TCP Flow Identification by Sequence and Acknowledgement Number

Peng Yanbing Gong Jian Ding Wei

(Department of Computer Science and Engineering, Southeast University, Nanjing, 210096, China)

Abstract: To reduce the TCP flow processing cost, some bit pattern selected from TCP/IP packet could be used as TCP flow identification. Based on the entropy and randomness analysis of the distribution of sequence number and acknowledgement number in the first packet of a TCP flow, this paper proposes a new uniformed TCP flow identification method (FIDSAN) to the heavy tailed IP or TCP traffic. The experiment results suggest that some bits in TCP Sequence Number (SN) and Acknowledgment Number (AN) can be selected out as flow ID with acceptable confliction probability. The bit length of flow ID selected under given confliction probability can be conducted from an equation deduced from observing window and flow ID range. FIDSAN has low computation cost in the comparison with the traditional methods , such as 5-tuple, CRC, and Checksum etc.

Keywords: flow labelling, flow ID, observing window, TCP, IP

基于顺序号和确认号的 TCP 流标识

彭艳兵 龚俭 丁伟

（东南大学计算机科学与工程系，南京 210096）

摘要：为了降低 TCP 流的处理开销，可以从 TCP/IP 报文中选取某些位串来作为流的标识。本文从位熵和随机性的角度分析了 TCP 流首报文的顺序号和确认号的分布，提出了一种从重尾的 IP 或 TCP 流里获得随机均匀的流标识的新方法（FIDSAN）。实验结果表明，在可以接受的冲突概率下，TCP 流首报文的顺序号（SN）和确认号（AN）的部分高位比特可以用来作为流标签。给定冲突概率时,这种流标识的比特长度可以根据一个由观察窗口和流 ID 值域导出的关系式求出。与 TCP 五元组、CRC、Checksum 等比较发现，FIDSAN 具有更低的计算开销。

关键词：流标识，流 ID，观察窗口，TCP，IP

中图分类号 TP393

Distinguishing TCP flow is a very basic network transmission mechanism in routers, e.g. used for congestion control. Generally, the 5-tuple of Source IP address, Destination IP address, and Protocol fields in IP header, and Source Port, Destination Port fields in TCP/IP header is used to label the different TCP flows. For example, Sarvotham et al. [1] introduced the concepts of “alpha flow” and “beta flow” based on the 5-tuple. Unless a specific flow label is defined for the purpose [5,6], a TCP flow must be identified by a 5-tuple. But in high-speed network, it might be burdensome to use 5-tuple to identify TCP flow because of the numbers of concurrent flow. Therefore, a number of transmutations which map the 96-bits 5-tuple into a shorter flow label, e.g. smaller than 32bits, have been defined and used[2]. These transmutations decreased the memory cost, but with higher calculation cost. Furthermore, for the fractal distributed flow rate in the IP address space [3], homogenization of the transmutation mapping should be considered to avoid a heavy conflict probability with some mapping methods.

Received 2005-03-29

The paper is supported by National Basic Research Program of China (2003CB314804)

Biographies: Peng Yanbing (corresponding author) , male, (1975--), doctor candidate, ybpeng@njnet.edu.cn; Gong Jian, male, professor; Ding Wei, female, professor,

TCP Sequence Number is generated by Initial Sequence Number Generator (ISNG) defined in RFC 793[4]. ISNG is designed as a 32bits clock which plus 1 per 4 microseconds and overflowed every 4.55 hours. "The initial send sequence number (ISS) is chosen by the data sending TCP, and the initial receive sequence number (IRS) is learned during the connection establishing procedure"[4]. According to the rule in RFC793, it is obviously that the Sequence Number (SN for abbreviation) and Acknowledgement Number (AN for abbreviation) is random distributed. Because of the independence of TCP connections, the SN and AN in the first acknowledgement packet of a TCP flow is independence, too. Although SN and AN increase as the data exchanged between the end hosts according to the moving window defined in TCP header, some high order bits within them remain the same, for most of the TCP flows do not last so long, and these bits could be candidates of TCP flow label different from the derivation from TCP 5-tuple.

