Enhancing Distributed Traffic Monitoring via Traffic Digest Splitting

LAM, Chi Ho

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
August 2009

Abstract of thesis entitled:

Enhancing Distributed Traffic Monitoring via Traffic Digest Splitting

Submitted by LAM, Chi Ho

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in August 2009

Recently, certain Distributed Traffic Monitoring schemes, DATALITE, Proportional Union Method (PU) and Quasi-Likelihood Approach (QMLE) have been proposed to support general Traffic Measurement and Analysis (TMA) functions for large scale, high-speed packet-switched networks. In these schemes, all traffic flows passing through a link are mapped into a single Traffic Digest (TD) for monitoring and analysis purpose. It has been observed that the relative estimation error caused by such single-TD-per-link approach can be significant for tiny traffic flows (mice) when they are sharing a link with other large volume flows (elephants) as commonly observed in practice.

In this thesis, we propose to enhance these distributed traffic digest schemes by taking an optimal traffic digest splitting strategy: flows sharing the same link are partitioned and mapped into different sub-TDs according to their previously estimated flow volume. By avoiding the mixing of "mice" and "elephant" flows in a single TD, we can significantly reduce the "noise-to-signal" ratio experienced by the former. Moreover, it can be shown that the reduction in such "noise-to-signal" ratio is more than enough to offset the negative effect caused by reduction in TD memory size for each sub-group (since the total memory size required across all sub-TDs is kept to be the same as that of the single-TD approach without splitting). We have derived analytical expressions of the optimal splitting threshold by minimizing the resultant root-mean-square (r.m.s.) (as well as maximum) relative error of the flows sharing a link under various traffic distributions. The extent of estimation improvement of TD-splitting depends on the estimation error characteristics of the TMA scheme and the flow-volume distribution. In most cases, the estimation error can be reduced. In particular, the r.m.s. estimation error of DATALITE can be reduced on average around 50% using single-level TD-splitting and it can be reduced by 80% using 2-level recursive TD-splitting which outperforms the non-splitting version of Quasi-Likelihood Approach by more than 20% in terms of r.m.s. relative estimation error. For PU and QMLE, the r.m.s. relative estimation error can be reduced by 30% using 2-level recursive TD-splitting. Equivalently, the memory size requirement of the TDs of these two schemes can be halved while keeping the same estimation error.

摘要

最近，分佈式流量測量計劃，例如 DATALITE, PROPORTIONAL UNION METHOD (PU) and QUASI- LIKELIHOOD APPROACH (QMLE) 均提出於大型及高速信息包交換網絡中支援一般流量測量和分析（TMA）功能。在這些計劃中，所有流量均通過單一流量摘要（Traffic Digest）以達到監測和分析的目的。我們觀察到當流量小之信息流跟其他流量大之信息流混合的時候，以單一流量摘要測量的話，會造成很大的測量誤差。

在本論文，我們建議以最佳流量摘要分割方法來提高這些分佈式流量測量計劃之測量準確度，如將流量摘要分割，並根據原先估計流量，將共享相同鏈接的信息流分開並將其分派到不同之次流量摘要（sub-TD）。從而避免在單一流量摘要之中同時混合大和小之信息流，大大減少"噪信"比。此外，有證據表明，減少"噪信"比能夠超過並足以抵消因減少每個次流量摘要內存大小所造成的負面影響（因為次流量摘要之總內存大小與單一流量摘要內存大小一樣）。我們以鏈接中所有流動之最少相對誤差之均方根（RMS）或最大值作目標，為不同流量之分佈得出最佳分割值的方程式。流量摘要分割後對測量準確度之改進主要是由於該測量計劃之誤差特性及鏈接中之流量分佈而有不同效果。在大多數情況下，測量誤差均得以減少。特別，在 DATALITE 使用最佳分割方法將單一流量摘要分成

兩個次流量摘要時，其相對誤差均方根值平均可以減少約百分之五十，而使用最佳分割方法分成四個次流量摘要時其相對誤差均方根值更可減少百分之八十。此外，後者之測量誤差更比沒有進行分割之 QMLE 低百分之二十。而 PU 及 QMLE，在使用最佳分割方法分成四個次流量摘要時，其誤差均方根值均相對減少了百分之三十。相對地，以保持同樣的測量誤差，這兩個測量方法所需之流量摘要所需之內存大小可減少百分之五十。

Acknowledgement

First of all, I have to express my gratitude to my supervisor Professor Wing-cheong Lau and co-supervisor Professor On-ching Yue, for their support and guidance in these two years. They gave me the freedom to explore various research areas and provided advices and suggestions which guide me to the right direction. Without their guidance and suggestion, the work could not be so fruitful. I also thank all my new friends in my graduate study, in particular, Wee-lum Tan, Chun-pong Luk, Kin-fai Li, Ho-yan Suen and Wai-kit Sze, for sharing their experience in research. I also wish to thank Ka-chun Lam and Michael Hui, for their encouragement in my final year of graduate study. Finally, I wish to thank my family for their support over the past two years.

This work is dedicated to my family and my friends for their supports.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

With the rapid expansion of high-speed data networks and the proliferation of new protocols and network services, the ability to provide real-time network traffic pattern monitoring and analysis becomes increasingly critical yet challenging. On one hand, there is an urgent demand on monitoring and analyzing the spatial and temporal behavior of traffic flow pattern in such networks. On the other hand, tracking the large number of traffic flows and the huge packet volume in such networks is a daunting task due to the stringent real-time wire-speed computation and large memory requirements. Towards this end, many streaming cardinality estimation algorithms have been proposed recently [7, 8, 10, 15, 16, 20]. They all suggested using Traffic Digest (TD) (or a "sketch") to summarize and track relevant streams of packets and applying probabilistic counting algorithm to estimate the cardinality of individual traffic streams. One of such scheme is DATALITE [20], a Distributed Architecture for Traffic Analysis via LIght-weight Traffic digEst, which performs Traffic Measurement and Analysis functions for large-scale, 10Gbps+ packet-switched networks. Its applications include: Traffic flow pattern/route monitoring, diagnosis and network forensic, Origination-to-Destination (OD) traffic load matrix estimation as well as trace-back on the origin(s) of attacking packets in a distributed denial of service (DDoS) attacks. Moreover, there are two other schemes, Proportional Union Method (PU) [8] and Quasi-Likelihood Approach (QMLE) [7] in stream cardinality. PU applied simple proportional estimation while QMLE applied Maximum Likelihood approach to perform traffic measurement, and both schemes provide more accurate results comparing to DATALITE.

In these schemes, as well as many other existing TD-based TMA schemes [9, 11, 24, 25], all traffic flows passing through a link (or a monitoring point) are mapped into a single TD for monitoring and analysis purpose. It has been observed that the relative estimation error caused by such single-TD-per-link approach can be significant for tiny traffic flows (mice) when they are sharing a link with other large volume flows (elephants). This is because interfering flows sharing the same TD behave like "noise" when one tries to measure the volume of a specific target flow. Unfortunately, the mixing of mice and elephant flows within a high-speed network is quite common given the heavy-tail distribution of traffic flow volumes observed in practice.

In this thesis, we propose to enhance the distributed traffic digest monitoring scheme by taking an optimal TD splitting strategy: flows sharing the same link are partitioned and mapped into different sub-TDs according to their previously estimated flow volumes. By avoiding the mixing of "mice" and "elephant" flows in a single TD, we can significantly reduce the "noise-to-signal" ratio experienced by the former. Moreover, it can be shown that the reduction in such "noise-to-signal" ratio is more than enough to offset the negative effect on estimation error caused by reduction in TD memory size for each sub-group (given the total memory size required across all sub-TDs is kept to be the same as that of the single-TD approach without splitting).

In the later part of this thesis, we demonstrate how we can apply our TD-splitting scheme in network-wide traffic measurements in practice. We also apply the common membership queries data structure, Bloom filter [4, 5], to store the mapping of the OD-pairs to the corresponding TDs. It has been shown that our proposed TD-splitting scheme can significantly reduce the estimation error of these TD-based TMA schemes to measure the network-wide traffic volume.

1.2 Organization

The thesis is organized as follows. In Chapter 2, we review several recent related works and TD-based TMA schemes which we are going to study in the thesis, including DATALITE [20], PU[8] and QMLE[7]. We first describe the traffic digest generation procedure and then describe the estimating procedure for each estimator.

In Chapter 3, we characterize the relationship of the estimation error and the NSR experienced by the target flow at different nodes and link for cardinality estimation of the 2-sets and 3-sets intersection case. Such characterization motivates us to suppress the estimation error by reducing the NSR.

Our TD-splitting scheme is described in Chapter 4. To make the problem analytically tractable, surrogate optimization objective functions are introduced based on the estimation error characterized in Chapter 3 and the local "bottleneck" assumption. We then derive analytical expressions of the optimal splitting threshold (for traffic flow volume) by minimizing the resultant root-mean-square (r.m.s.) (as well as maximum) relative error of the flows sharing a link under various traffic flow volume distributions. Recursive TD-splitting is discussed in the later section of this chapter.

We demonstrate how our scheme can be realized to support network-wide traffic measurement in Chapter 5. For network-wide traffic measurement, we distribute TDs to other nodes/links for flow-volume estimation. After splitting TDs into several sub-TDs, we need to choose the appropriate sub-TD for estimation. Therefore, we propose to apply Bloom filter to track the mapping between OD-pairs and their corresponding sub-TDs.

In Chapter 6, the performance of the proposed TD-splitting scheme is evaluated via comparison to the original DATALITE scheme as well as the PU [8] and QMLE [7] schemes. We also realize our TD-splitting scheme for network measurement in the Abilene Network.

Finally, Chapter 7 discusses the conclusions of our work.

In summary, the contributions of this thesis include:

1. We study the estimation error of several TD-based TMA schemes including DATALITE, PU and QMLE and derive the upper bound of the estimation errors of the OD-flow per link packet count estimation under those schemes. This motivates us to split the traffic digest to provide better estimation. We also extend QMLE to support the cardinality estimation of 3-sets intersection.

2. We propose a TD-splitting algorithm to reduce the estimation error of various TD-based TMA schemes. In particular, we propose an optimal splitting threshold for separating the flows according to the traffic volume and map them into different digests to reduce overall estimation errors. We define two different objectives, (1) minimizing the maximum estimation error; and (2) minimizing the r.m.s. estimation error across all the participating flows. We also realize the TD-splitting scheme for real network by applying Bloom filter [5] for sub-TD membership tracking.

□ **End of chapter.**

Chapter 2

Related Works and Background

2.1 Related Works

Much research has been focused on the problem of using traffic measurement to extract a spatial and temporal view of the traffic flow or routing pattern within a high-speed network [17, 19]. Specific applications include measuring the OD-flow packet (or byte) count to reconstruct a traffic matrix, or measuring the summaries of the network traffic and estimating individual flow volume by different stream cardinality estimator [8, 10, 12]. Ken Keys et al in [17] proposed a robust system to measure the network summary in terms of ports, IP addresses, etc. Abhishek K. et al in [18, 19] introduced a stream algorithm to estimate the flow size distribution with high estimation accuracy. Recently, measurement on the entropy of OD-flows was proposed in [23].

The advantages of the entropy-based approach include the reduction of CPU/memory resource consumption. The entropy of OD-flows also gives significant insight into network traffic dynamics which can serve as a more reliable and effective metric for anomaly detection and network diagnosis. However, most of the existing approaches are limited to estimate the OD-flow traffic matrix information, e.g., [8]. Direct extensions of their approaches to measure the OD-flow per link for every link within a network will lead to substantial increase in estimation errors due to various error aggregation and amplification problems to be illustrated in the next chapter. Nevertheless, much progress has been made for reducing the statistical variance of stream cardinality estimators under stringent memory constraints. In [6], Cai M. et al improved the $O(loglog\ N_{max})$ estimator [10] to provide more accurate estimation for small traffic flows within the network. The QMLE approach in [8] is extended to support per OD-pair byte-counting in [7]. Hyper-Loglog count [13] further improves the accuracy of the original FM Sketch [10] by computing the harmonic mean (instead of the geometric one) across an ensemble of estimates. By focusing on the reduction of the noise-to-signal ratio in each TD via TD-splitting (instead of variance reduction of the cardinality estimator), our approach is complementary to, and can be readily combined with, some of the variance-reduction techniques mentioned above, e.g., [13], to further reduce the overall traffic estimation error.

2.2 Background

In this thesis, our focus would be on studying the enhancement of several TD-based TMA schemes. In this section, we focus on reviewing these schemes, including DATALITE [20], Proportional Union Method (PU) [8] and Quasi-Likelihood Approach (QMLE) [7]. We first give a brief introduction for each scheme and then we describe how the traffic digest (or called "sketch") is generated. And finally, we explain how the scheme be used for network-wide traffic measurement.

2.2.1 DATALITE

DATALITE [20], a Distributed Architecture for Traffic Analysis via LIght-weight Traffic digEst, is a set of distributed algorithms and protocols to support general Traffic Measurement and Analysis (TMA) functions for large scale packet-switched networks. The authors formulate the network-wide TMA problem as a series of set-cardinality determination problems. By applying the probabilistic distinct sample counting technique [10] for set-cardinality, the network-wide traffic measurement can be computed in a distributed manner via the exchange of light-weight traffic digests (TDs) amongst all the nodes and links. In the following section, we give a brief introduction on how DATALITE supports network-wide measurement.

Network-wide traffic measurement and analysis as Set-Cardinality-Determination

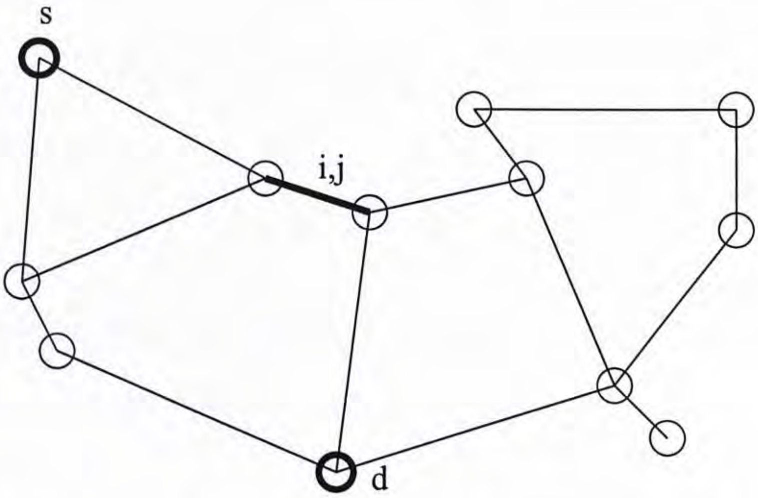


Figure 2.1: Estimating the number of packets originating from node s and terminating at node d passing through link (i, j)

Consider the network shown in Figure 2.1 as an example. Denote the sets of packets originated in node s and destined to node d by S and D respectively. Let L be the set of packets passing through link (i, j) and $k = (s, d)$ be the commodity index which indicates the OD-pair of a set of packets of interest.

Define $F^k_{i,j}$ as the set of distinct packets of commodity k, passing though link (i, j). The volume (in terms of packet count) for the traffic commodity k, passing through link (i, j) is given by:

$$|F^k_{i,j}| = |S \cap L \cap D| \tag{2.1}$$

where $|X|$ represents the cardinality (number of distinct element) of set X.

Based on elementary set theory, $|F^k_{i,j}|$ can be expressed as

$$\begin{aligned} |F^k_{i,j}| &= |S| + |D| + |L| - \\ & |S \cup D| - |D \cup L| - |S \cup L| \\ & +|S \cup D \cup L| \end{aligned} \tag{2.2}$$

In general, the cardinality of the intersection of multiple sets $\mathcal{S}_1$, $\mathcal{S}_2$, ..., $\mathcal{S}_\eta$ can also be computed based on the following expression:

$$\begin{aligned} |\bigcap_{i=1}^{\eta} \mathcal{S}_i| &= \sum_{i=1}^{\eta} |\mathcal{S}_i| - \sum_{i \neq j} |\mathcal{S}_i \cup \mathcal{S}_j| + \sum_{i \neq j \neq k} |\mathcal{S}_i \cup \mathcal{S}_j \cup \mathcal{S}_k| \\ & +... + (-1)^{(\eta-1)} |\bigcup_{i=1}^{\eta} \mathcal{S}_i|, \end{aligned} \tag{2.3}$$

Here, $O(Loglog\ N)$ distinct sample technique [10] is used to determine the cardinality of each of the union-terms in RHS of Equation 2.2.

Traffic Digest Generation for DATALITE

Consider a stream of packets, denoted by packet set $\mathcal{S}$, where each packet $s \in \mathcal{S}$ has a time-invariant, unique identifier PID_s. When the packet passes through a monitoring point p (which can be a link or a node), we use PID_s as input to a uniform hash function $h(\cdot)$ and store the hash output in an array of m buckets which serves as the traffic digest TD_p for this monitoring point. Specifically, we use the first l ($l = log_2 m$) bits of $h(PID_s)$ as the bucket key, and then count the index of the first occurrence of 1 in $h(PID_s)$ starting from the $(l+1)$-th bit in $h(PID_s)$ and store the value to the corresponding bucket. Each bucket stores the maximum value of the index of the first occurrence of 1. At the beginning of each measurement interval, we initialize a hash array Y with m buckets and set all the bucket value to zero. Each of them remains at 0 if no packet is hashed to it. Algorithm 1 summarizes this TD generation process.

The cardinality of the TD, i.e., $|TD|$, is calculated by:

$$|TD| = \tilde{\phi} = \alpha_m m 2^{\frac{1}{m}\sum_{i=1}^{m} Y_i} \tag{2.4}$$

where α_m is some correction factor derived in [10] and the error (σ) of this estimator for each TD, defined as the normalized root

Algorithm 1 Online traffic digest generation algorithm

```
Initialize a hash array $Y$ of size $m$ with values 0;
for each incoming packet $s$ do
    generate a uniform hash value $h(PID_s)$ with the packet unique identifier, $PID_s$
    pick $l$ bits as the bucket index ($idx$) for selecting the buckets from $h(PID_s)$, where $l = log_2\ m$
    count the index, $i$, of first occurrence of 1 in the remaining bits without overlapping the $l$ bits (i.e., counting the number of consecutive zero + 1)
    update $Y[idx] = \max[Y[idx], i]$
end for
```

mean square relative error (RMSRE), is given by,

$$\sigma = \frac{\sqrt{Var(\tilde{\phi})}}{\phi} = \frac{1.30}{\sqrt{m}} \tag{2.5}$$

where $\tilde{\phi}$, ϕ are the estimated and true value of the corresponding packet count respectively.

In addition, the cardinalities of the union-sets or R.H.S. in Equation 2.2 can be estimated by max-merging the corresponding TDs by setting $Y[i] = max_{p \in P}\ Y_p[i]$ where p is the location which we are interested in and estimate the value by Equation 2.4. For more details about $O(loglog\ N_{max})$ distinct sample counting, please refer to [10].

Via periodic broadcast of the TDs collected at each link (or node) within a network, a node can reconstruct the network-wide view of traffic flow and routing patterns by computing the estimates of $|F_{i,j}^k|$ locally.

2.2.2 Proportional Union Method

Aiyou Chen et al have proposed PU in [8], which is an efficient simple estimator for the cardinalities based on a continuous variant of Flajolet-Martin sketches [14]. Their estimator has been shown to have almost the same statistical efficiency as the Maximum Likelihood Estimator (MLE), which is known to be optimal in the sense of Cramer-Rao lower bounds [3] under regular conditions. For details, please refer to [8].

Traffic Digest Generation for PU

The difference of the traffic digest generation between the approach of PU [8] and DATALITE [20] is the value stored in the TD. At the beginning, we initialize a hash array Y with m buckets with value equal to 1. Consider a stream of packets, denoted by packet set $\mathcal{S}$, for each incoming packets $s \in \mathcal{S}$, we hash the packet with the packet identifier PID_s using a uniform hash function $h(\cdot)$ from 1 to m and use the output of $h(PID_s)$ as the bucket index. We then store the output of $u(PID_s)$ in that bucket, where $u(PID_s)$ is an uniform random variable between [0,1] using PID_s as its seed. For each bucket, we store the minimum value of $u(PID_s)$ for each packet s hashed to the corresponding bucket. Each of them remains at 1 if no packet is hashed to it. The online TD generation algorithm for PU is presented in Algorithm 2.

Algorithm 2 Online traffic digest generation algorithm for PU

```
Initialize a hash array $Y$ of size $m$ with values 1;
for each incoming packet $s$ do
    generate a uniform hash value $idx = h(PID_s)$ with the packet unique identifier using a uniform hash function $h(\cdot)$ from 1 to $m$
    generate a random number $u(PID_s)$ uniformly on the interval $[0, 1]$
    update $Y[idx] = \min[Y[idx], u(PID_s)]$
end for.
```

Estimation of the cardinality of a single stream

For a single stream of packets, denoted by a packet set $\mathcal{S}$, we let $\mu = |\mathcal{S}|/m$ be the average number of packets per bucket for the stream. According to the Lemma 1 in [8], the distribution of $Y[i]$ can be approximated to exponential distribution with rate μ, i.e.,

$$P(Y[i] \geq y) \approx e^{-\mu y}$$

for $y \in [0, 1]$ and $P(Y[i] = 1) \approx e^{-\mu}$.

