Sync-Codes Identification Based on Self-Similarity of Network

LiZheng Gao[1, a], JunYong Luo[2,b] ,Yan Liu[3,c] ,MeiJuan Yin[4,d] ,Kai Luo[5,e]

[1]State Key Laboratory of Mathematical Engineering and Advanced Computing

Zhengzhou, Henan, China

[a]nan1440429064@sina.com

Keywords: Bit-stream; Sync-codes; Self-similarity; Variance-feature

Abstract. How to correctly segment frames from bit-stream of unknown link-layer protocol is an important issue in the field of information security. Sync-codes identification is the essence of bit-stream segmentation, while the key of sync-codes identification is mining features of sync-codes distinguished from non-sync-codes. Inspired by self-similarity of network traffic, we infer that the bit-stream is also self-similar and infer the variance-feature of sync-codes based on the self-similarity of bit-stream. The sync-codes identification algorithm is designed in order to verify the correctness of sync-codes' variance-feature and the results of experiment prove the correctness of the method.

Introduction

Existing studies of frame synchronization are generally conducted based on the known frame synchronization method. Researchers can only observe mess bit-stream from data of unknown link-layer protocol which frame synchronization method is unknown and frame length is unfixed. It' s an important issue in the field of information security that how to accurately divide frames from bit-stream of unknown link-layer protocol without any prior knowledge.

Due to the lack of prior knowledge and a variety of synchronization methods, it is hard to present a suitable method for all unknown link-layer protocol. Bai et al.[1] calculate the partial TCF by operating the synchronization codes, then distinguish information codes and sync-codes by the similar peak feature and obtain the code length and the first bit of synchronization code. Lu et al.[2] model the frame synchronization and mine the relationship between the probability of detecting the sub synchronous sequence and the data length in the word frequency statistics with different word widths. Wang et al.[3] believe that sync-codes are with frequency attribute and use frequency statistic and association rules in the data mining study to identify sync-codes.

Discovering features of sync-codes is the key of bit-stream segmentation since sync-codes marks the begin and end position of a frame. Frames will be cut in the middle if sync-codes and non-sync-codes are mistaken. Inspired by the self-similarity of network traffic, we suppose the bit-stream data is also self-similar. Then we infer the variance-feature of sync-codes based on the self-similarity of bit-stream. In order to verify the correctness of sync-codes' variance-feature and self-similarity of bit-stream, we design a recognition algorithm of sync-codes. The result of the experiment proves the correctness of our method.

Problem Statement

In this article we study how to segment link-layer protocols bit-stream. Bit-stream refers to bit-sequence which contains multiple frames in one direction of the link-layer communication sent by one part of the communication. The length of a bit-stream is the sum length of all frames it contains. Each bit of bit-sequence can only take value either 0 or 1 (e.g.,"01111110"). In addition, frame and bit-stream are bit-sequences. There are 2^n possible values of a bit-sequence which length is n.

Each bit of bit-sequence can only take value either 0 or 1, we denote the alphabet as

$$B = \{0,1\} \tag{1}$$

The bit-sequence set over B, denoted by S, can be expressed as

$$S = \{s_k = \overline{b_0 b_1 ... b_l .. b_{n_k}} \mid b_l \in B, l = 0,1,2,...,n_k\} \quad (2)$$

The frame set over S , denoted by M , can be expressed as

$$M = \{m_k =< s_0, s_1, ... s_l ... s_{L_k} > \mid s_l \in S, l = 0,1,2,...,L_k\} \quad (3)$$

The bit-sequence frequency set, denoted by F , can be expressed as

$$F = \{ f_{s_k} \mid s_k \in S \wedge k = 0,1,2...\} \quad (4)$$

We denoted f as the sum of bit-sequence frequencies. The distribution of bit-sequence frequencies , denoted by D, can be expressed as

$$D = \{P(s = s_k) \approx \frac{f_{s_k}}{f} \mid f_{s_k} \in F \wedge s_k \in S\} \quad (5)$$

The Variance-feature of Sync-codes

Scientists have found that network traffic is statistical self-similar on a different time scale[5]. Such as campus network, the size of network traffic and the proportion of various protocols is similar at the same time every day. Researchers apply this finding to the prediction of some properties of network traffic[6][7]. In this paper, we argue the same link-layer protocol's actual probability distribution of the frame length is also self-similar, therefore we make the following inference accordingly

(1)Suppose $flow_1$, $flow_2$ and $flow_3$ as three bit-streams. Their length can be expressed as $len(flow_1)$, $len(flow_2)$ and $len(flow_3)$ respectively. The relationship between them can be expressed as an equation $len(flow_1) = len(flow_2) = len(flow_3)$.

