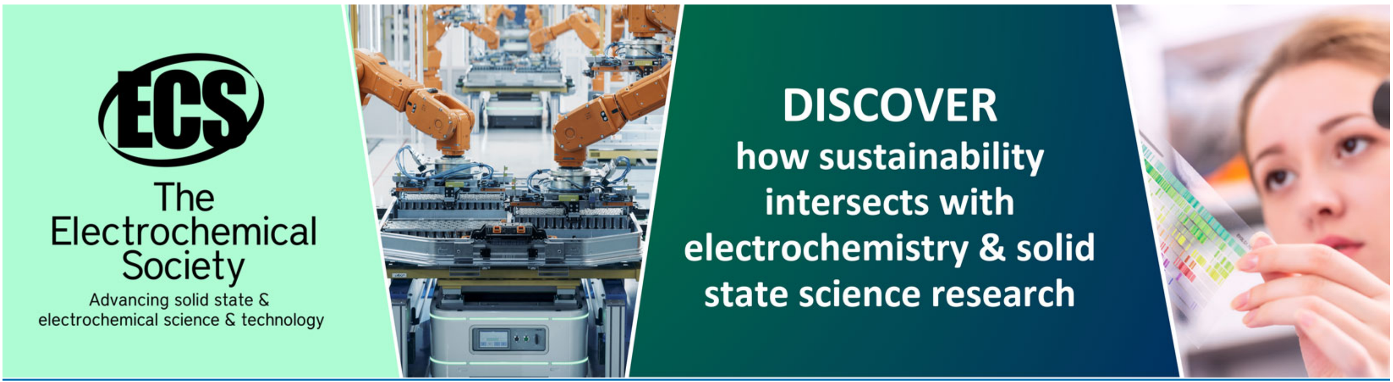
Journal of Physics: Conference Series

PAPER • OPEN ACCESS

The Design and Implementation of RTSP/RTP Multimedia Traffic Identification Algorithm

To cite this article: Liang Jianbing and Chen Shuhui 2019 *J. Phys.: Conf. Ser.* **1168** 052033


ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

The Design and Implementation of RTSP/RTP Multimedia Traffic Identification Algorithm

Liang Jianbing[1], Chen Shuhui[2]

[1]College of Computer Science, National University of Defense Technology, Chang Sha, Hunan Province, 410005, China

[2]College of Computer Science, National University of Defense Technology, Chang Sha, Hunan Province, 410005, China

*Corresponding author's e-mail: lsir_90jb@163.com

Abstract. In recent years, multimedia traffic has been growing rapidly in the network, and it is of great significance to effectively supervise and manage multimedia traffic. This paper not only analyzes some specific multimedia transmission protocols, but also studies their communication characteristics. We proposes a multimedia traffic identification framework as well, which has a great advantage in scalability. In addition, multimedia traffic identification algorithms for specific RTSP, RTP/RTCP traffic, which have high identification accuracy, are described and validated in this paper. More importantly, a solution is put forward, aiming to solve the problem that the start and end of the stream cannot be determined when the multimedia stream adopts UDP transmission.

1. Introduction

With the continuous development of network multimedia technology, the proportion of multimedia traffic in network traffic is increasing significantly. When multimedia data, such as live webcast video, video conferencing, webcam, is transmitted over the network, data compression technology is usually applied to compress the data to reduce the amount of data. Therefore, it is quite difficult to identify multimedia data by means of content feature matching. Thankfully, the study finds that multimedia data adopts specific multimedia transmission protocols when transmitted. The mainstream multimedia transmission protocols include Real Time Streaming Protocol (RTSP), Real Time Transport Protocol (RTP), and so on. The analysis of the characteristics of the multimedia transmission protocols can greatly improve the recognition accuracy.

