Cost-effective IP Trace Publishing Using Data Sketch

Li-hua MIAO, Wei DING, Hai-ting ZHU, Qing XIA
College of Computer Science and Technology
Southeast University
Nanjing, Jiangsu, China
{lhmiao, wding, htzhu, qxia}@njnet.edu.cn

Abstract—**IP Traces are sets of IP packets (or packet headers) captured at the measuring point. Their publishing, which is most challenged by massive size concern, is crucial for network research. In this paper, we propose a new scheme for IP Trace publishing which offers much smaller transportation quantity than the traditional methods. Based on Cisco's Netflow technique, the data provider first summarizes an original IP Trace to a sketch. During the summarizing process, extra statistics of certain fields in the original IP Trace are obtained. The sketch and the statistics, which are much smaller in size, are then published instead of the original IP Trace. Based on the Monte Carlo simulation technique, the data downloader can generate a synthetic IP Trace from the sketch and the statistics which preserves most of the statistical properties of the original IP Trace. According to our experiments, the transportation quantity of our scheme is only 3% of that in the traditional methods and meanwhile privacy is better protected. In the end, the utility of the synthetic IP Trace and that of the original IP Trace are compared using two network performance metrics (throughput and RTT). The result shows that this scheme is feasible.**

Keywords-IP Trace; flow outline; transportation quantity; privacy protection; utility; simulation algorithm

I. INTRODUCTION

*IP Trace*s are sets of IP packets (or packet headers) captured at the measuring point which provide the most authentic records of the local traffic's state. They are data sources for research in network management, network security, generation of simulation parameters, network behavior, QE assessment, etc. Stored IP Traces replayed in the memory could also be used as background traffic for software testing and assessment. Nevertheless, the gather, store and management of IP Traces require many conditions [1], such as powerful hardware support, etc. At present, only ISPs and a few institutions specialized in researching Internet have these conditions. More research teams usually obtain IP Traces by downloading them. This is also the typical technical route for the research work in this area.

The publishing of IP Traces is a significant challenge.

The first challenge is IP Traces' massive size. We gather IP packet headers from the border router of Jiangsu CERNET using flow sampling and the sampling ratio is 25%. Each sampled packet header, which is called a *watcher* record, contains an eight bytes' timestamp and the first forty bytes of the IP packet. Thus, an IP Trace in this paper is a set of watcher records for a given time horizon. According to our experience, the size of a one-hour uncompressed IP Trace is about 35 gigabytes. Assuming the downloading rate is 1 Mbps on average, this IP Trace can be downloaded in about 80 hours.

Second, IP Traces should be anonymized because real network hosts' behavior might be revealed from the fields contained by IP Traces, such as the source and destination IP address fields, the payload field, etc. For IP addresses, multiple anonymization tools have been developed, such as random permutation, prefix-preserving permutation and truncation [2]. The original work suggests anonymizing the source and destination IP address fields using prefix-preserving permutation technique [3]. Nevertheless, some recent work suggests that this strategy may not as secure as expected [4, 5]. An improved scheme is to anonymize IP addresses using truncation technique. Ref. [2] concludes that this scheme offers better privacy protection than the previous one. For the payload field, it should be removed because it contains the communication content. Other fields, such as source and destination port numbers, could be retained, replaced, recalculated or removed.

The third challenge is the utility reduction of published IP Traces. When IP Traces are anonymized, their utility will be reduced no matter what anonymization policy is applied. Generally the better the privacy is protected, the more the utility is reduced.

Faced with these concerns, some researchers suggest sharing IP Traces via secure queries [6, 7]. However, in these systems, the programs are relatively inflexible and the query languages are hard to meet all users' needs. Another suggestion is to upload executable programs to a public system [8], since the size of these programs is negligible in contrast to that of IP Traces. This system, which contains many IP Traces, will run the programs and send the results to its users afterwards. This makes a good solution except that some problems need to be solved first, such as trust mechanism and operating environment consistency.

