Trajectory Sampling with Unreliable Reporting

Nick Duffield
AT&T Labs—Research
180 Park Avenue
Florham Park NJ 07932, USA
Email: duffield@research.att.com

Matthias Grossglauser
School of Computer and Communication Sciences
EPFL
1015 Lausanne, Switzerland
Email: matthias.grossglauser@epfl.ch

***Abstract*— We define and evaluate methods to perform robust network monitoring using trajectory sampling in the presence of report loss. The first challenge is to reconstruct an unambiguous set of packet trajectories from the reports on sampled packets received at a collector. In this paper we extend the reporting paradigm of trajectory sampling to enable the elimination of ambiguous groups of reports, but without introducing bias into any characterization of traffic based on the surviving reports.**

Even after the elimination, a proportion of trajectories are incomplete due to report loss. A second challenge is to adapt measurement based applications (including network engineering, path tracing, and passive performance measurement) to incomplete trajectories. To achieve this, we propose a method to join multiple incomplete trajectories for inference, and analyze its performance. We also show how applications can distinguish between packet and report loss at the statistical level.

Keywords: Network Measurements, Statistics, Sampling

I. Introduction

A. Motivation

Trajectory Sampling (TS) has recently been proposed as a method to directly measure the spatial flow of traffic through an IP network at the packet level [3]. This is achieved by sampling a subset of packets consistently: each packet is sampled either at all routers it encounters, or at none. A router sends a report on each sampled packet to a collector. The reports contain sufficient information to distinguish different packets (with high probability); the collector is able to reconstruct the trajectory that the sampled packet took through the network.

The ability to reconstruct trajectories is impaired if reports are lost in transit; this consequently impairs the operation of measurement-based applications that exploit knowledge of the measured trajectories, either individually or through their statistical properties. In this paper we describe enhancements to reporting and reconstruction that enables measurement based applications to function even when reports are subject to loss.

B. Elements of Trajectory Sampling

TS is realized through hash-based selection of packets. While processing each packet, routers calculate a hash over a domain within the invariant portion of the packet, i.e., that part that does not change from hop to hop. (This excludes, e.g., the Time to Live field in the IP packet header, and the IP header checksum, which is recalculated at each hop). The packet is selected for reporting if its hash falls within a set known as the selection range. When all routers use the same hash function, domain and selection range, the selection decision for each packet is the same at all routers: the packet is sampled either everywhere or nowhere; see Figure 1.

Although hash-based selection is deterministic on packet content, selection can appear only weakly correlated with any field of the hash domain. This requires two things. Firstly, the hash function should be strong in the sense that small changes in the input (flipping a bit) generates large changes in the output. Secondly, there should be large variability in the content of each field in the hash domain. Under these conditions, hash values cover the range of the hash function nearly uniformly: hence the average sampling probability is the fraction of the range that is covered by the selection range.

C. Applications of Trajectory Sampling

A strength of TS is that since trajectories are measured directly, measurement-based applications do not need to join trajectory samples with network state data (such as routing tables) for interpretation. This eliminates uncertainties (e.g due to routing table fluctuations and transients) and can save significant computational and administrative cost associated with obtaining and joining with the routing data. Applications of TS include:

(i) *Network Engineering:* Reconstructed trajectories enable direct mapping of traffic onto the network topology. The actual traffic intensity of any class of traffic is estimated by dividing the intensity of the sampled traffic in that class by the sampling probability.

(ii) *Path Tracing:* the form of the trajectories themselves can be used to detect routing loops (manifest as self-intersecting trajectories) and to trace paths taken by network attack traffic when source address spoofing obscures the originating host of the attack.

(ii) *Passive Performance Measurement:* this is one of the major new applications of TS: passively measuring loss and delay attained by regular traffic, rather than that of probe traffic injected into the network. Trajectories that terminate before reaching their destination are interpreted as packet loss. (There is no confusion with a packet entering a tunnel provided TS enabled routers are able to look beyond encapsulation headers to locate the appropriate hash domain). If routers include synchronized timestamps in packet reports, the latency of

packets between routers can be found by subtraction. Sampling based on packet content is the only technique available for performing such measurements [10].

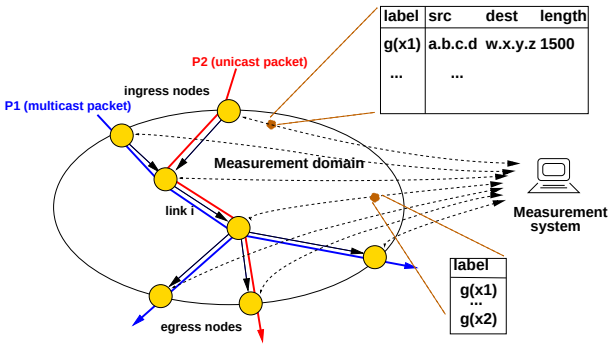

Fig. 1. SCHEMATIC REPRESENTATION OF TRAJECTORY SAMPLING. A measurement system collects packet *labels* from all the links within the domain. Labels are only collected from a pseudo-random subset of all the packets traversing the domain. Both the decision whether to sample a packet or not, and the packet label, are a function of the packet's invariant content.

D. Reporting and the Reconstruction of Trajectories

The packet reports contain a *key* and/or a *label*. The key contains fields from the invariant portion of the IP and transport headers, similar to the key used to distinguish packets within a raw IP flow. However, keys will not necessarily distinguish between different packets within a flow. For this purpose reports may also contain a label. This is a second hash (distinct from that used for selection) calculated over invariant part of the packet. We assume that the hash acts on fields that do vary between packets of a flow, e.g., IP identification, TCP sequence numbers, or even payload if available.

Our previous work [3], [4] has shown that a label length of about 4 bytes enables packets to be distinguished with high probability even in large networks. On the other hand, keys are expected to represent a significant portion of the IP and transport headers; ingress reports may also include routing state associated with the packet, such as routing prefix, and source and destination Autonomous System (AS). As a rule of thumb, we might expect keys to be $\alpha = 10$ times larger than labels. Thus, the use of labels offers considerable reduction in bandwidth consumed by trajectory samples. Consider the ideal situation that the hash is collision free and there is no report loss. Then it would be sufficient to associate the key with the label only once. This leads to the paradigm of Label Reporting, in which the ingress link reports both key and label, while core links report only labels. Core reports for which there is no ingress report with matching label are discarded at the collector.

Label reporting from a path of T hops consumes $T+\alpha$ units of bandwidth (measured in units of label size), as compared with $T(1+\alpha)$ if keys and labels are reported. The ratio of key to label bandwidth, $T(1+\alpha)/(T+\alpha)$ is increasing in T and α. For a long path (say $T = 30$) this ratio represents an order of magnitude difference.

Approaches to the collection and joining of individual packet reports in order to reconstruct trajectories are described in [4]. One of the main issues arising is collision of label hashes from different packets, and their resolution. A simple and robust approach that avoids introducing topological bias is to discard all reports within a given time window for which identical labels are observed at one or more ingress routers. Some renormalization of measured traffic intensities is then required in order to compensate for discarding measurements when estimating original traffic intensities.

A related problem is how to accommodate constraints on the bandwidth for reporting. As the label size increases, reporting bandwidth increases while the frequency of hash collisions decreases. Since labels do eventually repeat for different packets, a related question is how to group individual reports temporally in preparation for trajectory reconstruction.

E. The Need for Duplicate Elimination

We briefly comment on the need to eliminate all duplicate labels in a measurement period. Without duplicate elimination, if two (or more) packets happen to possess the same label, the corresponding trajectory appears as the composite of the individual trajectories followed by these packets.

If this occurs rarely, the reliability of some types of statistical estimators inferred from a set of measured trajectories may not be affected (e.g., a simple estimator of the rate of traffic between two routers). This might suggest that we should simply ensure that duplicate labels are rare, but tolerate the occasional composite trajectory that results. However, there are two reasons why we would like to ensure that composite trajectories never occur.