The Real Time Streaming Protocol (RTSP) defines a method for one-to-many application to transmit multimedia data over IP network, which controls the transmission of real-time data [10]. The RTSP, which uses TCP or UDP for data transfer, is architecturally located on top of RTP and RTCP. RTSP can control multiple transmission connections, and provide a way to select the transmission channel and a method for the RTP-based transmission mechanism. In other words, the RTSP acts as a network remote control for the multimedia server.

Real-Time Transport Protocol (RTP) is a network transport protocol that provides end-to-end delivery services for real-time data such as interactive video/audio or analog data under multicast or unicast network services [9]. The RTP protocol is commonly used in streaming media systems in conjunction with the RTSP protocol. Real-time transport control protocol (RTCP) is a twin protocol of

RTP, and its main function is to provide the feedback for the QoS provided by RTP. Compared to RTP, RTCP occupies very little bandwidth, usually only 5%.

This paper proposes a scalable multimedia traffic identification framework, and implements recognition algorithms for RTSP and RTP traffic. The identification result, called Identification-Result, consists of four types of multimedia data streams, which are RTSP Stream, Non-RTSP Stream, RTP Stream, and Non-RTP Stream.

2. Related Work

Traffic identification refers to determining the type of network traffic by analyzing traffic. In recent years, research on multimedia traffic identification and classification has gradually become a hot topic in related fields.

To improve the classification accuracy, a traffic classification method based on data stream fingerprint is introduced [7]. But the fingerprints of video and audio are too hard to extract. X Tian [1] proposes an integrated dynamic online traffic classification framework, which is called Data Stream based Traffic Classification(DSTC). This framework is to solve the problem of online identification of dynamic traffic, not the multimedia traffic. Meanwhile, machine learning technology is also applied to traffic identification. By analyzing the time level correlation of traffic flows, J Xue and G Wang [2] design a model of Competitive Artificial Neural Network to classify different traffic flows and modulate them, which can help mark the Ipv6 Flow Label. Wang Z [6] proposes a traffic identification framework based on deep learning. The framework includes automatic feature learning, protocol classification, anomalous protocol detection, and unknown protocol identification. K Oida [8] processes video stream recognition through an unsupervised learning algorithm based on packet arrival rate, variance and decay rate. S Kaoprakhon and V Visoottiviseth [3] have studied the methods of identifying video and audio data based on HTTP transmission. By combining keyword matching technique and statistical behavior profiles, they propose a classification method in which keywords of audio and video traffic are pre-defined. Behavior profiles consist of three attributes, which are the average received packet size, a ratio of number of server-client packets, and the flow duration. J Fan [4] adopts a statistical pattern classification technique to identify multimedia traffic, and proposes an automated self-learning system, called VOVClassifier, which bases on the packet inter-arrival times and the associated packet sizes. W Jiang [5] uses FPGA to realize multimedia traffic classification, which can accelerate the statistical identification of multimedia applications while maintaining certain classification accuracy. The identification method is based on the k-Nearest Neighbors (k-NN) algorithm.

However, the above researches have the following two restrictions. First, it requires a large amount of manual labour and plenty of time to label traffic. Second, it is basically coarse-grained identification, which cannot meet the requirements for monitoring multimedia content.

3. Research Methods

There are two ways to transmit RTP packets by RTSP, which are RTP/AVP/UDP and RTP/AVP/TCP. The default transmission mode is RTP/AVP, which is RTP/AVP/UDP [10]. The client uses the SETUP command to give the options to establish a data transfer connection, and the server determines the connection parameters based on the actual situation.

3.1Analysis of RTP

3.1.1RTP/AVP/UDP Transmission Method

Transport:	Value;	Parameter_1;	...	Parameter_n;

Figure 1. Transport header format

In the case of RTP/AVP/UDP [10], when the RTSP Client submits a SETUP request to the RTSP Server, the Transport header is used to specify the parameter options that the client can use to receive interactive data based on RTP/RTCP. When the RTSP Server responds, the Transport header is also used to confirm the parameters that are ultimately used for data transfer. These communication parameters include multimedia data transmission mode, transmission port, source IP address, destination IP address, etc. The Transport header format is shown in Figure1. Usually, RTCP packets are sent on the first available channel higher than the RTP channel. In the RTP/AVP/UDP transmission mode, RTP and RTCP connections are established separately for transmitting multimedia data and commands, in addition to the RTSP connection. Meanwhile, video and audio respectively execute the SETUP command, so they are two separate streams of data, and have their own RTP and RTCP ports.