In this paper, a new scheme is proposed for publishing IP Traces offering better privacy protection and smaller transportation quantity. In traditional methods, an IP Trace is anonymized and then published directly. In this paper, we suggest publishing a sketch and statistics of the IP Trace along with a universal reconstruction algorithm which can generate a new synthetic IP Trace from the sketch and the statistics. In the remainder of this paper, section Ⅱ concretely describes this new scheme. In section Ⅲ, the main difficulties of the reconstruction algorithm are discussed. In sectionⅣ, the transportation quantity and the privacy risk are analyzed quantitatively comparing to that in traditional methods. SectionⅤ compares the utility of a synthetic IP Trace with that of its original IP Trace using two network performance metrics relevant to the reconstruction algorithm. Finally, we conclude the paper in section Ⅵ.

II. SKETCH-BASED COST-EFFECTIVE SCHEME

A. Related Work in Another Area

The problem of how to publish data for scientific research is not new. Over the last several decades, scientists have developed many approaches to publish microdata, which are essential databases of attributes collected about individuals [9].

There are two kinds of ways to anonymize microdata for publishing. They are truncation and perturbation based methods. Besides them, two methods have been proposed which do not publish the microdata directly but instead provide statistics of the microdata in alternate ways. The first kind is synthetic microdata generation which models the original microdata and generates new microdata from the statistical model. As a result, this resultant microdata have no connection to real individuals and preserve the statistical properties of the original data simultaneously. The second kind is to store the microdata on a secure remote server where the microdata can only be accessed through a query interface. The users can get answers to their queries and the query interface ensures that no answers are harmful to privacy.

B. Sketch-based Cost-effective Scheme

In this paper, we use an idea similar to the synthetic microdata generation method. First, an IP Trace is processed; then, its sketch and statistics can be generated which are much smaller in size than the IP Trace. Users can download the sketch and statistics along with a universal algorithm which can generate a synthetic IP Trace from the sketch and statistics. In theory, the synthetic IP Trace preserves all statistical properties of the original IP Trace.

To generate the sketch, the IP Trace is first compressed into flow records using Cisco's Netflow technique. Then, flow records with the same source and destination IP addresses in a given duration are further compressed into a *RRE* record in table Ⅰ ［20］; the ID means the information element number in the IPFIX protocol [11] and the IP addresses are anonymized with truncation technique. The sketch consists of all RRE records generated from the IP Trace.

When generating the sketch, most fields in the watcher records are lost. Hence, statistics are needed besides the sketch for the reconstruction algorithm to reconstruct these fields correctly in synthetic IP Traces. Every lost field's characteristic is unique, hence different statistical metrics should be chosen for different field. In this paper, three fields' reconstruction is mainly discussed: packet length, timestamp and TCP flags. Statistics of these fields are generated when generating the sketch.

Let *Publishing RRE (PRRE)* be the sketch and the statistics. Let *PRRE* Generation Algorithm (*PRREGA*) be the algorithm which compresses an IP Trace into a PRRE. Let *Reconstruction Algorithm (RA)* be the algorithm which generates a synthetic IP Trace from the PRRE. Fig. 1 shows the framework of our scheme.

TABLE I. RRE RECORD FORMAT

ID	Length	Description
8	4 Bytes	RRE Head: Truncated Source IPv4 Address
12	4 Bytes	RRE Head: Truncated Destination IPv4 Address
163	4 Bytes	RRE Head: Flow count
182	2 Bytes	1st Flow: Source Port
183	2 Bytes	1st Flow: Destination Port
1	4 Bytes	1st Flow: Flow Total Bytes
2	3 Bytes	1st Flow: Flow Total Packets
4	2 bits	1st Flow: Protocol Identifier (00:TCP, 01:UDP, 10: ICMP)
240	6 bits	1st Flow: TCP Flags (customize)
150	4 Bytes	1st Flow: Flow Start Seconds
242	11 bits	1st Flow: Flow Duration
239	21 bits	1st Flow:Flight Count (customize)

