z}_a &= \tfrac{1}{K^2}[K_a^2\mathrm{VAR}Z_a + K_b^2/q^2\mathrm{VAR}Z_b + 2K_aK_b/q\mathrm{COV}(Z_a, Z_b)] \\
&= \tfrac{1}{K^2}[K_a^2(N-K_a)N/K_a\sigma^2 + K_b^2/q^2(Nq-K_b)Nq/K_b\sigma^2 + 2K_aK_b/q(N-1)q\sigma^2 \\
&= \tfrac{\sigma^2N^2}{K^2}[K_a(N-K_a)/N + K_b(N_b-K_b)/N_b + 2K_aK_b/N].
\end{aligned}$$

(A.8)

$$\begin{aligned}
\mathrm{COV}(\tilde{Z}_a, Z_b) &= \tfrac{K_a}{K}\mathrm{COV}(Z_a, Z_b) + \tfrac{K_b}{Kq}\mathrm{VAR}(Z_b) \\
&= \tfrac{K_a}{K}Nq\sigma^2 + \tfrac{K_b}{Kq}\tfrac{Nq(Nq-K_b)}{K_b}\sigma^2 \\
&= \tfrac{N\sigma^2}{K}[K_aq + N_b - K_b].
\end{aligned}$$

Section 4.2.2 also introduced a two-stage sampling procedure for the case of passing rate q unknown. In the first stage, q is estimated by $\frac{K_{1b}}{K_1}$, the ratio of the sampled packets seen on the second link to the sample size on the first link. Since K_{1b} follows Binomial(K_1, q), then $\mathrm{VAR}(K_{1b}) = K_1 q(1-q)$, and therefore we have $\hat{\mathrm{VAR}}(\hat{q}) = \frac{\hat{q}(1-\hat{q})}{K_1}$.

$\mathrm{VAR}(Z_a) = \frac{N_1(N_1-K_a)}{K_a}\sigma^2$ stays the same, while $\mathrm{VAR}(Z_b)$ changes and can be derived as following.

$$\mathrm{VAR}(Z_b) = \mathrm{E}_q\mathrm{VAR}(Z_b|q) + \mathrm{VAR}_q\mathrm{E}(Z_b|q),$$

where $\mathrm{VAR}_q\mathrm{E}(Z_b|q) = \mathrm{VAR}_q(\mu N(1+\rho)q) = \mu^2\frac{N^2(1+r)^2\hat{q}(1-\hat{q})}{K_1}$ because of the independence between flow length distribution and packet size distribution;

$$\mathrm{E}_q\mathrm{VAR}(Z_b|q) = \frac{N(1+\rho)\hat{q}[N(1+\rho)\hat{q} - K_b]}{K_b}\sigma^2.$$

Finally, we achieve

$$\mathrm{COV}(Z) = \begin{bmatrix} \frac{N_1(N_1-K_a)}{K_a}\sigma^2 & (N_1-1)\hat{q}\sigma^2 \\ (N_1-1)\hat{q}\sigma^2 & \frac{N_1(1+\hat{\rho})\hat{q}(N_1(1+\hat{\rho})\hat{q}-K_b)}{K_b}\sigma^2 + \frac{N_1^2(1+\hat{\rho})^2\hat{q}(1-\hat{q})}{K_1}\mu^2 \end{bmatrix}.$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Cantieni, Gion Reto, Iannaccone, Gianluca, Diot, Christophe, Thiran, Patrick (2004), Reformulating the Monitor Placement Problem: Optimal Network-Wide Sampling, *In proceeding of CoNEXT, 2006.*

[2] Chaudet, Claude, Fleury, Eric, Lassous, Isabelle Guerlin, Rivano, Herve, and Voge, Marie-Emilie (2005), Optimal Positioning of Active and Passive Monitoring Devices, *In proceeding of CoNEXT, 2005.*

[3] Choi,Baek-Young, Park, Jaesung, Zhang,Zhi-Li(2004),Adaptive Packet Sampling for Accurate and Scalable Flow Measurement, *Global Telecommunications Conference. GLOBECOM '04. IEEE.*

[4] Choi,Baek-Young and Bhattacharyya,Supratik (2005), On the Accuracy and Overhead of Cisco Sampled Netflow, *Sigmetrics Workshop on Large Scale Network Inference (LSNI).*

[5] Cisco NetFlow

[6] Claffy, K.C., Polyzos, G.C. and Braun, H.W. (1993), Application of Sampling Methodologies to Network Traffic Characterization, *Proceedings ACM SIGCOMM*, 13-17.

