# RTP Encapsulation for Scalable Video Stream and its Application in NS-2 Simulation

**Zhou Ying, Zhang Jihong, Liu Wei**

**Abstract.** Real-time Transport Protocol (RTP) is a widely used protocol providing end-to-end transportation for various media in the real time application. RTP encapsulation of H.264/SVC video data for RTP / RTCP protocol is described. The sequence number and timestamp in the RTP packet header are the basis for decoding correctly. So the format and encapsulation of them are analyzed. The NS-2 network simulation model is designed to transmit real video stream and verify the RTP encapsulation of H.264/SVC video stream. The simulation shows the correctness and validity of RTP encapsulation program proposed in this paper.

**Keywords:** Scalable Video Coding（SVC）• RTP encapsulation • NS-2 Simulation

## 1 Introduction

With the development and popularization of the Internet, there is a growing demand for real-time access to network multimedia, especially the audio and video information. Therefore, streaming media technology emerges. Traditional TCP / IP protocol is not suitable for real-time transmission due to great delay and jitteriness brought by retransmission mechanism. In order to achieve efficient and real-time streaming transmission, people pay more and more attention on RTP protocol (Real-Time Transport Protocol), which is widely used to provide an end-to-end transmission for real-time multimedia transmission.

SVC (Scalable Video Coding) provides hierarchical structure and spatial, temporal, and quality scalabilities. Therefore it is widely used to provide more

Zhou Ying(✉)
College of Information Engineering, Shen Zhen University
Shenzhen , China
e-mail:zhouying722@163.com

Zhou Ying
Key Laboratory of Visual Media Processing and Transmission, Shenzhen Institute of Information Technology
Shenzhen , China

Zhang jihong
Shenzhen Institute of Information Technology
Shenzhen , China

Liu Wei
Shenzhen Institute of Information Technology
Shenzhen , China

 

scalable video stream to meet the needs of various terminal users. In this paper, we present how to encapsulate H.264/SVC format video to RTP format and transmit it in NS-2 simulation in detail.

## 2 RTP Encapsulation and its Application in NS-2 Simulation

H.264/SVC contains two layers: VCL (Video Coding Layer) and NAL（Net Abstraction Layer）[1].Video encoding and decoding is usually finished in VCL, which includes motion compensation, transform coding, entropy coding and other compression units. NAL provides an independent and unified interface for VCL. Also NAL encapsulates and transmits the video data. Each NAL unit consists of a group of NAL header information corresponding to the encoded video data and a raw byte sequence payload (RBSP), as shown in Fig.1. With RTP / UDP / IP system, the encoded NAL units can be considered as the RTP payload directly.

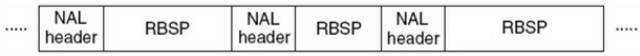


**Fig. 1** NAL unit sequence structure

Each NALU contains the start code with ‘0x00 00 01’ or ‘0x00 00 00 01’. The decoder detects each start code as the NAL initial identification and terminates the current NALU when the next start code is detected. Except ‘0x00 00 01’ or ‘0x00 00 00 01’, the other data is encapsulated into RTP packets.

For H.264/SVC video transmission, retransmission mechanism of TCP / IP protocol will bring latency and jitteriness. It is not conducive to the real-time transmission. So UDP protocol is widely used although it is connectionless and can not provide the quality assurance. While RTP / RTCP based on UDP can provide flow rate control and congestion control. Fig. 2 shows the video transmission framework based on RTP/UDP/IP, in which the multimedia data is firstly encoded, and secondly encapsulated into RTP packet, then loaded with UDP user datagram, finally transmitted into the IP layer.

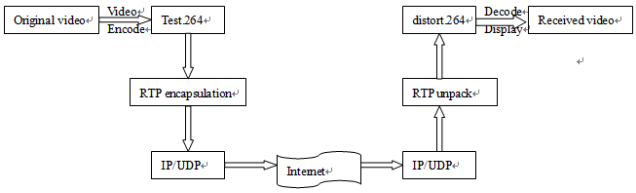


**Fig. 2** H.264/SVC video transmission framework

RTP is a protocol for the multimedia transmission on the Internet, which is defined working in one-to-one or one-to-many transmission case. RTP is used to provide time information and achieve flow synchronization. UDP only transmits data packets, regardless of the time sequence of the packet transmission. RTP together with UDP works as the transportation layer protocol. The data unit of RTP provides timestamp and sequence number for synchronous playback of media. When encapsulating RTP packets, sometimes a package is divided into several ones with the same timestamp.