3.1.2RTP/AVP/TCP Transmission Method

Sometimes considering the security, a firewall may require server to interleave RTSP methods and stream data [10]. In this circumstances, it needs to add an Interleaved Frame layer on top of the RTP layer to distinguish between the RTP channel and the RTCP channel. The structure of the Interleaved Frame is shown in Figure2.

The Interleaved Frame layer is equivalent to a supplement to the RTSP protocol and is used to interpret the data of the upper layer protocol. Interleaved binary data should only be used when RTSP is hosted over TCP [10]. Stream data (such as RTP packets) is encapsulated by an ASCII dollar sign (0x24 hexadecimal), followed by one byte of channel identifier, followed by two bytes of the length of the encapsulated binary data. The stream data follows immediately afterwards. Each $ block contains only one upper layer protocol data unit, such as an RTP packet. The channel identifier is defined in the Transport header with the interleaved parameter.

dollar sign (0x24)	channel identifier	data length
1Byte	1Byte	2Byte

Figure 2. RTSP Interleaved Frame

3.2Analysis of Protocol Characteristics

In this paper, we use a concept, called Life Cycle of Stream Characteristic, which refers to the number of packets transmitted from the time the recognition result is obtained to the time the feature disappears.

3.2.1Characteristics of RTP Stream

The RTP [9] stream in the RTP/AVP/UDP transmission mode is based on the C/S mode and transmitted in the form of a UDP packet. Multimedia data is sent from the server to the client, so it has obviously one-way characteristics. The RTP fixed header length is 12 bytes. The current version number of the RTP is located at the upper two bits of the first byte and has a fixed value of 2. Each multimedia data stream will establish its own unique RTP session, so the multimedia data types in an identical stream are the same, which means that the values of the PT fields in the RTP header are the same. In addition, the values of synchronization source(SSRC) in the identical RTP stream are also the same. The timestamp and sequence number ensure that the upper layer application can precisely decode the multimedia data, and the two have a certain correspondence. The sequence number is monotonically increasing. The data of one frame may be divided into multiple rtp blocks for transmission. The timestamps of the same frame are equal, but the timestamps of different frames are monotonically increasing. The data with the larger sequence number has a timestamp that is not smaller than the timestamp of the data with a smaller sequence number. Since the UDP packet is unordered, it is necessary to combine the values of the timestamp and sequence number fields together to make a judgment. If the sequence number and the timestamp violate the above rules, it can not be an RTP stream. The RTP header format is shown in Figure3. The formulas summarized, according to the RTP characteristics, are as follows.

$$\text{Payload}[0]\ \&\ 0xC0 = 2; \tag{1}$$

$$\mathrm{Direction(n)} = Direction(n-1); \tag{2}$$

$$\mathrm{PT(n)} = PT(n-1); \tag{3}$$

$$\mathrm{SSRC(n)} = SSRC(n-1); \tag{4}$$

$$\mathrm{TimeStamp(n)} \geq TimeStamp(n-1);\ \&\&\ SequenceNum(n) > SequenceNum(n-1); \tag{5}$$

The above conditions are all indispensable. So, if any of the conditions is not satisfied, the stream can be determined to be a Non-RTP Stream.

