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 patterns selected from the TCP/IP packet can be used as TCP flow identification. Based on the entropy and randomness analysis of the distribution of sequence number (SN) and acknowledgement number (AN) in the first packet of a TCP flow, this paper proposes a new uniform TCP flow identification by sequence and acknowledgement number (FIDSAN) to the heavy-tailed IP or TCP traffic. The experimental results suggest that some bits in the TCP sequence number and acknowledgment number 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 the 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.

Key words: flow labelling; flow ID; observing window; TCP; IP

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, protocol field in IP header, source port, and destination port fields in the TCP 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[2, 3], a TCP flow must be identified by a 5-tuple. But in a high-speed network, it might be burdensome to use the 5-tuple to identify TCP flow because of the large number of concurrent flows. Therefore, a number of transmutations which map the 96-bit 5-tuple into a shorter flow label, e. g. smaller than 32 bits, have been defined and applied[4]. These transmutations decreased the memory cost, but it enhanced the calculation cost. Furthermore, for the fractal distributed flow rate in the IP address space[5], homogenization of the transmutation mapping should be considered to avoid a heavy conflict probability with some mapping methods.

The TCP sequence number is generated by an initial sequence number generator (ISNG) defined in RFC793[6]. ISNG is designed as a 32-bit clock which pluses 1 per 4 ms and overflows every 4. 55 h. "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"[6]. According to the rule in RFC793, it is obvious that the sequence number (SN) and acknowledgement number (AN) is randomly distributed. Because of the independence of TCP connections, the SN and AN in the first acknowledgement packet of a TCP flow is also independent. Although SN and AN increase as the data is exchanged between the end hosts according to the moving window defined in the TCP header, some high order bits within them remain the same, since most of the TCP flows do not last so long, and these bits can be candidates of TCP flow labels different from the derivation from the TCP 5-tuple.

1 Relationship between Observing Window and Flow ID Range

Though 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 a given time. This portion of traffic is called an observing window, which comprises of packets belonging to each current flow.

The first packet of TCP flow (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 the sequence number for both sides of the TCP connection.

The 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 TCP flows.

Obviously, the range of flow ID is critical to the FIDSAN method. It cannot be so large that two longer

Received 2005-03-29.

Foundation item: The National Basic Research Program of China (973 Program) (No. 2003CB314804).

Biographies: Peng Yanbing (1975—), male, gradute; Gong Jian (corresponding author), male, doctor, professor, jgong@ njnet. edu. cn.

TCP connections may have different flow IDs. Neither can it be so small that it will cause the confliction probability that 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 the 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 are independent of 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 the statistics theory. A flow ID appears at the fixed location of the 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 of finding 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 $w \ll K$, this formulation can be simplified as w/K.

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

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

$$\begin{aligned}\text{bit\_length}(K) &= \lceil \log_2(K) \rceil = \\ &\lceil \log_2(1/(1-\log w(1-p))) \rceil = \\ &\lceil -\log_2(1-\log_w(1-p)) \rceil \qquad (1)\end{aligned}$$

When $w \ll K$, another equation can be used as a simplification:

$$\begin{aligned}\text{bit\_length}(K) &= \lceil \log_2(K) \rceil = \lceil \log_2(w/p) \rceil = \\ &\lceil \log_2 w - \log_2 p \rceil \qquad (2)\end{aligned}$$

According to RFC793, the SN and AN in the first acknowledgement packet of a TCP flow are generated homogeneously to the system clock. Generally speaking, every TCP connection builds randomly and independently, so the above assumption 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 Eq. (2).

For the packet window, the length of TCP flow is bigger than 3 packets resulting 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 an IPv6 network. There is a 24-bit flow label field defined in the IPv6 packet head. RFC1809 suggests that the flow label be a pseudo-random number between 0 and 0xFFFFFF and be random when combined with the source address. But it is a tentative field that an implementation can ignore[2, 3]. RFC3697 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 RFC1809 and RFC3697 are consistent with the assumption above; i. e., the flow label is independently generated and randomly distributed. IPv6 has not been widely deployed until now. For this reason, the validation of this conclusion will be reserved until a future day when the IPv6 is widely deployed.

2 Choice of Flow ID and Randomness of SN/AN

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

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

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

where p is the probability of a given bit, it can be 1 or 0. It is calculated from the rate between the counting of this bit when it is 1 and the counting of the total samples of FPTF.

The analysis was based on real traffic sampling in the 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. This means that the highest bit in AN is not random enough. Fig. 1 expresses the idea that SN has better performance of random distribution than AN field does, and the highest bit in AN field should be ignored in FIDSAN selection.