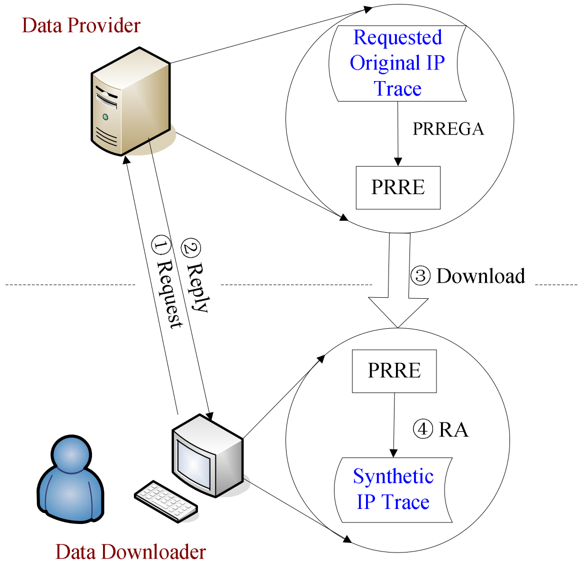


Figure 1. The Framework of the Cost-effective Scheme

III. STATISTICAL CALCULATION AND RECONSTRUCTION OF THERE SPECIAL FIELDS

Figure 1 shows that the RA can generate a synthetic IP Trace from the PRRE where all fields in watcher records are generated from their corresponding RRE records and the statistics. Statistical metrics kept in the statistics, which are calculated in the PRREGA for their corresponding fields, should be selected on two principles: (i) matching the reconstruction methods in the RA and (ii) minimizing the transportation quantity. Thus, the selection of the statistical metrics and the design of the reconstruction methods should be considered together to achieve a balance between privacy and utility. In this paper, we mainly discuss how to correctly reconstruct three fields: packet length, timestamp and TCP flags; the other fields will be discussed in our future work.

A. The Packet Length Field

1) The Statistical Metrics

All possible values of a UDP packet's length are among [28, 1500] and those of a TCP packet's length are among [40, 1500]. Divide [28, 1500] and [40, 1500] into *M* and *N* intervals where their lower and upper bounds are fixed, i.e. these intervals remain the same for every PRRE.

For every interval, let *Nsi* be the total number of TCP (or UDP) packets in it and *Nt* be the total number of TCP (or UDP) packets. Let *MERi* be *Nsi*/*Nt*. Then, *MERi* (for *i* = 1, …, *M*+*N*) are the statistical metrics we want.

2) For theData Provider

Calculate all the statistical metrics in the PRREGA and keep them in the PRRE.

3) For the Downloader

The downloader can generate the synthetic IP Trace using the RA.

For one watcher record, if it is a special one, reconstruct its packet length field according to its properties. For example, if this is a *syn* packet, its packet length field should be *40*Bytes.

Else, generate a random number *r* among [1, *M*] (or [1, *N*] if this is a TCP packet) first. Then, select an interval according to *r*. If the chosen interval's length > 1, reconstruct the packet length field according to the uniform distribution; else let the packet length field be the interval.

4) How to Divide the Intervals

For several IP Traces we gathered [12], let *TcpTotal* be the total number of TCP packets and *TCP[i]* be the number of TCP packets whose packet length field are *i* (for *i* = 40, …, 1500). Thus, *TcpTotal*=∑*TCP[i]*.

While *TCP[i]/TcpTotal* >= 1%, Make i an interval.

While for a left segment [i_0, i_1], if ∑*TCP[i]/ TcpTotal* > 5%, then [i_0, i_2] makes an interval where ∑*TCP[i]/ TcpTotal* = 5%; Else [i_0, i_1] makes an interval.

As for UDP packets, do the same way.

In this paper, five one-hour IP Traces are chosen, where *TcpTotal* = 1880906491 and *UdpTotal* = 1266240198. According to the principles above, *M* = 22 and *N* = 45. Thus, (22+45)**C* bytes are needed in the PRRE, where *C* <= 2.

B. The Timestamp and TCP Flags fileds

1) The Statistical Metrics

To reconstruct these fields more accurate, we use the *flight* in [13, 14]. All flows' flight counts and their TCP flags (if any) are the statistical metrics we want.

2) For theData Provider

In the PRREGA, keep every flow's flight count in its *RRE* record; if it is a TCP flow, set syn = 1 for the caller if there is any syn packet; set syn = 1 and ack = 1 for the callee if there is any syn and ack packet; set fin = 1 if there is any fin packet.

3) For the Downloader

Let I_1 be the time between two UDP packets in one flight and I_2 be that for TCP packets. In this paper, I_1 = 3200 microseconds, I_2 = 4100 microseconds.

