**DATALITE**: a Distributed Architecture for Traffic Analysis via Light-weight Traffic Digest

*Wing Cheong Lau*, Murali Kodialam**, T.V. Lakshman** and H. Jonathan Chao***

---

**Abstract** – In this paper, we propose DATALITE, a Distributed Architecture for Traffic Analysis via Light-weight Traffic digEst, which introduces a set of new distributed algorithms and protocols to support general Traffic Measurement and Analysis (TMA) functions for large-scale, 10Gbps+ packet-switched networks. We formulate the network-wide traffic measurement/analysis problem as a series of set-cardinality-determination (SCD) problems. By leveraging recent advances in probabilistic distinct sample counting techniques, the set-cardinalities, and thus, the network-wide traffic measurements of interest can be computed in a distributed manner via the exchange of extremely light-weight traffic digests (TD’s) amongst the network nodes. A TD for N packets only requires \(O(\log \log N)\) bits of memory storage.

**Keywords** — Traffic Measurement and analysis, traceback, digest.

---

**I. INTRODUCTION**

In recent years, we have witnessed the proliferation of high-speed data networks and the rapid expansion of the protocols/services supported by these networks. The development of network monitoring and traffic measurement techniques have so far failed to catch up with the operating speed and the large-scale deployment of these networks. There is an urgent need of a comprehensive, yet deployable, network monitoring and end-to-end traffic analysis infrastructure for large-scale, high-speed networks. Such infrastructure is particularly important for connectionless data networks such as the Internet, in which routes of traffic flows can change dynamically and unpredictably in middle of a session due to different types of expected or unexpected events. Such events include network component failures, non-deterministic load-balancing schemes (e.g. Equal Cost Multiple Path (ECMP)), software/hardware bugs and protocol mis-configurations. Currently, most operators can only rely on rudimentary diagnosis tools such as "traceroute", to obtain woefully inadequate samplings of end-to-end routes of individual traffic flows within the network.

In this paper, we propose **DATALITE**, a Distributed Architecture for Traffic Analysis via **Li**ght-weight **digEst**, which introduces a set of new distributed algorithms and protocols to support general Traffic Measurement and Analysis (TMA) functions for large-scale, 10Gbps+ packet-switched networks. These functions include, but are not limited to: Traffic flow pattern/route monitoring, diagnosis and network forensic; Estimation of Origination-to-Destination (OD) traffic load matrix for capacity planning and traffic engineering purposes; Traffic measurements for end-user accounting/billing as well as Inter-ISP (ASes) charge-settlement purposes; Traceback on the origin(s) of attacking packets in a distributed denial of service (DDoS) attacks.

We formulate the network-wide traffic measurement/analysis problem as a series of set-cardinality-determination (SCD) problems. By leveraging recent advances in probabilistic distinct sample counting techniques, the set-cardinalities, and thus, the network-wide traffic measurements of interest can be computed in a distributed manner via the exchange of extremely light-weight traffic digests (TD’s) amongst the network nodes, i.e. the routers. A TD for N packets only requires \(O(\log \log N)\) bits of memory storage. The computation of such \(O(\log \log N)\)-sized TD is also amenable for efficient hardware implementation at wire-speed of 10Gbps and beyond. Given the small size of the TD’s, it is possible to distribute nodal TD’s to all routers within a domain by piggybacking them as opaque data objects inside existing control messages, such as OSPF link-
ogy [8] can be used to collect fine grain per-flow traffic counters by most commercial gigabit routers. For example, while the Cisco Netflow technology enables per-flow traffic measurements, researchers [15, 19, 20] have resorted to combine link-load measurements with additional assumptions on O-D pair traffic distribution in order to estimate the required O-D pair traffic matrix. This approach has led to the new field of "Network Tomography". Unfortunately, most of the network tomography-based solutions proposed to-date are not robust w.r.t. the validity of their traffic distribution assumptions. They also heavily rely on the correctness, synchronization and consistency amongst multiple operational databases (e.g. router forwarding/ routing tables and configuration files, SNMP MIBs) from which measurements/ configuration information have to be extracted and collated. The modeling and operational assumptions also render the tomography-based approaches of little use for network failure detection/ diagnosis where neither the proper functioning of network elements/ databases nor the normality of traffic distribution can be assumed.