Section 1 of this paper describes the constraint between the flow ID and Observing Window. Section 2 validates the feasibility of this new TCP labeling method. Section 3 compares FIDSAN with the traditional 5-tuple method and its transmutation such as CRC32 and Checksum in their advantages and disadvantages. Section 4 proposes some potential applications of FIDSAN in router and web flow balance. Section 5 summaries some conclusions.

1. The relationship between Observing window and Flow ID Range

The following terms will be used in hereinafter discussion.

Through the data in network streams inexhaustibly, the study on traffic can only be carried out within the resource limitation, that is, only a portion of the traffic can be observed or processed at any given time. This portion of traffic is called an Observing Window, which composes of packets belonging to each current flow.

First packet of TCP flow(briefly, FPTF) is the packet in the TCP flow whose SYN and ACK Code Bits set to 1 at the same time, which should be the first acknowledgement packet from the receiving part. This special packet contains the start points of sequence number for the both sides of the TCP connection.

Range of Flow ID is the number of possible values expressed by a character string when it is taken as a flow ID to identify distinguishable TCP flows.

Obviously, the range of flow ID is critical to FIDSAN method. It cannot be too large because it will take too many bits from SN and AN field and that will make a longer TCP connection have different flow ID. It can neither be too small because it will use too few bits within SN and AN field that make the long continual TCP connections have identical IDs. A suitable selection should bring about an acceptable confliction probability of flow IDs. The other factors that affect this probability are the flow ID distribution of arrived flows and flow observing window.

Firstly, let us look at the relationship between flow ID and observing window to find a suitable range of flow ID. Suppose that the flow IDs obey to random distribution and be independent with each other, let K be the range of flow ID, and w be the size of the observing window.

Theorem 1: The probability for finding the given flow ID within an observing window is $1-(1-1/K)^w$.

Theorem 1 is tenable under the assumption above and can be deduced by statistics theory. A flow ID appears at the fixed location of observing window with a probability of $1/K$. Because generation of flow ID is independent, the probability of a given flow ID NOT existing in the observing window is $(1-1/K)^w$. So the probability to find a given ID in a w-sized observing window is 1 minus the probability of the given ID NOT appearing independently in the observing window, i.e. $1-(1-1/K)^w$. When $w<<K$, this formulation can be simplified as w/K.

For example, the 5-tuple is used to descript a certain TCP flow with 96bits. So K is $2^{96}=7.9\times10^{28}$, here. For a window expressed by 40bits character string, the size w is 1.1×10^{12}, it is large enough for today's device, and the conflicting probability in this window for a given flow ID is 10^{-16}. So the 5-tuple is a certainly uniqueness expression for any TCP flows.

The window size has an upper limit, generally. The acceptable conflicting probability of flow ID can be decided by the applications. Then we can calculate the bit length of flow ID by following equations:

$$bit\_length(K) = \lceil log_2(K) \rceil = \lceil log_2(1/(1-logw(1-p))) \rceil = \lceil -log_2(1-log_w(1-p)) \rceil \quad (1)$$

When $w<<K$, another equation can be used as a simplifying:

$$bit\_length(K) = \lceil log_2(K) \rceil = \lceil log_2(w/p) \rceil = \lceil log_2 w - log_2 p \rceil \quad (2)$$

According to RFC 793, the SN and AN in First acknowledgement packet of a TCP flow is generated homogeneous to the time. Generally speaking, every TCP connections build randomly and independently, so the assumption above is valid for FIDSAN. Therefore, if both the Observing Window and the acceptable flow ID conflicting probability are given, the shortest label length of FIDSAN can be calculated immediately according to Theorem 1 and Equation (2).

For the packet window, the length of TCP flow is bigger than 3 packets resulted from the 3-way shaking, so the de facto TCP flow number is smaller than the packet window. That is to say, the packet window will have a less conflicting probability of Flow ID descript in Theorem 1.