First, we can envision applications of trajectory sampling that check whether a particular condition *ever* happens in the network, e.g., a routing loop. Such applications could be very fragile to the occurrence of even a single unfortunate overlap of two or more trajectories (which could easily result in the appearance of a "phantom" routing loop). More generally, composite trajectories can be "physically impossible" for a single packet; this complicates the design of algorithms in measurement applications.

Second, allowing composite trajectories also complicates storing trajectory samples, because it will lead to a combinatorial explosion of the space of possible trajectories. In a trajectory database, significant compression and efficiency gains result from many packets following the same trajectory, which suggests data structures that either break out the observed trajectories into a separate table or memory structure, or explicitly enumerate the possible trajectories [4]. This table will grow significantly due to this combinatorial explosion.

F. The Case Against Reliable Reporting

Our previous work assumed that report packets are transported reliably from the observation points in the measurement domain (typically routers) to the collector. However, while this simplifies the task of the collector of reconstructing trajectories, this reliable transport has several disadvantages.

First, reliable transport requires that the observing device be addressable for feedback (ACKs or NACKs); while this is usually not a problem if the device is a router, it precludes transparent devices that simply inject report packets into the network without being addressable themselves (such a device might sit on a router's linecard, or it might be completely independent.).

Second, reliable transport requires that the measurement device buffer packets until they are acknowledged. This may be undesirable if the device has limited memory and processing power.

Third, it is necessary in the reliable scenario to match the report generation rate to the available transport rate, as any excess packets have to be buffered by the measurement device until they can be delivered. This in turn requires a well-designed outer control loop to quickly adjust the sampling rate in case a mismatch exists. In the unreliable scenario, the device has the additional option to react to a short-term overload condition simply by dropping some packets itself. Of course, appropriate congestion control is still required; we simply argue that in the unreliable case, we have more leeway to design this control (e.g., by averaging over longer timescales).

G. Scenarios for Information Loss

The most challenging scenario for TS is the export of reports across a wide area network that offers only best effort service. In this case report loss may be highly variable and essentially uncontrolled. A tamer scenario is to use a two stage export procedure. First, routers export reports to local staging servers, located in a routing center, for example. This initial export may take place out-of-band over dedicated management networks, or in-band over relatively tightly controlled network links, in which loss is rarer than in the WAN. Second, the staging servers export the reports reliably to a central collector. The staging servers may also play a larger role in distributed analysis, for example by performing local analysis. We refer the reader to [5] for description of such a multistage data collection infrastructure.

For the present work, we observe that even with good management of transport resources, data loss may still occur due expected changes in traffic load, or resource contention within the routers or other devices in the measurement infrastructure. For these reasons, it it necessary to make trajectory reconstruction and analysis robust with respect to report loss.

H. Complications for Reconstruction with Unreliable Reporting

Unreliable reporting complicates trajectory reconstruction and statistical inference from the packet reports. The methods must be well adapted to the requirements of the applications described in Section I-C.

The first problem is how to eliminate duplicate labels in the presence of report loss. With label reporting, loss of reports from ingress routers leaves "orphan" label-only reports from the core that cannot generally be distinguished from reports on other packets with the same label. This motivates a more robust method of duplicate elimination that degrades gracefully under report loss. Traffic volumes by path and class are an important input to network engineering. In estimating volumes, issues that arise during duplicate elimination are (a) how to avoid topological biasing against subsets of trajectories during elimination; and (b) how to correctly renormalize the surviving measurements in order to estimate the original traffic volumes that gave rise to the samples.

Once duplicates have been eliminated, there still remains the problem of adapting applications that perform analysis of packet trajectories to the occurrence of gaps in the reconstructed trajectories due to report loss. For path tracing, the problem is that measured trajectories may incomplete due to report loss. One way to obtain complete paths is to overlay trajectories of multiple packet that are expected to follow the same path. When routing is stable, packets sharing a common IP destination (or even prefix) will have this property.

For passive performance measurement, the problem is to distinguish report loss from packet loss. This is not always possible at the level of individual packets. For example, loss of a packet at a given link e of a path will produce the same set of packet reports at the collector as the loss of reports from all links subsequent to e on the path. Instead, we have to distinguish packet and report loss at the statistical level, employing trajectory samples from multiple packets.

I. Alternatives to Trajectory Sampling

We discuss alternatives to TS, and their drawbacks.

1) Ingress Packet Marking: In TS, a packet's hash value signals implicitly to the router whether the packet should be sampled. A alternative mechanism to consistently sample packets is to explicitly mark them for sampling on ingress, by randomly setting a bit in the packet. Marked packets are selected for reporting at all routers they encounter. But this approach has two disadvantages. Firstly, it requires allocating a bit for marking in the IP packet header. However, all bit positions in the IP4 are currently allocated, notwithstanding some proposals to overload header fields for path tracing applications [8].

Secondly, and more problematically, a domain that used packet marking to signal selection would have to filter the mark for all incoming packets. Otherwise, it would be possible to overload the measurement subsystem by injecting marked packets. Sealing the network against this attack would require all edge routers to have this filtering capability. Making such a change would be a formidable task in a large multi-vendor environment. This is not an issue for TS, since the use of a strong parameterizable hash function, with private parameter settings and selection range, makes it exceedingly difficult to craft streams of packets that would be selected. Also, TS can be deployed incrementally, enabling TS-based applications to operate for the logical overlay network spanned by TS enabled routers.

2) Independent Sampling (IS): This entails routers selecting packets in an uncoordinated manner, each router selecting

some proportion of the packets that pass through it, for example by periodic or simple random sampling. IS destroys the trajectory semantic, since a given packet is very unlikely to be sampled at all points in its trajectory. Thus with IS, passive performance measurement of individual packets, including loss and network latency, becomes practically impossible. Furthermore, it is not generally effective to substitute statistical performance measures. Section V shows that, for packet loss rates likely to be found in the Internet, statistical estimation of loss in a link of transmission rate q incurs a variance roughly $2/(1-q)$ times larger for IS than for TS, e.g, 50 times larger for a loss of 1%.

J. Outline of the Paper

We describe the TS architecture and record concepts, our model, and notation in Section II. In Section III we describe a method to deal with report loss than enables trajectory reconstruction to be performed in an unbiased manner. Our approach is for ingress nodes to record the presence of labels in Bloom filters [1], which are transmitted to the collector, where elimination is performed. The elimination procedure is unbiased, and robust with respect to partial loss of the Bloom filter in transit. This enables unbiased inference of original traffic intensities for network management applications. This is done transparently, using an *effective sampling rate* that is the product of the TS target sampling rate with the rate of duplicate elimination.

Even after duplicate elimination, other applications must be adapted to report loss. Section IV addresses the reconstruction of network paths from incomplete trajectories reconstructed from multiple packets in the same flow. Provided transmission rates are not identically zero, multiple packet reports that take the same network path eventually cover the path, in the sense that at least one report is received from each router on the path. We analyze the mean number of packets that must be reported to attain this coverage.

Section V shows how link loss rates can be inferred even in the presence of report loss. The main idea is that the collector can infer the loss rates of report packets if the reports include sequence numbers. In both Section IV and V we compare the performance of TS with applying the same methods with reports from independently sampled packets. In both cases, the performance is noticeably better for TS, and particularly so in the estimation of loss rate. We conclude in Section VI.

II. Overview and Notation

In this section, we give an overview of the proposed system architecture and methods we propose to deal with report loss.

We first define some notation:

- The measurement domain is a directed graph $G(V, E)$, where we refer to vertices as *routers* and to edges as *links*. The set of vertices comprises a set of *external routers*, a set of *edge routers*, and a set of *core routers*. External routers connect to edge routers through an *ingress link*. Links that are not ingress links are called *core links* or *internal links*.