(2)Given the actual probability distribution of frame length as φ_1, φ_2 and φ_3 respectively, and the relationship between the similarity of the probability distribution can be expressed as $sim(\varphi_1, \varphi_2) \approx 1 \wedge sim(\varphi_2, \varphi_3) \approx 1 \wedge sim(\varphi_1, \varphi_3) \approx 1$.

(3) The number of frames contained in bit-streams $flow_1$, $flow_2$ and $flow_3$ can be denoted by $num(flow_1)$ $num(flow_2)$ and $num(flow_3)$ respectively. Since only the length of bit-stream and the actual probability distribution of frame length have an effect on the number of frames, we infer $num(flow_1) \approx num(flow_2) \approx num(flow_3)$ by (1) and (2) .

(4) Suppose s_0 to be sync-codes. Its frequencies in $flow_1$, $flow_2$ and $flow_3$ can be expressed as f_{1s_0}, f_{2s_0} and f_{3s_0} . Since $f_{is_0} = k * num(flow_i)$ and (3), we can infer that $f_{1s_0} \approx f_{2s_0} \approx f_{3s_0}$. We denote $var(s_0)$ as the variance of the frequencies of s_0 , thus $var(s_0) \approx 0$.

We name this feature as the variance-feature of sync-codes and design an algorithm to validate its correctness. Experimental results not only demonstrate the correctness of the variance-feature but also demonstrate that the variance-feature is easy to identify under different conditions.

Sync-codes Identification Algorithm

Sync-codes identification algorithm contains following three steps. Figure 1 shows the overview of this algorithm.

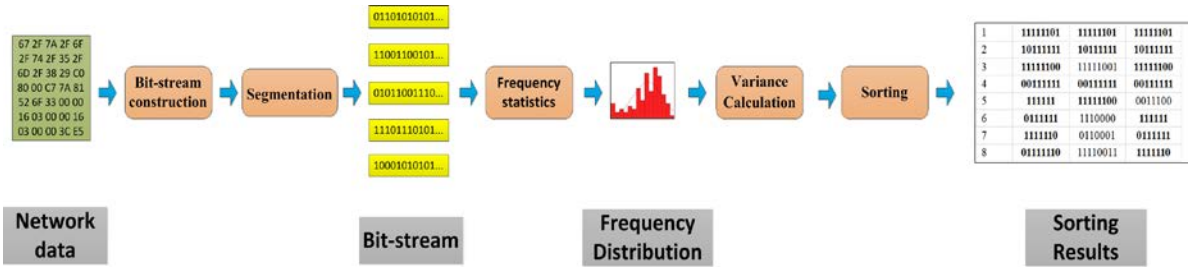

Figure 1: Architecture of sync-codes identification algorithm

Bit-stream Construction and Initial Segmentation

Due to the inconvenient access to link-layer bit-stream data, this paper selected packets captured by packet analyzer such as Wireshark and Tcpdump as the experimental data. These packets have common characteristics like.

1.The frame structure is not complete and the sync-codes have been removed.

2.Packets are stored in files in bytes.

The process of bit-stream construction and initial segmentation includes three parts. Firstly, we add the corresponding frame sync-codes for each packet to convert it into a complete frame. Then we convert frames to the corresponding string of 0 and 1, that is the bit-stream. At last we divide the bit-stream into several parts with equal length and get statistical frequency of bit-sequences in every part.

Frequency Statistics

There are three problems faced in the frequency statistics of bit-sequences.

1. Due to the lack of prior knowledge, the length and arrangement of the corresponding bit-sequence of sync-codes is uncertain.

2. With the increase of the length of bit-sequence, space and time complexity of the algorithm also increase dramatically.

3. The same bit-sequence may occur overlapping. For example in bit-stream fragment “11111”, the bit-sequence “1111” appears twice overlapping. Our method deals with this problem since sync-codes can not be overlapping.

Frequency statistics is a process of pattern matching. Due to the numerous types of bit-sequences and the single pattern matching is inefficient, we apply multi-pattern matching algorithm[8] to bit-sequence frequency statistics. We have to create a full binary tree to take account of all conditions since length and arrangement of bit-sequences is uncertain.

With the length of bit-sequence increases, space and time complexity of the algorithm will increase dramatically; besides, this article is only aims at verifying the correctness of sync-codes variance-feature in theory, for this reason, this article define a threshold for the depth of the full binary tree as $d_{\max}$ and a threshold for the length of the bit-sequence as $l_{\max}$.

$$d = l_{max} + 1 \qquad (6)$$

Each tree node’ corresponding bit-sequence takes value as follows:

1. The value of the root node is empty;

2. If a node is a left child of its parent node, its corresponding bit-sequence, expressed as s_l, has the relationship that $s_l = s_p + "0"$ with its parent’s corresponding bit-sequence, expressed as s_p.

3. If a node is a right child of its parent node, its corresponding bit-sequence, expressed as s_r, has the relationship that $s_r = s_p + "1"$ with its parent’s corresponding bit-sequence, expressed as s_p.