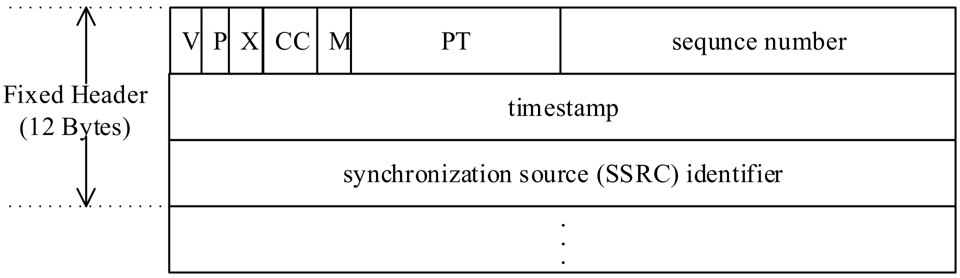


Figure 3. RTP header format

3.2.2Monitoring the identified UDP stream

In order to avoid repeating identification and affecting performance, the identification method in this paper will no longer identify the subsequently arriving packets in the stream that have obtained the recognition result. However, since there is no flag in the UDP stream to indicate the start and end of the stream, it is impossible to know when a UDP stream ends or when a new UDP stream starts, which may result in misidentification. This paper proposes a method that when a UDP stream obtains the Identification-Result, the first condition in Formula (1-5) will be used as a monitoring condition, and the stream will be monitored in two cases. The first case is that the Identification-Result of the stream is Non-RTP Stream and the monitoring condition is met, indicating that a new RTP stream may arrive. The second is that the Identification-Result of the stream is RTP Stream and the monitoring condition is not met, indicating that the stream has ended. Both cases require a new round of identification. In this way, misidentification of UDP streams can be effectively avoided.

3.2.3Characteristics of the RTSP stream

The messages of the RTSP stream during the interaction have fixed characteristics that can be used to identify the RTSP stream. During the interaction, the client sends requests (or commands) to the server. These requests (or commands) are indispensable in the RTSP interaction process, and these requests (or commands) usually contain strings with fixed characteristics. In addition, the response message sent by the server starts with the substring of "RTSP/1.0", which is unique to the RTSP server response message. The characteristics of RTSP are summarized in Table 1.

Table 1. RTSP Characteristics Strings.

Characteristics strings	Meaning
"RTSP/1.0 200 OK"	Response
"OPTIONS rtsp"	OPTIONS Request
"DESCRIBE rtsp"	DESCRIBE Request
"ANNOUNCE rtsp"	ANNOUNCE Request
"SETUP rtsp"	SETUP Request
"PLAY rtsp"	PLAY Request
"PAUSE rtsp"	PAUSE Request
"TEARDOWN rtsp"	TEARDOWN Request
"GET_PARAMETER rtsp"	GET_PARAMETER Request
"SET_PARAMETER rtsp"	SET_PARAMETER Request

4. Scalable Multimedia Traffic Identification Framework

4.1Description of the Identification Framework

This paper presents a scalable multimedia traffic identification framework, as shown in Figure 4. The framework consists of five modules, namely a global stream table management module, a UDP monitoring table, a UDP monitoring module, a UDP characteristics matching module, and a TCP characteristics matching module. The global stream table management module is responsible for managing the creation and deletion of stream nodes and recording the multimedia type of the stream. The UDP monitoring table, as shown in Figure 5, monitors the UDP stream through information fed back by other streams, which can realize the rapid identification of the UDP stream. The UDP monitoring module uses certain characteristics as the monitoring basis to monitor the stream,which has obtained the Identified-Result to determine the Life Cycle of Stream Characteristics. This module is used to avoid misidentification. The UDP characteristics matching module and the TCP characteristics matching module are responsible for performing characteristics match on the TCP stream and the UDP stream, and mapping the matching result into corresponding Identification-Results. The Identification-Result will be fed back to the stream table node. If a TCP stream is one that controls another UDP stream (for instance, an RTSP stream controlls an RTP stream), the TCP characteristics matching module will analyze the relevant information and feed them back into the UDP monitoring table to implement rapid identification of the UDP stream. The characteristics matching module includes the RTSP stream identification algorithm and the RTP stream identification algorithm. This framework has great scalability, so we can monitor different UDP multimedia streams via setting different multimedia stream flags on the UDP monitoring table. At the same time, the framework supports the extension of different identification algorithms to identify more other types of multimedia streams.