Since the correlation of each bucket can be neglected when μ is large, the likelihood functions of μ can be written as

$$L(\mu) = e^{-\mu \sum_{k=1}^{m} I(Y[k]=1)} \prod_{Y[i]<1} \mu e^{-\mu Y[i]}$$

where $I(\cdot)$ is the indicator function. Thus, the estimated average number of packets per bucket $\hat{\mu}$ and the estimated cardinality of the set of packets $\mathcal{S}$, i.e., $\widehat{|\mathcal{S}|}$, is given by

$$\hat{\mu} = \frac{\sum_{i=1}^{m} I(Y[i] < 1)}{\sum_{i=1}^{m} Y[i]},$$
$$\widehat{|\mathcal{S}|} = m\hat{\mu} \tag{2.6}$$

Estimation of per OD flow packet count

Suppose there are 2 sets of packets S and D at node s and d respectively and there is a set of packets $\mathcal{S}$ originating from node s and terminating at node d. Each packet set also contains other packet stream. Let Y_1 and Y_2 be the hash array used for node s and node d to capture the packet information. Let N be the number of packets of $S \cup D$, i.e., $N = |S \cup D|$ and let $Y_\cup$ be the hash array used for $S \cup D$ which can be obtained by min-merging the TD at node s and node d, i.e.,

$$Y_\cup[i] = min(Y_1[i], Y_2[i]), \quad \text{for } i = 1, ..., m$$

and the estimated value $\hat{N}$ of N can be obtained using Equation 2.6. The cardinality of $S \cap D$, i.e., $|S \cap D|$, can be estimated by

$$\widehat{|S \cap D|}^{(PU)} = \hat{N}\hat{P}(Y_1 = Y_2)$$

where $\hat{P}$ represents the empirical probabilities based on the observed traffic digests Y_1 and Y_2 and $\widehat{|X|}^{(PU)}$ represents the estimated cardinality of packet set X using PU estimator.

Estimation of OD flow per-link packet count

For estimating the cardinality of 3-sets intersection, we let $Y_1[i]$, $Y_2[i]$ and $Y_3[i]$, for $i = 1, ..., m$, be the TDs of packet set S, D and L respectively, where packet set S, D and L are the same as those described in Section 2.2.1 for DATALITE. We also let $\hat{N}$ be the estimated cardinality of the union-set $|S \cup D \cup L|$, i.e., $\hat{N} = |S \cup D \cup L|$, and let $Y_{\cup}$ be the hash array used for $S \cup D \cup L$ which can be obtained by min-merging the TDs at corresponding nodes, i.e.,

$$Y_{\cup}[i] = min(Y_1[i], Y_2[i], Y_3[i]), \quad \text{for } i = 1, ..., m$$

And, the cardinality of $|S \cap D \cap L|$ can be estimated by

$$|S \widehat{\cap D \cap} L|^{(PU)} = \hat{N}\hat{P}(Y_1 = Y_2 = Y_3)$$

where $\hat{P}$ represents the empirical probabilities based on the observed traffic digest Y_1, Y_2 and Y_3.

According to Theorem 3 in [8], the estimation error of this estimator σ_{PU} is given by,

$$\sigma_{PU} = \frac{\sqrt{Var(\hat{\phi})}}{\phi} \approx \frac{1}{\sqrt{m}} \cdot \sqrt{\frac{N}{\phi}} \tag{2.7}$$

where $\hat{\phi}$, ϕ are the estimated and true value of the corresponding packet count respectively and m is the number of buckets in the hash array.

2.2.3 Quasi-Likelihood Approach

Jin Cao et al have proposed a streaming algorithm for measuring the byte count of a packet stream in [7]. It provides accurate estimation on both packet and byte counting by Quasi-Likelihood Approach (QMLE). In the following, we describe the TD generation procedure and the OD packet count estimation using QMLE.

Traffic Digest Generation for QMLE

Unlike PU, in QMLE, we set the initial bucket value to ∞ and store the value of $g(PID_s)/v$, where v is the number of bytes in the packet, $g(\cdot)$ is an exponential random variable with mean equal to 1 and PID_s is the packet identifier. Since we focus on estimating the packet count, we simply set v equal to 1 for each incoming packet. Each of the bucket value remains ∞ when there is no packet hashed to it. Algorithm 3 shows the details of the TD generation procedure for QMLE.

Algorithm 3 Online traffic digest generation algorithm for QMLE

```
Initialize a hash array $Y$ of size $m$ with values $\infty$;
for each incoming packet $s$ do
    generate a uniform hash value $idx = h(PID_s)$ with the packet unique identifier using a uniform hash function $h(\cdot)$ from 1 to $m$
    generate an exponential random number $g(PID_s)$ of mean 1 using exponential random generator $g(\cdot)$
    update $Y[idx] = \min[Y[idx], g(PID_s)/v]$, where $v$ is the number of bytes in the packet
end for
```

Estimation of the cardinality of a single stream

Suppose there is a single stream of packets, denoted by packet set $\mathcal{S}$, we hash the packets to a TD Y according to Algorithm 3. By the Lemma 1 in [7], $Y[i]$ can be approximated by a Generalized Pareto distribution as,

$$P(Y[i] \geq y) \approx (1 + \beta y)^{-\alpha} \tag{2.8}$$

where $\mu = \alpha\beta$ is the average number of packets per bucket.

In the following, we omit $[i]$ when we refer to the traffic digest $Y[i]$ for simplicity. The quasi-density function, $P_Q(Y = y)$ can be written as:

$$P_Q(Y = y) = \alpha\beta(1 + \beta y)^{-(\alpha+1)}$$

Define $Y^* = YI(Y < \infty)$, where I is the indicator function. Let $l_{Q_1}(\mu, \beta)$ be the logarithmic quasi-likelihood function of $Y[i]$, for $i = 1, ..., m$, divided by m, i.e.,

$$\begin{aligned} l_{Q_1}(\mu, \beta) &= -\frac{1}{m}\sum_{i=1}^{m} log(\mu(1 + \beta Y^*[i])^{-(\frac{\mu}{\beta}+1)}) \\ &= -log(\mu) + \frac{1}{m}(\frac{\mu}{\beta} + 1)\sum_{i=1}^{m} log(1 + \beta Y^*[i]) \quad (2.9) \end{aligned}$$

The estimation $\hat{\mu}_{Q_1}$ can be obtained by optimizing the likelihood function $l_{Q_1}(\mu, \beta)$, i.e.,

$$(\hat{\mu}_{Q_1}, \hat{\beta}_{Q_1}) = \arg\max_{\mu,\beta} l_{Q_1}(\mu, \beta) \tag{2.10}$$

The problem can be solved by standard Newton-Raphson type algorithm [21]. Since the model in Equation 2.8 may not be the true model, such method is called Quasi Maximum Likelihood Estimation (*Quasi-MLE*) [22].

Estimation for per OD flow packet count

Suppose there is a stream of packets, denoted by packet set $\mathcal{S}$ originating from node s and terminating at node d. Let S and D be the set of packets at node s and d respectively.

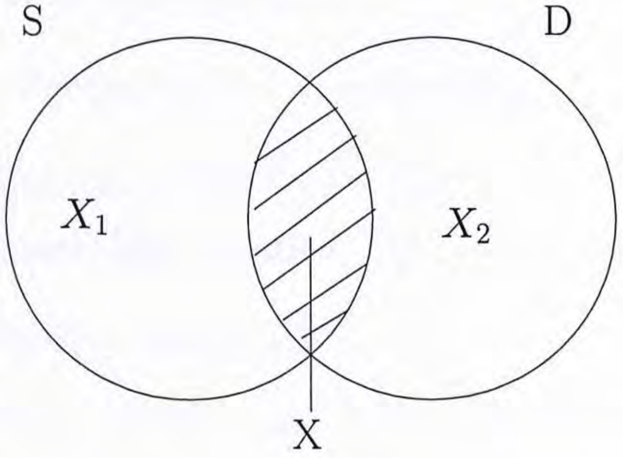


Figure 2.2: Set expression of 2 sets intersection

We let $Y_1[i], Y_2[i]$, for $i = 1, ..., m$, be a pair of hash arrays that store the information of the incoming packet stream at node a and node b using Algorithm 3. Let Λ_1, Λ_2 and μ be the average number of packet/byte count per bucket of the TDs of packet set S, D and $S \cap D$ respectively, where Λ_1, Λ_2 can be obtained by Equation 2.10 using TD Y_1 and Y_2. We let $X[i], X_1[i], X_2[i]$, for $i = 1, ..., m$, be the unobserved traffic digests, which are also generated by Algorithm 3, of the packet set $S \cap D$, $S \backslash D$ and $S \backslash D$ respectively as illustrated in Figure 2.2. To simplify the presentation, we omit $[i]$ when we refer the traffic digests $Y_1[i]$, $Y_2[i]$, $X[i]$, $X_1[i]$ and $X_2[i]$. We can see that

$$Y_j = \min(X, X_j), j = 1, 2 \tag{2.11}$$

According to Lemma 1 in [7], each of X, X_1, X_2 can be approximated by *Pareto Distribution*, i.e.,

$$X \approx Pareto(\alpha, \beta), \qquad X_j \approx Pareto(\alpha_j, \beta) \tag{2.12}$$

where $\alpha = \frac{\mu}{\beta}$ and $\alpha_j = \frac{\Lambda_j - \mu}{\beta}$, for $j = 1, 2$.

The quasi-probability function (P_Q) can be written as

$$\begin{aligned} & P_Q(Y_1 \geq y_1, Y_2 \geq y_2) \\ = \; & P_Q(X_1 \geq max(y_1, y_2), X_1 \geq y1, X_2 \geq y_2) \\ = \; & (1 + \beta max(y_1, y_2))^{-\alpha}(1 + \beta y_1)^{-\alpha_1}(1 + \beta y_2)^{-\alpha_2} \end{aligned}$$

And hence, the quasi-density function, $P_Q(Y_1 = y_1, Y_2 = y_2)$ can be derived as

$$\begin{cases} \text{Case 1 } (y_1 < y_2)\text{:} & \alpha_1(\alpha + \alpha_2)\beta^2(1 + \beta y_1)^{-(\alpha_1+1)} \\ & (1 + \beta y_2)^{-(\alpha+\alpha_2+1)} \\ \text{Case 2 } (y_1 > y_2)\text{:} & \alpha_2(\alpha + \alpha_1)\beta^2(1 + \beta y_2)^{-(\alpha_2+1)} \\ & (1 + \beta y_1)^{-(\alpha+\alpha_1+1)} \\ \text{Case 3 } (y_1 = y_2)\text{:} & \alpha\beta(1 + \beta y_1)^{-(\alpha+\alpha_1+\alpha_2+1)} \end{cases}$$

Define $Y_j^* = Y_j I(Y_j < \infty)$, for $j = 1, 2$, where I is the indicator function. Let $l_Q(\mu, \beta)$ be the logarithmic quasi-likelihood function of $(Y_1[i], Y_2[i])$ for $i = 1, ..., m$, divided by m, i.e.,

$$l_Q(\mu, \beta) = -\frac{1}{m}\sum_{i=1}^{m} log\left(P_Q(Y_1^*[i] = y_{1,i}, Y_2^*[i] = y_{2,i})\right)$$

where $y_{1,i}$ and $y_{2,i}$ are the empirical value of the i-th bucket of the hash array Y_1^* and Y_2^* respectively.

Optimizing the quasi-log-likelihood $l_Q(\mu, \beta)$, we obtain the QMLE estimate $\hat{\mu}_Q$ of μ, i.e.,

$$(\hat{\mu}_Q, \hat{\beta}_Q) = \arg\max_{(\mu,\beta)} l_Q(\mu, \beta) \tag{2.13}$$

for $\mu \leq \Lambda_1, \Lambda_2$ and $\beta > 0$.

This optimization can be solved by the standard Newton-Raphson type algorithm [21].

According to [7], the standard estimation error of QMLE for cardinality estimation of 2-sets intersection for packet set S and D, denoted by σ_{QMLE}, is $O(m^{1/2}(NSR_S + NSR_D - 2)^{1/2})$, i.e.,

$$\sigma_{QMLE} \propto \sqrt{\frac{NSR_S + NSR_D - 2}{m}} \tag{2.14}$$

where NSR_S and NSR_D are the noise-to-signal ratio of packet sets S and D described before, i.e., $\frac{|S|}{|S \cap D|}$ and $\frac{|D|}{|S \cap D|}$ respectively and m is the number of buckets of the hash array. Refer [7] for details.

We also extend the scheme for cardinality estimation of 3-sets intersection, e.g., to support OD-flow per-link packet count on individual intermediate links as described for DATALITE (and PU) in Section 2.2.1 (and 2.2.2). Refer to Appendix A for the detail derivation.

□ **End of chapter.**

Chapter 3

Estimation Error of Existing TD-based TMA schemes

We can see that the TD captures all the packets including both relevant and irrelevant packets. Via simulations, we found that the relative standard error of the estimator would be associated with the ratio of these two kinds of packets hashed in a TD. Define $\sigma_z(X)$ to be the standard error of cardinality estimation for packet set X using TD-based TMA scheme z, where $z = DL$, for DATALITE, $z = PU$ for PU and $z = QMLE$ for QMLE. In the following, we analyze the estimation error of the existing TD-based TMA schemes.

3.1 Error Accumulation and Amplification of Existing Schemes

Below we first discuss the effect of estimation error amplification of these three schemes for cardinality estimation of 2-sets intersection.

3.1.1 PU

According to Equation 2.7, the standard error σ_{PU} of the PU estimator for the intersection of the sets is directly proportional to the square root of the union-to-intersection ratio.

Suppose there are 2 sets of packets S and D. We are interested in the cardinality of the common packets of these 2 sets, i.e., $|S \cap D|$. The standard error of PU estimator for the cardinality estimation of $S \cap D$, denoted by $\sigma_{PU}(S \cap D)$, is given by:

$$\begin{aligned} \sigma_{PU}(S \cap D) &\approx \frac{1}{\sqrt{m}} \cdot \sqrt{\frac{|S \cup D|}{|S \cap D|}} \\ &= \frac{1}{\sqrt{m}} \cdot \sqrt{\frac{|S| + |D| - |S \cap D|}{|S \cap D|}} \\ &= \frac{1}{\sqrt{m}} \cdot \sqrt{NSR_S + NSR_D - 1} \end{aligned} \tag{3.1}$$

where NSR_S and NSR_D are defined as the noise-to-signal ratio of $S \cap D$ in set S and D, i.e., $\frac{|S|}{|S \cap D|}$ and $\frac{|S|}{|S \cap D|}$ respectively.

3.1.2 QMLE

Jin Cao et al have also analyzed the estimation error of QMLE in [7]. According to [7], the standard relative error of QMLE for the cardinality estimation of 2-sets intersection for packet set S and D is $O(\sqrt{\frac{NSR_S+NSR_D-2}{m}})$, i.e.,

$$\sigma_{QMLE}(S \cap D) \triangleq \frac{constant}{\sqrt{m}} \cdot \sqrt{NSR_S + NSR_D - 2} \qquad (3.2)$$

where NSR_S and NSR_D are the same as described in Equation 3.1. $\sigma_{QMLE}(S \cap D)$ is also similar to $\sigma_{PU}(S \cap D)$ as shown in Equation 3.1 when the noise-to-signal ratio of the 2 adjacent nodes are large.

3.1.3 DATALITE

According to Equation 2.2, the estimation errors associated with $|F_{i,j}^k|$ is the aggregated error of the cardinality estimators for each union-set on the R.H.S. of Equation 2.2. In general, the more number of stages of set-intersection, the greater the estimation error due to the increase in the number of union terms in the R.H.S. of Equation 2.3. Worse still, since the estimation error σ given in Equation 2.5 is a *relative* error with respect to each of the union-set cardinalities in the R.H.S. of Equation 2.3, the corresponding relative estimation error, i.e., percentage-wise, for the intersection set on the L.H.S. of Equation 2.3 can get "amplified" when the cardinality of the intersection set in the L.H.S. of Equation 2.3 is much smaller than that of the union-set terms on the R.H.S. of the equation.

Consider the intersection between packet sets S and D as an example. Denote the relative estimation error for $|S \cap D|$ and $|S \cup D|$ by $\sigma_{_{DL}}(S \cap D)$ and $\sigma_{_{DL}}(S \cup D)$ respectively. Here, $\sigma_{_{DL}}(S \cup D)$ corresponds to the relative error specified in Equation 2.5 resulted from the union-set cardinality estimation as proposed in [10]. Since $|S \cap D| = |S| + |D| - |S \cup D|$, even if we know the true values of $|S|$ and $|D|$, i.e., zero estimation error associated with those terms, we still have:

$$\sigma_{_{DL}}(S \cap D) \cdot |S \cap D| \;=\; \sigma_{_{DL}}(S \cup D) \cdot |S \cup D|$$

And thus,

$$\begin{aligned}\sigma_{_{DL}}(S \cap D) &= \sigma_{_{DL}}(S \cup D) \cdot \frac{|S \cup D|}{|S \cap D|} \\ &= \sigma_{_{DL}}(S \cup D) \cdot \frac{|S| + |D| - |S \cap D|}{|S \cap D|} \\ &= \frac{1.30}{\sqrt{m}} \cdot (NSR_S + NSR_D - 1) \qquad (3.3)\end{aligned}$$

where NSR_S and NSR_D are the noise-to-signal ratio of packet set A and B, i.e., $(NSR_S = \frac{|S|}{|S \cap D|}, NSR_D = \frac{|D|}{|S \cap D|})$.

The situation is particularly problematic when NSR_S, NSR_D

or both are large, which, unfortunately, is quite typical in the context of network-wide OD-flow-per-link monitoring.

Based on the above observations, we can see that the standard error of the cardinality estimation for 2-sets intersection is positively correlated to the noise-to-signal ratio of the corresponding nodes. In the next section, we will analyze the estimation error and derive the corresponding upper bound of the estimation error for cardinality estimation of 3-sets intersection.

3.2 Estimation Error of 3-sets intersection cases

In this section, we discuss the estimation error of cardinality estimation of 3-sets intersection, i.e., to support OD-flow per link packet count, of the TD-based TMA schemes. Unlike the cardinality estimation of 2-sets intersection, we can only derive the upper bound of the estimation error.

3.2.1 PU

According to the standard error of the PU estimator derived in [8] as shown in Equation 2.7, the upper bound of the estimation error of PU for the cardinality estimation of 3-sets intersection is derived as follows:

Suppose there is a target OD-flow $\mathcal{S}$ with volume $|\mathcal{S}|$ (in unit of packets per unit time) which originates from node a, terminates at node c and passes through a link b along its path. Denote the sets of packets observed at nodes s, d and link l by S, D and L respectively. In general, S, D and L can contain packets from other traffic flows besides the ones belonging to $\mathcal{S}$. We use NSR_i to represent the noise-to-signal ratio of $\mathcal{S}$ in each of these packet sets, i.e., $NSR_i = \frac{|i|}{|\mathcal{S}|}$ for $i = S, D, L$. The standard error of the PU estimator for the cardinality estimation of 3-sets intersection, denoted by $\sigma_{PU}(S \cap D \cap L)$, is given by:

$$\begin{aligned} & \sigma_{PU}(S \cap D \cap L) \\ \approx\ & \sqrt{\frac{N}{m \cdot |\mathcal{S}|}} \\ =\ & \sqrt{\frac{|S \cup D \cup L|}{m \cdot |\mathcal{S}|}} \\ =\ & \sqrt{\frac{|S| + |D| + |L| - |S \cap D| - |D \cap L| - |S \cap L| + |S \cap D \cap L|}{m \cdot |\mathcal{S}|}} \end{aligned} \tag{3.4}$$

Using the union-bound argument ($|S \cap D| \geq |\mathcal{S}|, |S \cap L| \geq |\mathcal{S}|$ and $|B \cap C| \geq |\mathcal{S}|$), we have:

$$\begin{aligned} & \sigma_{PU}(S \cap D \cap L) \\ \leq\ & \frac{1}{\sqrt{m}} \cdot \sqrt{\frac{|S| + |D| + |L| - |\mathcal{S}| - |\mathcal{S}| - |\mathcal{S}| + |\mathcal{S}|}{|\mathcal{S}|}} \\ =\ & \frac{1}{\sqrt{m}} \cdot \sqrt{NSR_S + NSR_D + NSR_L - 2} \\ \triangleq\ & \hat{\sigma}_{PU}(S \cap D \cap L) \end{aligned} \tag{3.5}$$

where NSR_S, NSR_D and NSR_L are defined as the noise-to-signal ratio of $\mathcal{S}$ in packet set S, D and L respectively, i.e., $\frac{|S|}{|\mathcal{S}|}$, $\frac{|D|}{|\mathcal{S}|}$ and $\frac{|L|}{|\mathcal{S}|}$. In special case, when $|S \cap D| = |S \cap L| = |D \cap L| = |\mathcal{S}|$, we have $\sigma_{PU}(S \cap D \cap L) \approx \hat{\sigma}_{PU}(S \cap D \cap L)$ according to Equation 3.5.

We let ϵ_{PU} be the empirical (actual) error in estimating the OD-flow per link packet count using PU. To demonstrate the relationship between $\hat{\sigma}_{PU}(S \cap D \cap L)$ and ϵ_{PU}, we apply PU to estimate the OD-flow per link packet count using different TD size $[60, 240, 960]$ KBytes in different NSR [10,20,50,80,100] situation. Here, we simulate the special case described above, i.e., $|S \cap D| = |S \cap L| = |D \cap L| = |\mathcal{S}|$ and $\sigma_{PU}(S \cap D \cap L) \approx \hat{\sigma}_{PU}(S \cap D \cap L)$. Figure 3.1 shows the strong linear relationship of the standard relative error, i.e., $\hat{\sigma}_{PU}(S \cap D \cap L)$ and the corresponding empirical root mean square relative error (RMSRE), i.e., ϵ_{PU} in all cases.

3.2.2 DATALITE

We consider the 3 sets of packets S, D and L for OD-flow per link packet count estimation using DATALITE mentioned in Section 3.2.1. We are interested in the cardinality estimation of the common packet set $\mathcal{S}$ of these 3 packet sets, i.e., $|\mathcal{S}| = |S \cap D \cap L|$.