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

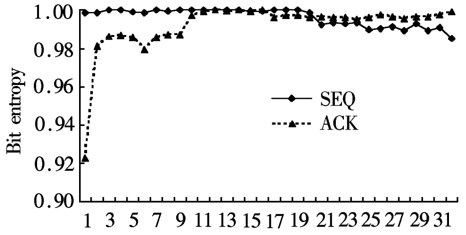

Fig. 1 Bit entropy of SN and AN of FPTF

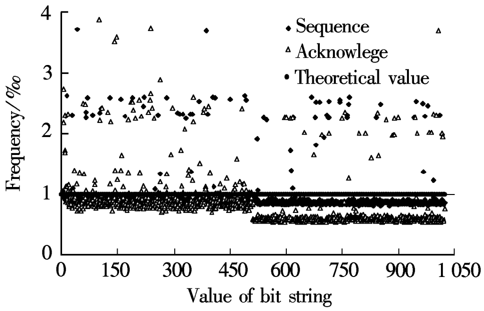

Fig. 2 Frequency analysis of the highest 10 bits in SN and AN

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 theoretical value $1/K$, which here is $9.765\,625 \times 10^{-4}$. 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 also very flat, 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 of the conclusion of Fig. 1.

The discrete entropy is calculated by Eq. (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 the one in the previous assumption. And $p_i \log_K p_i$ is the discrete entropy of each value i, and $H(\cdot)$ is the discrete entropy of the ID' s bit string. The discrete entropy is 1 of strictly uniform distributed random IDs. So the approximating degree between the really discrete entropy of the 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 the highest 10 bits of SN and AN from FPTFs is presented in Fig. 3, and the result is the same as that of Fig. 2.

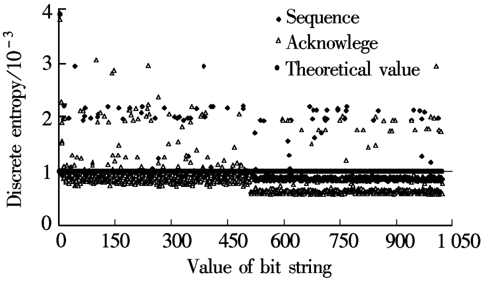

Fig. 3 Discrete entropy of the highest 10 bits of SN and AN

Using Eq. (4) over the dataset in Fig. 2, Fig. 4 and Fig. 5, we can obtain the results in Tab. 1. Tab. 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 the AN field is lower which confirms this conclusion. The 16-bit entropy is smaller than the 10-bit's since the sample capacity for each ID value in a 10-bit ID is larger than in a 16-bit ID.

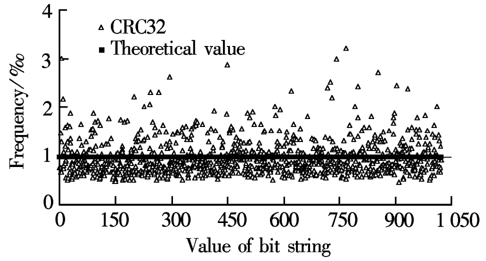

Fig. 4 Homogeneity of CRC32 highest 10 bits over the 5-tuple

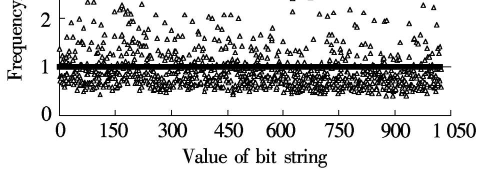

Fig. 5 Homogeneity of Checksum highest 10 bits over the 5-tuple

Tab. 1 Discrete entropy of the highest 10/16 bits of FPTF

ID length	10 bits	16 bits
SN	0. 987 24	0. 963 759
AN	0. 917 61	0. 905 687
CRC32	0. 986 27	0. 963 688
Checksum	0. 980 82	0. 963 688

The results of these experiments suggest that the SN field's highest 20 bits and AN field's 20 bits excluding the highest 1 bit of the first acknowledgement packet of TCP flow can be deployed as its flow ID. This 10-bit ID limits the flow length less than 2^{22} bytes, i. e., 4 Mbytes. This indicates a natural shortcoming of this original method; that is, the bit number of flow ID selected from SN/AN restricts the length of identified flows. The bit length of the TCP flow ID and the bit length sums up to 32 bits when the distinguishable TCP flow length is expressed as a bit string. It can be expressed by the following equation:

$$\log_2 \mathrm{flow\_length} + \log_2 K = 32 \qquad (5)$$

where flow_length is the byte-length of the labeled TCP flow, and K is ID_Range.

The longer the TCP flow length is, the higher the conflicting probability it will cause. 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 the AN field to obtain both longer ID length and longer flow length. The bit number of flow ID shares 64 bits here with the bit number when distinguishable flow length is expressed as a bit string. It can be expressed by the following equation:

$$\log_2 \mathrm{flow\_length1} + \log_2 \mathrm{flow\_length2} + \log_2 K = 64 \qquad (6)$$

where flow_length1 is the byte length of the TCP flow labeled by SN, and flow_length2 is the length of the TCP flow labeled by AN.

As a 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, 2004. The parameter of the observation window was chosen from 64 to 1 024, and a total of 1 179 450 FPTFs were gathered in Tab. 2, which fits the requirement of large number theorem. The confliction probability of flow ID in the given observing window was presented in Tab. 2. The conflicted probabilities increase as a response to the observation window size, and it is very close to the theoretical value.