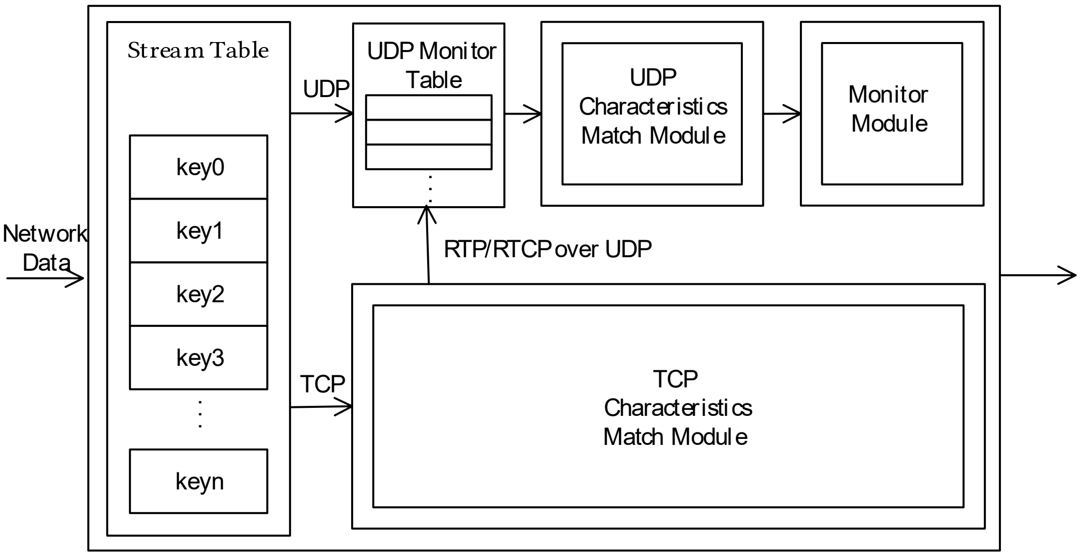


Figure 4. A Scalable Multimedia Traffic Identification Framework

4.2Identification Algorithm

4.2.1RTSP stream identification algorithm

Because RTSP is transmitted based on the TCP, so only TCP streams need to perform RTSP identification.

The quintuple is extracted from the data stream, and hash is calculated. The calculation result is used as the index value to quickly query the stream table. If the stream is in a state that Identification-Result has been obtained, the identification module is skipped directly. Otherwise, the characteristics of the first n messages of the stream will be matched for identification. If none of the n packets match successfully, the stream is determined to be a Non-RTSP stream,otherwise it is determined to be an RTSP stream. The Identification-Result is fed back into the stream table node.

After the RTSP stream is identified, the response message of the RTSP stream will be further analyzed to determine which method the multimedia data will be transmitted by. This paper proposes a method to find the response message with the "Transport" header in all response messages of the RTSP stream and parse the value of the header. If the value of the header contains the substring "/TCP", the multimedia data will be transmitted by RTP/AVP/TCP. Otherwise, the multimedia data will be transmitted by /RTP/AVP/UDP.

When the multimedia data is transmitted in /RTP/AVP/UDP mode, the RTSP stream negotiates the RTP/RTCP transmission parameters through the "Transport" header. Therefore, in this transmission mode, the RTP/RTCP transmission parameter can be obtained by parsing the "Transport" header, and then create a new monitoring node and insert it into the UDP monitoring table to implement monitoring of the subsequent UDP stream.