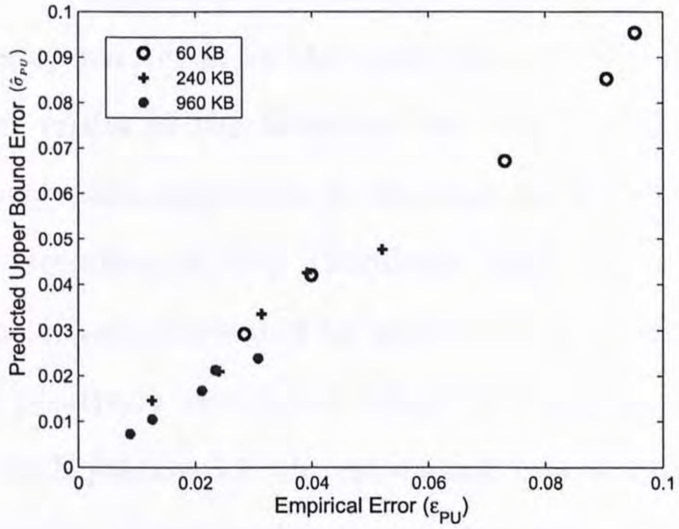


Figure 3.1: RMS Empirical Result vs the Predicted Upper Bound of Standard Estimation Error for PU.

If we know the true value of S, D and L, the estimate of $\mathcal{S}$, i.e., $\Delta|S \cap D \cap L|$ can be derived as follows:

$$\begin{aligned} & \widehat{|S \cap D \cap L|} \\ = \ & |S| + |D| + |L| - \widehat{|S \cup D|} - \widehat{|D \cup L|} - \widehat{|S \cup C|} + \widehat{|S \cup D \cup L|} \end{aligned}$$

where $\widehat{|X|}$ is the estimated value of packet set X using Equation 2.4. We defined $\Delta|X|$ to be the absolute estimation error of the cardinality estimation of packet set X, and thus, $\widehat{|X|} = |X| + \Delta|X|$. The absolute relative error of $\widehat{|S \cap D \cap L|}$, $\Delta|S \cap D \cap L|$, would be:

$$\begin{aligned} & \Delta|S \cap D \cap L| \\ = \ & -\Delta|S \cup D| - \Delta|D \cup L| - \Delta|S \cup L| + \Delta|S \cup D \cup L| \end{aligned}$$

However, when there are outliers[1] in the hash array of $S \cup D$, they may also exist in the hash array of $S \cup D \cup L$. The estimation errors of the terms in the R.H.S. of Equation 3.6 are positively correlated due to the max-merge property of the estimator described in [10]. (Similarly, $\Delta|D \cup L|$ and $\Delta|S \cup L|$ are also positively correlated to $\Delta|S \cup D \cup L|$ correspondingly). As these positively correlated terms do carry opposite signs, as shown in Equation 3.6, the estimation errors associated with these individual terms tend to cancel each other when compared to the case with independent assumption.

Therefore, by assuming the independence of the estimation errors for individual terms in the R.H.S. of Equation 3.6, the standard error of cardinality estimation of 3-sets intersection using DATALITE ($\sigma_{DL}(S \cap D \cap L)$), would be:

$$\begin{aligned} & \sigma_{DL}(S \cap D \cap L) \\ \leq\; & \frac{\Delta|S \cup D| + \Delta|D \cup L| + \Delta|S \cup L| + \Delta|S \cup D \cup L|}{|S \cap D \cap L|} \\ =\; & \frac{\sigma \cdot |S \cup D| + \sigma \cdot |D \cup L| + \sigma \cdot |S \cup L| + \sigma \cdot |S \cup D \cup L|}{|S \cap D \cap L|} \\ =\; & \sigma \cdot \frac{(|S \cup D| + |D \cup L| + |S \cup L| + |S \cup D \cup L|)}{|S \cap D \cap L|} \qquad (3.6) \end{aligned}$$

where σ is the common standard relative error for estimating

[1] A large number of consecutive zeros generated by the uniform hash function as described in Section 2.2.1

$|S \cup D|$, $|D \cup L|$, $|S \cup L|$ and $|S \cup D \cup L|$ under the same TD memory size m as given in Equation 2.5.

Using union-bound arguments, i.e., $|X \cup Y| \leq |X| + |Y|$, $\sigma_{DL}(S \cap D \cap L)$ can be upper further bounded by:

$$\begin{aligned} & \sigma_{DL}(S \cap D \cap L) \\ \leq \ & \sigma \cdot \frac{(|S| + |D|) + (|D| + |L|) + (|S| + |L|) + (|S| + |D| + |L|)}{|S \cap D \cap L|} \\ = \ & \frac{3.9}{\sqrt{m}} \cdot (NSR_S + NSR_D + NSR_L) \\ \triangleq \ & \hat{\sigma}_{DL}(S \cap D \cap L) \end{aligned} \tag{3.7}$$

Let ϵ_{DL} be the *actual* (empirical) relative error in estimating $|S|$ using DATALITE. The key observation from Equation 3.7 is that we can suppress $\hat{\sigma}_{DL}(S \cap D \cap L)$, therefore ϵ_{DL}, by reducing the noise-to-signal ratio associated with the corresponding packet sets (and their TDs).

To demonstrate the relationship between $\hat{\sigma}_{DL}(S \cap D \cap L)$ and ϵ_{DL}, we apply DATALITE to estimate the OD-flow per link packet count using different TD size $[40, 160, 640]$ KBytes in different NSR [10,20,50,80,100] situation. As shown in Figure 3.2, while the upper bound on the estimation error seems to be quite loose, there is a strong linear relationship between the upper bound of the standard relative error, i.e., $\hat{\sigma}_{DL}(S \cap D \cap L)$ and the corresponding empirical root mean square relative error (RMSRE)

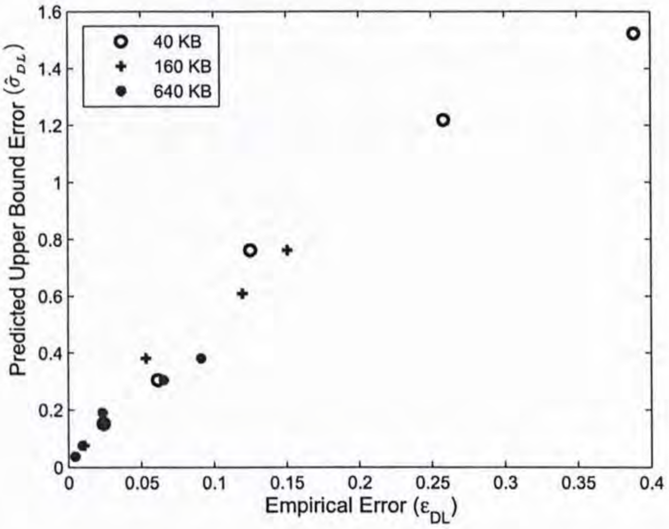


Figure 3.2: RMS Empirical Result vs the Predicted Upper Bound of Standard Estimation Error for DATALITE.

in all cases. Figure 3.2 plots the upper-bound of the standard relative error ($\hat{\sigma}_{DL}(S \cap D \cap L)$) for the cardinality estimation of 3-sets intersection against the empirical estimation error ϵ_{DL} achieved by DATALITE for different TD sizes (m).

Based on the observation of two schemes, the standard estimation error can be suppressed by reducing the noise-to-signal ratio (NSR) of the flows on different nodes/links, and thus, the actual error ϵ_{DL} and ϵ_{PU}. Therefore, we propose to separate the flows into groups and hash them into different TDs in order to reduce the total number of packets in one TD, and thus, reduce the noise-to-signal ratio. In the next chapter, we are going to propose a TD-splitting algorithm such that the flows can be separated according to their flow-volume in order to reduce the estimation error by keeping the same total traffic digest size.

□ **End of chapter.**

Chapter 4

Error Reduction Via Traffic Digest Splitting

4.1 Motivation

Our proposed Traffic Digest Splitting approach is motivated by the following 2 key observations: (i) Based on the estimation error characteristics derived in Equations 3.1 and 3.3, we can suppress estimation errors by reducing the noise-to-signal ratio associated with the corresponding TDs. (Similarly for the estimation errors of cardinality estimation for 3-sets intersection as shown in Equations 3.5 and 3.7) and (ii) The relative estimation error of the estimator used by TD-based TMA schemes only reduce at a rate of $O(\frac{1}{\sqrt{m}})$ where m is proportional to the memory size allocated for the TD. In other words, by halving the memory size allocated for a particular TD, the relative estimation error of the traffic flows sharing the TD will only increase by a factor of $\sqrt{2}$.

Based on these two observations, it becomes viable to split a TD at each monitoring point into separate sub-TDs, e.g., one for the mice flows and the other for the elephant ones while keeping the total amount of memory consumed by all sub-TDs unchanged. By doing so, the decrease of estimation error due to the reduction of noise-to-signal ratio (as shown in Equations 3.1 and 3.3) can outweigh the increase in estimation error due to a smaller memory-size of each sub-TD as pointed out in (ii).

In general, the mapping of individual flows into different sub-TDs along their end-to-end paths will interact with each other and impact the resultant estimation errors for flows sharing one or more sub-TDs. This implies that a globally optimal TD-splitting strategy would require tight coordination across many different monitoring points, which may not be feasible in practice. To avoid explicit network-wide coordination, we will focus on a local strategy by applying "optimal" TD-splitting within each individual monitoring point regardless of the TD-splitting decisions elsewhere during the current measurement interval. We also assume the local link to be the "bottleneck" which creates the dominant NSR for the passing flows, more details are discussed in Section 4.4.2.

Under this localized approach, the performance of a given TD-splitting strategy still depends on various factors including: the distribution of the flow volumes associated with the original TD; the splitting threshold (in terms of flow-volume); the amount of memory available for each sub-TD; the resultant noise-to-signal ratio experienced by individual traffic flows in their assigned sub-TD and the choice of the optimization objective, e.g., minimizing the maximum percentage error of any participating flows vs. minimizing the r.m.s. percentage error across all of the participating flows. To derive the optimal TD-splitting strategy, we have formulated the problem as a mathematical programme through which the optimal flow-to-sub-TD assignment (based on simple volume-based threshold) can be determined under various traffic flow distributions and optimization objectives. Furthermore, the TD-splitting strategy can be applied recursively as long as the number of packets assigned to each sub-TD is large enough to provide statistically valid measurements. Via simulation in Chapter 6, we compare the performance of TD-splitting scheme with that of the original non-splitting scheme and the preliminary results are quite promising.

4.2 Objective Functions for Optimal TD-splitting

Given the total amount of memory available for all sub-TDs of a target link (i,j), our local TD-splitting scheme maps the commodity flows on the link to different sub-TDs while minimizing the maximum (or alternatively, the root mean square) estimation error (of flow-volume) among all the participating flow commodities. In particular, the objective for this optimization can be expressed as the minimization of the following functions:

(1) *Maximum* estimation error across all the participating flows

$$\max_{k\in[1,n]} \epsilon_{z_{i,j}}^{k} \tag{4.1}$$

or

(2) *Root mean square* estimation error across all the participating flows

$$\sqrt{\sum_{k\in[1,n]} {\epsilon_{z_{i,j}}^{k}}^{2}} \tag{4.2}$$

where n is the number of flow commodities (noted as 1 OD-pair) passing through link (i,j), and $\epsilon_{z_{i,j}}^{k}$, is the *actual* relative estimation error of commodity k on link (i,j) using different schemes where $z = \{DL, PU\}$ for DATALITE and PU respectively. However, since there is no closed-form expression for $\epsilon_{z_{i,j}}^{k}$ in general, we resort to leveraging the strong linear relationship observed from Figures 3.2 and 3.1 between ϵ_{DL} and $\hat{\sigma}_{DL}(S \cap D \cap L)$ (as well as ϵ_{PU} and $\hat{\sigma}_{PU}(S \cap D \cap L)$) and use the expression of the latter (given in Equation 3.7 and 3.5) to establish the surrogate objective functions for the optimization. Due to the local nature of our approach, our optimization only needs to deal with each target link individually. Also, our objective functions can be applied to all TD-based TMA schemes with the same formulation for cardinality estimation of both 2-sets and 3-sets intersection. We therefore drop the per-link subscripts (i, j), the TD-based TMA scheme subscripts $z = \{DL, PU\}$ and the packet set parameter X from now on and replace the objective functions in Equations 4.1 and 4.2 by their respective surrogates:

$$\sigma_{max} \triangleq \max_{k \in [1,n]} \hat{\sigma}^{k} \tag{4.3}$$

and

$$\sigma_{rms} \triangleq \sqrt{\sum_{k=1}^{n} (\hat{\sigma}^{k})^{2}/n} \tag{4.4}$$

where, for symbol consistency, $\hat{\sigma}^k$ is defined to be equal to σ^k in Equations 3.1 and 3.3 for cardinality estimation of 2-sets intersection using DATALITE and PU respectively. For cardinality estimation of 3-sets intersection, $\hat{\sigma}^k$ is the upper bound of the standard estimation error for commodity k on the target link as defined in Equations 3.5 and 3.7.

4.3 Problem Formulation of Threshold-based Splitting

In general, there are many different ways to split the available memory amongst different sub-TDs and map different groups of commodity flows to each sub-TD. In this thesis, we only consider the flow-volume threshold-based mapping strategy. Furthermore, in order to facilitate the distributed merging of sub-TDs across different links in a network (max-merging for DATALITE and min-merging for PU and QMLE) while avoiding additional inter-link coordination, we will focus on the recursive binary splitting of the overall memory pool available for a link into equal-sized sub-TDs[1]. In the most basic form of such binary splitting approach, two equal-sized sub-TDs are introduced for each link, one for the small flows and the other for the large ones. Each flow on a link is classified as either a *small* or *large* flow by comparing its traffic volume, *as estimated during the*

[1] It is not straightforward to merge sub-TDs of *different* sizes generated at different monitoring points. This is left as a future work.

previous measurement interval, to a splitting threshold f_s and then mapped to the corresponding sub-TD. By separating small flows from large ones, we can reduce the noise-to-signal ratio of all the flows which is particularly important for reducing estimation errors for small flow volumes.

4.3.1 Minimizing Maximum Estimation Error

Under the "local-bottleneck" assumption, to minimize the maximum estimation error across all flows, i.e., σ_{max} as defined in Equation 4.3, we can substitute the expression of $\hat{\sigma}$ from Equations 3.1 and 3.3 into Equation 4.3 to determine the "optimal" splitting threshold $\hat{f}_s$ as follows:

$$\begin{aligned} \hat{f}_s &= arg\{\min_{f_s} \sigma_{max}\} \\ &= arg\{\min_{f_s} \max_{k \in [1,n]} \hat{\sigma}^k\} \\ &= arg\{\min_{f_s} \max_{k \in [1,n]} NSR(r_k, f_s)\} \end{aligned} \tag{4.5}$$

where n is the number of commodity flows on the target link. $NSR(r_k, f_s)$ is defined as the noise-to-signal ratio experienced by our target flow within its corresponding sub-TD of the target link when the splitting threshold is set to f_s. Such noise-to-signal ratio is given by:

$$NSR(r_k, f_s) = \frac{\mathcal{R}(r_{min}, f_s)}{r_k} I(r_k < f_s) + \frac{\mathcal{R}(f_s, r_{max})}{r_k} I(r_k \geq f_s) \tag{4.6}$$

where I is the Indicator function ;

$$\mathcal{R}(f_{low}, f_{up}) = \int_{f_{low}}^{f_{up}} r \cdot g(r) dr \cdot \frac{\mathcal{R}}{\int_{r_{min}}^{r_{max}} r \cdot g(r) dr} \tag{4.7}$$

is the total flow-volume in the sub-TD due to flows with volume between f_{low} and f_{up}. $g(r)$ is the p.d.f. of the flow volume carried by the link with range between $[r_{min}, r_{max}]$ and $\mathcal{R}$ is the total flow volume passing through the link or originating from (terminating at) the node.

Since the smallest flow in each of the "Small" and "Large" sub-TD always gives the largest noise-to-signal ratio, when the objective is to minimize the maximum noise-to-signal ratio encountered by any flow on the link, the optimization associated with Equation 4.5 can be solved by choosing f_s such that

$$NSR(r_{min}, f_s) = NSR(f_s, f_s) \tag{4.8}$$

where r_{min} and f_s are the respective volume of the smallest flow in the "Small" and "Large" sub-TD of the link.

$$NSR(r_{min}, f_s) = \mathcal{R}(r_{min}, f_s)/r_{min}$$

and

$$NSR(f_s, f_s) = \mathcal{R}(f_s, r_{max})/f_s$$

are the corresponding noise-to-signal ratio experienced by these two flows in their corresponding sub-TD. As a result, the optimal splitting threshold $\hat{f}_s$ in this case can be derived by solving the following equation:

$$\frac{\mathcal{R}(r_{min}, \hat{f}_s)}{r_{min}} = \frac{\mathcal{R}(\hat{f}_s, r_{max})}{\hat{f}_s} \tag{4.9}$$

Substituting Equation 4.7 into this, we have:

$$\int_{r_{min}}^{\hat{f}_s} r \cdot g(r) dr / r_{min} = \int_{\hat{f}_s}^{r_{max}} r \cdot g(r) dr / \hat{f}_s \tag{4.10}$$

Optimal Splitting Threshold Under Different Individual Flow-volume Distributions

In practice, $g(r)$, r_{min} and r_{max} can be estimated based on historical measurements. As we will show via our simulation results in the later chapter, the value of the optimal splitting threshold $\hat{f}_s$ derived from Equation 4.9 is quite robust with respect to the exact details of $g(r)$ as long as the overall shape of the distribution is captured. This observation motivates us to derive the analytical expressions for $\hat{f}_s$ by considering some common types of flow-volume distributions for $g(r)$. In the following, we derive the analytical expression for the optimal splitting threshold $\hat{f}_s$ for different types of flow-volume distribution including the Uniform, Gamma and Pareto distributions. In general, our scheme can be applied to any distributions.

(i) $g(r) \sim$ ***Uniform Distribution***

Consider $g(r)$ to follow a uniform distribution with range = $[r_{min}, r_{max}]$. By substituting $g(r)$ into Equation 4.10, we have:

$$\frac{(f_s^2 - r_{min}^2)\mathcal{R}}{(r_{max}^2 - r_{min}^2)r_{min}} = \frac{(r_{max}^2 - \hat{f}_s^2)\mathcal{R}}{(r_{max}^2 - r_{min}^2)\hat{f}_s}$$

And thus, $\hat{f}_s$ should satisfy the following cubic equation:

$$\hat{f}_s^{\ 3} + r_{min}\hat{f}_s^{\ 2} - r_{min}^2\hat{f}_s - r_{max}^2 r_{min} = 0 \qquad (4.11)$$

The optimal splitting threshold is given by the only real-valued positive root of Equation 4.11.

(ii) $g(r) \sim$ ***Gamma Distribution***

When the volume of individual flows on a target link follows a Gamma distribution $g(r)$ with the shape parameter k and scale parameter θ. By setting $k = 1$ (For demonstrating the equation clearly. In general, k can be any value.) and substitute $g(r)$ into Equation 4.10 followed by some algebraic manipulation, the optimal splitting threshold $\hat{f}_s$ can be obtained by numerically solving the following equation:

$$\begin{aligned} e^{-\frac{\hat{f}_s}{\theta}}\hat{f}_s^{\ 2} - e^{-\frac{r_{min}}{\theta}}(r_{min} + \theta)\hat{f}_s + e^{-\frac{\hat{f}_s}{\theta}}(\theta + r_{min})\hat{f}_s \\ + \theta r_{min} e^{-\frac{\hat{f}_s}{\theta}} - r_{min} e^{-\frac{r_{max}}{\theta}}(r_{max} + \theta) = 0 \qquad (4.12) \end{aligned}$$

(iii) $g(r) \sim$ ***Pareto Distribution***

For Pareto flow-volume distribution, $g(r) = \frac{k r_{min}^k}{r^{k+1}}$, where k is the shape parameter and r_{min} is the scale parameter corresponding to the minimum volume of an individual flow commodity. By setting $k = 2$ (For demonstrating the closed-form expression of the optimal splitting threshold. In general, k can be any value.) and substitute $g(r)$ into Equation 4.10 and solve the corresponding equation while considering the only real, positive root, we have:

$$\hat{f}_s = \frac{r_{max}r_{min} - r_{min}^2 + \sqrt{5r_{max}^2 r_{min}^2 + r_{min}^4 - 2r_{max}r_{min}^3}}{2r_{max}} \quad (4.13)$$

4.3.2 Minimizing R.M.S. Estimation Error

Instead of minimizing the maximum estimation error across all commodity flows on a target link, an alternative is to minimize the corresponding surrogate root mean square error as defined by Equation 4.4. For PU, according to the relationship between the estimation error and the noise-to-signal ratio as shown in Equation 3.5, under the "local-bottleneck" assumption, the optimal TD-splitting problem can be formulated as:

$$\begin{aligned}\hat{f}_s &= arg\{\min_{f_s} \sigma_{rms}\} \\ &= arg\{\min_{f_s} \sqrt{\sum_{k=1}^{n} (\sigma^k)^2/n}\} \\ &= arg\{\min_{f_s} \sqrt{\sum_{k=1}^{n} NSR(r_k, f_s)/n}\} \\ &= arg\{\min_{f_s} \sum_{k=1}^{n} NSR(r_k, f_s)\} \qquad (4.14)\end{aligned}$$

where $NSR(r_k, f_s)$ is the same noise-to-signal ratio expression defined in Equation 4.6.