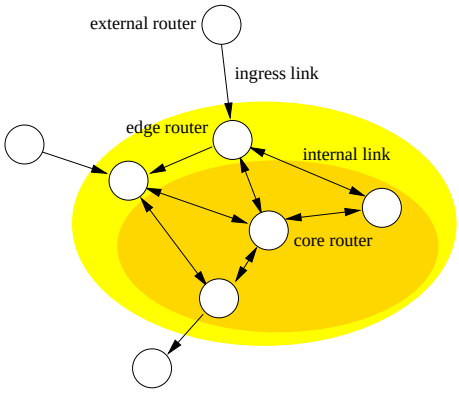

Fig. 2. A *measurement domain* consists of a set of routers under administrative control of a network operator, plus a set of external routers that act as sources and sinks for all traffic entering and leaving the network. The internal routers are further subdivided into *edge* and *core* routers, where edge routers are those having an incoming link (called *ingress link*) from an external router. In general, core routers only collect labels from their incoming links, while edge routers collect additional information from ingress links.

- A trajectory is a path, i.e., an ordered set of links $(e_1, \ldots, e_n)$ in the measurement domain G. $\mathcal{T}$ denotes the set of all valid trajectories. A *trajectory subset* is a set of links $\{e_1, \ldots, e_n\}$ that do not necessarily form a path, but which is a subset of at least one trajectory $\tau \in \mathcal{T}$. A trajectory subset arises when a trajectory is reported as a set of reports from each link it traverses, and some reports are lost.
- A *trajectory sample* is a trajectory subset that is reconstructed by the collector based on reports received from the links on a packet's trajectory. In general, trajectory samples are not used in raw form, but are aggregated into higher-level statistics as outlined in the introduction. Examples include inference of the traffic and path matrices (see Section III-D); passive measurement of packet loss rates (see Section V) and passive measurement of one-way packet delay (see e.g. [10]).
- The concepts of *label graph* and *lossy label graph* record how many times a particular label l has been observed on every link in the network. Specifically, a label graph C_p^l is a mapping $E \to \mathbb{Z}$; it denotes how many times a label l has been observed *for each link* $e \in E$, where p is the sampling probability. For the case where labels are transported unreliably to the collector, where q_v is the probability that a label from a router v is lost, we call the resulting label graph a *lossy label graph* and denote it with $C_{p,\mathbf{q}}^l$, where $\mathbf{q} = (q_1, \ldots q_{|V|})$ contains all the report loss rates for every router.
- A *path matrix* $PM(k, \tau)$ is a function that maps a *key* k and a path τ to a volume of traffic. The key k is itself a property of packets, e.g., the source or destination IP address or autonomous system (AS). Note that if the key is trivial (a constant), then the path matrix is simply the traffic volume for every possible path through the network. The path matrix provides the most fine-

grained spatial representation of the traffic that flows through a measurement domain over some time interval of interest. Trajectory sampling can essentially be viewed as estimating the path matrix (where the key can be any function of the packet collected by the method).

We discuss some of the assumptions underlying the description of the proposed method for inference from lossy reporting.

- Throughout this paper, we focus on a single reporting epoch, i.e., a time interval that is some upper bound of the lifetime of a packet in the network. We assume that two labels received within such an epoch may stem from the same packet or from different packets; if two labels are from different epochs, then they must stem from different packets. Therefore, the main challenge is to reconstruct the set of trajectories of sampled packets within an epoch, and the entire discussion in this paper focuses on a single epoch[1].
- We assume in the performance analysis that hash functions are *perfect*, i.e., we can view a hash function computed over some set of objects (e.g., packets) as generating a set of i.i.d. uniform random variables over the range of the function. This assumption is justified by the properties of hash functions and by our earlier work [3], where we show that there is enough entropy, i.e., variability from packet to packet, to ensure that sampling decisions and labels essentially appear random.

The basic trajectory sampling architecture that is able to cope with report loss in shown in Figure 3. First, for every link in the measurement domain (internal and ingress links), reports are generated from sampled packets and transported to a collector[2].

TS reports include one of both of the following:

- *Key:* fields from the invariant portion of the IP and transport headers. We assume that keys will not distinguish between different packets within a flow.
- *Label:* a hash calculated over invariant part of the packet. We assume that the hash acts on fields that vary between packets of a flow (e.g. IP identification, TCP sequence numbers) in order the packets may be distinguished. This is important for applications such as passive delay measurement that must distinguish individual packets.

The basic reporting paradigm for TS is:

- *Label Reporting:* The ingress link reports both key and label, while core links report only labels. Core reports for which there is no ingress report with matching label are discarded at the collector.

The above is identical to the reporting paradigm in the reliable case, as proposed in [3]. We now add a new type of report that allows the collector to eliminate all duplicate labels, i.e., all reports from multiple packets that happened to generate identical labels. This is necessary because the collector has no guarantee to receive all the reports of these packets, with a possibility of missing these duplicates. The details of this process are explained in the next section.

[1]We discuss some methods to deal with the various packet and reporting delays in [4].

[2]In a typical scenario, multiple reports are aggregated into a single *report packet* for efficiency; when there is no danger of confusion, we do not make this distinction explicit.

III. Unbiased Duplicate Elimination under Loss

A. Challenges for Unbiased Duplicate Elimination

As we have mentioned previously, there is a nonzero probability that two or more packets produce identical labels because the hash function is many-to-one [3], [4]. When more than one packet has the same label, it would sometimes be possible to disambiguate them. However, this disambiguation is costly, and may introduce bias into estimators if care is not taken. Therefore, in [4] we have proposed a different approach: eliminating all duplicate labels, whether they can be disambiguated or not. While this is slightly suboptimal because we ignore useful information, it greatly simplifies reconstructing trajectories and avoids bias in estimators.

If we assume that reports are carried reliably from routers to the collector, we can simply use the labels collected from ingress routers to detect and discard duplicate labels, because if a label is observed multiple times on an ingress router, it is necessarily a duplicate.

In the unreliable case, this approach cannot be used directly because ingress reports may be lost, which can result in undetected duplicates. Consider the set of labels received from some ingress node over a time period of interest. If we simply transfer these labels as normal reports, i.e., as sequences of labels, then in the case where one or several reports are lost, we have no idea what the lost labels were. We could try to FEC-encode the set of labels using erasure codes in order to tolerate a certain loss rate, but this is computationally expensive, and would only work if the actual loss rate is smaller than the predefined target that the code was designed for. Another option would be to send the complement of the observed labels, i.e., all the label values *not* observed at each ingress link. However, this approach is very wasteful, as the complement set is much larger than the label set.

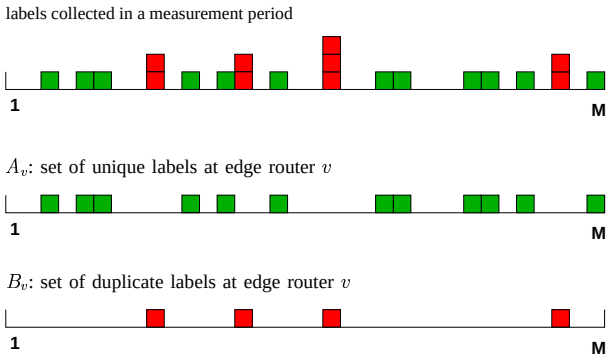

Fig. 4. At every ingress router v, the set of labels is partitioned into a set of unique labels A_v and a set of duplicate labels B_v, which are then separately encoded using a packetized Bloom filters.