Theorem 1 can also be applied to IPv6 network. There is a 24bits Flow Label field defined in IPv6 packet head. RFC 1809 suggests that the Flow Label be a pseudo-random number between 0 and 0xFFFFFF and be randomness when combined with the source address. But it is a tentative field that an implementation could ignore it [5,6]. RFC 3697 suggests that "The Flow Label value set by the source MUST be delivered unchanged to the destination node(s)." and "To enable Flow Label based classification, source nodes SHOULD assign each unrelated transport connection and application data stream to a new flow." Such prerequisites implied by RFC 1809 and 3697 are consistent with the assumption above, i.e. Flow Label is independently generated and random distributed. IPv6 is not widely deployed until now, for this reason, the validation of this conclusion will be reserve to the future day when IPv6 is widely deployed.

2. The choice of Flow ID and the randomness of SN/AN

It seems obviously that the high order bits of SN/AN field should be taken as the flow ID because they are comparatively more stable. However, Cheng G. *et al.* [7] found that the higher randomness of a field in packet header can minimize the confliction among its values when deployed to identify flows. It suggests that the randomness of TCP fields should be studied either, to seek the possibility to reduce the confliction probability further.

The concept of bit entropy is used for the randomness analysis, which is calculated by Equation 3:

$$H(\cdot) = -p\log_2 p - (1-p)\log_2(1-p) \quad (3)$$

Where p is the probability a given bit gains 1 or 0. It is calculated from the rate between the counting of this bit when it is 1 and the total counting while dealing with the total samples of FPTF.

The analysis was based on real traffic sampling in CERNET(China Education and Research Network) backbone. The sample capacity is 119,170,048 FPTFs.

Fig. 1 shows Bit Entropy of SN and AN fields in FPTFs. It can be found that the bit entropy of the high order bits located in SN is very close to 1; the bit entropy of the low order bits locations is smaller, but still higher than 0.98. For AN field, the highest bit gained the lowest bit entropy of 0.92, which is distinctly different from those other bits whose entropies

are greater than 0.98, means that the highest bit in AN is not random enough. Fig. 1 expresses that SN has better randomness than AN field, and the highest bit in AN field should be ignored for FIDSAN selection.

To verify the finding above, consider the highest 10bits in SN and AN fields of those samples as a number smaller than 2^{10}, and count each numbers' hits, and the hit rate (Frequency) of each value can be calculated by divided the sum of samples from this value's hits. Fig. 2 discovers the hit rate of those samples in Fig. 1.