Given the flow-volume distribution $g(r)$ on the target link, $NSR(r_k, f_s)$ can again be determined via Equation 4.6 and Equation 4.7. The corresponding optimal splitting threshold $\hat{f}_s$ can then be expressed as:

$$\begin{aligned}\hat{f}_s &= arg\{\min_{f_s} \sum_{k=1}^{n} NSR(r_k, f_s)\} \qquad (4.15) \\ &= arg\{\min_{f_s} \ [\int_{r_{min}}^{f_s} \frac{\mathcal{R}(r_{min}, f_s)}{r} g(r)dr + \int_{f_s}^{r_{max}} \frac{\mathcal{R}(f_s, r_{max})}{r} g(r)dr]\}\end{aligned}$$

where $\mathcal{R}(f_1, f_2)$ are the same as those specified in Equation 4.7.

For DATALITE, under the aforementioned "local-bottleneck" assumption, the problem can be formulated as:

$$
\begin{aligned}
\hat{f}_s &= arg\{\min_{f_s} \sigma_{rms}\} \\
&= arg\{\min_{f_s} \sum_{k=1}^{n} NSR(r_k, f_s)^2\} \\
&= arg\{\min_{f_s} \; [\int_{r_{min}}^{f_s} \frac{\mathcal{R}(r_{min}, f_s)^2}{r^2} g(r)dr + \int_{f_s}^{r_{max}} \frac{\mathcal{R}(f_s, r_{max})^2}{r^2} g(r)dr]\}
\end{aligned} \tag{4.16}
$$

Unlike the case of minimizing maximum estimation error where the optimization problem can be transformed to a simple equation specified in Equation 4.10, there is no counterpart for Equation 4.10 in the case of minimizing rms estimation error. As a result, we often have to solve Equations 4.15 and 4.16 numerically using line-search to determine the corresponding optimal splitting threshold value $\hat{f}_s$ even under typical distributions of $g(r)$.

4.4 Analysis of Estimation Error Reduction Via Single-Level TD-splitting

In Equations 3.1 (Equation 3.2), we have already discussed that the standard error of PU (as well as QMLE) is linearly proportional to the square root of the noise-to-signal ratio of the flow while that of DATALITE is proportional to the square root of the noise-to-signal ratio as shown in Equation 3.3 . With the same splitting threshold, the estimation error reduction would be different across different schemes. In the following, we first describe the reduction of the noise-to-signal ratio of various flow-volume distributions for cardinality estimation of 2-sets intersection and then analyze the error reduction via single-level TD-splitting.

4.4.1 Noise-to-signal Ratio Reduction

Suppose our monitoring point is located at node s, and we are interested in estimating the volume of flows originating from node d and terminating at node d. The corresponding noise-to-signal ratio of the OD-flow at node s and node d for flows with flow-volume r_k would be defined as:

$$NSR_S(r_k) = \frac{|S|}{r_k} \text{ and } NSR_D(r_k) = \frac{|D|}{r_k}$$

where S and D are the set of packets originating from node s and terminating at node d respectively. r_k is the volume of the k-th flow of node s. We define the optimal splitting threshold at node s and node d as $\hat{f}_s^S$ and $\hat{f}_s^D$, which are estimated by Equations 4.10 or 4.16 (4.15), respectively. The new noise-to-signal ratio of the flows at node s and node d after TD-splitting, denoted by $NSR'_S(r_k, \hat{f}_s^S)$ and $NSR'_D(r_k, \hat{f}_s^D)$, would be obtained by substituting $\hat{f}_s^S$ and $\hat{f}_s^D$ to Equation 4.6, i.e.,

$$\begin{aligned} & NSR'_S(r_k, \hat{f}_s^S) \\ = & \begin{cases} \frac{|S|}{r_k \cdot \int_{r_{min}}^{r_{max}} r \cdot g(r) dr} \cdot \int_{r_{min}}^{\hat{f}_s^A} r \cdot g(r) dr, & \text{if } r_k < \hat{f}_s^S \\ \frac{|S|}{r_k \cdot \int_{r_{min}}^{r_{max}} r \cdot g(r) dr} \cdot \int_{\hat{f}_s^S}^{r_{max}} r \cdot g(r) dr, & \text{if } r_k \geq \hat{f}_s^S \end{cases} \end{aligned} \tag{4.17}$$

and

$$\begin{aligned} & NSR'_D(r_k, \hat{f}_s^D) \\ = & \begin{cases} \frac{|D|}{r_k \cdot \int_{r_{min}}^{r_{max}} r \cdot g(r) dr} \cdot \int_{r_{min}}^{\hat{f}_s^D} r \cdot g(r) dr, & \text{if } r_k < \hat{f}_s^D \\ \frac{|D|}{r_k \cdot \int_{r_{min}}^{r_{max}} r \cdot g(r) dr} \cdot \int_{\hat{f}_s^D}^{r_{max}} r \cdot g(r) dr, & \text{if } r_k \geq \hat{f}_s^D \end{cases} \end{aligned} \tag{4.18}$$

And hence, the noise-to-signal ratio reduction factor for flow with flow volume r_k at node a, denoted by $\gamma_A(r_k)$, would be defined as

$$\begin{aligned} \gamma_S(r_k) &= \frac{NSR_S(r_k)}{NSR'_S(r_k, \hat{f}_s^S)} \\ &= \begin{cases} \int_{r_{min}}^{r_{max}} r \cdot g(r) dr / \int_{r_{min}}^{\hat{f}_s^S} r \cdot g(r) dr, & \text{if } r_k < \hat{f}_s^S \\ \int_{r_{min}}^{r_{max}} r \cdot g(r) dr / \int_{\hat{f}_s^S}^{r_{max}} r \cdot g(r) dr, & \text{if } r_k \geq \hat{f}_s^S \end{cases} \end{aligned} \tag{4.19}$$

Similar to the noise-to-signal ratio reduction factor of node d, denoted by $\gamma_D(r_k)$, we can obtain a similar result.

Based on Equation 4.19, we can see that the noise-to-signal ratio reduction factor only depends on the flow-volume distribution, minimum and maximum flow volume. Since it is difficult to express $\gamma_S(r_k)$ in closed-form, we compare the reduction factor numerically using various flow volume distributions, namely

Flow Volume Distribution	NSR Reduction Factor	
	$r_k < \hat{f}_s$	$r_k \geq \hat{f}_s$
Uniform	7.35	1.16
Gamma ($k = 4, \theta = 3.5 \times 10^5$)	4.95	1.25
Pareto ($k = 2, r_{min} = 5 \times 10^5$)	2.59	1.63

Table 4.1: NSR reduction factor of various flow-volume distributions

with Uniform, Gamma and Pareto distribution, which are used for performance evaluation in Chapter 6. Table 4.1 shows the noise-to-signal ratio reduction factor for the various cases at a TD, where $\hat{f}_s$ is the optimal splitting threshold.

The single-level TD-splitting gives the greatest noise-to-signal ratio reduction factor with Uniform distributed flow-volume while it gives the least reduction factor with Pareto distributed flow-volume for flows with flow volume less than the optimal splitting threshold. It gives the least noise-to-signal ratio reduction factor with Uniform distributed flow-volume while it gives the greatest reduction factor with Pareto distributed flow-volume for flows with flow volume less than the optimal splitting threshold.

Here, we can see that the noise-to-signal ratio reduction factors are different across different flow volume distributions. In the next section, we will analyze the changing ratio of the estimation error across different TD-based TMA schemes after

TD-splitting is applied.

4.4.2 Estimation Error Reduction

Suppose the noise-to-signal ratio reduction factor for flows with flow volume less than the optimal splitting threshold in the local TD (at node s) and the remote TD (at node d) are $\gamma_S(r_k)$ and $\gamma_D(r_k)$ respectively, as shown in Equation 4.19. Since all the flows in the same sub-TD suffer the same noise-to-signal ratio reduction factor and splitting threshold, we drop the r_k and $\hat{f}_s^S$ (as well as $\hat{f}_s^D$) when we are referring NSR_S and NSR'_S (NSR_D and NSR'_D) as well as γ_S (γ_D) in the following. According to our locally-optimal strategy stated in Section 4.1, we treat the noise-to-signal ratio at the remote node b before (NSR_D) and after TD-splitting (NSR'_D) being constant and $\gamma_D = \frac{NSR_D}{NSR'_D}$ for some constant γ_D.

Let ξ_{PU}, ξ_{QMLE} and ξ_{DL} be the changing ratio of the estimation error after TD-splitting for cardinality estimation of 2-sets intersection. We let $\sigma_z(S \cap D)$ and $\sigma'_z(S \cap D)$, for $z = \{PU, QMLE, DL\}$, be the standard estimation error of PU, QMLE and DATALITE before and after splitting respectively. According to Equations 3.1, 3.2 and 3.3, we have

$$\begin{aligned}
&\xi_{PU} \\
=\;& \frac{\sigma'_{PU}(S \cap D)}{\sigma_{PU}(S \cap D)} \\
=\;& \frac{\frac{1}{\sqrt{m/2}} \cdot \sqrt{NSR'_S + NSR'_D - 1}}{\frac{1}{\sqrt{m}} \cdot \sqrt{NSR_S + NSR_D - 1}} \\
=\;& \frac{\sqrt{2} \cdot \sqrt{NSR'_S + NSR'_D - 1}}{\sqrt{NSR_S + NSR_D - 1}} \\
\approx\;& \begin{cases} \frac{\sqrt{2} \cdot \sqrt{NSR'_S}}{\sqrt{NSR_S}} = \sqrt{\frac{2}{\gamma_S}}, & \text{if } NSR_S >> NSR_D \text{ and } NSR'_S >> NSR'_D \\ \frac{\sqrt{2} \cdot \sqrt{NSR'_D}}{\sqrt{NSR_D}} = \sqrt{\frac{2}{\gamma_D}}, & \text{if } NSR_S << NSR_D \text{ and } NSR'_S << NSR'_D \\ \frac{\sqrt{2} \cdot \sqrt{2NSR'_S}}{\sqrt{2NSR_S}} = \sqrt{\frac{2}{\gamma_S}}, & \text{if } NSR_S \approx NSR_D \text{ and } NSR'_S \approx NSR'_D \end{cases}
\end{aligned} \tag{4.20}$$

$$\begin{aligned}
&\xi_{QMLE} \\
=\;& \frac{\sigma'_{QMLE}(S \cap D)}{\sigma_{QMLE}(S \cap D)} \\
=\;& \frac{\sqrt{2} \cdot \sqrt{NSR'_S + NSR'_D - 2}}{\sqrt{NSR_S + NSR_D - 2}} \\
\approx\;& \begin{cases} \frac{\sqrt{2} \cdot \sqrt{NSR'_S}}{\sqrt{NSR_S}} = \sqrt{\frac{2}{\gamma_S}}, & \text{if } NSR_S >> NSR_D \text{ and } NSR'_S >> NSR'_D \\ \frac{\sqrt{2} \cdot \sqrt{NSR'_D}}{\sqrt{NSR_D}} = \sqrt{\frac{2}{\gamma_D}}, & \text{if } NSR_S << NSR_D \text{ and } NSR'_S << NSR'_D \\ \frac{\sqrt{2} \cdot \sqrt{2NSR'_S}}{\sqrt{2NSR_S}} = \sqrt{\frac{2}{\gamma_S}}, & \text{if } NSR_S \approx NSR_D \text{ and } NSR'_S \approx NSR'_D \end{cases}
\end{aligned} \tag{4.21}$$

$$
\begin{aligned}
& \xi_{_{DL}} \\
= \; & \frac{\sigma'_{_{DL}}(S \cap D)}{\sigma_{_{DL}}(S \cap D)} \\
= \; & \frac{\frac{1}{\sqrt{m/2}} \cdot (NSR'_S + NSR'_D - 1)}{\frac{1}{\sqrt{m}} \cdot (NSR_S + NSR_D - 1)} \\
= \; & \frac{\sqrt{2} \cdot (NSR'_S + NSR'_D - 1)}{NSR_S + NSR_D - 1} \\
\approx \; & \begin{cases} \frac{\sqrt{2} \cdot NSR'_S}{NSR_S} = \frac{\sqrt{2}}{\gamma_S}, & \text{if } NSR_S >> NSR_D \text{ and } NSR'_S >> NSR'_D \\ \frac{\sqrt{2} \cdot NSR'_D}{NSR_D} = \frac{\sqrt{2}}{\gamma_D}, & \text{if } NSR_S << NSR_D \text{ and } NSR'_S << NSR'_D \\ \frac{\sqrt{2} \cdot 2NSR'_S}{2NSR_S} = \frac{\sqrt{2}}{\gamma_S}, & \text{if } NSR_S \approx NSR_D \text{ and } NSR'_S \approx NSR_D \end{cases}
\end{aligned}
\tag{4.22}
$$

From the above equations, we can see that when the NSR of the remote node dominates that of the local node, the estimation error reduction only depends on the NSR reduction of the remote node which is consistent to our local "bottleneck" assumption. Moreover, the relationship between the estimation error changing ratio and the NSR reduction ratio is as follow:

$$
\xi_{_{DL}} \approx \frac{\sqrt{2}}{\gamma}, \; \xi_{_{PU}}, \xi_{_{QMLE}} \approx \sqrt{\frac{2}{\gamma}} \tag{4.23}
$$

for γ either equal to γ_S or γ_D which depends on the situations stated above. Similar to the flows with flow-volume larger than optimal splitting threshold, we can obtain the same results.

From Equation 4.23, we can observe that the reduction in the estimation error of DATALITE after TD-splitting is greater than that of PU and QMLE given the same NSR reduction factor γ

Flow volume Distribution	$\gamma = \frac{NSR}{NSR'}$		Estimation Error Changing Ratio $(\xi_{DL} = \frac{\sigma'}{\sigma})$		(ξ_{PU}, ξ_{QMLE})	
	$r_k < \hat{f}_s$	$r_k \geq \hat{f}_s$	$r_k < \hat{f}_s$	$r_k \geq \hat{f}_s$	$r_k < \hat{f}_s$	$r_k \geq \hat{f}_s$
Uniform	7.35	1.16	0.19	1.22	0.52	1.31
Gamma	4.95	1.25	0.29	1.13	0.64	1.26
Pareto	2.59	1.63	0.55	0.87	0.88	1.11

Table 4.2: The changing ratio of the estimation error before and after splitting of various flow-volume distributions. (The distribution parameters of Gamma are $k = 4, \theta = 3.5 \times 10^5$ and Pareto are $k = 2, r_{min} = 5 \times 10^5$.)

(using the optimal splitting threshold obtained by Equation 4.9) due to the characteristic of the estimation scheme. The estimation error would only be reduced when the NSR reduction factor is larger than 2 for PU and QMLE, i.e., $\gamma > 2$ while the minimum NSR reduction factor of DATALITE for estimation error reduction is $\sqrt{2}$, i.e., $\gamma > \sqrt{2}$. Therefore, we can conclude that DATALITE takes greater benefit in estimation error reduction by applying TD-splitting scheme than PU and QMLE under the same TD-splitting strategy.

The comparison of the changing ratio[2] of the estimation error before and after splitting with splitting threshold $\hat{f}_s$ using

[2] In general, the estimation errors of flows with flow volume greater than $\hat{f}_s$ are just few percents and thus, the estimation errors are still small even though they are slightly increased. While the estimation errors of flows with flow volume smaller than $\hat{f}_s$ are very large for DATALITE, they are greatly reduced.

various schemes is shown in Table 4.2. We can observe that the estimation errors can be reduced for flows with volume less than the optimal splitting threshold (i.e., $\xi_{DL}, \xi_{PU}, \xi_{QMLE} < 1$, for $r_k < \hat{f}_s$) and the estimation error reduction of DATALITE using TD-splitting is greater than that of PU (as well as QMLE). The estimation errors are increased for flows with volume greater than or equal to the optimal splitting threshold (i.e., $\xi_{DL}, \xi_{PU}, \xi_{QMLE} > 1$, for $r_k \geq \hat{f}_s$), except Pareto distributed flow-volume for DATALITE. Such increase in estimation errors, due to the dominated effect of the TD size reduction, for DATALITE is less than PU (and QMLE). In conclusion, the estimation error improvement for DATALITE is better than for PU and QMLE which will be validated via simulations in Chapter 6.

4.5 Recursive Splitting

In Section 4.3, we have discussed the single-level TD-splitting scheme which reduces the noise-to-signal ratio by partitioning the flows according to the flow volume and map them into corresponding sub-TDs. However, in some cases, there are large numbers of tiny flows in a link/node and the noise-to-signal ratios of the flows are still large even though the large and small flows are separated into two different sub-TDs. This motivates us to recursively split the TD into even more equal-sized sub-TDs, so that the flows are further separated into smaller groups. This is to further reduce the noise-to-signal ratio experienced by those *mice* flows.

4.5.1 Minimizing Maximum Estimation Error

By fixing the total memory size of the traffic digests on a particular link (node), we split a TD into 4 small pieces with equal memory size and the flows are assigned to 4 different sub-TDs according to 3 splitting thresholds. Similar to the formulation of single-level TD-splitting discussed in Section 4.3.1, we assume uniform loading of the node and links. And therefore, according to Equation 4.8, we are going to minimize the NSR of the smallest flow in each sub-TD after splitting.

We define $\mathcal{R}$ as the total traffic volume passing through the link (node) and $\mathcal{R}(f_{low}, f_{up})$ be the total traffic volume of each sub-TD, as shown in Equation 4.7. The noise-to-signal ratio of the k-th flow, $NSR(r_k, f_1, f_2, f_3)$, are defined as,

$$\begin{aligned} & NSR(r_k, f_1, f_2, f_3) \\ = \; & \frac{\mathcal{R}(r_{min}, f_1)}{r_k} I(r_k < f_1) + \frac{\mathcal{R}(f_1, f_2)}{r_k} I(f_1 \leq r_k < f_2) \\ & + \frac{\mathcal{R}(f_2, f_3)}{r_k} I(f_2 \leq r_k < f_3) + \frac{\mathcal{R}(f_3, r_{max})}{r_k} I(f_3 \leq r_k \leq r_{max}) \end{aligned} \tag{4.24}$$

where $[r_{min}, r_{max}]$, $g(r)$ are the range and the distribution of the traffic volume of the participating flows respectively. I is the indicator function; r_k is the flow volume of the k-th flows, $[f_{low}, f_{up}]$ are the flow volume range of the flows which are mapped to the corresponding sub-TD.

According to Equation 4.8 in Section 4.3, the NSR of the smallest flows in the 4 sub-TDs are minimized when their NSRs become equal. And so, the problem can be formulated as follows:

$$\begin{cases} NSR(r_{min}, f_1, f_2, f_3) & = & NSR(f_1, f_1, f_2, f_3) \\ NSR(r_{min}, f_1, f_2, f_3) & = & NSR(f_2, f_1, f_2, f_3) \\ NSR(r_{min}, f_1, f_2, f_3) & = & NSR(f_3, f_1, f_2, f_3) \end{cases} \tag{4.25}$$

where $NSR(r, f_1, f_2, f_3)$ is defined in Equation 4.24, for $r \in [r_{min}, r_{max}]$

By re-arranging the terms, we have,

$$\begin{cases} \frac{\mathcal{R}(r_{min},f_1)}{r_{min}} - \frac{\mathcal{R}(f_1,f_2)}{f_1} & = & 0 \\ \frac{\mathcal{R}(r_{min},f_1)}{r_{min}} - \frac{\mathcal{R}(f_2,f_3)}{f_2} & = & 0 \\ \frac{\mathcal{R}(r_{min},f_1)}{r_{min}} - \frac{\mathcal{R}(f_3,r_{max})}{f_3} & = & 0 \end{cases} \tag{4.26}$$

By solving the above equations, the solutions $(\hat{f}_1, \hat{f}_2, \hat{f}_3)$ gives the optimal splitting thresholds which minimized the maximum estimation error of all the participating flows.

4.5.2 Minimizing R.M.S. Estimation Error

Similar to the single-level TD-splitting algorithm, instead of minimizing the maximum upper bound of the estimation error, we also formulate our problem as minimizing the r.m.s. upper bound of the estimation error by varying different thresholds.

To determine the optimal TD-splitting thresholds for PU, similar to Equation 4.14, the problem can be formulated as:

$$\min_{f_1,f_2,f_3} \sum_{k=1}^{n} NSR(r_k, f_1, f_2, f_3) \tag{4.27}$$

Given the flow-volume distribution $g(r)$ on the link, the corresponding optimal splitting thresholds $(\hat{f}_1, \hat{f}_2, \hat{f}_3)$ can be expressed as:

$$\begin{aligned} & (\hat{f}_1, \hat{f}_2, \hat{f}_3) \qquad (4.28) \\ = \; & arg\{ \min_{f_1,f_2,f_3} \sum_{k=1}^{n} NSR(r_k, f_1, f_2, f_3)\} \\ = \; & arg\{ \min_{f_1,f_2,f_3} \; [\int_{r_{min}}^{f_1} \frac{\mathcal{R}(r_{min}, f_1)}{r} g(r)dr + \int_{f_1}^{f_2} \frac{\mathcal{R}(f_1, f_2)}{r} g(r)dr \\ & + \int_{f_2}^{f_3} \frac{\mathcal{R}(f_2, f_3)}{r} g(r)dr + \int_{f_3}^{r_{max}} \frac{\mathcal{R}(f_3, r_{max})}{r} g(r)dr]\} \end{aligned}$$

Similar to the formulation of single-level TD-splitting for DATALITE as shown in Equation 4.16, we assume the nodes and the links are uniformly loaded, the corresponding optimal splitting thresholds $(\hat{f}_1, \hat{f}_2, \hat{f}_3)$ of DATALITE can be expressed as:

$$\begin{aligned}
& (\hat{f}_1, \hat{f}_2, \hat{f}_3) \qquad (4.29) \\
= \; & arg\{ \min_{f_1,f_2,f_3} \sum_{k=1}^{n} NSR(r_k, f_1, f_2, f_3)^2 \} \\
= \; & arg\{ \min_{f_1,f_2,f_3} \; [\int_{r_{min}}^{f_1} \frac{\mathcal{R}(r_{min}, f_1)^2}{r^2} g(r)dr + \int_{f_1}^{f_2} \frac{\mathcal{R}(f_1, f_2)^2}{r^2} g(r)dr \\
& + \int_{f_2}^{f_3} \frac{\mathcal{R}(f_2, f_3)^2}{r^2} g(r)dr + \int_{f_3}^{r_{max}} \frac{\mathcal{R}(f_3, r_{max})^2}{r^2} g(r)dr] \}
\end{aligned}$$

As a result, we can solve Equations 4.29 and 4.29 numerically using line-search [21] to obtain the optimal splitting thresholds by given any distribution $g(r)$.