4.2.2Creating and Querying UDP Monitoring Table Nodes

The node of the UDP monitoring table consists of two parts, which are the quintuple and the Flags. The flags are the types of multimedia of the node. When a new node is inserted, the quintuple will be calculated by the Hash algorithm, and the result will be used as the table index value -- key, and the new node is inserted into the linked list indicated by the key. When querying, first perform a hash calculation on the quintuple to get the table index value -- key, and then perform quintuple matching on the linked list indicated by the key. If the match is successful, the flag of the node is returned, otherwise it returns 0. The Hash algorithm with low collision rate can effectively reduce the length of the linked list, and accelerate the query process in some degree.

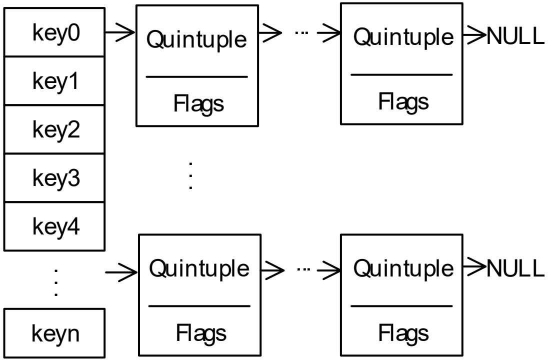


Figure 5. UDP Monitoring Table

4.2.3RTP Stream Identification Algorithm

Step 0. The algorithm will query the UDP monitoring table. If the query is successful, it returns the stream type flag in the monitoring table, and deletes the monitoring node from the monitoring table. And then it goes to step5. If the query fails, the UDP payload length is first determined. If the length is less than 12 bytes, no judgment is made. Otherwise, it goes to step1.

Step 1. The algorithm will check whether the upper two bits of the first byte of the UDP payload is 2 or not. If yes, it goes to step2. Otherwise, it goes to step6.

Step 2. The algorithm will record the value of PT, SSRC, timestamp, sequence number, and packet direction into the stream table for comparison with subsequent packets. And then, it goes to step3.

Step 3. The algorithm will extract the PT, SSRC, timestamp, sequence number and direction of the subsequent packet, and compare them with the saved values. If all the conditions in Formula (1-5) are satisfied, it goes to step4. Otherwise, it goes to step6.

Step 4. The algorithm will compare the following n-1 packets according to the method of step3. If there is a packet that does not satisfy the Formula (1-5), it goes to step6. If all the packets satisfy the Formula (1-5), it goes to step5.

Step 5. The stream can be determined as an RTP stream. And the the algorithm proceeds to step7.

Step 6. The stream can be determined as a Non-RTP stream. And the the algorithm proceeds to step7.

Step 7. The first condition in Formula (1-5) is used as a monitoring condition, and it starts to monitor the UDP stream. If the stream has been determined as a Non-RTP stream and the monitoring condition is satisfied, or it has been determined as an RTP stream but does not satisfy the monitoring condition, a new round of identification is restarted.

5. Experiment and Results

5.1Experiment Data

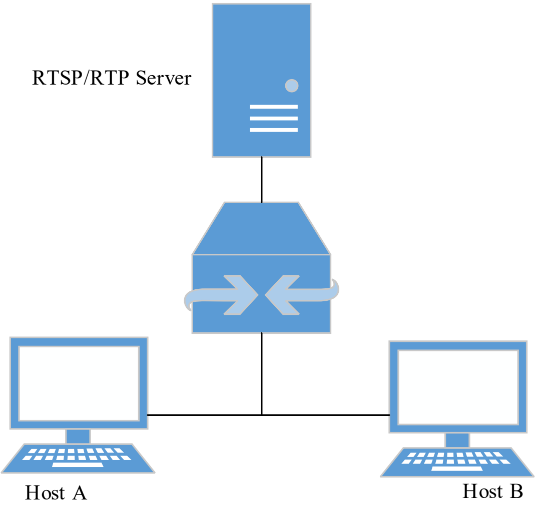


Figure 6. Network topology of the experimental environment

At the end of the paper, the above identification method is verified by experiments. The specific experimental method is as follows. The identification method proposed in this paper is deployed to host B in the experimental network environment. The network topology is shown in Figure 6. The RTSP/RTP server is a multimedia streaming server that provides RTSP/RTP audio and video data. Host A acts as a client and establishes a connection with the RTSP/RTP server. Host B connects to the switch through port replication to capture traffic flowing to host A. The traffic types and quantity statistics are shown in the Table 2.