[7] Cochran, W.G. (1977), *Sampling Techniques*, 3rd edition, John Wiley, New York, NY

[8] Cohen, Edith, Duffield, Nick, Kaplan, Haim, Lund, Carsten, and Thorup, Mikkel (2007), Algorithms and Estimators for Accurate Summarization of Internet Traffic, *IMC'07, October 24-26, 2007, san Diego, California, USA.*

[9] Cohen, Edith, Duffield, Nick, Kaplan, Haim, Lund, Carsten, and Thorup, Mikkel (2007), Algorithms and Estimators for Accurate Summarization of Internet Traffic, *PODS'07, June 11-13, 2007, Beijing, China.*

[10] De Leeuw, J. (1994), Block Relaxation Algorithms in Statistics, in *Information Systems and Data Analysis*, Bock et al. (eds)., Springer, Heidelberg

[11] Diebolt, J. and Robert, C. (1994), Estimation of Finite Mixture Distributions through Bayesian Sampling, *Journal of the Royal Statistical Society, B*, 56, 363-375

[12] Duffield, Nick, Lund, Carsten, and Thorup, Mikkel (2005), Optimal Combination of Sampled Network Measurements, *IMC'05, October 19–21, 2005, New Orleans, LA.*

[13] Duffield, N.G., Lund, C. and Thorup, M. (2005), Estimating Flow Distributions from Sampled Flow Statistics, *IEEE/ACM Transactions on Networking*, 13, 325-336

[14] Duffield, N.G., Lund, C. and Thorup, M. (2003), Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastruction, *Proc ACM SIGCOMM nternet Measurement Conference 179-191.*

[15] Duffield, N.G., Lund, C. and Thorup, M. (2005) Learn More, Sample Less: Control of volume and variance in network measurement, *IEEE Transactions on Information Theory.*

[16] Duffield, N.G., Lund, C. and Thorup, M. (2002) Properties and Prediction of Flow Statistics from Sampled Packet Streams, *Proc. ACM SIGCOMM Internet Measurement Workshop 159-171.*

[17] Duffield, N. (2004), Sampling for Passive Internet Measurement: A Review, *Statistical Science*, 19, 472-498.

[18] Duffield, N.G., Lund, C. and Thorup, M. (2001),Charging from Sampled Network Usage, *ACM SIGCOMM Internet Measurement Workshop 2001.*

[19] Duffield, N.G., Lund, C. and Thorup, M. (2004), Flow Sampling Under Hard Resource Constraints, *ACM SIGMETRICS/Performance'04* New York, NY.

[20] Duffield, N.G. and Grossglauser, M., Trajectory Sampling for Direct Traffic Observation.

[21] Estan, C. and Varghese, G. (2003), New Directions in Traffic Measurement and Accounting: Focusing on the elephants, ignoring the mice, *ACM Transactions on Computer Systems*, 21 270-313.

[22] Faraway, J. (2005), *Extending the Linear Model with R*, CRC Press.

[23] Hohn, M. and Veitch, D. (2003), Inverting Sampled Traffic, *Proceedings of Internet Measurement Conference.*

[24] Internet Protocol Flow Information Export, IETF Working Group

[25] Inmon Corporation, *SFlow accuracy and billing.*

[26] Kamiyama, T. (2005), Identifying High-rate Flows with Less Memory, *Proceedings IEEE Infocom*, 2781-2785

[27] Keener, Robert W, *Statistical Theory: A Medley of Core Topics*, in preparation.

[28] Kompella, Ramana Rao, Estan, Cristian (2005), The Power of Slicing in Internet Flow Measurement, *IMC'05, Proceeding of the 5th ACM/USENIX Internet Measurement Conference, October 2006.*

[29] Kodialam,M., Lakshman,T.V. and Mohanty, S. (2004), Runs Based Traffic Estimator(RATE): A simple, memory efficient scheme for per-flow rate estimation, *Proc. IEEE INFOCOM 2003 3 1808-1818.*

[30] Kumar, Abhishek, Sung, Minho, Xu, Jun, Wang, Jia (2004), Data Streaming Algorithms for Efficient and Accurate Estimation of Flow Size Distribution, *SIGMETRICS/Performance'04, June 12-16, 2004, New York, NY.*

[31] Kumar, Abhishek, Sung, Minho, Xu, Jun, Zegura, Ellen (2005), A Data Streaming Algorithm for Estimating Subpopulation Flow Size Distribution, *SIGMETRICS'05, June 6-10, 2005, Banff, Alberta, Canada.*

[32] Kumar, Ritesh and Kaur, Jasleen (2004), Efficien Beacon Placement for Network Tomography, *IMC'04, October 25-27, 2004, Taormina, Sicily, Italy.*

[33] Kumar, V., Grama, A., Gupta, A. and Karypis, G. (1994), *Introduction to Parallel Programming - Design and Analysis of Algorithms*, Benjamin Cummings Publishing.