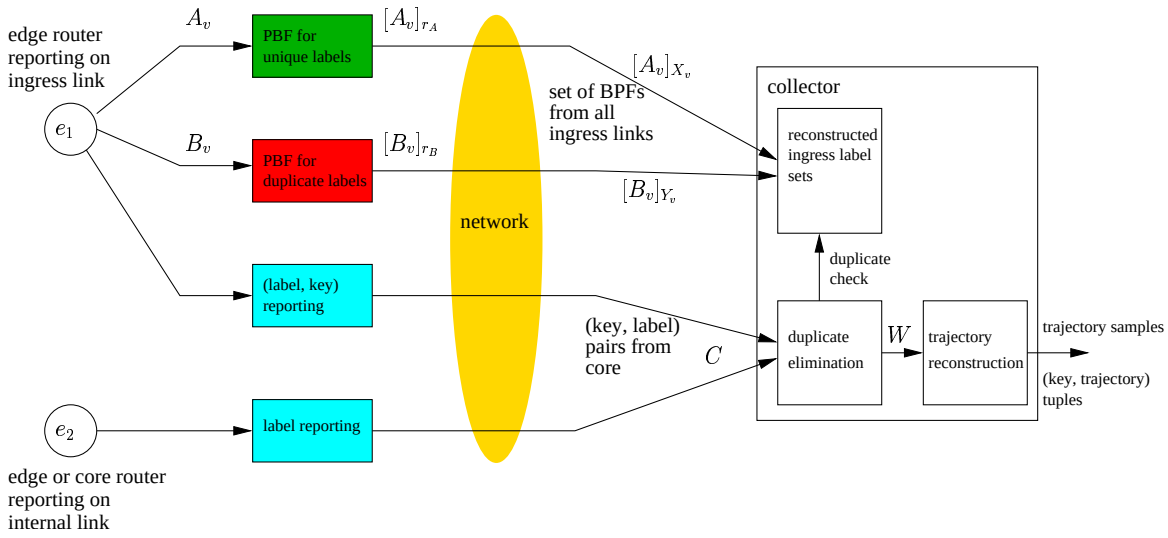

Fig. 3. The type of report generated for a sampled packet depends on the link on which a packet was observed. For an ingress link, a report contains the packet's label along with other information of interest (the key); separately, the label is reported as part of a set A_v or B_v, depending on whether the label was unique or duplicate. Both A_v and B_v are encoded as packetized Bloom filters $[A_v]_{X_v}, [B_v]_{Y_v}$ for transport. The collector tests every label l received from internal links against the PBFs, eliminating every label from consideration that has been observed multiple times.

B. Packetized Bloom Filters

We instead propose a data structure based on *Bloom filters* to encode the set of labels observed on ingress links. A *Bloom filter* [1], [2] is a data structure to compress a set membership function. We essentially encode the set of labels as a Bloom filter and transport it to the encoder as a sequence of separate packets that we refer to as a *packetized Bloom filter (PBF)*.

More formally, let A be a set of labels out of an alphabet of size M, and let $n = |A|$ be its size. We denote by $[A]_r$ the r-packet PBF for A obtained as follows. The PBF uses a set of hash functions $\{h_i, i = 1, \ldots, k\}$, where $h_i : \{1, \ldots, M\} \to \{1, \ldots, rs\}$, with r the number of report packets generated from A (cf. Fig. 5) and s the size of a report packet. Specifically, each report packet j is a bit array x_j of length s, where $x_j(y - (j-1)r) = 1$ if and only if $h_i(l) = y$ for at least one $i \in \{1, \ldots, k\}$ and any $l \in A$. Each packet also includes some control information (time-stamp, j, etc.). In essence, this amounts to first generating a large Bloom filter of size rs, which we then transmit as r individual packets of size s each.

To check whether a candidate element y is in the compressed set, we check whether all the bit positions $\{h_1(y), \ldots, h_k(y)\}$ are set to one. Thus, a Bloom filter achieves compression of a set at the expense of only false positives (i.e., adding some elements to the set), but no false negatives. This is because the bit positions corresponding to an element y present in the set are guaranteed to be set to one, while there is no guarantee that at least one bit position corresponding to an element y' not in the set is set to zero.

Now note that we can ensure the same property (only false positives) with any received subset of packets of a PBF, if we simply replace each missing packet with a vector of length s of only 1's. Obviously, the probability of false negatives increases with the fraction of lost PBF packets, but we never falsely reject a label. We will see that this property allows us to ensure that despite the lossy transport of the PBFs from ingress routers, any duplicate labels are *guaranteed* to be eliminated by the collector, because we cannot miss a label that has been observed more than once. Of course, some unique labels are also eliminated due to false negatives, and the probability of elimination increases with the number of lost PBF packets. However, we show below that these false positives can easily be compensated for.

C. Duplicate Elimination and the Elimination Rate

We use the PBF in our proposed architecture in the following way. Consider an edge router v, and the set of packets sampled on the ingress links connected to v. This set of packets generates a set of labels. We partition the set of labels into two subsets A_v and B_v, where A_v contains the set of *unique* labels at v (i.e., only a single packet gave rise to each label), and B_v contains the set of packets with *duplicate* labels at v (i.e., multiple packets gave rise to each label).

We now generate two PBFs $[A_v]_{r_A}$ and $[B_v]_{r_B}$ from these sets and transmit them unreliably to the collection system. As some of the packets may be lost, only a subset X_v, Y_v of the original packets is received. We denote the resulting partial Bloom filter by $[A_v]_{X_v}$ and $[B_v]_{Y_v}$. Once the collector has received $([A_v]_{X_v}, [B_v]_{Y_v})$ for all edge routers v as well as explicit label report (packets containing sets of labels) from core routers, it proceeds as follows.

First, the collector matches the loss rates for all Blooom filters $\{[A_v]_{X_v}\}$. It achieves this by selecting a subset X'_v of received packets X_v for every PBF, such that $|X'_v| = \min_{v \in \{\text{edge routers}\}} |X_v|$, i.e., all PBFs have the same length, equal to the length of the smallest received PBF. This will

ensure that the duplicate elimination probability does not depend on what edge router a label has been observed on[3].

Second, for every label l received explicitly from a core or edge router, the collector eliminates l if

$$\text{if} \quad l \in [B_v]_{Y_v} \text{ for some } v \tag{1}$$

$$\text{or if} \quad l \in [A_{v_1}]_{X_{v_1}} \cup [A_{v_2}]_{X_{v_2}} \text{ for some } v_1 \neq v_2. \tag{2}$$

In other words, any label l that has either been observed more than once at a single ingress router, and/or been observed at multiple ingress routers, is eliminated from the pool of labels. We call C the set of explicit labels received from internal links by the collector, and W the set of labels after duplicate elimination (cf. Fig. 3).

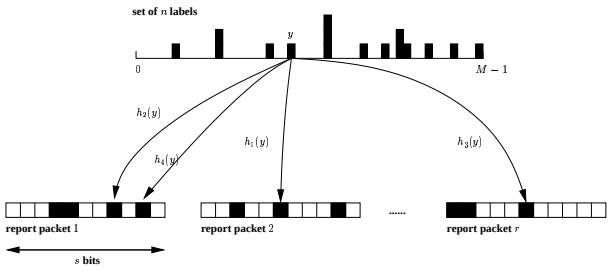

Fig. 5. A *packetized Bloom filter (PBF)* encodes the set of labels received at edge links into r packets of length s each.

Note that because of the possibility for a Bloom filter to produce false positives, some globally unique labels can also be eliminated. However, we next show that the elimination process is unbiased. This implies that the set of labels left after the duplicate elimination can be regarded as having been produced by a label assignment process that assigns a *unique* label to every sampled packet, but at a lower sampling rate.

Consider a fictitious system in which labels are a-priori unique (e.g., by selecting them out of a very large alphabet, or through some global coordination). We denote by $C^*_{p,\mathbf{q}}$ the label subgraph obtained with sampling rate p and report loss probabilities $\mathbf{q}$.

Theorem 1: Assume a network $G(V, E)$ and a set of packets with associated trajectories. Then the label subgraph $C^l_{p,\mathbf{q}}$ for every received label l satisfies

$$\mathsf{P}\left[C^l_{p,\mathbf{q}}\right] = \mathsf{P}\left[C^*_{\beta p,\mathbf{q}}\right], \tag{3}$$

where $\beta = \mathsf{P}[l \text{ eliminated}]$.