Table 2. Experimental Stream Statistics.

Type	Number of Packets
RTSP	5832
RTP	569562
RTCP	2991
Others	3443119
Total	4021492
Total streams	26091

5.2Analysis of Results

The experimental results show that the proposed method of multimedia traffic identification can achieve high identification accuracy of RTSP, without leakage identification and misidentification. At the same time, it can accurately identify RTP packets based on RTSP control transmission, without leakage identification and misidentification. For a separate RTP stream, there is a very small amount of leakage identification. This is because when RTP is identified, a certain number of packets are needed to match before the RTP stream can be confirmed. The RTP leakage identification rate is within the controllable range. In addition, the experimental results show that the proposed method can accurately determine the termination of UDP multimedia stream and the beginning of a new multimedia stream. The experimental results are shown in Table 3.

Table 3. Experimental Results.

Type	Number of identified packets	Actual number of packages	Accuracy
RTSP	5832	5832	100%
RTP	569506	569562	99.99%

6. Conclusions

In response to the developing trend of multimedia traffic, this paper proposes a scalable multimedia traffic identification framework. Meanwhile, we design and implement the identification algorithm for mainstream multimedia protocols.These algorithms enable fine-grained identification of multimedia traffic, laying the foundation for in-depth research in the future. Finally, the feasibility and accuracy of the framework are verified by experiments. In the future research, we will further study multimedia traffic identification algorithms and implement multimedia content analysis, making the multimedia traffic identification framework more complete.

References

[1] Tian, X., Sun Q., Huang, X.H., Ma, Y. (2009) A Dynamic Online Traffic Classification Methodology Based on Data Stream Mining. World Congress on Computer Science & Information Engineering. 1: 298-302.

[2] Xue, J.S., Wang, G.X. (2005) A method of classifying multimedia traffic flows based on neural network. International Conference on Wireless Communications. 2: 1249-1252.

[3] Samruay, K., Vasaka, V. (2009) Classification of audio and video traffic over HTTP protocol. In: International Symposiumon on Communication & Information Technology. pp: 1534-1539.

[4] Fan, J.Y., Wu, D.P., Nucci, A., Keralapura, R., Gao, L.X. (2011) Protocol oblivious classification of multimadia traffic. Security & Communication Networks. 4(4): 357-371.

[5] Weirong, J., Maya, G. (2010) Real-Time Classification of Multimedia Traffic Using FPGA. In: International Conference on Field Programmable Logic & Applications. Milano, Italy. pp: 56-53.

[6] Wang, Z.Y. (2015) The Applications of Deep Learning on Traffic Identification. BlackHat USA.

[7] Cheng, K.F., Wei, G.H., Ma, X.J. (2016) Traffic Classification Method Based On Data Stream Fingerprint. In: International Conference on Advanced Materials and Computer Science. Qingdao, China. pp: 735-739.

[8] Oida, K., Nakayama, N. (2013) Video Stream Identification for Traffic Engineering. International Journal of Future Computer and Communication. 2(4): 275-280.

[9] RFC 3550. RTP: A Transport Protocol for Real-Time Applications. http://www.faqs.org/rfcs/rfc3550.html.

[10] RFC 2326. Real Time Streaming Protocol (RTSP). http://www.faqs.org/rfcs/rfc2326.html.

[11] Moore, A.W., Papagiannaki, K. (2005) Toward the Accurate Identification of Network Applications. International Conference on Passive & Active Network Measurement. 3431: 41-54