Recently, an alternative packet trajectory-based traffic monitor/ analysis approach has been proposed by [9, 17] in which each node (router) maintains a compressed summary, or a digest, of all the packets it recently handled. In [17], the digest is in form of a Bloom filter [2, 5] which is updated for every packet arriving at the node and periodically uploaded to some centralized server to support future offline traffic analysis as well as archival purposes. Armed with these very informative nodal traffic digests, the centralized server can not only construct the traffic flow pattern and per-flow/commodity measurements throughout the network, but also answer queries regarding the end-to-end path, or the so-called trajectory, of any given packet traversing the network in the (recent) past. The ability of answering trajectory query for any given individual packet does come with a heavy cost: the Bloom filter has to be big enough to store sufficient information for every individual incoming packets. Even with the efficient memory Vs. false-positive-trade-off of a Bloom filter, it still requires O(N) bits of memory to capture and correctly distinguish the signatures of N different packets with high probability. In [17], it is estimated that the system requires approximately 0.5% of link capacity of the node per unit time in storage. For a 10Gbps link, this translates to 50Mbits of storage for every one second of monitoring time. Such a heavy weight traffic digest approach not only stresses the memory storage and communication requirements of the system but also scales poorly as the link speed and/ or monitoring duration increases. The key ideas of the current paper, i.e. the DATALITE algorithms, protocols and system architecture, together with its various

The remainder of this paper is organized as follows. We review related work in Section II. Section III describes the formulation of the traffic measurement/ analysis problem as a series of set-cardinality-determination (SCD) problems. We also show how the SCD problem can be solved in a distributed manner with minimal communications overhead by leveraging recent advances in probabilistic distinct sample counting techniques. In Section IV, we discuss the overall system architecture of DATALITE and introduce additional techniques and refinements to further enhance the system scalability in terms of communication bandwidth and memory requirements. The paper ends with a preliminary conclusion in Section V.

II. RELATED WORK

Recent research in traffic measurement/ analysis methodologies and infrastructures has been strongly driven by the demands of a couple of critical real-life applications such as the Origination-to-Destination (O-D pair) traffic matrix estimation for large scale ISPs and the support of traceback services in IP-based networks to tackle spoofed DDoS attacks [14, 16, 17]. The crux of the problem is the lack of support of inexpensive, scalable per-flow counters by most commercial gigabit routers. For example, while the Cisco Netflow technology [8] can be used to collect fine grain per-flow traffic statistics, its formidable storage and bandwidth requirements make it unsuitable for 10Gbps networks. To address the inadequacy of the measurement infrastructure, researchers [15, 19, 20] have resorted to combine link-load measurements with additional assumptions on O-D pair traffic distribution in order to estimate the required O-D pair traffic matrix. This approach has led to the new field of "Network Tomography". Unfortunately, most of the network tomography-based solutions proposed to-date are not robust w.r.t. the validity of their traffic distribution assumptions. They also heavily rely on the correctness, synchronization and consistency amongst multiple operational databases (e.g. router forwarding/ routing tables and configuration files, SNMP MIBs) from which measurements/ configuration information have to be extracted and collated. The modeling and operational assumptions also render the tomography-based approaches of little use for network failure detection/ diagnosis where neither the proper functioning of network elements/ databases nor the normality of traffic distribution can be assumed.

The requirement of O-D pair traffic distribution in order to estimate the system architecture of DATALITE and introduce additional techniques and refinements to further enhance the system scalability in terms of communication bandwidth and memory requirements. The paper ends with a preliminary conclusion in Section V.

II. RELATED WORK

Recent research in traffic measurement/ analysis methodologies and infrastructures has been strongly driven by the demands of a couple of critical real-life applications such as the Origination-to-Destination (O-D pair) traffic matrix estimation for large scale ISPs and the support of traceback services in IP-based networks to tackle spoofed DDoS attacks [14, 16, 17]. The crux of the problem is the lack of support of inexpensive, scalable per-flow counters by most commercial gigabit routers. For example, while the Cisco Netflow technology [8] can be used to collect fine grain per-flow traffic
applications in network-wide traffic monitoring and packet-flow tracking were first proposed and filed for patent in [21,22]. Subsequently, Kwok et al [23] used some of the ideas in [21,22] and further extended them in [24] for similar applications.