The frequency statistics phase reads into bit-stream after the full binary tree have been created and we set the beginning of bit-stream as $startoffset$. The algorithm of frequency statistics performs the

following steps recursively:

1.Read into bits from *startoffset* of the bit-stream and point the pointer of the tree to the root node;

2.Read a bit of the bit-stream orderly.If the bit takes value of “0”, the algorithm jumps to step 3; if the bit takes value of “1”, the algorithm jumps to step 4;

3.The corresponding bit-sequence frequency of the root node of the left sub-tree add one. Point the pointer to the left sub-tree’ root node and continue step 5.

4.The corresponding bit-sequence frequency of the root node of the right sub-tree add one. Point the pointer to the right sub-tree’ root node and continue step 5.

5.If the pointer reaches the leaf node of the binary tree , add 1 to *startoffset* and continue step 1; otherwise continue step 2.

In order to eliminate the effect of "overlapping sequence" , this article adds two items for each node of the full binary tree. These two items record the length (expressed as len) of the node’s corresponding bit-sequence s and the position (expressed as len) of the last match to s in the bit-stream.If the bit-sequence s is matched once again, the current position of s, expressed as $curindex$, must have the relation with $preindex$ and len like:

$$curindex - preindex >= len \tag{7}$$

Variance Calculation and Result Sorting

We divide bit-stream into n equal length parts and denote the frequency of the bit-sequence s in the i-th sub bit-stream as f_{is}. The average frequency can be denoted as f_{as} and

$$f_{as} = \frac{1}{n}\sum f_{is} \tag{8}$$

The variance of the frequency of bit-sequence s can be expressed as $var(s)$ and

$$var(s)^2 = \frac{1}{n}\sum (f_{is} - f_{as})^2 \tag{9}$$

We rank the results in ascending order after we have calculated each bit-sequence’ variance of frequency. Because of the variance-feature of the sync-codes, the corresponding result should be kept at the front of the list.

Experiment

Test Data.

In this paper, we simulate unknown link-layer data using bit-stream data constructed by hand. To reduce the complexity of the algorithm, we select a short sync-codes just like the sync-codes of HDLC, that is “01111110” . We fill the rest of the fields in the new frame with a common Ethernet frame data. To avoid sequence “111111” appearing in other fields except the sync-field, the new frame need the bit-stuffing operation.

Evaluation Criteria.

This article computes the variance of frequency of each bit-sequence and rank the results in ascending order. The variance of the sync-codes close to 0 according to the proposed theory of sync-codes’ variance-feature. Therefore, the ideal result is that sync-codes’ corresponding result should be kept at the front of the list.

Since we take the bit-stuffing operation on bit-stream, sync-codes must appear at the position where sequences “??111111??” (? takes value of “0” or “1” or empty) appear. Meanwhile, sync-codes “01111110” is also “??111111??” sequence. According to the theory of self-similarity, the frequencies of the sequences “??111111??” are also self-similar in different bit-streams, so the variance-feature is also the feature of sequences “??111111??” thus their corresponding variance will be kept at the front of the list.

Experimental Results.

The fundamental reason of sync-codes to appear the variance-feature is that the length of the frame follow the same probability distribution in different bit-streams. Factors such as the size of bit-stream, the type of frame payload and stages of communication may affect the actual probability distribution of the frame length, thereby affect the results of the experiment. This paper designs three sets of experiments to analyze the influence of the three factors to the variance-feature of sync-codes and the capability of anti-jamming of the variance-feature.

Different bit-stream size: The size of the three initial bit-streams are 27.2MB, 2.72MB and 272KB respectively. The number of times of bit-stream segmentation is 8. The top 10 results is displayed as shown in Table 1.

Table 1: Experiment results of different bit-stream size

Order	27.2MB	2.72MB	272KB
1	**11111101**	**11111101**	**11111101**
2	**10111111**	**10111111**	**10111111**
3	**11111100**	11111001	**11111100**
4	**00111111**	**00111111**	**00111111**
5	**111111**	**11111100**	0011100
6	**0111111**	1110000	**111111**
7	**1111110**	0110001	**0111111**
8	**01111110**	11110011	**1111110**
9	1110000	0011100	**01111110**
10	0011100	0011011	1100010

When the data size is 27.2MB, “??111111??” sequences occurs 8 times in the top 10 bit-sequences and sync-codes “01111110” ranks in 8th place; when the data size is 2.72MB, the “??111111??” sequences occurs 4 times in the top 10 bit-sequences and sync-code isn't one of them; when the data size is 272KB, the “??111111??” sequences occurs 8 times in the top 10 bit-sequences and sync-codes “01111110” ranks in 9th place.