Tab. 2 Conflicting probability of 10-bit flow ID %

Observation window size	Confliction probability	Theoretical value
64	6. 113 1	6. 062
128	11. 811 0	11. 756
256	22. 541 0	22. 129
512	40. 170 0	39. 362
1 024	63. 880 0	63. 230

Tab. 3 lists the 16-bit ID length selected from SN and AN in various observing windows. The 16-bit ID is composed of the highest order 8 bits in SN and the high order 8 bits in AN (from the 2nd bit to the 9th bit in Fig. 1). Observing time was April 17, 2004, with a sample capacity of 29 554 155. The result in this table is better than that in Tab. 2. From the comparison of Tab. 2 and Tab. 3, it suggests that if K is much larger than w, the effect will be more suitable for applications. Tab. 2 and Tab. 3 validate theorem 1. It can be predicted that if a 32-bit FIDSAN will work well in an observing window containing 65 536 pieces of flows, then it is accurate enough for most of applications.

Tab. 3 Conflicting probability of 16-bit flow ID %

Window size	Confliction probability	Theoretical value	Simplified value w/K
64	0. 056 64	0. 097 61	0. 097 65
128	0. 109 32	0. 195 12	0. 195 31
256	0. 194 32	0. 389 87	0. 390 63
512	0. 410 10	0. 778 21	0. 781 25
1 024	0. 916 50	1. 550 37	1. 562 50

3 Comparison among FIDSAN and Traditional Hash Algorithms

Let us compare the advantages and disadvantages among FIDSAN and 5-tuple, CRC32 and Checksum from different points of view. From Tab. 4, it is obvious that FIDSAN has some advantages over the traditional 5-tuple and their HASH. Fig. 4 and Fig. 5 show

Tab. 4 Comparison between the FIDSAN and traditional HASH

Comparison item	Traditional 5-tuple	CRC32 and Checksum	FIDSAN
ID length/bit	96	32/16	<32
Operations when generating	5 times of location and copy	More than 100/7 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

the high order 10 bits' hit rate of CRC32 and Checksum over the 5-tuple. The flow sample capacity was 29 554 155, which started on 2004-04-17. The high order 10/16 bits show entropy of CRC32 and Checksum was listed in Tab. 1. It implies that FIDSAN owns better randomicity, so it has better performance than the CRC32 and Checksum operation when they are selected to form a TCP flow ID. The highest 16-bit frequency figures of FIDSAN, CRC32 and Checksum are ignored here for their similar appearance to the 10-bit ones.

4 Conclusion

In this research, a novel method is proposed based on SN and AN of the TCP fields to label TCP flows. A theoretical model is also built for the design, the selection and the application of flow ID based on the relationship between the range of flow ID and the observing window. If the range of ID is K, the window size is w, the ID is random and independent enough, then ID conflicting probability is w/K when $w \ll K$. The conclusion was validated by experiments.

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

The validation of the IPv6 flow label is not provided in this paper and is left as a future work. Further investigation will be conducted in the application of FIDSAN. In those resource restricted system such as router, the 5-tuple have great memory overhead; the CRC32 and Checksum has greater operations and less randomness than FIDSAN. In order to label quantity packets with lower operation and lower memory resource, the more effective 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 with stable balance performance as well. We are surprised by the evenness of the curve in this paper. It can also be employed in high-speed backbone routers to control the congestion and QoS.

References

[1] Sarvotham S, Riedi R, Baraniuk R. Connection-level analysis and modeling of network traffic [A]. In: *ACM SIGCOMM Internet Measurement Workshop* [C]. New York: ACM Press, 2001. 99 - 103.

[2] Kohler E, Li J Y, Paxson V, et al. Observed structure of addresses in IP traffic [A]. In: *Internet Measurement Workshop*[C]. New York: ACM Press, 2002. 253 - 266.

[3] Cao Z, Wang Z, Zegura E. Performance of hasing-based schemes for Internet load balancing [A]. In: *Proceedings of IEEE Infocom* [C]. Tel Aviv, Israel, 2000. 332 - 341.

[4] Rajahalme J, Conta A, Carpenter B, et al. RFC3697 IPv6 flow label specification [S]. Internet Society, 2004.

[5] Partridge C. RFC1809 Using the flow label field in IPv6 [S]. Internet Society, 1995.

[6] Postel J. RFC793 Transmission control protocol [S]. Internet Society, 1981.

[7] Cheng G, Gong J, Ding W. Network traffic sampling measurement model on packet identification [J]. *Chinese Journal of Electronics*, 2002, **30**(12A): 83 - 89. (in Chinese)

[8] Zhu X L. *Fundamentals of applied information theory* [M]. Beijing: Tsinghua University Press, 2001. 16. (in Chinese)

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. 从位熵和随机性的角度分析了 TCP 流首报文的顺序号（SN）和确认号（AN）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 流首报文的顺序号和确认号的部分高位比特可以用来作为流标签. 给定冲突概率时，该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