III. THE DATALITE APPROACH FOR TRAFFIC MEASUREMENT/ANALYSIS (TMA)

While the ability of answering a trajectory query for any given individual packet was considered to be necessary for the designers in [17] to support IP traceback, we argue that it is an overkill for most TMA applications. As demonstrated below, for most TMA applications, it suffices to know the trajectory and/or traffic aggregate of interest is clearly defined by the context of the measurement application.

Consider the directed graph representation of a network \( G = (V,E) \) where \( V \) is the set of nodes and \( E \) is the set of directional links. Let \((i,j) \in E\) be the directional link from node \( i \) to node \( j \). Let \( L_{ij} \) be the set of packets traversing over link \((i,j)\) during a given measurement period of length \( T \) seconds. For now, let's assume the measurement period to be much longer than the maximum end-to-end delay within the network so that the fringe effects caused by in-flight packet can be neglected\(^1\). Let \( O_i \) (or \( D_i \)) be the set of packets originated (or terminated) at node \( i \) during the same measurement period\(^2\). During the given measurement period, the traffic aggregates of our interest can be readily represented as the intersection of the packet sets defined above. To illustrate our approach, let's consider the following common (and fundamental) TMA tasks:

Sample TMA Task #1: Route Monitoring

To determine the routes and volume of traffic between all O-D node-pairs in the network. Consider the set of packets, \( F^{k}_{i,j} \), that pass through link \((i,j) \in E\), with \( k=\left(s,d\right) \in V \times V \) as their O-D node pair. Notice that \( F^{k}_{i,j} \) can be expressed as an intersection set, namely, \( F^{k}_{i,j} = O_i \cap L_{i,j} \cap D_{d} \). The key observation is that, for Task#1, and a wide range of TMA applications, those mentioned in Section I, it suffices to know the cardinality of \( F^{k}_{i,j} \), i.e., \( |F^{k}_{i,j}| \), instead of its full details.

Sample TMA Task #2: Traffic Traceback

Consider the problem of determining the origins and traffic flow volume/routes towards a downstream node \( d \) of interest. It suffices to determine \( |F^{k}_{i,j}| \) for every link \((i,j) \in E\) where \( F^{k}_{i,j} = L_{i,j} \cap D_{d} \) and \( k=\left(s,d\right) \), where * is the wildcard. Similarly, one can trace the destination, downstream route pattern and flow volume for of packets originating from a given node \( s \) by determining \( |F^{k}_{i,j}| \) for every link \((i,j) \in E\) where \( F^{k}_{i,j} = O_i \cap L_{i,j} \) and \( k=\left(s,*\right) \).

A. Traffic Measurement/Analysis as an Intersection-Set-Cardinality-Determination (ISCD) Problem

The main idea of DATALITE is to provide an infrastructure to support the distributed estimation of the cardinality (i.e., \( |F^{k}_{i,j}| \)'s) of some sets \( F^{k}_{i,j} \)'s for a network, where \( F^{k}_{i,j} \) is the intersection of some packet relevant sets such as the \( O_i \)'s, \( D_d \)'s and \( L_{i,j} \)'s defined above. By focusing on the cardinality instead of the full details of \( F^{k}_{i,j} \)'s, storage and communication bandwidth requirements for DATALITE can be much reduced.

B. Distributed Intersection-Set-Cardinality-Determination via Distinct Sample Counting

To solve the distributed ISCD problem, we first transform the ISCD problem to one or more union-set cardinality determination (USCD) problems using elementary set theory. We then apply the \( O\log\log N \) distinct sample counting algorithm proposed in [10] to solve the USCD problem in a distributed manner. As an illustration, recall Sample TMA Task #2 where \( F^{k}_{i,j} = O_i \cap L_{i,j} \). Based on elementary set theory, \( |F^{k}_{i,j}| = |O_i \cap L_{i,j}| = |O_i| + |L_{i,j}| - |O_i \cup L_{i,j}| \)\text{Eq.(1)}