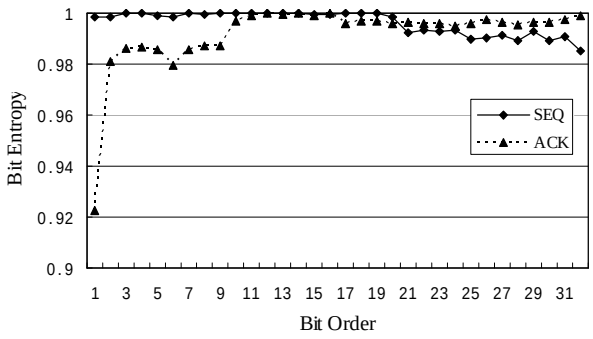

Fig 1. Bit Entropy of SN and AN of FPTF

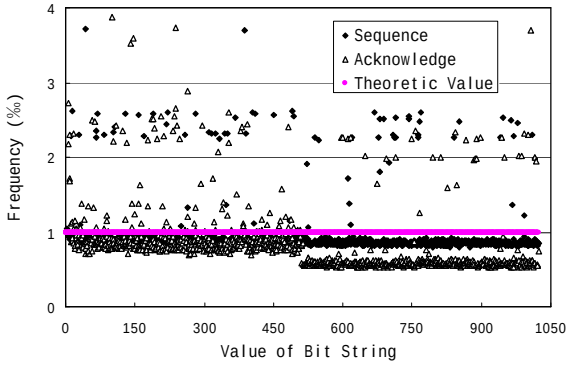

Fig. 2 Frequency analysis of highest 10bits in SN and AN

In Fig 2, the hit rate distribution of those samples is very flat, especially for SN. It is very uniform and very close to the theoretic value *1/K*, here is 0.0009765625. For AN, it got a stage-like curve, and the hit rates of the lower values are very close to the theoretical value. The hit rate curve of greater values is very flat, too, though it is smaller than the low order bins. The critical point is just at the border of the greater values and the lower values, which shows that the highest bit in AN is not very random, so it is the substantial evidence to the conclusion of Fig. 1.

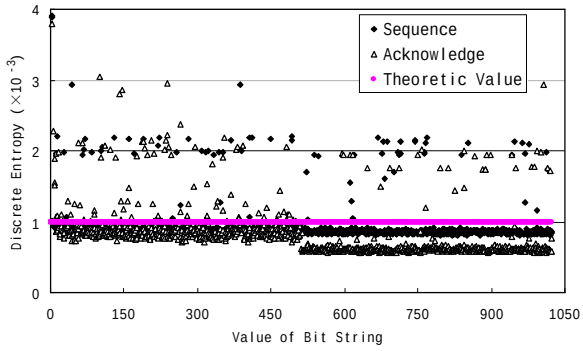

Fig. 3 Discrete Entropy of highest 10bits of SN and AN

Table 1 Discrete Entropy of Highest 10 bits/16bits of FPTF

ID Length	10bits	16bits
SN	0.98724	0.963759
AN	0.91761	0.905687
CRC32	0.98627	0.963688
Checksum	0.98082	0.963688

The discrete entropy is calculated by the Equation 4[8], which is close to 1 when every discrete item is strictly random:

$$H(\cdot) = -\sum p_i \log_K p_i \qquad (4)$$

Where p_i is the hit rate in Fig. 2 and K is the same as which in the previous assumption. $p_i \log_K p_i$ is the Discrete Entropy of each value i, H(·) is the Discrete Entropy of the ID's bit string. The Discrete Entropy is 1 of strictly uniform distributed random ID. So the approximating degree between the really discrete entropy of ID string and 1 can be used as a rule to measure the randomness without considering the effect of ID length.

The Discrete Entropy Analysis of highest 10bits of SN and AN from FPTFs is presented in Fig. 3, and the result is the same as that from Fig. 2.

Use the Equation (4) over the dataset in Fig. 2, Fig. 4 and Fig. 5, we can obtain the results in Table 1. Table 1 shows that the Discrete Entropy of SN is very close to the theoretical value '1' which means high randomness of these 10 bits; the randomness of AN field is lower which confirms the conclusion. The 16bits entropy is smaller than 10bits for that the sample capacity for each ID value in 10 bits ID is larger than which in 16bits ID.

These experiments suggest that the SN field's highest 20 bits and AN field's 20bits except the highest 1bit of First Acknowledgement packet of TCP flow can be deployed as its flow ID. This 10bits ID limits the flow length less than 2^{22} Byte, i.e. 4MBytes. This makes a natural shortcoming to this new method, that is, the bit number of flow ID selected from SN/AN limits the length of identified flows. The bit length of TCP flow ID and bit length when the distinguishable TCP flow length expressed as a bit string, which sum up to 32 bits. It can be expressed by the following equation:

$$log_2 Flow\_length + log_2 K = 32 \quad (5)$$

Where *Flow_length* is the byte length of the labeled TCP flow, *K* is *ID_Range*.