The typical applications of RTP are built on UDP. RTP can only guarantee real-time data transmission. It can not provide reliable delivery mechanism for tactic transmitted packets or flow control or congestion control. It relies on the RTCP to provide these services.

## *2.1 The Serial Number and Timestamp of the RTP Header*

The RTP packet header structure is shown in Fig. 3. The RTP header is comprised of 12 bytes of fixed fields and 4 bytes of the CSRC selectable fields. CSRC fields appear only in the mixer inserted [2]. Where, the sequence number and timestamp in the RTP packet header are the basis for decoding correctly.

| 1 | 2 | 3 | 8 | 9 | 16bit |
|---|---|---|---|---|---|
| V | P | X | CSRC Count | M | Payload Type |
| Sequence number | | | | Timestamp | |
| SSRC | | | | CSRC (variable 0 - 15 items 32bits each) | |

**Fig. 3** RTP packet header structure

1) Sequence number：16 bits. Sequence number increases by 1 with each RTP packet, which is used for receiver to detect packet loss and to restore packet sequence. For the protection of the encrypted data, the initial value of the sequence number is random set.

When the packet size is larger than 255, the serial number will overflow. At this time, the low bit of serial number is storied in the first byte and the high bit in the second byte.

2) Timestamp：32bit.Timestamp reflects the sampling time of the first byte in the RTP packet, which must be exported from the monotonic and linear increasing clock for synchronization and jitter calculation. The initial value of timestamp should be randomly generated as the serial number. RTP packets belonging to the same video frame should have the same timestamp. The timestamp increases by one increment while the frame number increases by one. If the sampling frequency is 90000Hz, the frame rate is 30fps, then the timestamp increments is 90000/30 = 3000.

The timestamp plays three roles as follows:

1）Recording the local time of the current video packet, which is mainly used for audio and video synchronization, real-time monitoring and so on.

2）Getting the correct broadcast time of the video for the receiver. Decoding deadline is used to determine whether to decode the received packets. If timestamp goes beyond the decoding deadline, the packets with this timestamp are considered arriving late and can not be decoded. So there is no need to decode these packets.

3) The video stream trace file is generated based on the timestamp with the NS-2 network simulation. During the NS-2 video transmission simulation, the corresponding virtual traffic trace file needs be generated according to the real video stream. Then a proxy is used to read and transfer this trace file. The format of trace file is as <packet identify, packet transmission time, packet size>. Wherein, the packet transmission time is acquired according to the timestamp in the RTP file.

## *2.2 RTP Encapsulation*

RTP coding standard (RFC3550) provides three package methods[2]:

1) Single NAL unit mode

A RTP packet only contains a complete NALU. The packet of which NALU length is less than MTU (maximum transmission unit) is encapsulated with a single NAL unit mode. In this case, the NALU header type field of the RTP is the same as the one of original H.264/SVC.

Among the SVC stream, the SEI and parameter set information of each frame is so important that if they are lost, the entire frame data can not be correctly decoded. That is why we encapsulate parameter set information as a separate RTP packet in this paper, which is conducive to the protection and control of important data.

2) Combination Units mode

A RTP packet is composed of multiple NAL units. There are four kinds of combinations: STAP-A, STAP-B, MTAP16 and MTAP24. The types of the corresponding values are 24, 25, 26, and 27.

3) Fragmentation Units mode

The NALU unit whose length exceeds the MTU must be split. There are two kinds of FUs: FU_A and FU_B. The types of the corresponding values are 28 and 29.

When the RTP packet payload is small, the packet loss rate is relatively low, and the transmission process is easy to control, while the transfer rate is relatively low, because the overhead of protocol header is relatively large. With the increase of RTP packet payload, the payload utility ratio increases, accompanied by the higher packet loss rate.