where \(|O_i|\) is the no. of distinct packets originated at node \( s \) during the measurement period. By definition, every packet generated is distinct and thus, \(|O_i|\) can be maintained as a single packet counter for every origi-
nating network node. \( |L_{i,j}| \) is the no. of distinct packets traversing link \((i, j)\). We will apply the probabilistic distinct sample counting techniques in [10] to keep track of \( |L_{i,j}| \) for every link \((i, j) \in E\). A key advantage of such technique is that it only requires the maintenance of an \( O(\log \log N_{\text{max}}) \)-bit digest to summarize the necessary information of the packet set of interest, i.e. \( L_{i,j} \) in this example, where \( N_{\text{max}} \) is the maximal number of distinct samples in the set. In DATALITE, we will refer this digest as the traffic digest (TD) of \( L_{i,j} \), denoted by \( TD_{i,j} \). Besides maintaining \( TD_{i,j} \), we also introduce a simple packet counter \( C_{i,j} \) for every link \((i, j) \in E\) to track the simple count of packets (including duplicates) passing through the link during the same measurement period. A large discrepancy between the values of \( C_{i,j} \) and \( |L_{i,j}| \) would indicate potential routing problems as link \((i, j)\) may have become part of a routing loop. Thus, the remaining challenge in obtaining \( F_{i,j} \) is to compute \( |O_{i} \cup L_{i,j}| \), i.e. the no. of distinct packets in the union of \( O_{i} \) and \( L_{i,j} \). Incidentally, the techniques in [10] can also be used to compute \( |O_{i} \cup L_{i,j}| \) in a distributed manner. In particular, it only requires the exchange of \( O(\log \log N_{\text{max}}) \)-sized TD’s for the packet sets \( O_{i} \) and \( L_{i,j} \), denoted by \( TD_{O_{i}} \) and \( TD_{L_{i,j}} \), and maintained locally by node \( s \) and node \( i \) respectively. Similarly, Sample Task \#1 can also be solved in by expressing:

\[
\begin{align*}
F_{i,j}^{k} &= |O_{i} \cap L_{i,j} \cap D_{j}| \\
&= |O_{i}| + |L_{i,j}| + |D_{j}| - |O_{i} \cup L_{i,j}| \\
&- |L_{i,j} \cup D_{j}| - |D_{j} \cup O_{i}| + |O_{i} \cup L_{i,j} \cup D_{j}|
\end{align*}
\]

Here, the \( O(\log \log N_{\text{max}}) \) distinct sample counting technique of [10] can be used to determine the cardinality of each of the union-sets in the R.H.S. of Eq.(2) while requiring only a single light-weight TD per link, plus one simple packet counter per link\(^4\). This is sufficient to determine the network-wide route-patterns and per-link traffic volumes for the \( |V|^{2} \) types of packets based on O-D node-pair classification.

In general, one can also compute the cardinality of the intersection of multiple sets \( S_{1}, S_{2}, \ldots, S_{n} \) based on the following expression:

\[
\left| \bigcap_{i=1}^{n} S_{i} \right| = \sum_{i=1}^{n} |S_{i}| - \sum_{i\neq j} |S_{i} \cup S_{j}| + \ldots + (-1)^{n-1} \left| \bigcup_{i=1}^{n} S_{i} \right|
\]

Eq.(3) will become useful when we apply additional set intersections to refine the definition the traffic aggregate of interest, e.g. all 40-byte TCP packets with O-D pair \((s,d)\) traversing link \( l_{i,j} \).

Our approach can be summarized by the following steps:

(i) Transform the TMA problem of interest to the problem of determining the cardinalities of some intersection packet sets of interest.

(ii) Using Eq.(3), transform the Intersection-Set-Cardinality-Determination problem in (i) to the problem of determining the cardinality of some union-sets of interest, noting that the elements (packets) of those union-sets are observed at geographically distributed locations.

(iii) By exchanging \( O(\log \log N_{\text{max}}) \)-sized traffic digests maintained locally by each node, we can determine the cardinality of the union sets in (ii) in a distributed manner using distinct sample counting techniques in [10].

C. Applying Loglog Distinct Sample Counting in a Distributed manner

Consider a set of packets \( S \) where each packet \( s \) has an unique identifier \( PID_{s} \). The packets of the set \( S \) may go through separate locations and be observed thereof. Let \( S = \bigcup_{p=1}^{P} S_{p} \) where the sets \( S_{p} \)'s are maintained in \( P \) separate locations and constructed based on local observations. We can, in a distributed manner, estimate the number of distinct packets in \( S_{p} \) by applying techniques in [10].