The longer TCP flow length shorts the valid bits for flow ID, and then causes higher conflicting probability of shorter ID length. The bit length of ID shares 32 bits here with the binary length of distinguishable flow length. An approach is to employ the high entropy bits in AN field to obtain both longer ID length and longer flow length. The bit number of flow ID share 64bits here with the bit number when distinguishable flow length is expressed as a bit string. It can be expressed by the following equation:

$$log_2 Flow\_length1 + log_2 Flow\_length2 + log_2 K = 64 \quad (6)$$

Where *Flow_length1* is the byte length of the TCP flow labeled by SN, *Flow_length2* is the length of the TCP flow labeled by AN, *K* is *ID_Range*.

As the result of the bits sharing, high order bits are reasonably selected from SN and AN as the TCP flow ID in the next experiments. The Observing time was March 18, 2004The parameter of Observation Window was chosen from 64 to 1024, and total 1,179,450 FPTFs were gathered for Table 2, which fit the requirement of Large Number Theorem. The confliction probability of flow ID in the given observing window was presented in Table 2. The conflicted probabilities increase as a response to the observation window size, and it is very close to the theoretical value.

Table 2 Conflicting Probability of 10-bit flow ID

Observation window size	Confliction probability	Theoretical value
64	0.061131	0.06062
128	0.11811	0.11756
256	0.22541	0.22129
512	0.4017	0.39362
1024	0.6388	0.63230

Table 3 Conflicting Probability of a 16-bit flow ID

Window size	Confliction probability	Theoretical value	Simplified value w/K
64	0.0005664	0.0009761	0.0009765
128	0.0010932	0.0019512	0.0019531
256	0.0019432	0.0038987	0.0039063
512	0.004101	0.0077821	0.0078125
1024	0.009165	0.0155037	0.015625

Table 3 lists the 16bits ID length with selected from SN and AN in various observing windows. The 16bits ID is composed by the highest order 8bits in SN and the high order 8bits in AN (from the 2nd bit to the 9th bit in Fig. 1). Observing time was April 17, 2004, sample capacity of 29,554,155. The result in this table is better than which in Table 2.

From the Comparison of Table 2 and Table 3, it suggests that if the *K* is very larger than *w*, the effect will be more suitable for applications. The Table 2 and Table 3 validate the Theorem 1. It can be predicted that if a 32 bits FIDSAN will work well in an observing window contained 65536 pieces of flows, and it is accurate enough for most applications.

Table 4 The comparison between the FIDSAN and traditional HASH

	Traditional 5-tuple	CRC32 and Checksum	FIDSAN
ID length	96bit	32/16	<32bit
Operations when generating	5 times of location and copy	More than 100 times/More than 7 times	2 times of location and copy, shift once
Operations cost	comparing 4 times	Comparing one time	Comparing one time
Advantage	Without conflicting	Lower memory overhead	Lower calculating times, lower memory overhead
Disadvantage	Higher memory overhead, Higher calculating times	Higher calculating times Given conflicting probability, work well with small observing window	Given conflicting probability, work well with small observing window
Tuple involved	5-tuple	5-tuple	Transport protocol, SN and/or AN

3. The comparison among FIDSAN and traditional Hash algorithms

Let's compare the advantage and disadvantage among FIDSAN and 5-tuple, CRC32 and Checksum in several sides. From table 4, it is obviously that FIDSAN have some advantage than traditional 5-tuple and their HASH. Fig. 4 and Fig. 5 discover the high order 10 bits' hit rate of CRC32 and Checksum over 5-tuple. The flow sample capacity was 29,554,155, which started at 2004-04-17. The high order 10bits/16bits Discrete Entropy of CRC32 and Checksum was listed in the Table 1. It implies that FIDSAN owns better randomicity and better performance than the CRC32 and Checksum operation when they are selected to form a TCP flow ID. Highest 16bits frequency figures of FIDSAN, CRC32 and Checksum are ignored here for they similar appearance to the 10bits ones.