[34] Lawrence, E, Michailidis, G., Nair, Vijay, Xi, Bowei(2006), Network Tomography: A Review and Recent Developments, *Frontiers in Statistics*

[35] Louis, T.A. (1982), Finding the Observed Information Matrix When Using the EM Algorithm', *Journal of the Royal Statistical Society, Series B*, 44, 226-233.

[36] McLachlan, G.J. and Krishnan, T. (1997), The EM algorithm and Extensions, John Wiley and Sons, Inc.

[37] Mori, T., Uchida, M. and Kawahara, R. (2004), Identifying Elephant Flows Through Periodically Sampled Packets, *Proceedings ACM SIGCOMM*, 115-120.

[38] Ribeiro, Bruno, Towsley, Don, Ye, Tao, Bolot, Jean (2006), Fisher Information of Sampled Packets:an Application to Flow Size Estimation, *IMC'06, October 25–227, 2006, Rio de Janeiro, Brazil.*

[39] Rice, John A. (1994) *Mathematical Statistics and Data Analysis*, Duxbury Press.

[40] Ross, Sheldon M. (2003), *introduction to Probability Models*, Academic Press.

[41] Roughan Matthew (2006), A Comparison of Poisson and Uniform Sampling for Active Measurements, to appear in *IEEE Journal on Selected Areas in Communications*

[42] Sarndal, Carl-Erik, Swensson, Bengt, and Wretman, Jan (1991), *Model Assisted Survey Sampling*, Springer-Verlag

[43] SAVAGE, S., WETHERALL, D., KARLIN, A. and ANDERSON, T. (2000). Practical network support for IP traceback. *In Proc. ACM SIGCOMM* 2000 295-306.

[44] Scott,David, *Multivariate Density Estimation : Theory, Practice, and Visualization*, Wiley

[45] Sekar, Vyas, Reiter, Michael, Willinger, Walter, Zhang, Hui (2007) Coordinated Sampling: An Efficient, Network-Wide Approach for Flow Monitoring, *July 16, 2007, Technique Report, School of Computer Science, Carnegie Mellon University* .

[46] Singhal, Harsh, Michailidis, George (2008), Optimal Sampling in State Space Models with Applications to Network Monitoring, *SIGMETRICS08, June 2-6, 2008, Annapolis, Maryland, USA*.

[47] SNOEREN, A. C., PARTRIDGE, C., SANCHEZ, L. A., JONES, C. E., TCHAKOUNTIO, F., KENT, S. T. and STRAYER, W. T. (2001). Hash-based IP traceback. *In Proc. ACM SIGCOMM* 2001 3-14.

[48] Süli,Endre and Mayers David (2003), *An Introduction to Numerical Analysis*, Cambridge University Press.

[49] Suh, Kyoungwon, Guo, Yang, Kurose, Jim, Towsley, Don (2006), Locating Network Monitors: Complexity, Heuristics, and Coverage, *Computer Communications, 2006*.

[50] Uhlig, Steve, Quoitin, Bruno, Balon, Simon and Lepropre, Jean. Providing public intradomain traffic matrices to the research community. *ACM SIGCOMM Computer Communication Review, 36(1), January 2006*.

[51] The ns-simulator, Information Sciences Institute, available at www.isi.edu/nsnam/ns

[52] Vardi, Y (1982), Nonparametric Estimation in the Presence of Length Bias, *The Annals of Statistics*, Vol.10, No.2, 616-620.

[53] Willinger, W., and Paxson, V.(1997), Discussion of 'Heavy Tail Modeling and Teletraffic Data', *The Annals of Statistics*, Vol.25, No.5, 1856-1866.

[54] Wu, C. F. Jeff, Hamada, Michael (2000), *Experiments: Planning, Analysis, and Parameter Design Optimization*, Wiley.

[55] Xing, Song and Paris, Bernd-Peter (2003), Measuring the Size of The Internet via Importance Sampling, *IEEE Journal on Selected Areas in Communications* Vol.21, No.6, 922-933 x

[56] Yang, L and Michailidis, G. (2006), Estimation of Flow Lengths from Sampled Traffic, *Proceedings of Globecom*, San Francisco, CA.

[57] Yang, L and Michailidis, G. (2007), Sampled Based Estimation of Network Traffic Flow Characteristics, *Proceedings of Infocom*, Anchorage, Alaska.

[58] Zacks, S. (1991), *Theory of Statistical Inference*, John Wiley, New York, NY & Sons.

[59] http://www.geant.net/