For every flow, let t_1 be the flow duration, t_2 be the average flight duration, n_1 be the flight count, n_2 be the packet count and n_3 be the packet count in a flight.

For a UDP flow or a TCP flow where syn = 0, $t_2 = t_1/n_1$, $n_3 = n_2/n_1$. The generating method is: for packets inside one flight, timestamp is generated according to I_1 and I_2; for every first packet in adjacent flights, generate timestamp according to t_2.

For a TCP flow where syn = 1 and ack = 1, $n_3 = (n_2\text{-}1) / (n_1\text{-}1)$. The generating method is the same as above except there is only one packet (syn = 1 and ack = 1) in the first flight.

For a TCP flow where syn = 1 and ack = 0, $n_3 = (n_2\text{-}2) / (n_1\text{-}2)$. The generating method is the same as above except there is only one packet (syn = 1) and one (ack = 1) in the first and second flight respectively.

For a TCP flow, set the last packet's fin = 1 when fin = 1.

IV. QUANTIFICATION OF TRANSPORTATION QUANTITY AND RISK

In this section, the transportation quantity and the privacy risk are analyzed quantitatively. The analyses show that the scheme in this paper offers better privacy protection and smaller transportation quantity than traditional methods.

A. Transportatin Quantity Quantification

We compare the transportation quantity of our scheme with that of the traditional methods from two aspects where we publish PRRE and the RA while the traditional methods publish anonymized IP Trace directly.

On one hand, first cut a one-hour IP Trace into six IP Trace segments where they all begin at 14:00 and last for 5, 10, 15, 20, 25 and 30 minutes respectively. Then compress these IP Trace segments into six PRREs. The size of the IP Trace segments and that of their PRREs are showed in table Ⅱ.

TABLE II. PRRE VS. IP TRACE SEMENT

Duration (minutes)	IP Trace Segment (Megabytes)	PRRE (Megabytes)	PRRE / IP Trace Segment
5	2891	84	2.91%
10	5421	159	2.93%
15	8674	240	2.77%
20	11565	321	2.78%
25	13734	402	2.93%
30	16625	480	2.89%

On the other hand, first chose five 20-minute IP Traces. Then, compress them into PRREs. Take one IP Trace and its PRRE as an example. Let *m[i]* be the watcher record count, *n[i]* be the flow count, *k[i]* be the *RRE* record count, *R[i]* be the ratio of the size of the PRRE to that of the IP Trace. Let *a* be the length of a watcher record, *b* be the length of flow in a *RRE* record, *c* be the length of a *RRE* head, *d* be the length of the statistics in the PRRE. Then $R[i] = (n[i]*b+k[i]*c+d) / (m[i]*a)$. According to the above sections, $a = 48$ bytes, $b = 20$ bytes, $c = 12$ bytes and $d = 134$ bytes. According to our experiments, *R[i]* is 3.03% on average.

Hence, the transportation quantity of our scheme is only 3% of that in traditional methods.

B. Risk Quantification

The anonymization policy of our scheme is to anonymize IP addresses with truncation, remain the protocol type and ports fields, remove the TTL and checksum fields, reconstruct the timestamp and packet length fields; while in the traditional methods, the anonymization policy is the same except for remaining the timestamp and packet length fields.

Ten fields in watcher records are classified into three categories: high sensitive fields (source and destination IP addresses), middle sensitive fields (source and destination ports, checksum) and low sensitive fields (timestamp, protocol type, TTL and packet length). Then, these fields are given different weights according to their importance to privacy protection showed in table Ⅲ.

Table Ⅲ shows the Probability of Being Identified (*PBI*) of those fields where the *PBI* of the IP addresses is calculated based on [2] using a one-hour IP Trace we gathered. The *PBI* of the packet length field is obtained according to section Ⅲ using the same one-hour IP Trace.

TABLE III. PBI ANALYSES

Fields	Weight	This Paper	Traditional Methods
		PBI	
Source IP Address	16%	Internal IP Address: 0.00449; External IP Address: 0.0037	
Destination IP Address	16%		
Source Port	11%	1	
Destination Port	11%	1	
Checksum	11%	0	
Timestamp	9%	0	1
Protocol type	8%	1	
TTL	9%	0	
Packet Length	9%	0.61	1

Thus, the risk of our scheme can be compared with that of the traditional methods in table Ⅳ.