Proof: Partition the set of labels of sampled packets into two sets U and V, where U denotes the set of unique labels, and where V denotes the set of labels that occur more than once. Note that a label $l \in U$ occurs only in the set $A_{v(l)}$, where $v(l)$ is the ingress router where the corresponding packet entered.

[3]Note that this procedure is adopted for simplicity, although it has the disadvantage of increasing the duplicate elimination probability. An alternative, more complex approach would consist in not equalizing the PBF lengths and to renormalize the weights of different trajectories.

Consider an arbitrary duplicate label $l \in V$. By definition, this label either (a) occurs in the set $A_{v(l)}$ and at least one other set A_w or B_w with $w \neq v(l)$, or (b) it occurs in the set $B_{v(l)}$ and possibly one or more sets A_w or B_w with $w \neq v(l)$. This implies that a label in $l \in V$, by the fundamental property of Bloom filters (no false negatives), will be correctly eliminated by the criteria (1) and (2).

Next, consider an arbitrary unique label $l \in U$. Define the following events. For an ingress node w, $E_w = \{l \in [A_w]_{X'_w}\}$, and $F_w = \{l \in [B_w]_{Y_w}\}$. Note that $E_{v(l)}$ is always true, while E_w, $w \neq v(l)$, and F_w can be true only due to a false positive match in the corresponding PBF.

It follows from the perfect hashing assumption for the sets of hash functions $\{h_i\}$ used to compute the PBFs that for any $w \neq v(l)$, the events $\{E_w : w \neq v(l)\}$ and $\{F_w\}$ are independent of each other and of l. Furthermore, the $\mathsf{P}[E_w]$ depend only on $|X'_w|$, which by definition are equal. Both $\cup_{w \neq v(l)} E_w$ and $\cup_w F_w$ are independent of each other and do not depend on $(l, v(l))$. Therefore, the events $\{l \text{ eliminated}\}$ for all $l \in U$ are equiprobable and mutually independent.

Given the perfect hashing assumption on the label hash l, the label l of each packet is independent of the packet itself, and it follows that every packet is eliminated independently with equal probability β. This is equivalent to sampling the packet population with sampling probability βp. ■

D. Effective Sampling Rate and Applications

The above theorem implies that the set of labels W after duplicate elimination can be regarded as resulting from a trajectory sampling process that avoids label collisions altogether, i.e., where every label is unique. This simplifies the statistical inference of estimators, such as the loss rates on a set of links, as there is no need to explicitly account for the possibility of label collisions to avoid bias in constructing these estimators. Rather, we consider sampling to have taken place with the *effective sampling rate* βp. We give two examples of inference using the effective sampling rate:

- *Inference of Packet Loss Rates:* An example is provided in Section V, where we construct estimators for link loss rates: we can simply work on the set W and assume a-priori unique labels when constructing an estimator. The only correction is to assume that the sampling rate was βp.
- *Inference of Path Matrix Elements:* let K_{in} be a packet input key, i.e., some function of the packet key that does not depend explicitly on the destination IP address or destination TCP/UDP port numbers. An example would be source Autonomous System (AS). Likewise, let K_{out} be a packet output key, i.e, some function of the packet key that does not depend explicitly on the source IP address or source TCP/UDP port numbers, e.g., the destination AS. Note the usual traffic matrix elements are

$$TM(k_{\text{in}}, k_{\text{out}}) = \sum_{\tau \in \mathcal{T}} PM(k_{\text{in}}, k_{\text{out}}, \tau). \tag{4}$$

Depending on the application, $\mathcal{T}$ might be the set of all paths in a domain, or the set of all paths in the domain that connect specific ingress and egress links.

Let $PM^{\mathrm{TS}}(k_{\mathrm{in}}, k_{\mathrm{out}}, \tau)$ denote the a path matrix element of trajectory sampled traffic after duplicates have been eliminated. Then the corresponding path matrix element of the original traffic is estimated by dividing by the effective sampling rate:

$$\widehat{PM}^{\mathrm{TS}}(k_{\mathrm{in}}, k_{\mathrm{out}}, e) = PM^{\mathrm{TS}}(k_{\mathrm{in}}, k_{\mathrm{out}}, \tau)/(\beta p) \qquad (5)$$

Note that in practice, β can be easily derived from auxiliary information in report packets giving the size of the sets of labels before (C) and after (W) elimination. It follows from the law of large numbers that the ratio $|W|/|C|$ converges a.s. to β when the number of received labels grows large.

E. Parameter Settings for the PBF

Finally, we discuss parameter settings in the PBF. First, assume that all the labels are observed at a single point, where they are encoded in a PBF. We encode both sets A_v and B_v into PBFs; note that $|A_v| \gg |B_v|$, therefore the cost is dominated by r_A.

Assume a fraction α of the r_A report packets is received, i.e., the report loss rate is $1-\alpha$. When k is chosen optimally [2], a bit in the Bloom filter is zero or one with equal probability $1/2$, and the false positive probability is given by

$$f = (p_1)^k = (1-\alpha/2)^k = (1-\alpha/2)^{r_A s \ln 2/n}, \qquad (6)$$

where $p_1 = (1-\alpha/2)$ is the probability that a bit in the *received* PBF is one. Therefore, to ensure a reasonably small error probability, a set of n elements has to be encoded into $r_A s = O(n/\alpha)$ bits. As s should reasonably lie within the range of 10^3 to 10^4 bits in IP, this determines the number r_A of PBF packets that should be used to encode the labels received during an epoch. A similar reasoning gives r_B.

In our previous work [3], we computed the optimal number of sampled labels n^* and the optimal alphabet size M^*, given a constraint c on the total number of bits to be collected from the network in one measurement period. We showed that $M^* = c \log 2$ and $n^* = M^*/\log M^*$. Therefore, $r_A s = O(c/\log c)$, which is considerably cheaper than explicitly sending the complement of observed labels, which costs $O(c)$.

In general, of course, labels are observed at multiple ingress links. The above dimensioning argument should then be applied on a per-ingress basis, i.e., the number of exported PBF packets r_A and r_B should depend on the expected number of labels n_e observed on ingress link e. This avoids that resources are wasted by transmitting underpopulated Bloom filters from slow links.

In summary, the duplicate elimination process described in this section ensures that duplicates are guaranteed to be eliminated. The robustness of this process to report loss was bought at the expense of the loss of a small number of unique labels. Appropriate dimensioning of the PBF sizes ensures that this additional loss rate remains small. The overhead of collecting PBF from ingress links is small with respect to the total overhead due to label collection from the entire network. Most importantly, the duplicate elimination process can be treated as simply subsampling the set of sampled packets. Therefore, the possibility of duplicates can be ignored by statistical estimators.

IV. Path Coverage and Lossy Reporting

A. Coverage Count and Loss Model

In the introduction we stated that one of the new applications of TS is the ability to trace packet paths through a network. With lossy reporting, the set of links for which a packet is received may not form a contiguous path through the network. Nevertheless, when routing is stable, packets from a given traffic flow are expected to follow the same path (or set of paths if load balancing is used). Provided the report loss rate is not one on any link on a path, eventually a report will be received from every link on the path. Thus taking the union of label subgraphs derived from multiple packets from the same flow, will eventually cover each link on the the path or set of paths followed by the flows packets.

The covering approach requires that packets report a quantity that enables them to be identified as members of the flow of one or more packets, and that which uniquely determines the path taken by the packet. Here we will assume that the key (or more generally, some subfield of the key) serves this purposes. Thus, packets are grouped into flow on the basis of key value, and the key contains the IP destination address.

We shall assume that the set of packet reports has already undergone duplicate elimination as described in Section III. We derive expressions for the mean number of packets required to cover a path. Let τ be a trajectory, and let $T = \#\tau$ denote the number of links in τ. We assume that packets on the trajectory are sampled and reports dispatched to a collector from each link e in the trajectory. Consider a sequence of packets labeled by $n = 1, 2, \ldots$. The *coverage count* for τ is the smallest integer for which a report has been received from each link in τ; at this time we say that τ has been covered.