Different number of times of bit-stream segmentation: This article studies the influence of communication phases by changing the segmentation times of the initial bit-stream. The size of initial bit-stream data is 27.2MB and the number of times of bit-stream segmentation are 4, 8 and 16 respectively. The top 10 results is displayed as shown in Table 2.

Table 2: Experiment results of different number of times of bit-stream segmentation

Order	4 Times	8 Times	16 Times
1	**11111101**	**11111101**	**11111101**
2	**10111111**	**10111111**	**10111111**
3	**11111100**	**11111100**	**11111100**
4	**00111111**	**00111111**	**00111111**
5	0110110	**111111**	1110000
6	0011100	**0111111**	0011100
7	**111111**	**1111110**	0100001
8	**0111111**	**01111110**	0110001
9	**1111110**	1110000	**111111**
10	**01111110**	0011100	**0111111**

When the number of times of bit-stream segmentation is 4, the “??111111??” sequences occurs 8 times in the top 10 bit-sequences and sync-codes “01111110” ranks in 10th place. When the number of times of bit-stream segmentation is 8, the “??111111??” sequences occurs 8 times in the top 10 bit-sequences and sync-codes “01111110” ranks in 8th place. When the number of times of bit-stream segmentation is 16, the “??111111??” sequences occurs 6 times in the top 10 bit-sequences and sync-code isn't one of them.

Different types of frame payload: Ethernet frames’ payloads are usually single IP packets and cannot be used in controlled experiments. Therefore, this article carries the controlled experiments by distinguishing the transport-layer protocol type. We select the common transport-layer protocols TCP and UDP as the payloads and the size of bit-stream is 26.9MB and the number of times of bit-stream segmentation is 8. The top 10 results is displayed as shown in Table 3.

Table 3: Experiment results of different types of frame payload

Order	TCP	UDP
1	**11111101**	**11111101**
2	**10111111**	**00111111**
3	**00111111**	**10111111**
4	**11111100**	**11111100**
5	00110001	**111111**
6	11101010	**0111111**
7	11111001	**1111110**
8	00011100	**01111110**
9	11000101	01110100
10	01001110	10100100

When the payloads' type is TCP, the "??111111??" sequences occurs 4 times in the top 10 bit-sequences and sync-codes "01111110" isn't one of them; when the payloads' type is UDP, the "?111111??" sequences occurs 8 times in the top 10 bit-sequences and sync-codes "01111110" ranks in 8th place.

Above all, the experiment results prove that frame length follows specific probability distribution in the bit-stream and sync-codes' variance-feature is anti-jamming under different conditions.

Summary

How to precisely cut frames from the bit-stream data of unknown link-layer protocol without prior knowledge is an important issue that must be solved in the field of information security. Sync-codes identification is the essence of bit-stream cutting, while the key of sync-codes identification is mining features of sync-codes distinguished from non-sync-codes. We infer that the bit-stream is self-similar inspired by the self-similarity of network traffic. Further, we infer the variance-feature of sync-codes based on the self-similarity of bit-stream. The sync-codes identification algorithm is designed in order to verify the correctness of sync-codes' variance-feature and the results of experiment prove the correctness of the method.

Reference

[1] Bai Yu, Yang Xiaojing, Zhang Yu. A Recognition Method of m-sequence Synchronization Codes Using Higher-order Statistical Processing[J]. Journal of Electronics & Information Technology, 2012, 34(1): 33-37.

[2] Lu Xizai, Su Shaojing, Huang Zhiping. Research on blind detection of the sub synchronous sequence based on word frequency statistic[J]. Journal of Xidian University, 2011, 38(3): 189-196.

[3] Wang Hezhou, Xue Kaiping, Hong Peilin, et al. An unknown link protocol bit stream segmentation algorithm based on frequent statistics and association rules[J]. Journal of University of Science and Technology of China, 2013, 43(007): 554-560.

[4] Bai Yu, Yang Xiaojing, Wang Mao. Recognition Method of Frame Synchronization Codes Based on Relativity Filter and Hadamard Transformation Algorithm[J]. Journal of Detection & Control, 2011, 33(3): 69-72.

[5] Song C, Havlin S, Makse H A. Self-similarity of complex networks[J]. Nature, 2005, 433(7024): 392-395.

[6] Ren Xunyi,Wang Ruchuan,Wang Haiyan. Wavelet analysis method for detection of DDoS attack based on self-similar[J]. Journal on Communications, 2006, 27(5): 6-11.

[7] Dasari R, Renikunta R, Perati M R. Self-Similar Network Traffic Modelling Using Fractal Point Process-Markovian Approach[M]//Fractals, Wavelets, and their Applications. Springer International Publishing, 2014: 413-425.

[8] Yuan Z, Yang B, Ren X, et al. TFD: A multi-pattern matching algorithm for large-scale URL filtering[C]//Computing, Networking and Communications (ICNC), 2013 International Conference on. IEEE, 2013: 359-363.