□ **End of chapter.**

Chapter 5

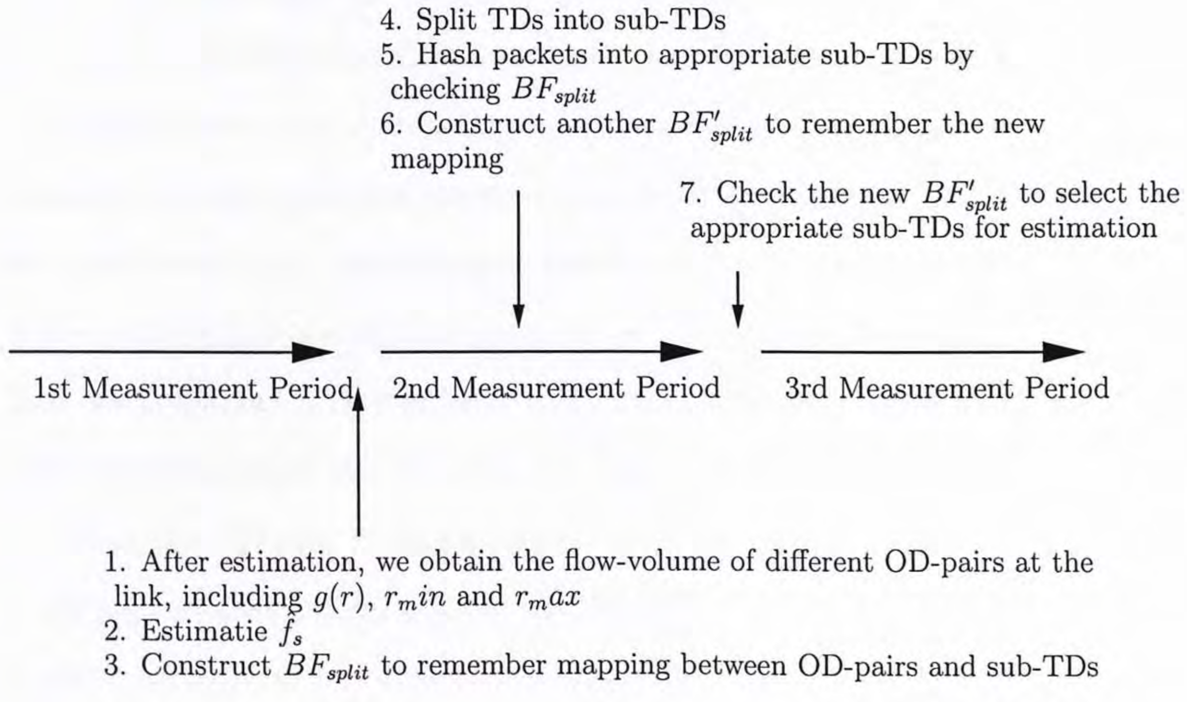
Realization of TD-splitting for Network Traffic Measurement

In the previous chapter, we study how to determine optimal TD-splitting threshold for each traffic digest (of a link). In this chapter, we discuss how to realize the proposed TD-splitting scheme for network-wide traffic measurement and monitoring.

As discussed in Chapter 4, Section 4.1, we take a loosely-coupled, decentralized approach where the TD-splitting threshold is determined on a per-digest (i.e., per-link) basis. In particular, the TD-splitting threshold value for a target link is updated periodically at the end of each measurement period, T, say, = 60 seconds, according to the estimated per-flow volume as well as overall traffic flow distribution on this link during the past measurement period.

Consider the case where a flow is defined by its origin and destination node-pair (OD-pair). Figure 5.1 depicts the procedure for applying TD-splitting on practical network. We can determine the mapping between OD-pairs and different sub-TDs associated with the target link according to the volume of each OD-pair and the TD-splitting threshold value based on the past measurement period (Step 1-2). After that, we use a Bloom filter (BF_{split}) to remember the mapping (Step 3). In particular, BF_{split} is described in Section 5.1.2. When a new measurement period begins, the TDs are split into different sub-TDs and the OD-pair of each incoming packet will be used to check against this flow-to-sub-TD mapping with BF_{split} to determine the sub-TD that the packet should be hashed into (Step 4-5). After TD-splitting is applied, we also need to remember this flow-to-sub-TD mapping by another BF'_{split} to enable correct estimation of individual traffic volume at the end of the measurement period (Step 6-7).

In the following sections, we describe how to maintain the sub-TD membership mapping of individual flows using the Bloom filter [4, 5] data structure. We will also describe the overall procedure to apply the TD-splitting scheme for network traffic measurement in detail.



4. Split TDs into sub-TDs
5. Hash packets into appropriate sub-TDs by checking BF_{split}
6. Construct another BF'_{split} to remember the new mapping

7. Check the new BF'_{split} to select the appropriate sub-TDs for estimation

1. After estimation, we obtain the flow-volume of different OD-pairs at the link, including $g(r)$, r_min and r_max
2. Estimatie f_s
3. Construct BF_{split} to remember mapping between OD-pairs and sub-TDs

Figure 5.1: Procedure for applying TD-splitting on practical network.

5.1 Tracking Sub-TD Membership

5.1.1 Controlling the Noise due to Non-Existent Flows on a Target Link

First, even though there is no packet passing through a target link for a particular OD-flow during a measurement period, the estimation of the intersection of the corresponding packet sets may still be non-zero due to the estimation error. Such small by non-zero noisy measurements are particularly problematic when we want to use the measurement results to determine network-wide routing pattern of individual flows. To address this problem, we construct a Bloom filter (BF_{mon}) to represent the set of OD-pairs hashed to the TD/sub-TD. For each incoming packet, we treat the OD-pair as an element of a set and add it to BF_{mon} during the measurement period. At the end of the measurement period, BF_{mon} will be broadcasted with the TDs to other nodes. We can then check the existence of the OD-pair in a TD before estimation. If the OD-pair exists in the TD, we do estimation; otherwise, we return zero for that OD-flow per link count. Since Bloom filter allows false positive, the non-zero link will still return estimation value. By applying this membership queries, we do not need to estimate all the combinations of the OD-pairs and links, and hence, the estimation time is shortened.

5.1.2 Sub-TD Membership Tracking for Single-level TD-splitting

Consider the single-level TD-splitting case where the TD of the target link is split into 2 sub-TDs and the packets are hashed to different sub-TDs according to their flow volume and the splitting thresholds obtained in the previous measurement period. We construct a Bloom filter (BF_{split}) to represent the set of OD-pairs which are mapped to the large sub-TD and use the source and destination address of the incoming packets as the key for BF_{split} to remember the set of the OD-pairs hashed to the large sub-TD in the current measurement period. Even though a Bloom filter may have false positive errors, i.e., OD-pairs with small flow-volume got mapped by mistake to the large sub-TD, the noise-to-signal ratio (and thus, the flow-volume estimation error) of other flows in the large sub-TD would only be increased slightly. This also explains why we use the Bloom filter to track the membership of the large sub-TD rather than that of the small sub-TD: in the latter alternative, false positives due to Bloom filter will lead to the incorrect mapping of a large flow into the small sub-TD, resulting in significant increase of estimation errors of the *mice* flows.

5.1.3 Sub-TD Membership Tracking under Recursive Splitting

For recursive TD-splitting, we further divide the flows into smaller groups and split the TDs into smaller sub-TDs. Therefore, more Bloom filters are needed. We construct $num\_of\_subtd/2$ Bloom filters to capture the membership information. For example, we split the TD into 4 sub-TDs and we only use 2 Bloom filters $\{BF_A, BF_B\}$ to remember the mapping of the OD-pairs of the corresponding sub-TDs $\{TD_1, TD_2, TD_3, TD_4\}$ which are ordered by the flow volume, (i.e., the flow volume of the OD-pair mapped to TD_1 are smaller than those in TD_2 and those in TD_2 are smaller than those in TD_3, and so on). The OD-pairs existence results are shown in Table 5.1. If the OD-pair exists in both Bloom filter, the OD-pair belongs to TD_4; if the OD-pair exists in BF_A but not in BF_B, it belongs to TD_3; if the OD-pair exists in BF_B but not in BF_A, it belongs to TD_2; otherwise, it belongs to TD_1. With this mapping scheme, only small flows are mapped to those sub-TDs with flow-volume larger than themselves due to the false positive error. For example, the OD-pairs belongs to TD_1 may wrongly be mapped to TD_2, TD_3 and

TD_4 while the OD-pairs belongs to TD_3 may only wrongly be mapped to TD_4, etc.

	BF_A	BF_B
TD_1	Does Not Exist	Does Not Exist
TD_2	Does Not Exist	Exists
TD_3	Exists	Does Not Exist
TD_4	Exists	Exists

Table 5.1: Mapping of Bloom filters and the corresponding sub-TDs for 2-level recursive TD-splitting

Once the Bloom filters are defined for a link as described above, we can use them to track the OD-flows hashed to each TD and sub-TD as illustrated. In the next section, we describe the complete procedure of applying our TD-splitting with TD-based TMA scheme to provide network-wide traffic measurement.

5.2 Overall Operations to support TD-splitting for Network-wide Traffic Measurements

Figure 5.2 depicts the flowchart of applying TD-splitting to a TD-based TMA scheme to support network-wide traffic measurement. In practice, we construct 2 TDs on each link for packets passing through in both directions and construct 2 TDs on each node for packets originating from and terminating at the node at the beginning of the measurement interval. During the first measurement period, since we do not have the measurement history and the prior splitting threshold, we apply the original (non-TD-split) TMA scheme as described in Chapter 1 and hash all incoming packets to a single TD and BF_{mon} in order to capture the OD-pair members of the local TD. At the end of the measurement period, we distribute the TDs and BF_{mon} to other nodes to obtain an initial estimation. For every measurement period, we can then obtain the network-wide OD-flow per link count based on the cardinality estimation of the intersection sets using the TDs.

With the network-wide traffic aggregates, we can estimate the distribution, minimum and maximum value of the flow volume for each TD and hence, the splitting thresholds can be calculated by the surrogate methods introduced in Chapter 4. Based on the splitting thresholds, we split the TD and construct BF'_{split} to remember the OD-pair membership of each sub-TD (For recursive splitting of TD into 4 pieces, $BF'_{split} = \{BF'_A, BF'_B\}$ as described above).

Starting from the second measurement period, we hash the packets to the corresponding sub-TD by checking the membership of BF'_{split}. Since some of the OD-pairs may not exist in this measurement period, we construct another Bloom filter, BF_{split} to remember the actual OD-pairs membership of the sub-TDs in order to reduce the false positive rate during estimation. At the end of the measurement period, we select the appropriate sub-TD for estimation by checking the membership through BF_{mon} and BF_{split}. With the new traffic aggregates, we estimate the new splitting thresholds and re-initialize BF'_{split} for the next measurement period.

5.2.1 Computation Time for TD-splitting

At the end of each measurement period, the computation time needed for TD-splitting would be consisting of three different processes: (i) the time used for the single parameter line search [21] for the optimal splitting threshold estimation; (ii) the time used for checking the existence of an OD-pair at the link or node for each OD-pair; and (iii) the time used for the sub-TD membership checking for the OD-pairs existed at the link/node. For a given precision requirement, the single-parameter line-search in (i) can be completed in constant time. To keep the false positive probability of the Bloom filters to be less than $p = 0.6185^{m/n}$, the time-complexity of (ii) and (iii) are both dominated by the evaluation of $k = m \log 2/n$ hashes where m is the number of

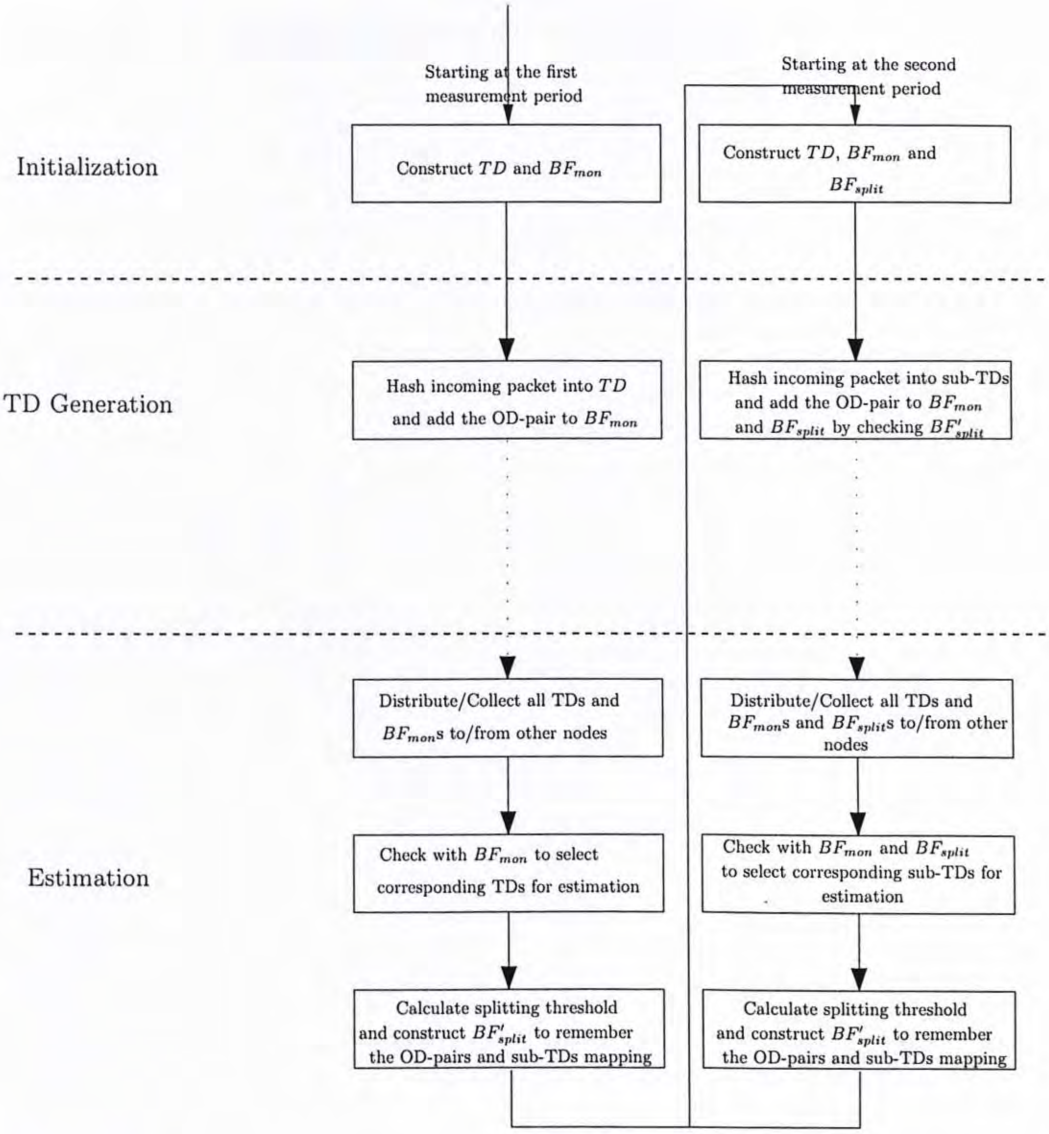

Initialization
TD Generation
Estimation
Starting at the first measurement period
Starting at the second measurement period
Construct TD and BF_{mon}
Construct TD, BF_{mon} and BF_{split}
Hash incoming packet into TD and add the OD-pair to BF_{mon}
Hash incoming packet into sub-TDs and add the OD-pair to BF_{mon} and BF_{split} by checking BF'_{split}
Distribute/Collect all TDs and BF_{mon}s to/from other nodes
Distribute/Collect all TDs and BF_{mon}s and BF_{split}s to/from other nodes
Check with BF_{mon} to select corresponding TDs for estimation
Check with BF_{mon} and BF_{split} to select corresponding sub-TDs for estimation
Calculate splitting threshold and construct BF'_{split} to remember the OD-pairs and sub-TDs mapping
Calculate splitting threshold and construct BF'_{split} to remember the OD-pairs and sub-TDs mapping

Figure 5.2: Overall Procedure to apply TD-splitting scheme in practice

bits in the Bloom filter and n is the number of distinct OD-pairs present. In practice, k is typically chosen to be a small constant, e.g., < 10, by provisioning m with respect to n.

□ **End of chapter.**

Chapter 6

Performance Evaluation

In this chapter, we present the simulation study of the proposed TD-splitting scheme for both generic and practical networks. We first demonstrate the validity and the superior estimation error performance of the proposed TD-splitting scheme. We then apply the proposed scheme on real traffic traces and topology of the Abilene Network [1].

6.1 Applying TD-splitting on Generic Network Topology

In this section, we construct a generic network with different flow-volume distributions to demonstrate the validity of our surrogate optimization objectives introduced in Equation 4.3 and Equation 4.4 via simulation. We then compare the estimation error performance of optimal TD-splitting to that of the original DATALITE scheme, the Proportional Union Method proposed in [8] and the Quasi-Likelihood Approach introduced in [7] by extending the latter to support OD-flows-per-link cardinality estimation as described in Appendix A.

6.1.1 Simulation Settings

Consider a set of OD-commodity flows where commodity k originates from source node S_k, terminates at different destination node D_k, and passes through link L with flow volume r_k, for $k \in [1, n]$, as illustrated in Figure 6.1. Each node (or link) also contains cross-traffic of other commodities (shown as dashed lines in Figure 6.1) with same traffic volume distribution as link L such that the total flow volume of each node/link is the same. This also results in the same noise-to-signal ratio for each commodity as it passes through different nodes. In the experiment, we estimate the traffic volume of the flow which passes through link L within each measurement period using different estimation schemes. To demonstrate the compatibility of our surrogate methods to different distributions, we evaluate the performance of TD-splitting under Uniform, Gamma and Pareto flow volume distributions. To ensure the loading of each traffic digest

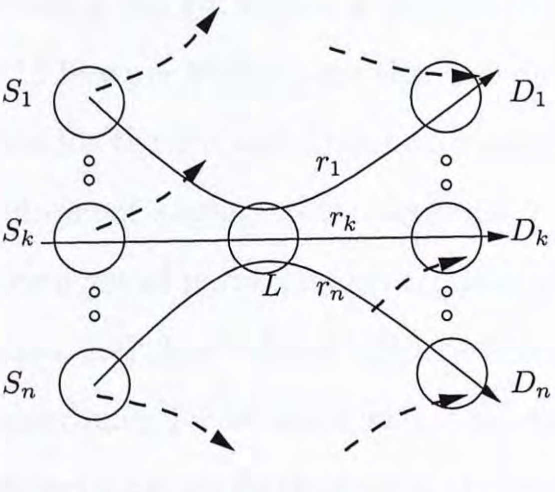


Figure 6.1: Flow pattern configuration for simulations

to be sufficiently high [6], we generate the total traffic volume of nodes/links to be around 8.2×10^7 (packets per measurement interval) with individual flow volume ranges between [1×10^5, 4×10^6], [2×10^5, 4.1×10^6] and [5×10^5, 1×10^7] packets per measurement interval for *Uniform Distribution*, *Gamma Distribution* with $k = 4$ and $\theta = 3.5 \times 10^5$ and *Pareto Distribution* with $k = 2$ respectively. In the following, we first show the validity of the proposed surrogate objective functions and then we show the estimation error across different flow-volume distribution under Single-level and 2-level recursive splitting.

We use $m = 2^{20}$ buckets and allocate 5 bits per bucket for DATALITE, and thus the memory size would be 640 KBytes, which is sufficient to measure upto 2^{32} packets per observation interval. Following the guidelines in [8], we set $m = 2^{19}$ buckets and allocate 15 bits per bucket, and thus the memory size would be 960 KBytes for the PU and QMLE to yield comparable total memory requirement among all the schemes.

For any given set of parameters (i.e., estimation scheme, objective function and flow-volume distribution), we repeat the simulation experiment for 10 times with different seeds and compute the root mean square relative error of flow volume estimate for each flow across the 10 experiments as follows:

$$RMSRE = \sqrt{\frac{1}{10}\sum_{t=1}^{10}(percentage\_error_t)^2}$$

where $percentage\_error_t$ is defined as the percentage error of the estimation during the t-th trial.