\(^3\) DATALITE actually uses this property to support routing-loop detection/identification.

\(^4\) By identifying the links of a router \( i \) through which packets actually enter (depart) the network, one can derive the TD’s for the originating (terminating) packet sets of the router \( i \) based on the TD’s of those links. It is therefore no need to maintain \( T_{O_{i}} \) and \( T_{D_{i}} \) explicitly.
digest of $S_p$ is maintained in form of a collection of $R_j^p$, for $1 \leq j \leq m$. Figure 1 depicts how to compute $R_j^p$ using packet identifiers observed at location $p$.

The estimate of $|S_i|$ is denoted by $\hat{n}$, which can be computed via:

$$\hat{n} = \alpha_m m^2 \sum_{i=1}^{n} \frac{1}{R_i}$$

where $\alpha_m$ is a correction factor as a function of $m$.

Further, the standard error $\sigma$ of $\hat{n}$ is given by:

$$\sigma = 1.05 / \sqrt{m}$$

See [10] for the theory behind $R_j^p$’s and their use in estimating the number of distinct samples in a set $S$.

As an illustration, Figure 2 summarizes the operational steps performed by a DATALITE-enabled node to realize route/traffic monitoring as described in Sample TMA Task #1. Here, each node $i \in V$ maintains the light-weight traffic digests (TD) for each of its local packet sets of interest: namely, $O_i, D_i$ and $L_{i,j}$ for $(i, j) \in E$. Denote the corresponding TD’s of these packet sets by $TD_{i,o}, TD_{D_i}$ and $TD_{L_{i,j}}$ respectively. In addition to use simple counters, $C_{i,j}$, to track per-link packet counts (without considering packet duplicates), and packet generation counts (|$O_i$|), each node tracks the local distinct packet counts of interest, $|L_{i,j}|$’s and $|D_i|$’, and maintains $TD_{L_{i,j}}$’s and $TD_{D_i}$’. Node $i$ can estimate the values of $|F_{i,j}^k|$’s in Eq. (2) for the links $(i, j) \in E$ and all $k \in V \times V$ after receiving $TD_{L_{i,j}}$ and $TD_{D_i}$ from other nodes $j \in V$. Once the estimates of $|F_{i,j}^k|$’s are computed locally by Node $i$, they can be distributed to other network nodes via periodic broadcast or on an on-demand, query-and-answer basis. Each node can then construct the network-wide view of the traffic aggregates of interest based on the knowledge of $|F_{i,j}^k|$’s. To further reduce the measurement/estimation errors on the $|F_{i,j}^k|$’s (due to the probabilistic nature of the schemes in [10]), each node performs a minimum square error (MSE) optimization based on nodal and network-wide flow conservation constraints (described in the next section).
E. Considerations and Optimization of Memory and Communication Bandwidth Requirements for DATALITE

One of the key design/engineering challenges is to maintain (1) the local memory requirement for the TD’s and (2) the inter-node communication bandwidth requirements, to an acceptable level, while satisfying the estimation error requirements for the TMA application of interest. Towards this end, we propose the following multi-prong strategy:

1. Judicious Control of Memory Size Per TD

Consider the memory requirement of a TD to support TMA tasks in 10Gbps+ networks. Since a 40Gbps link can transfer a maximum of 125 millions of 40-byte packets every second, a value of $10^{12}$ or $2^{40}$ should be adequate for $N_{\text{max}}$ (in Eq.(6)) to support measurement periods up to 8000 seconds long. According to Eq.(5), in order to achieve a standard error $\sigma \leq 2\%$ for the distinct sample count estimate, $m$ should be $\geq 2048$. Substituting $N_{\text{max}} = 2^{40}$ and of $m = 2048$ into Eq. (6) yields $R_{\text{max}} = 32 = 2^5$. In other words, it is sufficient to allocate 5-bits to encode each of the $R_j$’s in a TD. Thus, the memory requirement