For analysis we assume the following simple statistical model of trajectory sampling: with probability p a packet is selected at all links on its trajectory; otherwise at none. We ignore transmission loss, and focus instead on report loss and assume that reports are independently successfully transmitted with probability q.

For this model, coverage time and its asymptotic behavior is characterized using the following result:

Theorem 2: Let $\{x_{ei} : \ e = 1, \ldots, T; i = 1, 2, \ldots\}$ be i.i.d. indicator random variables with $\mathsf{P}[x_{ei} = 1] = q$. Let $N = \inf\{i : \min_e \max_{j \le i} x_{ej} = 1\}$.

(i) $\mathsf{E}[N] = F(T,q) := \sum_{k=1}^{T} \binom{T}{k} \frac{(-1)^{k-1}}{1-(1-q)^k}$
(ii) $F(T,q) \to 1$ and $\partial F(T,q)/\partial q \to -T$ as $q \nearrow 1$.
(iii) As $q \to 0$, $F(T,q) \sim F_1(T,q) = H_T/q$ where H_T is the harmonic number $\sum_{i=1}^{T} 1/i$.
(iv) $qF_1(T,q) \sim \gamma + \log T$ as $T \to \infty$, where γ is the Euler constant $\lim_{T\to\infty}(H_T - \log T) = 0.577216\ldots$.

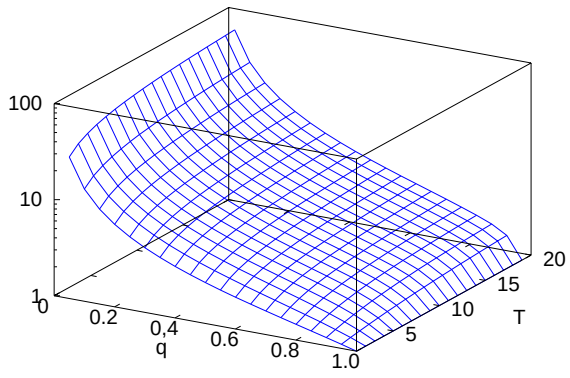

Fig. 6. Surface plot of mean coverage count $F(T,q)$ as function of path length T and report transmission probability q. Note logarithmic vertical axis. $F(T,q)$ grows as $1/q$ for small q, but growth with path length T is relatively slow.

Proof: (i) $N = \max_e N^{(e)}$ where $N^{(e)} = \inf\{i : x_{ei} = 1\}$. The $N^{(e)}$ are i.i.d geometrically distributed random variables, with $\mathsf{P}[N^{(e)} > n] = (1-q)^n$. Hence $\mathsf{P}[N > n] = 1 - (1-(1-q)^n)^T$ and $\mathsf{E}[N] = \sum_{n\geq 0} \mathsf{P}[N > n] = \sum_{n\geq 0} \sum_{k=1}^{T} \binom{T}{k} (-1)^{k+1} (1-q)^{nk} = F(T,q)$, after binomial expansion of $\mathsf{P}[N > n]$. (ii) follows simply from (i). (iii) $\lim_{q\to 0} qF(T,q) = \sum_{k=1}^{T} \binom{T}{k}(-1)^{k+1}/k = \int_0^1 dx(1-(1-x)^T)/x = \int_0^1 dx(1-x^T)/(1-x) = H_T$. (iv) follows from the definition of γ. ∎

In the current context, N is the number of sampled packets needed to receive at least one packet report form each of T links on a path. The form of $F(T,q)$ is displayed in Figure 6. As predicted from Theorem 2, $F(T,q)$ grows as $1/q$ for small q, but growth with path length T is relatively slow.

B. Reporting Strategies and Mean Coverage

The mean coverage count for label reporting is calculated as follows. Only packets which generate a label report that reaches the collector contribute. These occur at a rate pq relative to the original packet stream. The mean coverage count within these packets for the $T-1$ core links is $F(T-1,q)$. Hence the overall mean coverage count is

$$N_{\mathrm{TS,L}} = (pq)^{-1} F(T-1,q) \tag{7}$$

From Theorem 2 the range of behavior as a function of the report loss rate q is

$$N_{\mathrm{TS,L}} \approx \begin{cases} \frac{1+T(1-q)}{p}, & q \approx 1 \\ H_{T-1}/(pq^2), & q \approx 0 \end{cases} \tag{8}$$

Note from (ii) that for nearly lossless reporting ($q \approx 1$) the mean coverage count grows affinely with the path length T. The rapid q^{-2} growth for small q can be tempered by adjusting the reporting strategy:

- *Key Reporting:* All routers report keys; labels may be reported as well if it is desired to distinguish different packets within a flow.

For key reporting, the mean coverage count is

$$N_{\mathrm{TS,K}} = p^{-1} F(T,q) \tag{9}$$

which behaves as $H_T/(pq)$ for small q, reducing the growth by q relative to label reporting. When report loss is small ($q \approx 1$), The behavior for small report loss ($q \approx 1$) is the same as for TS, to leading order in $(1-q)$.

C. Comparison with Independent Sampling

Routers which do not offer TS may still be able to sample packets; see e.g. [7]. With key reporting, trajectory coverage can then be performed at the collector. We model this with independent sampling (IS) of packets at probability p. The mean coverage count in this case is

$$N_{\mathrm{IS,K}} = F(T,pq) \tag{10}$$

We compare with key reporting for TS in two regimes. Suppose there is no report loss: $q = 1$. By Theorem 2:

$$\frac{N_{\mathrm{IS,K}}}{N_{\mathrm{TS,K}}} = pF(T,p) \approx H_T \tag{11}$$

since p is assumed small. For example, for the longest paths typically observed in the wide area Internet, we take $T = 30$ [6]; then the above ratio is 4.0.

In the lossy regime $q \to 0$ then by Theorem 2,

$$\frac{N_{\mathrm{IS,K}}}{N_{\mathrm{TS,K}}} = \frac{pqF(T,pq)}{qF(T,q)} \approx 1 \tag{12}$$

since p is small. The decreasing advantage of TS relative to IS as q becomes small stems from the fact that increasingly, once a report on a given packet has been received from one link, reports are reasonably likely to have been received from other links.

A fuller comparison of the sampling methods should also take account of the reporting bandwidth. Recall labels have the advantage of being smaller than keys, and hence consume less bandwidth. Let $\alpha > 1$ denote the ratio of the size of a key to the size of a label. Label sizes of 4 bytes have been found to be sufficient to distinguish packet in TS. On the other hand, a flow key may include a significant proportion of the IP and UDP/TCP packet headers. We assume α to be about 10.

Measuring in units of label size, key reporting consumes $T\alpha$ while label reporting consumes $T + \alpha$ without keys, and $2T\alpha$ keys. Comparing TS without keys and IS, the ratio of the mean bandwidth required to cover a trajectory of length T is

$$\frac{N_{\mathrm{IS,K}}}{N_{\mathrm{TS,L}}} \frac{T\alpha}{T+\alpha} = \frac{pqF(T,pq)}{F(T-1,q)} \frac{1}{T^{-1}+\alpha^{-1}} \tag{13}$$

With lossless reporting ($q = 1$) and small p this ratio is approximately $H_T/(T^{-1}+\alpha^{-1})$. Taking $T = 30$ and $\alpha = 10$ yields a ratio of 30.0. For lossy reporting, the ratio behaves as $q/(T^{-1}+\alpha^{-1})$. Thus for sufficiently small q, IS can have a bandwidth advantage, i.e., the ratio becomes less than 1.