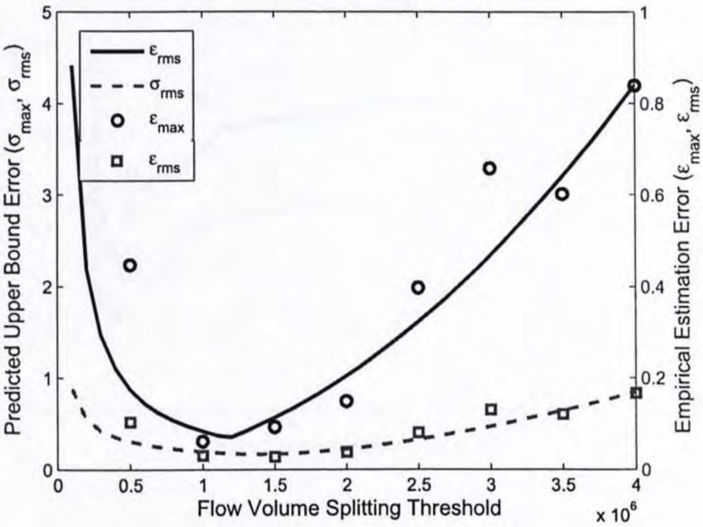
6.1.2 Validity of the Proposed Surrogate Objective Functions

In this section, we demonstrate the validity of our surrogate optimization objectives introduced in Equation 4.3 and Equation 4.4 via simulation. By varying the value of splitting threshold, the 3 panels of Figure 6.2 plot the corresponding value of σ_{max} (σ_{rms}) computed from Equation 4.3 (4.4) and compare it to its empirical RMSRE counterpart under the same splitting threshold value for the cases of Uniform, Gamma and Pareto flow-volume distributions. In particular, Figure 6.2 shows that the optimal splitting threshold value obtained via the minimization of these two surrogate objective functions also minimize the maximum (as well as r.m.s.) empirical RMSRE (ϵ_{max} and ϵ_{rms}) across all the participating flows under various flow-volume distributions for DATALITE. For example, with Uniform flow-volume distribution in Figure 6.2a, the minimum value of both of the computed and empirical maximum (as well as r.m.s.) error curves is achieved by setting f_s to 1.2×10^6 and 1.5×10^6 (packets per measurement interval) respectively. For Gamma flow-volume distribution, Figure 6.2b shows that a value of $f_s = 10^6$ (packets per measurement interval) can minimize both the computed and empirical maximum errors while $f_s = 1.2 \times 10^6$ (packets per measurement interval) would minimize both the computed and empirical r.m.s. errors. For Pareto flow-volume distribution in Figure 6.2c, the minimum value of computed and empirical maximum (as well as r.m.s.) error curves is achieved by setting f_s to 7.9×10^6 (9×10^6) (packets per measurement interval). Note also that the optimal splitting threshold values under Uniform and Gamma flow distributions are very close to each other, which suggests that a rough estimation of $g(r)$ would already serve well for the purpose of determining a near-optimal splitting threshold value.

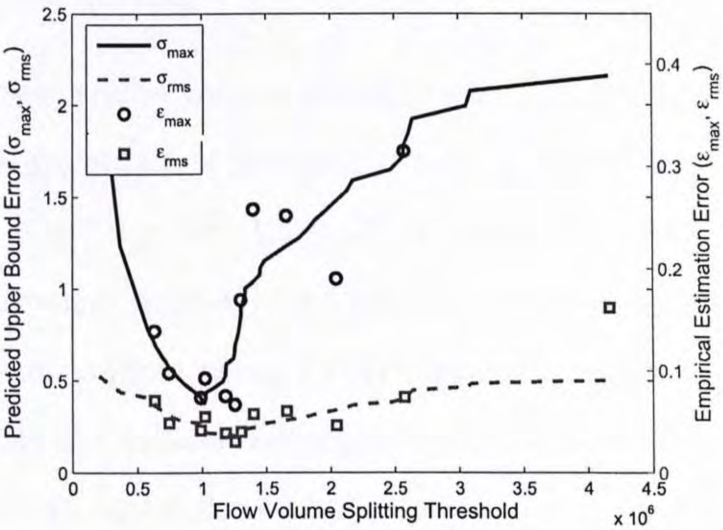
The figures also demonstrate that the splitting threshold is insensitive to the estimation error. By varying the splitting threshold, the maximum (as well as r.m.s.) estimation error of all the participating flows is increased slightly for Uniform and Gamma distributed flow-volume. Therefore, it also suggests that a rough estimation of $g(r)$ can help us to determine a near-optimal splitting threshold value.

6.1.3 Performance of Single-level TD-splitting

In this section, we compare the estimation error of different flow-volume estimation schemes including the original DATALITE, PU and QMLE schemes and a basic single-level TD-splitting for all schemes (i.e., 2 equal-sized sub-TDs per monitoring point). Since the memory size of PU (as well as QMLE) is $\frac{60}{40} = 1.5$ times of DATALITE, due to the different of the bucket size, we scale the estimation error of DATALITE by $\frac{1}{\sqrt{1.5}} \approx 0.8165$ in order to provide a fair comparison between the schemes. Figure 6.3 and 6.4 shows the empirical error (of PU) and scaled empirical RMSRE (of DATALITE) of different flows at a link

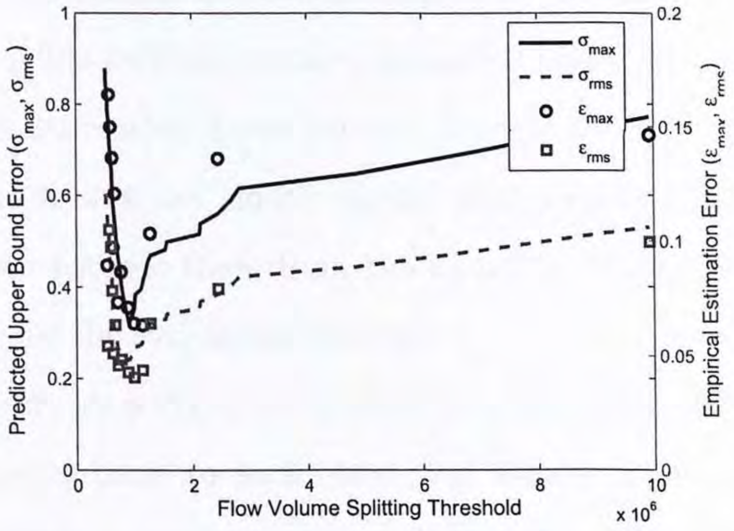


(a) Uniform Distribution



(b) Gamma Distribution

Figure 6.2: Comparison between the calculated σ_{max} (σ_{rms}) and the empirical estimation error.



(c) Pareto Distribution

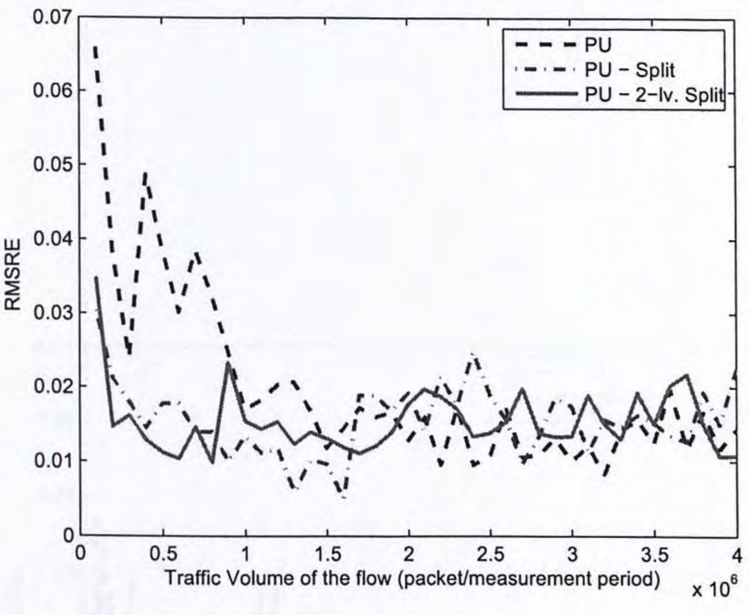
Figure 6.2: Comparison between the calculated σ_{max} (σ_{rms}) and the empirical estimation error. (Cont'd)

with different traffic volume distribution. The optimal splitting thresholds are obtained by minimizing σ_{rms} which are 1.6×10^6, 1.3×10^6 and 7.9×10^5 (1.5×10^6, 10^6 and 7.9×10^5) (packets per measurement interval) for Uniform, Gamma and Pareto distributed flow-volume using PU (DATALITE) respectively. The figure shows the substantial improvement in estimation error of the flows with small flow-volume for both PU and DATALITE after single-level TD-splitting (dot-dashed line) and the estimation errors of flows with large flow-volume are close to or slightly larger than the estimation error of original scheme.

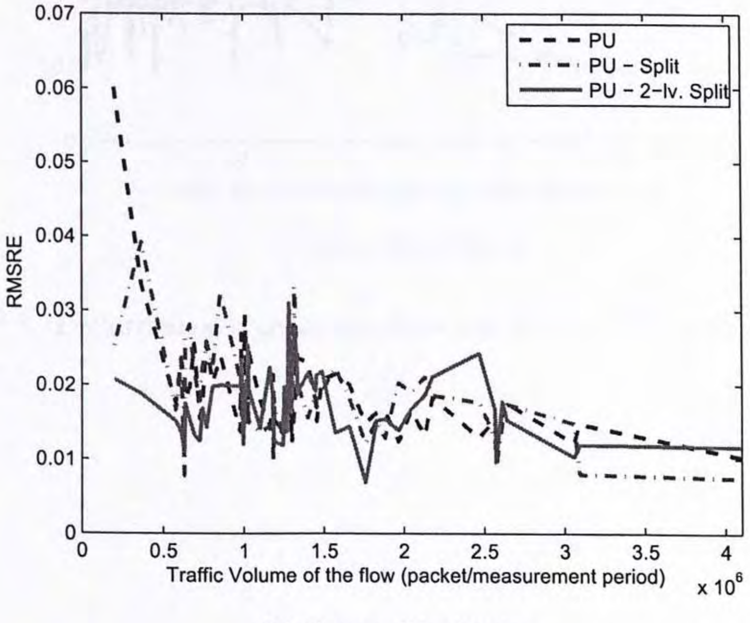
Figure 6.5 compares the estimation error of DATALITE using single-level TD-splitting to the original PU and QMLE schemes. Since the estimation error curves of single-level TD-splitting of PU and QMLE are closely clutter with that of the original scheme, we exclude them from the figure for clarity. Refer to Table 6.1 for the estimation results.

The figure shows that PU (dashed line) and QMLE (thin solid line) are very close to each other and always achieve smaller estimation error than the original DATALITE. The single-level TD-splitting scheme for DATALITE performs better than QMLE for small traffic flows and vice versa for large flows.

This is due to the significant reduction of the noise-to-signal ratios for small flows while the corresponding noise-to-signal reduction of large flows cannot outweigh the impact of reduction of memory size for the large-flow sub-TD as described in Section 4.4.2. Table 6.1 compares the r.m.s. error across all the participating flows under various estimation schemes. The single-level TD-splitting scheme reduce the r.m.s. estimation error of the original DATALITE by about 40-60% while PU and QMLE are only reduced by 10-20%. However, QMLE still gives the smallest r.m.s. estimation error comparing to all other schemes.

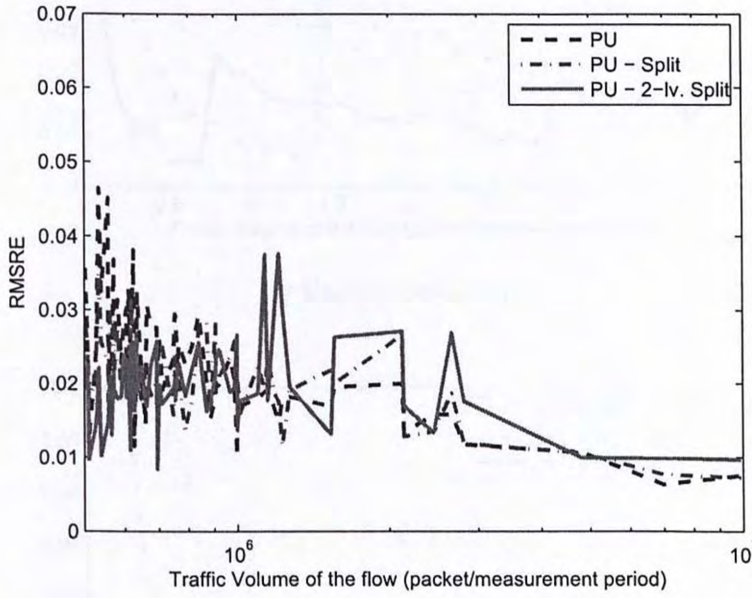


(a) Uniform Distribution



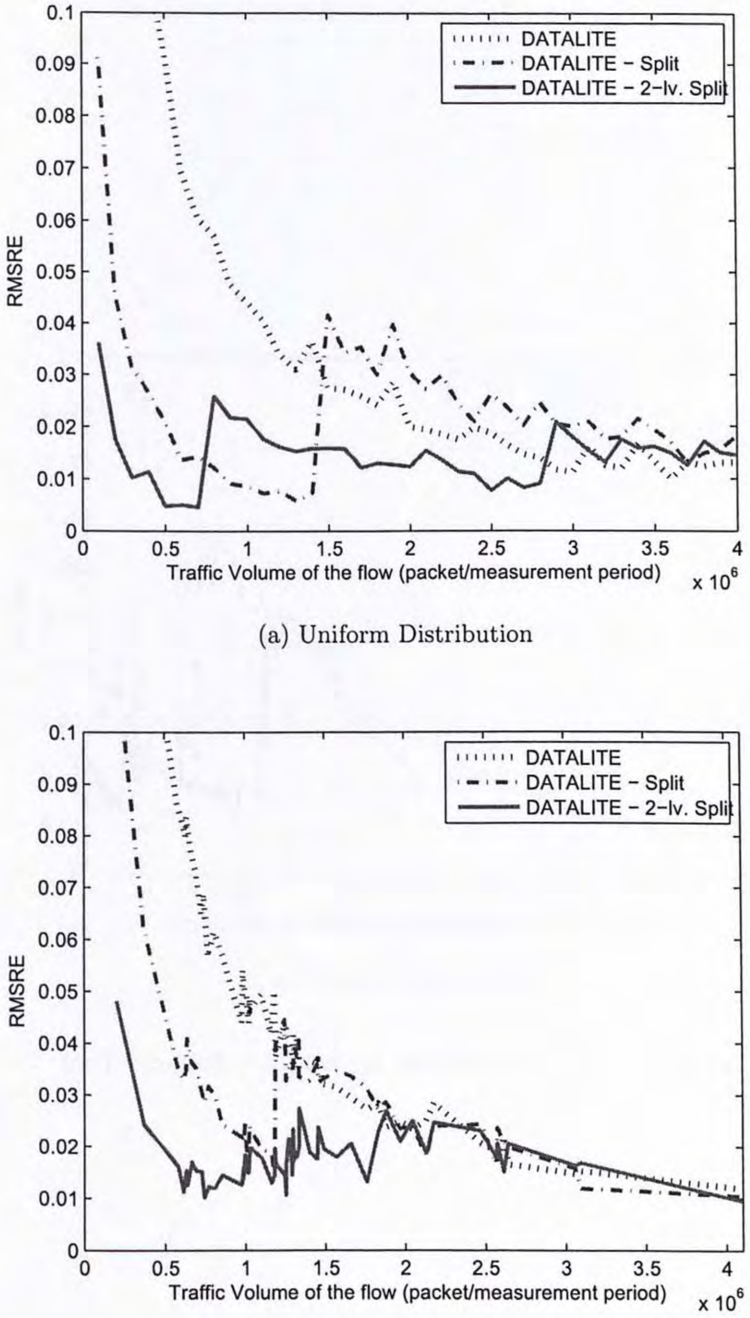
(b) Gamma Distribution

Figure 6.3: Performance comparison between different TD-splitting for PU.



(c) Pareto Distribution

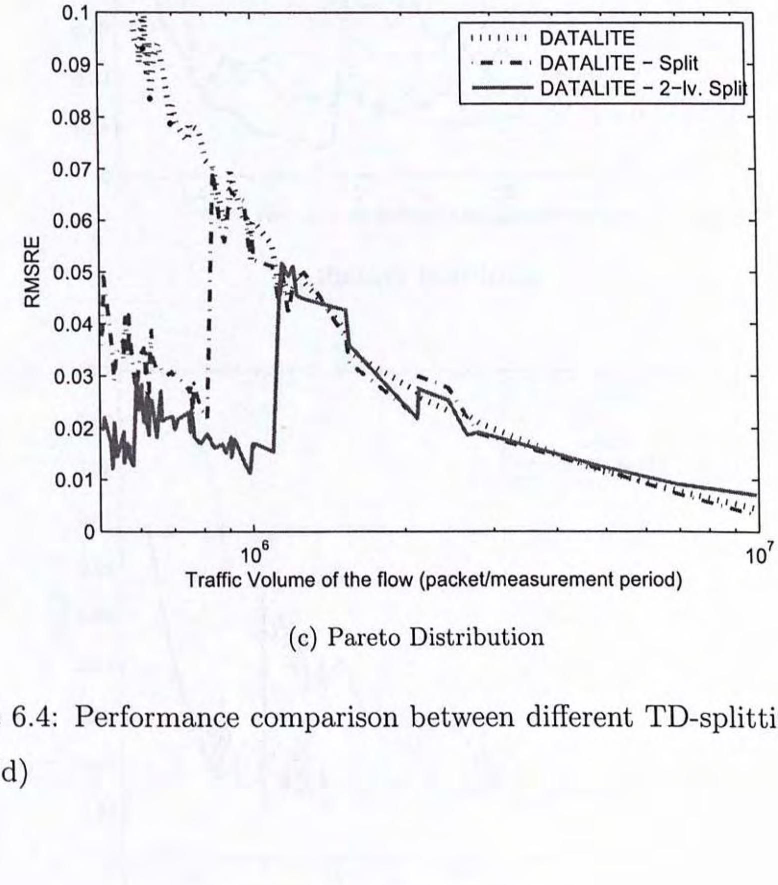
Figure 6.3: Performance comparison between different TD-splitting for PU. (Cont'd)



(a) Uniform Distribution

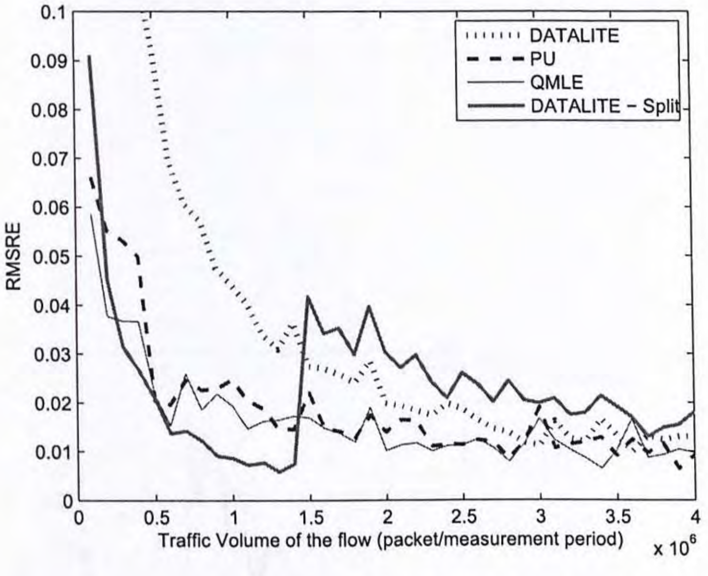
(b) Gamma Distribution

Figure 6.4: Performance comparison between different TD-splitting for DATALITE.

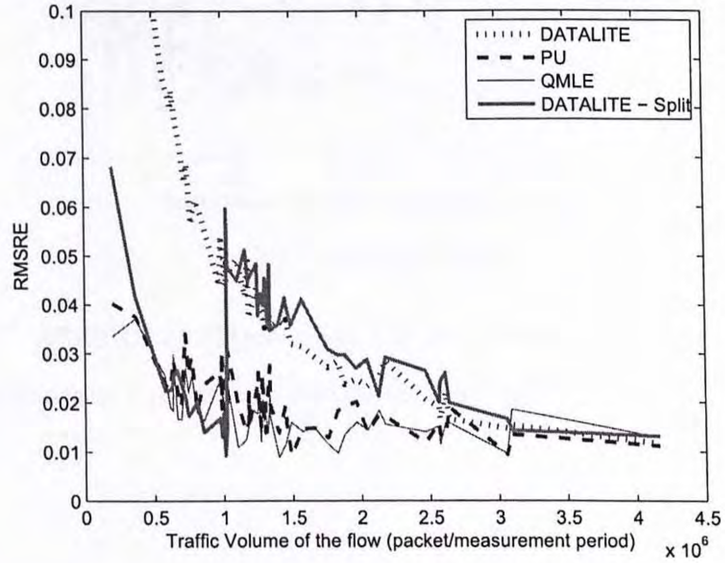


(c) Pareto Distribution

Figure 6.4: Performance comparison between different TD-splitting for Pu. (Cont'd)

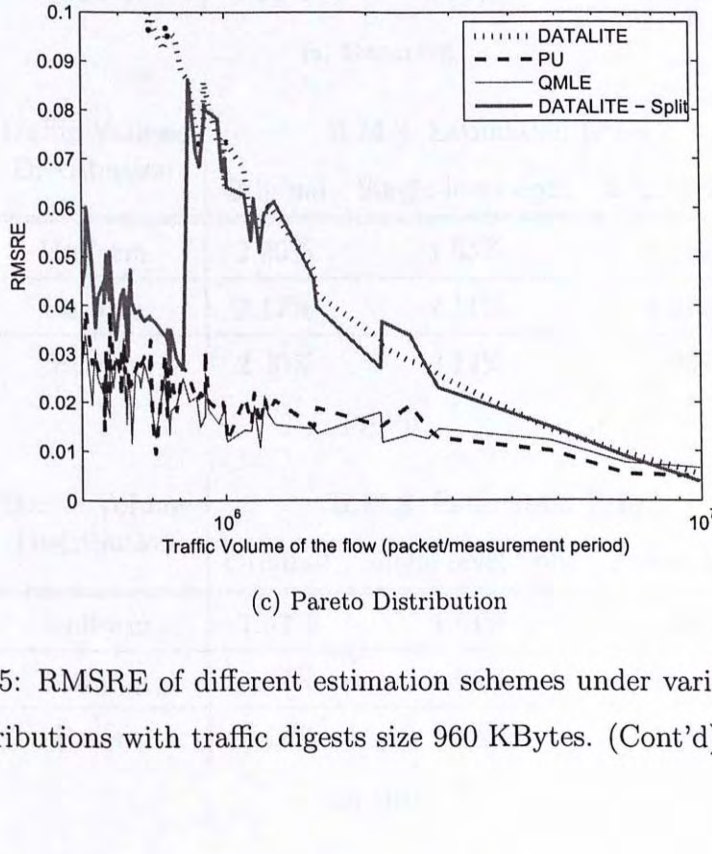


(a) Uniform Distribution



(b) Gamma Distribution

Figure 6.5: RMSRE of different estimation schemes under various flow volume distributions with traffic digests size 960 KBytes.