$$M = m \log_2 R_{\text{max}} \text{ (bits)}$$

of each TD is dictated by the value of $m$. For $m = 2048$, which corresponds to a standard error of 2% for the corresponding distinct sample count estimate, the size of a TD is about 1.7KByte. We consider this to be a lower bound on the memory requirement of each TD in order to account for possible estimation error accumulation and/or "amplification" during the evaluation of the ultimate metric of interest, e.g., the $|P^{k}_{i,j}|$’s of the sample TMA tasks discussed in Section III.A. This is because, according to Eq. (3), the estimation error of the term on the L.H.S. of Eq.(3) is the sum of the estimation errors of each term on the R.H.S. Thus, the more stages of set-intersection in the L.H.S. term, the larger the estimation errors as the estimation errors for the union-set cardinalities on the R.H.S. of Eq.(3) accumulate. Furthermore, since $\sigma = 1.05 / \sqrt{m}$ is a "relative" estimation error with respect to each of the union-set cardinalities in the R.H.S. of Eq. (3), the corresponding relative estimation error, i.e., percentage-wise, for the intersection-term on the L.H.S. of Eq.(3) can get "amplified" especially when the cardinality of the intersection set on the L.H.S. of Eq.(3), is, in absolute terms, much smaller than that of the union-set terms on the R.H.S..

In practice, the actual value of $m$, and hence the memory-size per TD, is determined based on the estimation error requirements of the TMA application. For those TMA applications where rough estimates/measurements are sufficient, we will use a coarser-grain, smaller-sized TD’s, such as the 1.7Kbyte ones may be sufficient. Examples of this type of TMA applications include route failure/ mis-configuration/ change detection as well as DDoS attack traceback where the event of interest will typically cause drastic changes in traffic flow pattern. As a result, and the values of $|P^{k}_{i,j}|$’s would deviate substantially from their nominal values and the estimation error is insignificant with respect to the sudden deviation. For the TMA applications where quantitative, highly accurate measurements are required, we can increase $m$ (and thus decrease $\sigma$ ) during the estimation of each of "union-wise" terms on the L.H.S. of Eq.(3). However, since a linear decrease in $\sigma$ requires a quadratic increase in the size of TD, such memory vs. accuracy trade-offs should be conducted in a judicious manner. Fortunately, due to the inherent light-weight nature of the TD, there is considerable room for one to scale up its memory size. For instance, a 512-fold increase of TD size from 1.7KByte to 0.87MByte can reduce $\sigma$ to below 0.08%. Yet, a 0.87MByte TD is still very reasonable with current fast memory technologies.

In this research, we will investigate the means to support multiple sizes of TD’s in parallel. We will also explore how to adaptively control the size of the TD’s according to the applications/query-precision requirements. The necessary protocols to coordinate across network nodes in order to support such dynamic adaptation will also be considered.

2. Efficient support of Multiple Traffic Aggregates (or Packet Sets) per Link

In practice, instead of maintaining a single packet set, i.e. $L_{i,j}$, per link, some TMA applications may demand finer-grain definitions of packet sets, e.g. based on the protocol-type and/or port number of the packets. Another interesting use of multiple finer-grain packet sets per link is the use of "time-indexed" packet sets in which the original measurement period is divided into multiple smaller intervals so that one can estimate, admittedly with limited resolution, the path delay within a network by computing the cardinality of the intersection of the time-indexed packet sets belonging to different links within the network.

To support multiple packet sets per link, a naïve approach would assign a separate TD for each of the packet sets. However, this approach is not only inefficient, but also wasteful of memory. To overcome this problem, we propose a multi-prong strategy:

(1) Judicious Control of Memory Size Per TD

(2) Inter-node Communication Bandwidth Requirements

...
finer-grain packet set of interest. However, by introducing a generalized packet-set-intersection technique, we can support the network-wide TMA for $O(Q^2)$ types of finer-grain traffic aggregates using only $Q$ TD’s per line-card. The basic idea of this technique is as follows:

Assume we need to support $K = 2^k$ packet sets per link. Instead of assigning a TD for each of the $K$ packet sets, denoted by $P_1, P_2, ..., P_k$, we construct a list of $Q$ artificial packet sets:

$$S_1, S_2, ..., S_Q$$

where $$\left( \frac{Q}{2} \right) = Q(Q-1)/2 \geq K$$. The $S_1, S_2, ..., S_Q$ are defined s.t., $\forall i, 1 \leq i \leq K$, there exists $1 \leq q_1, q_2 \leq Q$ where $P_i = S_{q_1} \cap S_{q_2}$. In other words, every $P_i$’s can be “recovered” from the intersection of a pair of artificial sets. Thus, by maintaining only the TD’s for each of the $Q$ artificial sets $S_1, S_2, ..., S_Q$, denoted by $TD_{S_1}, TD_{S_2}, ..., TD_{S_Q}$ respectively, we can compute the cardinality of any intersection or union set with $P_i$ as one of its component.