The weights in table Ⅲ are set according to our experience. However, our scheme always offers a better privacy protection no matter how the weight is set.

TABLE IV. RISK QUANTIFICATION

	Risk
Traditional Methods	0.16*0.00819+0.11*2+0.09+0.08+0.09*1= 0.48131
This Paper	0.16*0.00819+0.11*2+0.08+0.09*0.61= 0.35621

V. UTILITY QUANTIFICATION

Utility is very sensitive to the applications that the synthetic IP Traces are used for. They can perfectly serve as background traffic. However, some fields in the synthetic IP Traces are different from that of the original IP Traces after all. Hence, further discussion is necessary if the synthetic IP Traces are treated as data sources of certain research. Apparently, analyzing metrics based on fields which are not reconstructed in this paper can never be accomplished correctly.

In this paper, the synthetic IP Traces preserve only part of the statistic properties of the original IP Traces in packet count, byte count, flow count, protocol-level metrics and port-level metrics. The main differences between the synthetics IP Traces and the original IP Traces are three fields (timestamp, packet length and TCP Flags) which are simulated in the synthetic IP Traces.

To verify the utility of the synthetic IP Traces, two network performance metrics which are related to those reconstructed fields are chosen: throughput and RTT. They are calculated respectively for a six-minute IP Trace gathered on 08/03/2010 and its corresponding synthetic IP Trace.

A. Throughput

Throughput is calculated respectively for TCP and UDP traffic from two aspects: BPS (Bytes Per Second) and PPS (Packets Per Second). The results are shown in Fig. 2 and 3 where the red line stands for the synthetic IP Trace and the blue one stands for the original IP Trace (the same in the following text).

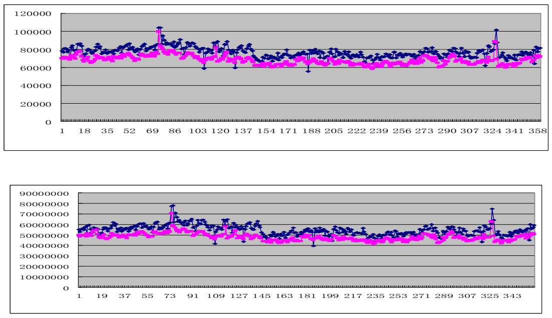
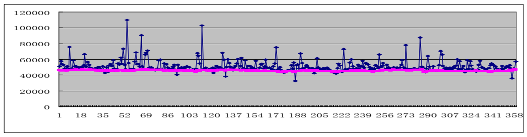


Figure 2. PPS (upper) & BPS (lower) analyses for TCP Traffic



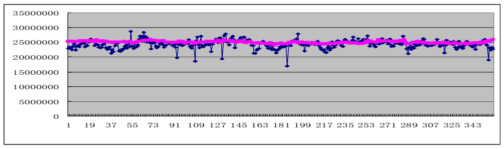


Figure 3. PPS (upper) & BPS (lower) analyses for UDP Traffic

These figures show that the simulation of the TCP traffic is better than that of the UDP traffic whose reason might be that real UDP traffic fluctuates more abruptly.

B. RTT

Usually, RTT is a metric for flow. Thus extra work needs to be done in order to apply it to IP Traces. First, the experiment data in last subsection are cut into five-second segments. In addition, SYN-ACK [15] and PRE [16] algorithms are improved so that an average RTT can be obtained for every segment. Fig. 4 shows the result.

In Fig. 4, the RTT of the synthetic IP Trace is bigger than that of the original IP Trace in a smooth manner. The reasons would be: (i) the timestamp field is reconstructed using average flight duration; (ii) the timestamp and packet length fields are reconstructed in an independent manner.