(c) Pareto Distribution

Figure 6.5: RMSRE of different estimation schemes under various flow volume distributions with traffic digests size 960 KBytes. (Cont'd)

Traffic Volume Distribution	R.M.S. Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	8.99%	2.72%	1.56%
Gamma	5.80%	3.37%	2.57%
Pareto	7.98%	3.91%	2.42%

(a) DATALITE

Traffic Volume Distribution	R.M.S. Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	2.29%	1.63%	1.60%
Gamma	2.12%	2.11%	2.27%
Pareto	2.40%	2.14%	2.06%

(b) PU

Traffic Volume Distribution	R.M.S. Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	1.93%	1.61%	1.39%
Gamma	1.87%	2.04%	1.95%
Pareto	2.12%	1.86%	1.62%

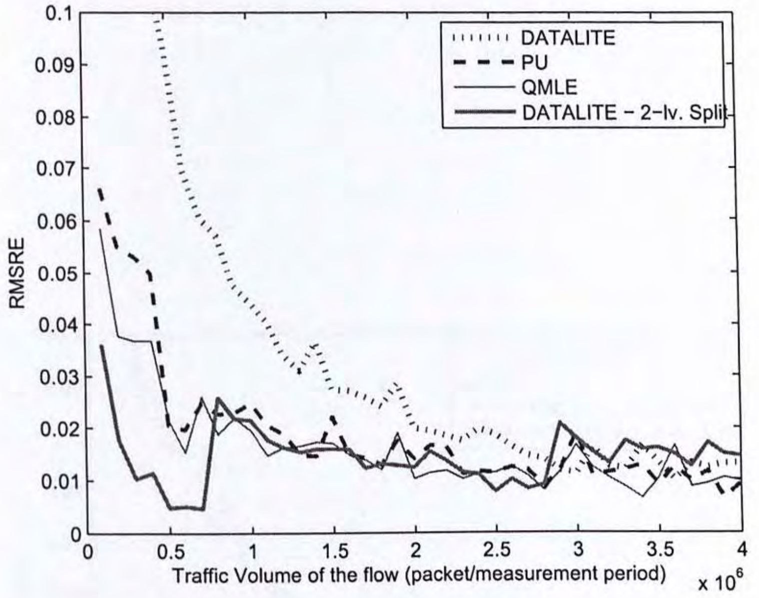
(c) QMLE

Table 6.1: Comparison of empirical root-mean-square relative error across all the participating flows by applying TD-splitting (using minimizing r.m.s. estimation error objective) to different TD-based TMA scheme.

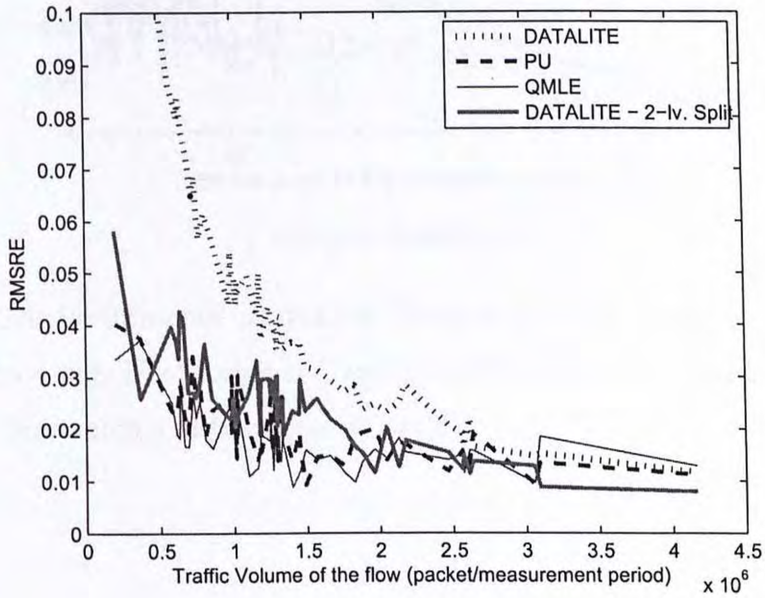
6.1.4 Performance of Recursive TD-splitting

To further improve the performance of estimation, we perform recursive splitting of the overall TD memory pool into equal-sized sub-TDs. For the case of 2-level recursive TD-splitting, it results in 4 equal-sized sub-TDs where the number of buckets per sub-TD becomes $m = 2^{18}$ for DATALITE and $m = 2^{17}$ for PU and QMLE. We can then partition the flows by calculating the corresponding optimal splitting thresholds recursively. Figures 6.3 and 6.4 show that the estimation errors of flows with small flow-volume (solid line) are further reduced for both PU and DATALITE. Figure 6.6 compares the error performance of the 2-level recursive TD-splitting (using minimization of r.m.s. error as the objective function) with PU and the QMLE scheme. As shown in the figure, the estimation error for large flows are very close under 2-level recursive TD-splitting of DATALITE (thick solid line) or PU (dashed line) (as well as QMLE (thin solid line)) while the former achieves much smaller estimation errors for small flows under Uniform Distribution.

The r.m.s. estimation error across all the participating flows are shown in Table 6.1. The r.m.s. estimation error of DATALITE is reduced by 80% after using 2-level recursive TD-splitting for Uniform distributed flow-volume. In other words, the 2-level re-

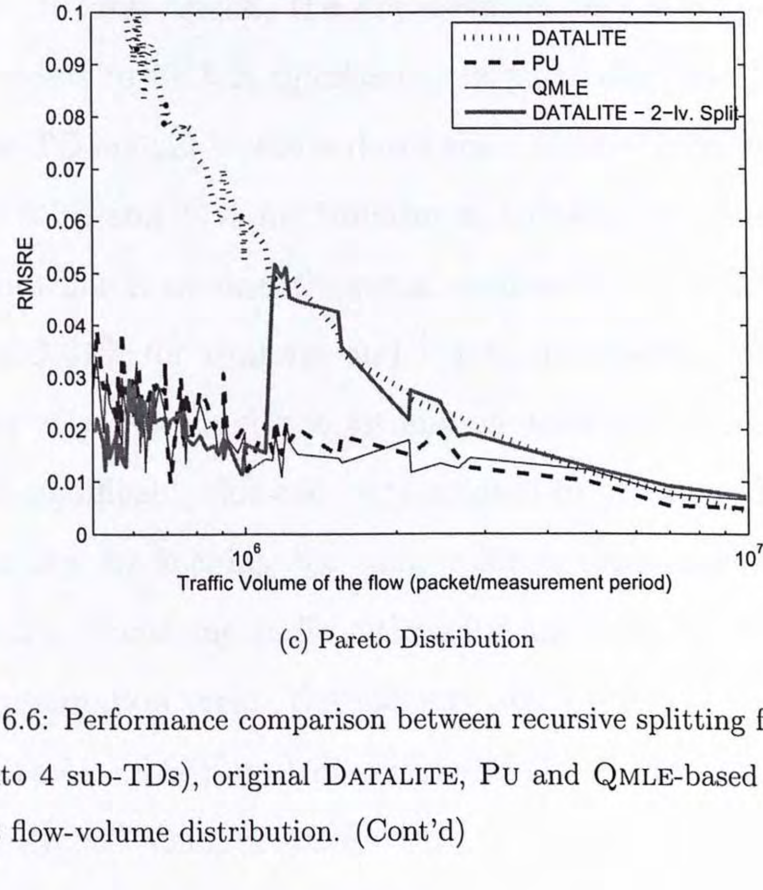


(a) Uniform Distribution



(b) Gamma Distribution

Figure 6.6: Performance comparison between recursive splitting for DATALITE (into 4 sub-TDs), original DATALITE, PU and QMLE-based one across various flow-volume distribution.



(c) Pareto Distribution

Figure 6.6: Performance comparison between recursive splitting for DATALITE (into 4 sub-TDs), original DATALITE, PU and QMLE-based one across various flow-volume distribution. (Cont'd)

cursive TD-splitting of DATALITE outperforms the state-of-the-art QMLE scheme [7] by about 20% in terms of r.m.s. relative estimation error in this case. The 2-level recursive TD-splitting also reduce the r.m.s. estimation of DATALITE by 55% and 70% for Gamma and Pareto distributed flow volume.

For PU and QMLE, the improvement in r.m.s. estimation error seems to be less significant. In particular, the 2-level recursive TD-splitting only reduces the r.m.s. estimation error of PU by 27% and 14% for Uniform and Pareto distributed flow-volume while it reduces the r.m.s. estimation error of QMLE by 28% and 31% for Uniform and Pareto distributed flow-volume respectively. Although the estimation error reduction seems to be less significant, this can be translated to substantial decrease in memory by keeping the same relative estimation error. In particular, according to Equations 2.7 and 2.14, by keeping the same estimation error, the memory size of the TD can be reduced to $(1-0.27)^2 \approx 0.53$ (53%) and $(1-0.14)^2 \approx 0.74$ (74%) of the original memory size for PU with Uniform and Pareto distributed flow-volume; and for QMLE, the memory requirement of the TD can be reduced to 52% and 48% of the original requirement for Uniform and Pareto distributed flow-volume which is approximately half of the original memory. However, in Gamma distributed flow-volume situation, the r.m.s. estimation errors of both schemes are slightly increased.

In practice, we are more concerned about the large estimation errors across all the participating flows rather than the average one. We compare the 95-th percentile of the estimation error across all the participating flows in different cases as shown in Table 6.2. Similar to the r.m.s. estimation error, the TD-splitting scheme results in the largest benefit for DATALITE. The 2-level recursive splitting reduces the 95-th percentile estimation error of DATALITE by 86%, 53% and 59% for Uniform, Gamma and Pareto distributed flow-volume respectively. For PU (and QMLE), the 95-th percentile estimation error is reduced by 58% (40%) and 17% (25%) for Uniform and Pareto distributed flow-volume respectively. It validates our analysis in Section 4.4.2 that the TD-splitting gives better estimation error improvement for DATALITE than PU and QMLE due to the estimation error characteristics of the scheme.

Table 6.3 shows the root-mean-square and 95-th percentile estimation error across all the participating flows in different flow-volume distribution using minimizing σ_{max} be the objective for optimal splitting threshold estimation. The single-level TD-splitting reduces the r.m.s. estimation error of DATALITE about

Traffic Volume Distribution	95-th Percentile Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	15.61%	4.16%	2.17%
Gamma	8.36%	4.83%	3.97%
Pareto	11.02%	6.39%	4.55%

(a) DATALITE

Traffic Volume Distribution	95-th Percentile Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	3.86%	2.24%	2.17%
Gamma	2.69%	3.10%	2.41%
Pareto	3.69%	2.84%	2.71%

(b) PU

Traffic Volume Distribution	95-th Percentile Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	3.67%	2.19%	2.21%
Gamma	2.75%	2.89%	2.99%
Pareto	3.00%	2.67%	2.26%

(c) QMLE

Table 6.2: Comparison of empirical 95-th percentile relative error across all the participating flows by applying TD-splitting (using minimizing r.m.s. estimation error objective) to different TD-based TMA scheme.

40-70% while the 2-level TD-splitting reduces the r.m.s. estimation error of DATALITE 57-80% . For PU, the r.m.s. estimation errors are reduced by 11-25% using single-level TD-splitting while the 2-level TD-splitting reduces the r.m.s. estimation error about 16-27%. The estimation error reduction across different schemes are similar to those using minimizing r.m.s. objectives for optimal splitting threshold estimation.

Traffic Volume Distribution	R.M.S. [95-th Percentile] Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	8.99% [15.61%]	2.83% [5.63%]	1.63% [2.65%]
Gamma	5.80% [8.36%]	3.58% [5.55%]	2.47% [3.60%]
Pareto	7.98% [11.02%]	3.91% [6.39%]	2.43% [3.67%]

(a) DATALITE

Traffic Volume Distribution	R.M.S. [95-th Percentile] Estimation Error		
	Original	Single-level Split	2-level Split
Uniform	2.29% [5.31%]	1.71% [2.35%]	1.69% [2.49%]
Gamma	2.02% [3.04%]	2.14% [3.10%]	1.92% [2.85%]
Pareto	2.42% [3.25%]	2.15% [2.84%]	2.01% [2.71%]

(b) PU

Table 6.3: Comparison of empirical root-mean-square and 95-th percentile relative error across all the participating flows by applying TD-splitting (using minimize maximum estimation error objective) to different TD-based TMA scheme.

6.1.5 Heterogeneous NSR Loading

In this section, we demonstrate the estimation error performance of different schemes under heterogeneous NSR loading, which the flow-volume distribution at the link/nodes are different, and thus the total flow-volumes are also different for different nodes and link. Therefore, the NSRs experienced by the flows at different nodes or link are different.

Figure 6.7 compares the error performance of the single-level TD-splitting (using minimization of r.m.s. error as the objective function) with PU scheme for heterogeneous NSR loading network. In Figure 6.7, the estimation error for small flows are greatly reduced under single-level TD-splitting of DATALITE (thick solid line) or PU (dashed line) while the former achieves much smaller estimation errors for small flows under Uniform Distribution.

The r.m.s. and the 95-th percentile estimation error across all the participating flows are shown in Table 6.4. The r.m.s. estimation error of DATALITE is reduced by 72%, 62%, 65% after using single-level TD-splitting for Uniform, Gamma and Pareto distributed flow-volume respectively. For PU, the reduction in r.m.s. estimation error is about 40%, 47% and 49%. The 95-th percentile estimation error of DATALITE is reduced from 62% to

77% while that of PU is reduced from 40% to 48% across different distributions. We show that our TD-splitting can reduce the estimation error of different flows in a heterogeneous NSR loading topology from 40% to 72% using different estimation schemes under single-level TD-splitting.

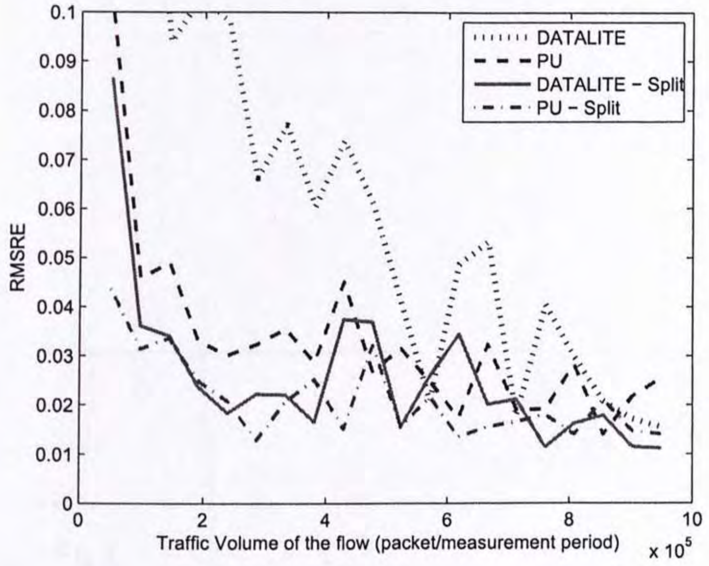
Traffic Volume Distribution	R.M.S. [95-th percentile] Estimation Error	
	Original	Single-level Split
Uniform	8.85% [13.36%]	2.51% [3.05%]
Gamma	5.32% [8.25%]	2.01% [3.17%]
Pareto	7.01% [10.93%]	2.42% [3.55%]

(a) DATALITE

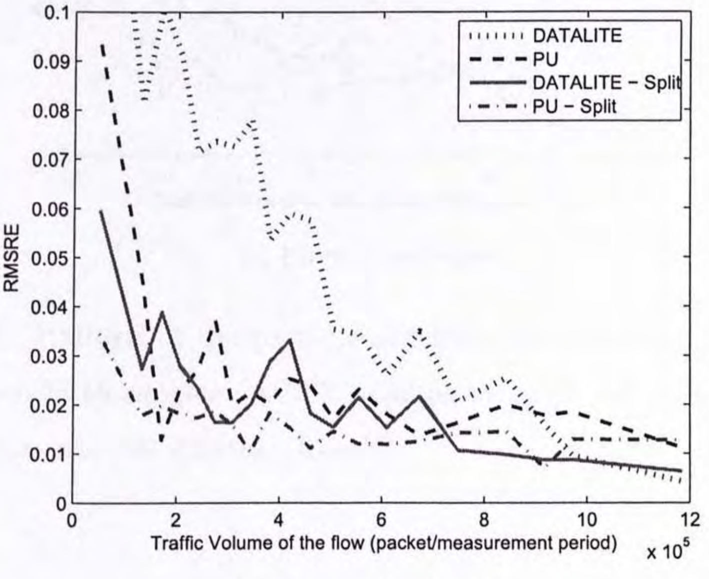
Traffic Volume Distribution	R.M.S. [95-th percentile] Estimation Error	
	Original	Single-level Split
Uniform	3.78% [4.90%]	2.27% [3.36%]
Gamma	3.07% [4.53%]	1.64% [1.97%]
Pareto	3.46% [5.14%]	1.77% [2.92%]

(b) PU

Table 6.4: Comparison of empirical r.m.s. and 95-th percentile relative error across all the participating flows by applying single-level TD-splitting to different TD-based TMA scheme under Heterogeneous NSR Loading Topology.

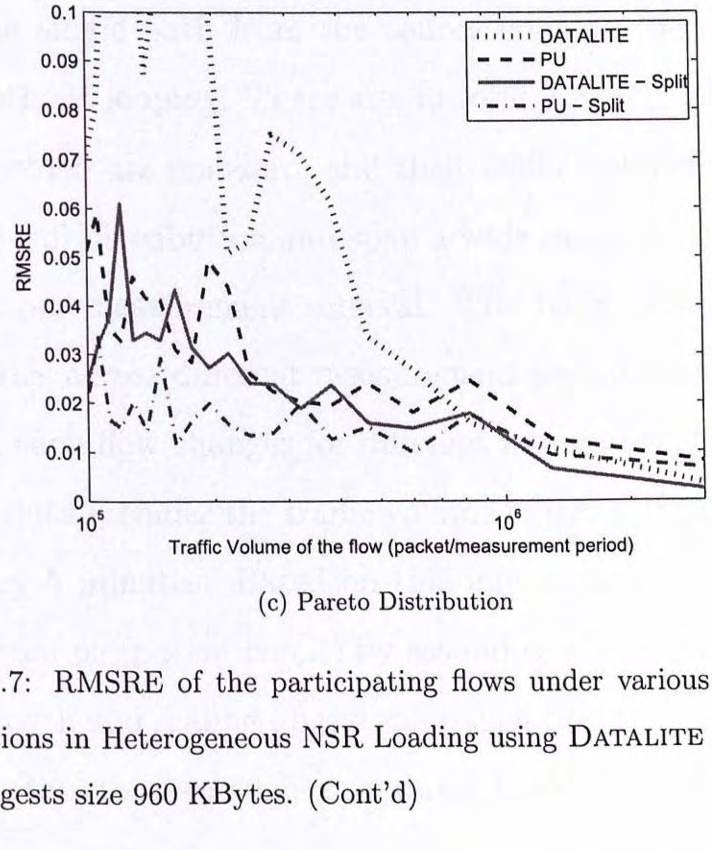


(a) Uniform Distribution



(b) Gamma Distribution

Figure 6.7: RMSRE of the participating flows under various flow volume distributions in Heterogeneous NSR Loading using DATALITE and PU with traffic digests size 960 KBytes.



(c) Pareto Distribution

Figure 6.7: RMSRE of the participating flows under various flow volume distributions in Heterogeneous NSR Loading using DATALITE and PU with traffic digests size 960 KBytes. (Cont'd)

6.2 Internet Trace Evaluation

We also evaluate our proposed scheme using a traffic matrix collected from the core network of the Abilene Network [1], which we obtained from [2]. In the network, there are 12 nodes and 30 links. For each commodity (noted as 1 OD-pair), there is only one single path from the source node to the destination node without looping. There are, in total, 620 OD-flow per link counts which are non-zero, and their traffic volume are follows a heavy-tail distribution and span a wide range = $[10^5, 2 \times 10^9]$ packets per measurement interval. The traffic volume of each flow varies across different measurement period, and hence, the NSR of each flow changes for different measurement period.

The data includes the traffic volume in bytes of each OD-pair for every 5 minutes. Based on this information, we derive the traffic volume (packet count) by assuming all the packets in the same length and scaling up the value such that the loading factor of the estimator is enough for unbiased estimation [6], which we use as ground truth.

Since there is no flow or packet level information from the data as the input content to our algorithms, therefore we use the output of a uniform random number generator as the packet identifier. For each OD-pair, we assign the packets to the TDs according to its path until the total number of packets for that OD-pair reached the corresponding value obtained from the traffic matrix.