In this paper FU_A mode is selected to split and encapsulate the NALU whose size is greater than 1500 bytes. The RTP payload format of FU_A is shown in Fig. 4. FU_A fragmentation unit is composed of one byte indicator, one byte slice unit head and slice unit load.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| FU indicator  |   FU header   |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                         FU payload                            |
|                                                               |
|                              +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              :...OPTIONAL RTP padding         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

**Fig. 4** RTP payload format of FU_A

In which, the format of the slice unit for FU_A is as shown in Fig. 5:

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+---------------+
```

**Fig. 5** Format of the slice unit for FU_A

FU_A indication type is equal to 28. NRI and F value must be set in accordance with the corresponding value of the split NAL unit. FU_A unit header format is shown in Fig. 6：

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|S|E|R|  Type   |
+---------------+
```

**Fig. 6** FU_A unit header format

Where，S: 1 bit. 1 signals that the start bit indicates the beginning of the NAL fragment unit.

E: 1 bit. 1 signals that the stop bit indicates the end of the NAL fragment unit. The last byte of the load also is one of the NAL fragment unit.

R: 1 bit. Reserved bit must be set to 0, which the receiver must ignore.

Type: 5 bits. They are the same with the NAL unit payload type.

Assuming that among the original encoded SVC/H.264 video, a NALU unit begins with 0x74, and greater than 1500 bytes, such as"0x74 C0……BB 5A……12 C4……EB". It needs to be split. 0x74=01110100, F=0, NRI=11, type=20. The unit is divided into three RTP packets with FU_A mode as: "0x7C 94 C0……BB","0x7C 14 5A……12","0x7C 54 C4……EB".

All the received FU_A fragmented packets at the receiver need to be combined to the original NALU. Eight bits of the restored NALU header are composed of top three bits of FU_A indicator and last five bits of FU_A header, that is nal_unit_type = (FU_A indicator & 0xE0) | (FU_A header & 0x1F).

## 3 RTP Protocol Based NS-2 Video Transmission Simulation

In this paper, the network video transmission simulation platform is established based on the NS-2 network simulation software. Real video stream will be transmitted in the network simulation, so we extend and modify he NS-2 structure, including adding required network elements and agent.

At first the network topology model is established, and the YUV format video source is encoded according to the configuration file, then the compressed data is encapsulated into RTP packet. Secondly, the trace file of the network traffic is produced according to the encoded video stream. The format of trace file is <packet identify, packet transmission time, packet size> [5]. The basic Converting principle is to read the timestamp and the packet size of RTP packet, and store them in the trace file, and then input them into the NS-2 simulation network transmission.

We design simulation process as shown in Fig. 7.

After simulation, we can get the Send file (sender trace file) and Receive file (receiver trace file), which record the serial number of the data packet, the transmission / reception time and size respectively. The missing data packets after the transmission can be obtained by comparing two files.

At receiver all of the received packets are reordered according to the timestamp and the missing packets are discarded. The generated video file can be played, and we can get the intuitive effect diagram of network transmission. The analysis result obtained from such video stream transmission is more reliable. And this

transmission of the video stream in NS-2 simulation is basically the same as one in real network.

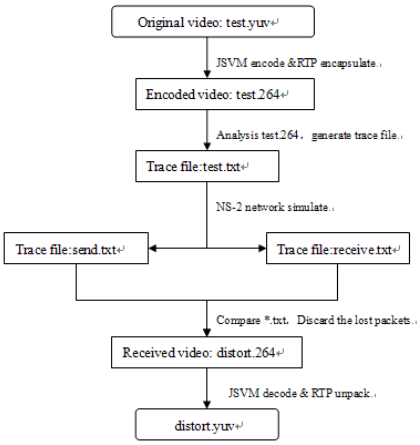


**Fig. 7** Simulation process

## 4 Experiment Results and Analysis

A network topology consisting of eight nodes is conFig.d as shown in Fig. 8. And the characteristics of the link is determined as follows: the network includes three transmission nodes S1, S2 and S3, two routers R1 and R2, and three receiving ends d1, d2, and d3. The bandwidth between s1 and r1 is 10Mbps, while the ones between s2, s3 and r1 are 5Mbps with transmission delay of 1ms. The bandwidth bottleneck between r1 and r2 is 0.3Mbps, with propagation delay of 10ms. All link management mechanism is droptail, and the maximum queue length between r1and r2 is the length of 10 packets. A UDP connection and a video UDP stream are established between s1 and r1. The video trace file is input into the NS-2simulation. A UDP connection and a video CBR stream are established between S2, S3 and r1 respectively as a background flow to disturb the video transmission. The start time of transmission is 0s, and the termination time is that required for 16 frames transmission with 30 frames / sec.