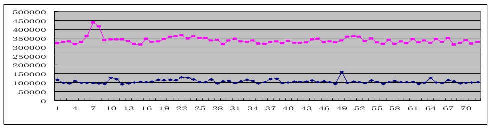


Figure 4. RTT analyses

VI. CONCLUSION AND FUTURE WORK

Since 2005, our lab has started to gather and store IP Traces from the border router of Jiangsu CERNET and done some research on them. To share IP Traces with more researchers, we propose a sketch-based cost-effective scheme of publishing IP Traces. The scheme and its main difficulties are discussed in section Ⅱ and Ⅲ. Section Ⅳ shows that the transportation quantity of our scheme is only 3% of that in the traditional methods and also privacy is better protected. In addition, three fields in watcher records (timestamp, packet length and TCP Flags) are well reconstructed according to analyses based on throughput and RTT metrics in section Ⅴ. The future work would be: (i) improving the reconstruction algorithm, (ii) seeking a more efficient *PRREGA*, (iii) developing the complete system and (iv) reconstructing other fields such as the sequence number and TTL fields.

ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program of China under Grant No. 2009CB320505 and the National Key Technology R&D Program of China under Grant No.2008BAH37B04.

REFERENCES

[1] Mark Alllman, Vern Paxson. Issues and etiquette concerning use of shared measurement data, Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, October 24-26, 2007, San Diego, California, USA

[2] Martin Burkhart, Daniela Brauckhoff, Martin May, Elisa Boschi. The risk-utility tradeoff for IP address truncation, Proceedings of the 1st ACM workshop on Network data anonymization, October 31-31, 2008, Alexandria, Virginia, USA

[3] Bing Shi, Wei Ding, Ya-dong Gao, Jian Gong. IP Trace data based on a CERNET backbone, Journal on Communication, 2006, 27(11A).

[4] S. Coull, C. Wright, F. Monrose, M. Collins, and M. Reiter. Playing devil's advocate: inferring sensitive information from anonymized network traces. In Proceedings of the Network and Distributed System Security Symposium, Feb. 2007.

[5] D. Koukis, S. Antonatos, and K. G. Anagnostakis. On the privacy risks of publishing anonymized IP network traces. In Communications and Multimedia Security, volume 4237 of Lecture Notes in Computer Science, pages 22–32.

[6] J C Mogul, M Arlitt. SC2D: An alternative to trace anonymization. In Proceedings of the SIGCOMM 2006 Workshop on Mining Network Data, 2006.

[7] Jelena Mirkovic, Privacy-safe network trace sharing via secure queries, Proceedings of the 1st ACM workshop on Network data anonymization, October 31-31, 2008, Alexandria, Virginia, USA

[8] ZHU Hai-Ting, DING Wei, XIA Zhen, CAO Xu, CHEN Xing-Qi. IP TASCM: An open system for network measurement analysis, CERNET 2009, pp228-231, Tianjin, China, Dec 2009.

[9] Coull SE, Monrose F, Reiter MK, Bailey M. The challenges of effectively anonymizing network data, Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security, IEEE Computer Society, Washington, DC, USA, pp 230–236

[10] The IPFIX protocol in RFC5101, RFC5102 and RFC5470. http://www.ietf.org

[11] The IPTAS project. http://iptas.edu.cn

[12] Srinivas Shakkottai, R. Srikant, Nevil Brownlee, Andre Broido, kc claffy. The RTT distribution of TCP flows in the Internet and its impact on TCP-based flow control. http://www.caida.org

[13] Ming-zhong Zhou. Study of large-scale network IP flows behavior characteristics and measurement algorithms: doctor paper. Nanjing: Southeast University, 2006

[14] Hao Jiang, Constantinos Dovrolis. Passive estimation of TCP round-trip times. ACM SIGCOMM Computer Communication Review 32(3), 75–88 (2002)

[15] Yibo Zhang, Zhenming Lei. A passive RTT estimate algorithm for TCP. Journal of Beijing University of Posts and Telecommunications. Oct 2004, Vol27, No.5.

[16] Ruoming Pang, Mark Allman, Vern Paxson, Jason Lee, The devil and packet trace anonymization, ACM SIGCOMM Computer Communication Review, v.36 n.1, January 2006

[17] A. Parate, G. Miklau. A framework for safely publishing communication traces. Umass Computer Science Technical Report 2009-040, 2009.

[18] Joel Sommers , Paul Barford. Selfconfiguring network traffic generation, IMC'04, October 25–27, 2004, Taormina, Sicily, Italy.

[19] Kashi Venkatesh Vishwanath, Amin Vahdat. Realistic and responsive network traffic generation, SIGCOMM'06, September 11–15, 2006, Pisa, Italy.