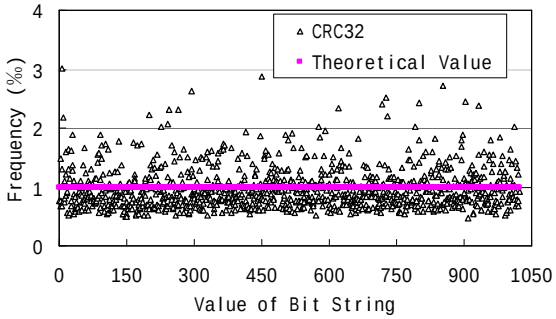

Fig. 4 The homogeneity of CRC32 highest 10bits over the 5-tuple

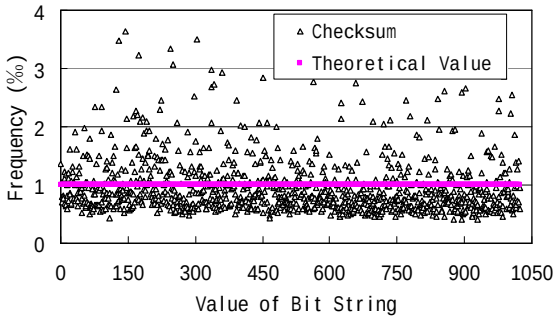

Fig. 5 The homogeneity of Checksum highest 10bits over the 5-tuple

4. Conclusion

In this research, it proposed a new type of method based on SN and AN of the TCP fields to label TCP flows. A theoretical model is also built for the designing, the selecting and applying of flow ID based on the relationship between the Range of flow ID and Observing Window. If the Range of ID is *K*, the window size is *w*, the ID is randomness and independent enough, the ID conflicting probability is w/K when $w<<K$. The conclusion was validated by experiments.

FIDSAN has lower operations and better random than the traditional HASH algorithms such as CRC32 and Checksum in IPv4. The flow ID bits shares the 32/64bits in SN/SN+AN with the bit expression of TCP flow length in FIDSAN.

The validation of IPv6 Flow Label is not provided in this paper and left as a future work. Further investigation will be implemented in the application of FIDSAN. In those resource restricted system such as router, the 5-tuple has great memory overhead; The CRC32 and Checksum has greater operations and less random than FIDSAN. In order to labeling quantity packets with lower operation and lower memory resource, the suitable method is FIDSAN.

Another potential application field of FIDSAN is the session-based web flow balance for the websites with huge burst access in short time, such as the homepages of Olympics, which balance the hosts' session handling capacity by their clusters. It will provide us stable balance performance as well as we are surprised by the evenness of the curve in this paper. It can also be employed in high-speed backbone router to control the congestion and QoS.

References:

[1] Shriram S., Rudolf R., Richard B. Connection-level Analysis and Modeling of Network Traffic [A]. *ACM SIGCOMM Internet Measurement Workshop*[C]. November 2001 San Francisco, USA.

[2] Rajahalme J., Conta A., RFC 3697, Carpenter B. et al., IPv6 Flow Label Specification [S], March 2004

[3] Partridge C., RFC 1809, Using the Flow Label Field in IPv6 [S] , June 1995

[4] Cao Z., Wang Z., Zegura E., Performance of Hasing-Based Schemes for Internet Load Balancing[A], *In Proceedings of IEEE Infocom*[C], March 2000, Tel Aviv, Israel. 332-341

[5] Kohler E., Liy J., Paxson V., Shenker S., Observed Structure of Addresses in IP Traffic[A], *Internet Measurement Workshop*[C]. November 2002. Marseille, France

[6] Postel J., RFC 793, Transmission Control Protocol[S], Sep-01-1981

[7] 程光, 龚俭, 丁伟, 基于分组标识的网络流量抽样测量模型[J], 电子学报 Vol.30, No.12A, 2002:83-89

Cheng G., Gong. J., Ding W., Network Traffic Sampling Measurement Model on Packet Identification[J]. *Chinese Journal of Electronics*, Vol.30, No.12A, 2002:83-89

[8] 朱雪龙, 应用信息论基础, 清华大学出版社 2001.3, p16.

Zhu X., *Fundamentals of applied information theory*[M]. (in Chinese) Publisher of Tsin ghua University, 2001.3, p16