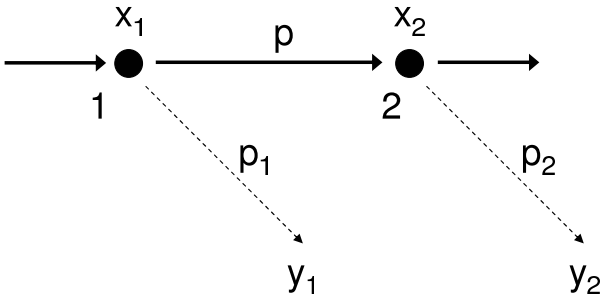

Fig. 7. Network and measurement collection configuration for loss estimation

This happens when $q < T^{-1} + \alpha^{-1}$, i.e., about 0.13 using above values for T and α. This would be an extremely low transmission ambient transmission rate in the Internet, so TS would be expected to have the bandwidth advantage over IS.

V. Loss Inference in the Presence of Report Loss

A. Distinguishing Packet and Report Loss

Passive measurement of packet loss is one of the most attractive new applications of TS. With reliable reporting, packet loss is manifest by trajectories that terminate without the packet reaching its destination. Consider the simplest case that traffic of a given key class is routed along a single path (that we assume stable routing and no load balancing). The loss rate for packet of that class at a link on that path is estimated as the proportion of packet reports for the class that terminate at that link.

With unreliable reporting, reports may be lost. In order to estimate packet loss, we must disentangle the effect of report loss. Clearly this is not possible at the level of single packets. Loss of a packet at a given link e of a path will produce the same set of packet reports at the collector as loss of reports from all links subsequent to e on the path. Instead, we have to distinguish packet and report loss at the statistical level, employing reports from multiple packets. In the following we shall assume that the set of packet reports has already undergone duplicate elimination as described in Section III.

B. Estimating Packet Loss with Known Report Loss

The generic configuration for loss estimation is show in Figure 7. We wish to estimate the packet loss rate on along a path $1 \to 2$. Packet reports are collected from both nodes. We make no assumptions concerning the collection paths: they may have subpaths in common, and may encompass the paths $1 \to 2$ or $2 \to 1$.

Consider some number n of packets in a class of interest. The presence of absence of a given packet, or its reports, is indicated by the random variables x_i and y_i respectively. A given packet k is present at node i iff $x_i^{(k)} = 1$, with $x_i^{(k)} = 0$ otherwise. A report for that packet is received at the collector from node i if $y_i^{(k)} = 1$, with $y_i^{(k)} = 0$ otherwise. Thus, $m_i = \sum_k y_i^{(k)}$ is the number of packets reaching the collector from node i. We denote by q_i the conditional probability for a report to reach the collector from node i, given that the underlying packet is sampled at i; the probability is assumed uniform over all packets.

Suppose the transmission rates q_i for the packet reports were known. Then we would estimate the transmission rate q on the link $1 \to 2$ by

$$\widehat{q} = \frac{m_2 q_1}{m_1 q_2} \tag{14}$$

This estimator is consistent for a stationary loss process, provided the law of large numbers holds for the numbers of packet and reports transmitted. This holds, for example, if the sequence of random variables $(x_1^{(k)}, y_i^{(k)})$ labeled by packet k sampled at node 1, forms a stationary and ergodic process. Let there be are n_i sampled packets in the class of interest present at node i. As $n_1 \to \infty$, then $n_2/n_1 \to q$, $m_i/n_i \to q_i$, and hence

$$\widehat{q} = \frac{m_2}{n_2}\frac{n_2}{n_1}\frac{n_1}{m_1}\frac{q_1}{q_2} \to q. \tag{15}$$

The numbers of packets n_i in the class that are sampled at nodes i do not enter explicitly into the estimator $\widehat{q}$.

C. Estimating Packet Loss with Unknown Report Loss

If the q_i are not known, they may be estimated from the streams of packet reports. For example, assume that reports are transmitted individually to the collector, and that they carry transmission sequence numbers. Suppose M_i successive trajectory samples reach the collector from node i in a given period. We assume that the collector performs any reordering of samples with respect to transmission sequence number, if required. Adding 1 to the difference between the transmission sequence numbers of the first and last of these packets yields N_i, the number of trajectory samples transmitted between transmission of the first and last received samples. The transmission rate for trajectory samples from node i is estimated by $\widehat{q}_i = M_i/N_i$.

The statistics of the packet process may potentially influence the transmission rate of reports. For example, burstiness in the packet stream of a traffic class can be inherited to some extent by the sampled packet stream. However, we expect sampling rates to be quite small, hence "taming" the burstiness by spacing out packets in the sampled stream, as compared with the original stream. Beyond this, there is no reason to assume that packet selection and transmission of packet reports will be coupled with packet content. For these reasons we assume that the transmission rate of packet reports for the class under study is the same as for all traffic. Thus we are free to employ transmission sequence numbers applied to packet reports from the whole packet stream, rather than those from the traffic class under study. Thus we estimate the transmission rate q by replacing q_i with $\widehat{q}_i$ in (14):

$$\widehat{q} = \frac{m_2 M_1 N_2}{m_1 N_1 M_2} \tag{16}$$

D. Loss Estimation Variance

Correlation between loss of reports (from different packets and/or different nodes) does not effect estimator consistency, but it can effect estimator variance. Now, correlation between loss reports is manifest as non-zero conditional covariance between $y_1^{(k)}$ and $y_2^{(k)}$, conditional on packet k having being sampled. But since terms $y_1^{(k)}$ appear in the denominator, while terms $y_2^{(k)}$ appear in the numerator, positive correlations between appear actually reduce estimator variance.

To show this, we compute the variance of the estimator (16), asymptotically for a large number of samples. For simplicity we ignore the variability in the numbers M_i of reports and packet N_i in all classes. This is a reasonable approach if the traffic under study forms a small proportion of the total, and is equivalent to treating the ratios M_i/N_i as fixed numbers q_i as in (14).

We analyze the asymptotic variance of $\widehat{q}$ from (14), as $n \to \infty$, using the Delta method [9]. This derives the asymptotic behavior of functions of sums of random variables that obey the central limit theorem, as we now summarize:

Lemma 1: Let Z and Z_n be any vector-valued random variables such that $\sqrt{n}(Z_n - Z)$ has asymptotically, as $n \to \infty$, a multivariate Gaussian distribution with mean zero and covariance matrix C. Then for any real function f of the random variables, differentiable at Z, $\sqrt{n}(f(Z_n) - f(Z))$ has asymptotically, as $n \to \infty$, a multivariate Gaussian distribution with mean zero and covariance $c = v \cdot Cv$ where $v = \nabla f(Z)$ is the gradient of f at Z.

We model TS as selecting trajectories independently with probability p, and apply the Delta method using

$$Z_n = n^{-1}(\textstyle\sum_k z^{(k)} y_1^{(k)}, \sum_k z^{(k)} y_2^{(k)}) \quad (17)$$

$$Z = \lim_{n\to\infty} \mathsf{E} Z_n = (pq_1, pqq_2) \quad (18)$$

$$f(m_1, m_2) = m_2/m_1 \cdot q_1/q_2 \quad (19)$$

One finds $v = (-q/(pq_1), 1/(pq_2),)$, and C is the covariance matrix of (zy_1, zy_2) namely,

$$C = \begin{pmatrix} pq_1(1-pq_1) & pd_{12} + p(1-p)qq_1q_2 \\ pd_{12} + p(1-p)qq_1q & pqq_2(1-pqq_2) \end{pmatrix} \quad (20)$$

where d_{12} is the covariance of $y_1^{(k)}$ and $y_2^{(k)}$, conditional on packet k having been sampled. Summarizing:

Theorem 3: The distribution of $\sqrt{n}(\widehat{q} - q)$ converges as $n \to \infty$ to a Gaussian random variable with mean 0 and variance $c = v \cdot Cv$, equal to

$$c^{\mathrm{TS}} = \frac{q(q_1(1-qq_2) + qq_2(1-q) - 2d_{12})}{pq_1q_2}. \quad (21)$$

This establishes the earlier claim that positive correlation between transmission of reports reduced estimator variance. Conversely, negative correlation increases estimator variance. Negative correlation may occur if the reports from different routers compete for transmission resources along a common path to the collector.