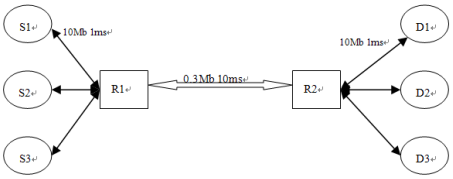


**Fig. 8** Network topology

In this paper, the foreman video sequence is encoded with the frame rate of 30fps and 16 frames. After transmission, the YUVviewer is used to observe the difference between the reconstructed video and the original video stream, as shown in Fig. 9：

(a) Original video (b) Compressed video

(c) Adding one UDP flow (d) Adding two UDP flow

**Fig. 9** Foreman video sequence comparison

Fig. 9 (a) is the original image without compression. Fig. 9 (b) is the reconstructed image at the receiver after compression and transmission through the network. Fig. 9(c) is the reconstructed image after compression and transmission through the network together with a CBR background flow. Fig. 9 (d) is the reconstructed image after compression and transmission through the network together with two CBR background flows.

The objective video quality at receiver through network transmission is shown in Fig. 10.

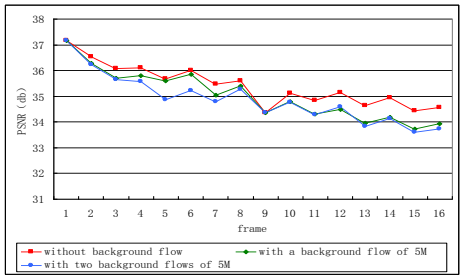


**Fig. 10** PSNR comparison of the received foreman sequence

From Fig. 9 and Fig. 10, we can get the following conclusions: the quality of Fig. 9 (a) is the best, because there is no compression or any lost information. The quality of Fig. 9 (b) is a little poor because during transmission there is some packet loss which leads to some screen unsolvable. The quality of Fig. 9 (c) and Fig. 9 (d) is poor because under the same network structure, the added CBR data streams with video stream through the router transmission will seize the bandwidth resources, so that the packet loss rate in the network transmission increases, and the reconstructed video quality of the receiver is the worst.

The experiments show that the video stream can be reconstructed using the RTP encapsulation strategy and NS-2 simulation model transmission mentioned above. According to the subjective and objective evaluation, the reconstructed video stream is little different from the original video sequence. If there is background noise

added, which affects the transmission of the video stream. The subjective and objective evaluation decline, which are consistent with the reality. It proves the validity and correctness of the RTP encapsulation strategy.

## 5 Conclusions

In this paper RTP encapsulation of H.264/SVC video data for RTP / RTCP protocol is described in detail. In order to transmit real video stream in the NS-2 network simulation, the trace file must be generated according to the RTP packet and the proxy must be added. This article presents a RTP encapsulation program for NS-2 network simulation. The sequence number and timestamp in the RTP packet header and RTP encapsulation method are elaborated. By simulating network video transmission, the correctness and effectiveness of the program proposed in this paper is verified.

**Acknowledgments** This project is supported by the National Natural Science Foundation of China (61172165) and National Natural Science Foundation of Guangdong（S2011010006113）.

## References

1. Kwang-deok Seo, Jin-soo Kim, Soon-heung Jung, and Jeong-ju Yoo.(2010). A Practical RTP Packetinzation Scheme for SVC Video Transport over IP Networks. *ETRI Journal*, 32(2), 281-291.
2. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson.(2003). RTP: a Transport Protocol for Real-Time Applications. IETF/STD 0064, RFC3550.
3. Wenger, S., Ye-Kui Wang, Schierl, T.(2007).Transport and Signaling of SVC in IP Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 17( 9), 1164 – 1173.
4. Renzi, D., Amon, P., Battista, S.(2008). Video Content Adaptation Based on SVC and Associated RTP Packet Loss Detection and Signaling. *Ninth International Workshop on Image Analysis for Multimedia Interactive Services*, 97 – 100.
5. Chih-Heng Ke(2012).myEvalSVC: an Integrated Simulation Framework for Evaluation of H.264/SVC Transmission. *KSII Transactions on Internet and Information Systems* ,6(1),379-393.
6. Detti, A., Bianchi, G., Pisa, C., Proto, F.S., Loreti, P., Kellerer, W., Thakolsri, S., Widmer, J.(2009). SVEF: an open-source experimental evaluation framework for H.264 scalable video streaming. *IEEE Symposium on Computers and Communications*, 36 – 41.