6.2.1 Simulation Results

To measure the OD-flow per link count of the Abilene Network, we setup 40 KBytes memory (array size of $m = 2^{16}$) for DATALITE and setup 60 KBytes memory (array size of $m = 2^{15}$) for PU and QMLE to store the traffic digests and measure the traffic from mid-night to 1 a.m. on 1^{st} Match, 2004 for every 5 minutes. At time = 0, we use the original scheme to take an initial measurement for the OD-flow per link count and estimate the optimal splitting threshold based on this measurement. To provide fair comparison, we scale the estimation error of DATALITE by $1/\sqrt{\frac{60}{40}} = \frac{1}{1.5} \approx 0.8165$ to compensate for the larger memory advantage of PU (and QMLE). Figure 6.8 shows how the estimation errors of each measurement interval varies during this 1 hour period. TD-splitting is applied starting from the second measurement period (at 12:05 a.m.). (Since there are marginal improvement for PU and QMLE after applying TD-splitting, we exclude the curves from the figure.) After applying TD-splitting, the r.m.s. estimation error of DATALITE (thick solid line) across all flows is reduced to around 30% of the non-splitting version. The resultant estimation errors are close to the non-splitting version of PU and also QMLE. Figure 6.9 compares the quantiles of the relative estimation errors of different schemes at the second measurement period: the three vertical lines are the 75%, 90% and 99% quantiles of the estimation. The 75% and 90% quantiles for the relative errors of the original DATALITE is 9% (0.09) and 25% (0.25) respectively while those of single-level TD-splitting for DATALITE is dropped to 5% (0.05) and 9% (0.09) which is slightly less than the estimation errors of the original PU and QMLE schemes. This demonstrates the estimation for each measurement interval of applying the proposed TD-splitting scheme to practical network.

Table 6.5 shows the r.m.s. estimation error across all the measurement over all the measurement period (excluding the first measurement period). The r.m.s. estimation error of DATALITE is reduced by 76% while that of PU and QMLE are reduced by 29% and 8% respectively after applying single-level TD-splitting. The 95-th percentile estimation error of DATALITE also reduced by 80% while there is just slightly reduction for PU and QMLE which is around 30% and 20%.

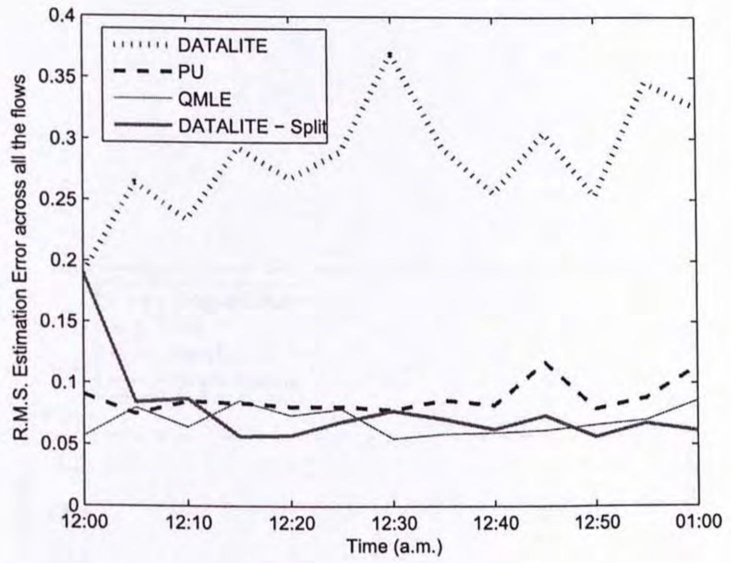


Figure 6.8: R.M.S. estimation error across all the flows in Abilene Network using various measurement schemes at different time

	R.M.S. [95-th percentile] Estimation Error	
TMA Scheme	Original	Single-level Split
DATALITE	29.38% [72.12%]	6.93% [13.09%]
PU	7.85% [16.93%]	5.57% [11.60%]
QMLE	7.01% [14.62%]	6.43% [11.66%]

Table 6.5: Comparison of the estimation error after applying single-level TD-splitting to different TD-based TMA scheme in Abilene Network

Figure 6.9: Error distributions amongst different measurement schemes at 12:05 a.m.. The vertical lines represents the 75%, 90% and 99% quantiles of the relative estimation error and the horizontal lines represent the relative estimation errors equal 0.05, 0.09, 0.26 and 0.33

□ **End of chapter.**

Chapter 7

Conclusion

In this thesis, we have proposed a traffic digest splitting approach to enhance TD-based TMA schemes by partitioning the flows carried by a link into different sub-TDs. By avoiding the mixing of "mice" and "elephant" flows in a single traffic digest, we can significantly reduce the "noise-to-signal" ratio experienced by the smaller flows. Such reduction in "noise-to-signal" ratio is more than enough to offset the negative effect on estimation error caused by reduction in TD memory size for each subgroup. Based on this TD-splitting approach, we have derived analytical expressions for the optimal splitting threshold which minimizes the resultant maximum (or r.m.s.) relative error of the flows sharing a link under various flow volume distributions. Our analysis shows that the estimation error improvement depends on the estimation error characteristics of the specific TMA scheme and the traffic flow-volume distribution. Simulation results show that our proposed scheme can significantly reduce the estimation error in most cases.

In particular, 2-level recursive TD-splitting of DATALITE reduces the r.m.s. estimation error by on average 80% which also outperforms the QMLE scheme by 20%. Although the benefits of applying the 2-level recursive TD-splitting to PU and QMLE seem to be less significant, due to the small estimation errors of these two schemes, we can still halve the memory size requirement of the TDs while keeping the same estimation error after applying 2-level recursive TD-splitting.

□ **End of chapter.**

Appendix A

Extension of QMLE for Cardinality Estimation of 3-sets Intersection

To provide network-wide OD-flows per link count estimation, we further extend QMLE by deriving the set expressions over three packet sets as follows. We let Y_i, for $i = 1, 2, 3$ to represent the traffic digests of 3 sets of packets, S, D and L, where packet set S, D and L are the same as those described in Section 2.2.1 for DATALITE. Let Λ_1, Λ_2, Λ_3, Λ_{12}, Λ_{23}, Λ_{31} and μ be the average number of packets/bye count per bucket of the traffic digests for packet set S, D, L, $S \cap D$, $D \cap L$, $S \cap L$, $S \cap D \cap L$. Λ_1, Λ_2, Λ_3, Λ_{12}, Λ_{23}, and Λ_{31} can be estimated using Equations 2.10 and 2.13, by estimating the cardinality of the corresponding packet sets divided by m.

Here, we further let $X, X_1, X_2, X_3, X_{12}, X_{23}, X_{31}$, be the unobserved traffic digests of the packet sets $S \cap D \cap L$, $S \backslash (D \cup L)$, $D \backslash (S \cup L)$, $L \backslash (S \cup D)$, $(S \cap D) \backslash L$, $(D \cap L) \backslash S$ and $(S \cap L) \backslash D$, generated by Algorithm 3, as illustrated in Figure A.1. And then, we have

$$Y_i = min(X, X_i, X_{ij}, X_{li}) \tag{A.1}$$

where $i = 1, 2, 3$ and $ij, li \in \{12, 23, 31\}$.

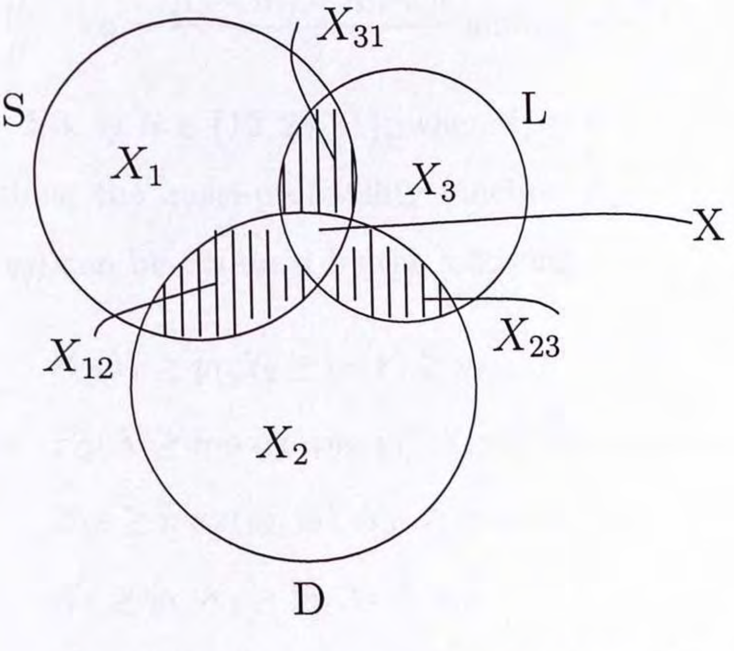


Figure A.1: Set expression of 3 sets intersection

According to Lemma 1 in [7], $X, X_1, X_2, X_3, X_{12}, X_{23}, X_{31}$ can be approximated by *Pareto* distribution, i.e.,

$$X = Pareto(\alpha, \beta), \quad X_i = Pareto(\alpha_i, \beta)$$
$$X_{ij} = Pareto(\alpha_{ij}, \beta)$$

where $\alpha\beta$, $\alpha_1\beta$, $\alpha_2\beta$, $\alpha_3\beta$, $\alpha_{12}\beta$, $\alpha_{23}\beta$ and $\alpha_{31}\beta$ are the average number of packets/byte count per bucket of packet set $S \cap D \cap L$, $S \backslash (D \cup L)$, $D \backslash (S \cup L)$, $L \backslash (S \cup D)$, $(S \cap D) \backslash L$, $(D \cap L) \backslash S$ and $(S \cap L) \backslash D$ respectively, i.e.,

$$\alpha = \frac{\mu}{\beta}, \quad \alpha_i = \frac{\Lambda_i - \Lambda_{ij} - \Lambda_{li} + \mu}{\beta} \text{ and } \alpha_{ij} = \frac{\Lambda_{ij} - \mu}{\beta}$$

for $i = 1, 2, 3$, $ij, li \in \{12, 23, 31\}$, where $ij \neq li$.

And thus, the quasi-probability function $P_Q(Y_1 \geq y_1, Y_2 \geq y_2, Y_3 \geq y_3)$ can be obtained by the following:

$$\begin{aligned}
& P_Q(Y_1 \geq y_1, Y_2 \geq y_2, Y_3 \geq y_3) \\
= \;& P_Q(X \geq max(y_1, y_2, y_3), X_{12} \geq max(y_1, y_2), \\
& X_{23} \geq max(y_2, y_3), X_{31} \geq max(y_1, y_3), \\
& X_1 \geq y_1, X_2 \geq y_2, X_3 \geq y_3) \\
= \;& (1 + \beta max(y_1, y_2, y_3))^{-\alpha}(1 + \beta max(y_1, y_2))^{-\alpha_{12}} \\
& (1 + \beta max(y_2, y_3))^{-\alpha_{23}}(1 + \beta max(y_1, y_3))^{-\alpha_{31}} \\
& (1 + \beta y_1)^{-\alpha_1}(1 + \beta y_2)^{-\alpha_2}(1 + \beta y_3)^{-\alpha_3}
\end{aligned}$$

The quasi-density function, $P_Q(Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$, can be derived into 13 cases as follows:

$$\left\{\begin{array}{l}
\text{Case 1 } (y_1 > y_2 > y_3): \\
\quad (\alpha+\alpha_1+\alpha_{12}+\alpha_{31})(\alpha_2+\alpha_{23})\alpha_3\beta^3(1+\beta y_1)^{-(\alpha+\alpha_{12}+\alpha_{31}+\alpha_1+1)} \\
\quad (1+\beta y_2)^{-(\alpha_2+\alpha_{23}+1)}(1+\beta y_3)^{-(\alpha_3+1)} \\
\text{Case 2 } (y_1 > y_3 > y_2): \\
\quad (\alpha+\alpha_1+\alpha_{12}+\alpha_{31})(\alpha_3+\alpha_{23})\alpha_2\beta^3(1+\beta y_1)^{-(\alpha+\alpha_{12}+\alpha_{31}+\alpha_2+1)} \\
\quad (1+\beta y_2)^{-(\alpha_2+1)}(1+\beta y_3)^{-(\alpha_3+\alpha_{23}+1)} \\
\text{Case 3 } (y_2 > y_1 > y_3): \\
\quad (\alpha+\alpha_2+\alpha_{12}+\alpha_{23})(\alpha_1+\alpha_{31})\alpha_3\beta^3(1+\beta y_1)^{-(\alpha_1+\alpha_{31}+1)} \\
\quad (1+\beta y_2)^{-(\alpha+\alpha_2+\alpha_{12}+\alpha_{23}+1)}(1+\beta y_3)^{-(\alpha_3+1)} \\
\text{Case 4 } (y_2 > y_3 > y_1): \\
\quad (\alpha+\alpha_2+\alpha_{12}+\alpha_{23})(\alpha_3+\alpha_{31})\alpha_1\beta^3(1+\beta y_1)^{-(\alpha_1+1)} \\
\quad (1+\beta y_2)^{-(\alpha+\alpha_2+\alpha_{12}+\alpha_{23}+1)}(1+\beta y_3)^{-(\alpha_3+\alpha_{31}+1)} \\
\text{Case 5 } (y_3 > y_1 > y_2): \\
\quad (\alpha+\alpha_3+\alpha_{23}+\alpha_{31})(\alpha_1+\alpha_{12})\alpha_2\beta^3(1+\beta y_1)^{-(\alpha_1+\alpha_{12}+1)} \\
\quad (1+\beta y_2)^{-(\alpha_2+1)}(1+\beta y_3)^{-(\alpha+\alpha_3+\alpha_{23}+\alpha_{31}+1)} \\
\text{Case 6 } (y_3 > y_2 > y_1): \\
\quad (\alpha+\alpha_3+\alpha_{23}+\alpha_{31})(\alpha_2+\alpha_{12})\alpha_1\beta^3(1+\beta y_1)^{-(\alpha_1+1)} \\
\quad (1+\beta y_2)^{-(\alpha_2+\alpha_{12}+1)}(1+\beta y_3)^{-(\alpha+\alpha_3+\alpha_{23}+\alpha_{31}+1)}
\end{array}\right.$$

$$\begin{cases}
\text{Case 7 } (y_1 > y_2 = y_3): \\
\quad (\alpha + \alpha_1 + \alpha_{12} + \alpha_{31})\alpha_{23}\beta^2(1+\beta y_1)^{-(\alpha+\alpha_1+\alpha_{12}+\alpha_{31}+1)} \\
\quad (1+\beta y_2)^{-(\alpha_2+\alpha_3+\alpha_{23}+1)} \\
\text{Case 8 } (y_2 > y_3 = y_1): \\
\quad (\alpha + \alpha_2 + \alpha_{12} + \alpha_{23})\alpha_{31}\beta^2(1+\beta y_1)^{-(\alpha_1+\alpha_3+\alpha_{31}+1)} \\
\quad (1+\beta y_2)^{-(\alpha+\alpha_2+\alpha_{12}+\alpha_{23}+1)} \\
\text{Case 9 } (y_3 > y_1 = y_2): \\
\quad (\alpha + \alpha_3 + \alpha_{23} + \alpha_{31})\alpha_{12}\beta^2(1+\beta y_1)^{-(\alpha_1+\alpha_2+\alpha_{12}+1)} \\
\quad (1+\beta y_2)^{-(\alpha+\alpha_3+\alpha_{23}+\alpha_{31}+1)} \\
\text{Case 10 } (y_1 = y_2 = y_3): \\
\quad \alpha\beta(1+\beta y_1)^{-(\alpha+\alpha_1+\alpha_2+\alpha_3+\alpha_{12}+\alpha_{23}+\alpha_{31}+1)} \\
\text{Case 11 } (y_1 = y_2 > y_3): \\
\quad (\alpha+\alpha_{12})\alpha_3\beta^2(1+\beta y_1)^{-(\alpha+\alpha_1+\alpha_2+\alpha_{12}+\alpha_{23}+\alpha_{31}+1)}(1+\beta y_3)^{-(\alpha_3+1)} \\
\text{Case 12 } (y_2 = y_3 > y_1): \\
\quad (\alpha+\alpha_{23})\alpha_1\beta^2(1+\beta y_1)^{-(\alpha_1+1)}(1+\beta y_2)^{-(\alpha+\alpha_2+\alpha_3+\alpha_{12}+\alpha_{23}+\alpha_{31}+1)} \\
\text{Case 13 } (y_3 = y_1 > y_2): \\
\quad (\alpha+\alpha_{31})\alpha_2\beta^2(1+\beta y_2)^{-(\alpha_2+1)}(1+\beta y_3)^{-(\alpha+\alpha_1+\alpha_3+\alpha_{12}+\alpha_{23}+\alpha_{31}+1)}
\end{cases}$$

Define $Y_j^* = Y_j I(Y_j < \infty)$, for $j = 1, 2, 3$, where I is the indicator function. Let $L_{Q_3}(\mu, \beta)$ be the quasi-log-likelihood of (μ, β) for (Y_1, Y_2, Y_3) and it is expressed as

$$L_{Q_3}(\mu,\beta) = -\tfrac{1}{m}\textstyle\sum_{i=1}^{m} log\{P_Q(Y_1^*[i] = y_{1,i}, Y_2^*[i] = y_{2,i}, Y_3^*[i] = y_{3,i})\}$$

where $y_{1,i}$, $y_{2,i}$ and $y_{3,i}$ are the empirical value of the i-th bucket of TD Y_1^*, Y_2^* and Y_3^* respectively.

The QMLE estimate $\hat{\mu}_{Q_3}$ of μ can be obtained by the solution of the following optimization problem

$$(\hat{\mu}_{Q_3}, \hat{\beta}_{Q_3}) = \arg\max_{(\mu,\beta)} L_{Q_3}(\mu, \beta) \tag{A.2}$$

for $\mu > \Lambda_{ij} + \Lambda_{li} - \Lambda_i$ and $\mu < \Lambda_{ij}$, for $i, j, l = 1, 2, 3$, $ij, li = \{12, 23, 31\}$, $i \neq j \neq l$, $ij \neq li$ and $\beta > 0$.

□ **End of chapter.**

Bibliography

[1] Abilene Internet2 Network. [Online]. Availabe: http://abilene.internet2.edu/.

[2] Traffic Matrix Dataset. [Online]. Availabe: http://www.cs.utexas.edu/~yzhang/research/AbileneTM/AbileneTM-all.tar.

[3] P. Bickel and K. Doksum. *Mathematical statistics: basic ideas and selected topics.* Prentice Hall, 2001.

[4] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13:422–426, 1970.

[5] A. Broder and M. Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004.

[6] M. Cai, J. Pan, Y. Kwok, and K. Hwang. Fast and accurate traffic matrix measurement using adaptive cardinal-

ity counting. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 205–206. ACM New York, NY, USA, 2005.

[7] J. Cao, A. Chen, and T. Bu. A Quasi-Likelihood Approach for Accurate Traffic Matrix Estimation in a High Speed Network. In *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, pages 21–25, 2008.

[8] A. Chen, J. Cao, and T. Bu. A simple and efficient estimation method for stream expression cardinalities. In *Proceedings of the 33rd international conference on Very large data bases*, pages 171–182. VLDB Endowment, 2007.

[9] A. Dobra, M. Garofalakis, J. Gehrke, and R. Rastogi. Processing complex aggregate queries over data streams. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 61–72. ACM New York, NY, USA, 2002.

[10] M. Durand and P. Flajolet. Loglog Counting of Large Cardinalities. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 605–617, 2003.

[11] C. Estan, G. Varghese, and M. Fisk. Bitmap algorithms for counting active flows on high speed links. In *Proceedings of*

the 3rd ACM SIGCOMM conference on Internet measurement, pages 153–166. ACM New York, NY, USA, 2003.

[12] C. Estan, G. Varghese, and M. Fisk. Bitmap algorithms for counting active flows on high speed links. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 153–166. ACM New York, NY, USA, 2003.

[13] P. Flajolet, É. Fusy, O. Gandouet, and F. Meunier. HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm. In *Proceedings of the 13th conference on analysis of algorithm (AofA 07)*, pages 127–146, 2007.

[14] P. Flajolet and G. Martin. Probabilistic counting. In *Foundations of Computer Science, 1983., 24th Annual Symposium on*, pages 76–82, 1983.

[15] S. Ganguly. Counting distinct items over update streams. *Theoretical Computer Science*, 378(3):211–222, 2007.

[16] S. Ganguly, M. Garofalakis, and R. Rastogi. Tracking set-expression cardinalities over continuous update streams. *The VLDB Journal*, 13(4):354–369, 2004.

[17] K. Keys, D. Moore, and C. Estan. A robust system for accurate real-time summaries of internet traffic. In *Proceedings*

of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 85–96. ACM New York, NY, USA, 2005.

[18] A. Kumar, M. Sung, J. Xu, and J. Wang. Data streaming algorithms for efficient and accurate estimation of flow size distribution. In *Proceedings of the joint international conference on Measurement and modeling of computer systems*, pages 177–188. ACM New York, NY, USA, 2004.

[19] A. Kumar, M. Sung, J. Xu, and E. Zegura. A data streaming algorithm for estimating subpopulation flow size distribution. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 61–72. ACM New York, NY, USA, 2005.

[20] W. Lau, M. Kodialam, T. Lakshman, and H. Chao. DATALITE: a distributed architecture for traffic analysis via light-weight traffic digest. In *Broadband Communications, Networks and Systems, 2007. BROADNETS 2007. Fourth International Conference on*, pages 622–630, 2007.

[21] J. Nocedal and S. Wright. *Numerical optimization.* Springer, 1999.

[22] H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

[23] H. Zhao, A. Lall, M. Ogihara, O. Spatscheck, J. Wang, and J. Xu. A data streaming algorithm for estimating entropies of od flows. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 279–290. ACM New York, NY, USA, 2007.

[24] Q. Zhao, A. Kumar, J. Wang, and J. Xu. Data streaming algorithms for accurate and efficient measurement of traffic and flow matrices. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 350–361. ACM New York, NY, USA, 2005.

[25] Q. Zhao, A. Kumar, and J. Xu. Joint data streaming and sampling techniques for detection of super sources and destinations. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 7–7. USENIX Association Berkeley, CA, USA, 2005.