Correlations between different probes do not qualitatively change the results. Gaussian asymptotics still prevail, although the asymptotic covariances are in general different. For a given mode, e.g., the $(x^{(k)}, y^{(k)})$ form a Markov process, the covariances can be calculated using generalizations of the Central Limit Theorem for dependent variables.

Finally, although we have estimated the loss rate on a single path, one could perform joint estimation of the loss rates on a number of (possibly) intersecting paths. It can be shown that the loss rate estimators remain consistent under the previous assumptions. Variance is calculated using a higher dimensional analog of the matrix (20), whose elements take into account potential correlations of loss between export from different routers, e.g., due to intersecting export paths.

E. Comparison with Independent Sampling

We now isolate the difference in estimator variance due to sampling method. For reference, consider first TS with the simplifying assumption of no report loss: $q_i = 1, d_{12} = 0$, Then c^{TS} reduces to

$$c^{\mathrm{TS}} = q(1-q)/p. \quad (22)$$

For IS, it can be shown that the only change to the calculation behind Theorem 3 is to set the diagonal terms in (20) to 0 since independent selection renders report transmission independent under the assumption of no report loss. This yields asymptotic variance

$$c^{\mathrm{IS}} = q(1+q-2pq)/p. \quad (23)$$

Both variances are inversely proportional to p: fewer samples mean higher variance.

One sees easily that $c^{\mathrm{IS}} \geq c^{\mathrm{TS}}$, with equality only when $p = 1$ or $q = 0$. The expected physical regime is small sampling probability p and small report loss probability $1 - q$. In this regime, the expressions for variance simplify:

$$c^{\mathrm{TS}} \approx (1-q)/p, \quad \text{while} \quad c^{\mathrm{IS}} \approx 2/p. \quad (24)$$

The notable property is that the ratio of the variances of the two estimators is driven by loss rate to be estimated, with $c^{\mathrm{IS}}/c^{\mathrm{TS}} \approx 2/(1-q) = 50$ when estimating a 1% loss. We display the ratio $\rho = c^{\mathrm{IS}}/c^{\mathrm{TS}}$ as function of p and q in Figure 8. Observed the rapid growth of ρ as $1/1-q$ for $q \approx 1$. Dependence on p is mild by comparison.

F. Loss Estimation under Load Balancing

The techniques of this section apply to estimation of loss on a point-to-point path. In practice, load balancing may be employed, giving rise to point-to-multipoint paths. The above technique can be applied provided the problem can be reduced to that of estimation on a set of point-to-point paths. This is possible if trajectory sampling is employed on both inbound and outbound interfaces at nodes in which load balancing takes place.

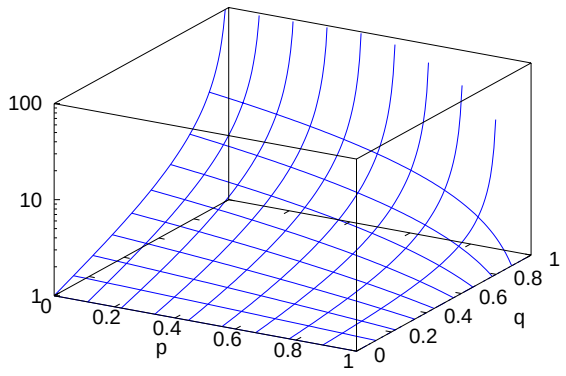

Fig. 8. Surface plot of the ratio $c^{\mathrm{IS}}/c^{\mathrm{TS}}$ of variances of transmission rate estimators for independent and trajectory sampling, as a functions of the sampling probability p and the path transmission rate q under estimation.

VI. Conclusion

In a network measurement system such as the one described in this paper, there exists a tradeoff between the complexity of the measurement devices and the complexity of the central collector. In our previous work, we assumed that measurement devices are capable of reliably exporting measurements to the collector, which simplifies the task of the collector in reconstructing a statistically representative set of trajectory samples. However, there are circumstances where such reliable export is either not desirable or not possible. Therefore, in this paper, we assume that measurement devices export measurement report packets unreliably, which relieves them of the burden of buffering and processing acknowledgments. But dealing with missing reports complicates the task of the collector. In this paper, we propose methods for the collector to deal with such loss.

The first aspect of trajectory reconstruction that is complicated by report loss is duplicate elimination, i.e., the elimination of reports that happened to map to identical labels. We wish to eliminate such duplicates in order to ensure that the final set of trajectories is not polluted by composite trajectories resulting from multiple packets. This may have various undesirable side effects when estimating quantities of interest from these trajectories and monitoring the correct network behavior.

With reliable reporting, a straightforward approach to eliminate duplicate labels is to rely on auxiliary information reported about sampled packets from ingress links, or simply to assume that a given packet should not be observed on ingress links more than once. With unreliable reporting, this approach is not guaranteed to catch all duplicates, because ingress reports may be lost.

We have proposed an approach based on Bloom filters, a data structure that compressed a set membership function into a bit array. Bloom filters are appropriate because the only error they incur are false positives, which may lead to the elimination of some unique labels in addition to actual duplicates. While this represents a small loss of measurement data, the main property of this approach, given in Theorem 1, is that the duplicate elimination essentially behaves like subsampling the original set of packets. Therefore, by applying a correction factor, the resulting set of trajectories can be treated as if it had been obtained in a collision-free way. This insulates the estimation and detection procedures fed by trajectory samples from the intricacies of duplicate elimination.

Once duplicate labels have been eliminated, the resulting report stream can be passed to applications. In general, applications must be adapted to report loss. Path tracing applications must amalgamate reports from several packets in order to reconstruct complete trajectories. Passive loss measurement applications must distinguish report loss from packet loss by exploiting transmission sequence numbers in the reports to estimate report loss rates. The performance analysis of these applications shows that trajectory sampling brings substantial advantages over independent packet sampling, reducing both estimator variance and reporting bandwidth.

Future work includes unification of the current work on report loss with the work of [4] in grouping reports temporally for reconstruction. Some timeout must be applied to packet grouping, both to manage collector memory and to reduce label recurrence. On the other hand, this inevitably sunders some reports from trajectories; an efficient way is required to manage this without discarding stranded reports too aggressively.

References

[1] B. Bloom, “Space/time tradeoffs in hash coding with allowable errors”, CACM, 13(7),422-426, 1970.

[2] A. Broder and M. Mitzenmacher, “Network Applications of Bloom Filters: A Survey”, In *Proc. Allerton Conference*, Monticello, IL, October 2002.

[3] N. G. Duffield and M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. *IEEE/ACM Transactions on Networking*, 9(3):280–292, June 2001.

[4] N. G. Duffield, A. Gerber, and M. Grossglauser. Trajectory Engine: A Backend for Trajectory Sampling. In *Proc. Network Operations and Management Symposium (NOMS)*, Florence, Italy, April 2002.

[5] N.G. Duffield, C. Lund, “Predicting Resource Usage and Estimation Accuracy in an IP Flow Measurement Collection Infrastructure, Preprint, 2003.

[6] “Packet Wingspan Distribution”, NLANR. See `http://www.nlanr.net/NA/Learn/wingspan.html`

[7] P. Phaal, S. Panchen, N. McKee, “InMon Corporation’s sFlow: A Method for Monitoring Traffic in Switched and Routed Networks”, RFC 3176, September 2001

[8] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. ”Practical network support for IP traceback.” In Proc. of ACM SIGCOMM 2000, pages 295-306, Aug.2000.

[9] M.J. Schervish, “Theory of Statistics”, Springer, New York, 1995.

[10] T. Zseby, “Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements”, Proceedings of Passive and Active Measurement Workshop (PAM 2002), Fort Collins, CO, USA, March 25-26, 2002