This is by first expressing $P_i$ as $S_{q_1} \cap S_{q_2}$ and then applying Eq.(3) and the distinct sample counting techniques as discussed before. As an example, with this set-intersection techniques, we only need to maintain 24 TD’s per link in order to simultaneously support TMA of 276 finer-grain traffic aggregates network-wide. Even with the high-resolution, 0.87MByte TD’s, the total TD memory requirement per line-card for such configuration is less than 21MByte or 8.4% of the amount memory (2Gbits, or 200msec-worth of buffering) found in a typical 10Gbps line-card today. In this research, we will also explore the possibility of further increasing the supported types of traffic aggregate by dynamically altering the composition of the $S_1, S_2, ..., S_Q$’s in an on-demand basis.

In theory, we can further reduce the number of required TD’s to $2 \log_2 K$ per line-card by applying $\log_2 K$ stages of intersections among $2 \log_2 K$ artificial sets (each corresponds to the bit-value of the $\log_2 K$ bits binary representation of $K$) to recover each $P_i$. However, the accumulated estimation errors may be excessive due to the large number of terms on the R.H.S. of Eq.(3). This, in turn, would increase the memory requirement of each TD in order to reduce per-stage estimator error. The detail trade-offs between number of stages of intersection and the increase memory requirement per TD would be a subject of our investigation.

3. Further Estimation Error Reduction by Considering Network-wide Flow-conservation Constraints

Another way to conserve TD memory is by reducing the effective estimation errors in the $|F^k_{i,j}|$’s via post-processing of the initial estimates. In this scheme, after a node receives the initial estimates of the $|F^k_{i,j}|$’s from all nodes within the network, it will perform a minimum square error (MSE) optimization based on nodal flow conservation constraints expressed in terms of the $|F^k_{i,j}|$’s. Let $\hat{F}^k_{i,j}$ be the initial estimated value of $|F^k_{i,j}|$ received. The motivation of this optimization is to try to reduce the collective errors in the initial estimates $\hat{F}^k_{i,j}$’s by reconciling their inconsistencies in view of the nodal flow conservation constraints. Let $f^k_{i,j}$ be the new estimated value of $|F^k_{i,j}|$ which would satisfy nodal flow conservation constraints. Denote by $e^k_{i,j} = f^k_{i,j} - \hat{F}^k_{i,j}$ which is the perturbation required for changing $\hat{F}^k_{i,j}$ to $f^k_{i,j}$. Consider the following MSE optimization problem:

$$\text{Minimize } \left\{ \sum_{\forall (i,j \in E)} (e^k_{i,j})^2 \right\}$$

subject to

$$\sum_{(i,j \in E)} (\hat{F}^k_{i,j} + e^k_{i,j}) = \sum_{(j,i \in E)} (f^k_{j,i} + e^k_{j,i}), \quad \forall i \in V, \forall k = (s,d) \in V \times V, s \neq i, d \neq i$$

$$\sum_{(i,j \in E)} (\hat{F}^k_{i,j} + e^k_{i,j}) = |\Omega|, \quad \forall i \in V, \forall k = (i,d), d \in V$$

$$\sum_{(i,j \in E)} (f^k_{j,i} + e^k_{j,i}) = |\Omega|, \quad \forall i \in V, \forall k = (s,i), s \in V$$

The solution to the above optimization would yield the “collectively” minimum perturbations on $\hat{F}^k_{i,j}$’s, i.e. the $e^k_{i,j}$’s, so that the new estimates $f^k_{i,j} = \hat{F}^k_{i,j} + e^k_{i,j}$ will, at least, satisfy the nodal flow conservation constraints as in the case of the true values of $|F^k_{i,j}|$’s. We conjec-
ture} that by considering the flow conservation constraints, (which must be satisfied by the true values of $F_{i,j}$’s), we will bring our estimates closer to the their true values.

4. Inter-node Communication Bandwidth Optimization

The dominant controlling factors on communication bandwidth requirement of DATALITE include:

1. The number and size of the TD’s to be exchanged,
2. The frequency of the exchange of the TD’s and the resultant traffic flow estimators and
3. The way through which the TD’s and the traffic flow estimators are distributed throughout the network.

We have already described previously how (1) can be controlled and reduced. Here, we will note a few additional optimization opportunities regarding (1). First, depending on the need of the TMA application, i.e. the traffic measurements of interest, we may only need to distribute a selected set of TD’s to other nodes in the network. For example, while Sample TMA Task#1 in Section IIIA requires the full-mesh exchange of $TD_{O_i}$ and $TD_{O_j}$ among all nodes within the network, Sample Task#2 only requires the distribution of the $TD_{I_i}$ of the downstream DDoS attack victim node. In this project, we will investigate distributed query optimization techniques in order to reduce the amount of information exchange in supporting a given network TMA tasks. We can also compress the TD’s at the end of a measurement period before distributing them to other relevant nodes.

The dual periodic broadcast and on-demand query-and-answer modes of operations supported by DATALITE also help to control (2). In fact, the frequency of exchange of the TD’s and resultant traffic flow estimators is largely dictated by the need of the application. For example, for change-detection type of applications, e.g. the detection of route mis-configuration/ network failure, the frequency should be much higher in order to reduce detection time. Fortunately, these are also the applications where lower-precision measurements/estimates, i.e. smaller TD’s, may be sufficient because the resultant changes on flow patterns and per-link flow values caused by the event of interest tend to be significant. On the other hand, the TMA applications which require higher precision measurements/estimates (and thus larger TD’s) tend to be meaningful only over a longer measurement interval, which in turn, helps to reduce the bandwidth requirement. Another advantage of the DATALITE scheme is that the TD memory requirement grows very slowly ($O(loglogT)$) with measurement period $T$. As stated above, we can effectively keep the size of TD constant for measurement period up to 8000 seconds long, which should be adequate for most traffic measurement applications.

Finally, the distribution pattern of the TD’s should be decided based on the number and size of the TD’s as well as the required distribution frequency. To put things into perspective, let’s consider 2 extreme scenarios (or applications requirements). In the first scenario, the 1.7KByte TD’s are used in traffic-pattern-change/failure detection applications or DDoS traceback. Even if the uncompressed $TD_{O_i}$ and $TD_{O_j}$ of a each node are distributed to all other nodes every one second using flooding (the most expensive option bandwidth-wise), the total TD-traffic generated on every link in the network will not exceed $2*1.7*8*|V|$ Kbps, i.e. 2.72 Mbps per link for a 100-node network, or less than 0.03% of the capacity of a 10Gbps link. In the second scenario, special hourly traffic study is conducted to measure network-wide time-of-the-day statistics and route behavior of fine-grain traffic types. Here, 24*2 higher-resolution, uncompressed TD’s of 0.87MByte each are distributed, every hour, using a full I-BGP mesh between all nodes within the same network to analyze the behavior of 276 types of traffic aggregates network wide (using generalized set-intersection technique described above). The corresponding incoming TD’s bandwidth per-node is $0.87*8*24*100/3600$ Mbps = 9.28Mbps which is still less than 0.1% of the capacity of a 10Gbps link.

IV. CONCLUSIONS

In summary, by taking a direct-measurement approach, DATALITE avoids the problems caused by invalid traffic modeling or operational assumptions which plague the network tomography approaches. Although there are some high-level commonalities between the DATALITE scheme and the existing trajectory-based ones, there are key differences between them: First, by formulating the traffic measurement problem as a series of set-cardinality-determination problems, we can leverage recent advances in distinct sample counting to perform traffic analysis in a distributed manner with minimal communications overhead. Second, by focusing on the measurements and analysis of traffic aggregate behavior instead of individual packet ones, the system memory and communication bandwidth requirements for DATALITE are much reduced. As a result, it is possible for DATALITE to adopt a distributed computational model as opposed to the heavy-weight, centralized approach taken by existing trajectory-based systems.
References


