Synchronous playback technology of airborne network video based on RTP

Ying An*, Zhao Hao, and GuoWang Zhang
Chinese Flight Test Establishment, Xi’an, Shaanxi, 710089, China
*Corresponding author’s e-mail: 201721013@stumail.nwu.edu.cn

Abstract. In the flight test, the airborne video data is an important part of the flight test data. In order to solve the problem of synchronous playback of multiple channels of airborne network video data, a synchronous playback scheme of multiple channels of airborne network video data based on the RTP/RTCP protocol is proposed. The network video data streaming and preprocessing mechanism are used in this paper. Firstly, by identifying the data packets, the network video data packets are extracted from the data recorded on the airborne. Then, after unpacking and time calculation of the RTP-encapsulated H.264 video data, it uses FFmpeg's good decoding capabilities to decode a frame of image, and uses SDL's excellent video performance to display YUV data. The synchronization mechanism based on timestamp is adopted to realize the fast synchronized playback of multiple airborne network videos. The software adopts modular and multi-threaded design ideas to improve processing efficiency and ensure functional scalability. At present, the system has been successfully applied to the simultaneous playback of multi-channel airborne network video. The actual application effect shows that the system works stably and can effectively improve the efficiency of model flight tests.

1. Introduction
Airborne video data is an important part of the flight test data. The video image is an important basis for the flight test engineer to give the aircraft test conclusions because of its comprehensive, intuitive, and true reflection of the current aircraft and equipment working status [1].

At present, in the flight test, video telemetry and monitoring still adopt the mainstream PCM[2] architecture, and the airborne digital video signal is transmitted to the ground in the form of PCM stream for analysis. Generally, the multi-channel video data of the avionics system is separately recorded by the on-board recorder. After the flight, the video data of each channel is restored to standard video format files (*.m2t) through ground unloading, and general video playback software such as Baofengyingyin can play it. With the emergence of Internet radio stations based on Mesh architecture [3] [4], due to mission requirements, airborne video realizes networked telemetry and recording through Internet radio stations in the form of network data packets of the RTP/RTCP protocol. Different from the existing PCM structured telemetry digital video transmission system, the video data and other test parameters are all recorded in a network data file (*.enet) in the form of network data packets. Because of the changes in the airborne acquisition system and telemetry transmission mode, the recorded video data format has changed, and the existing video playback software cannot meet the new requirements of the model task.

According to the above requirements, in order to ensure the airborne multi-channel network video data playback task, the airborne network video synchronization playback technology based on...
RTP/RTCP[5][6] is proposed. Through the extraction of the airborne recorded network data, the network video data is split and pre-processed, the image is decoded using FFmpeg, and displayed using SDL.

2. RTP/RTCP protocol
The airborne webcam sends the video image in the form of RTP/RTCP protocol network data packets to the recorder through the switch for recording. The recorder encapsulates RTP/RTCP network data packets, that is, adds UDP header, IP header, and MAC header, and writes them into the file.

Real-time transport protocol (RTP) is used to provide end-to-end real-time transport services for multimedia data such as audio and video. The transmission service quality of RTP real-time transmission of audio and video streams is controlled by the corresponding real-time transport control protocol (RTCP). During the RTP session, the webcam periodically sends RTCP packets in the form of multicast, and the packets contain information such as the number of sent video data packets and the number of lost packets. Therefore, the webcam can use this information to dynamically change the transmission rate of video images.

When the webcam is turned on, an RTP session is established at the same time, and a destination transmission address consisting of a network address and a parity port is determined. The even-numbered UDP port 2n is allocated to RTP video packets, and the adjacent odd-numbered UDP port 2n + 1 is allocated to RTCP packets, forming a UDP port pair. RTP/RTCP data packets are sent through this UDP port pair.

The RTP protocol encapsulates the video image stream received from the network camera to obtain RTP data packets. The RTCP protocol receives control information from the network camera and encapsulates it into RTCP data packets. The combination of RTP and RTCP can optimize the transmission efficiency of video data packets through effective feedback and minimal overhead. When using RTP to package and encapsulate the H.264 code stream, the encapsulation format defines three different basic structures, and the receiving end recognizes the current package structure through the last 5 bits of the first byte of the RTP packet.

1) Single packet mode: An independent NALU is encapsulated with an RTP packet. The sequence parameter set SPS and the image parameter set PPS are smaller than the MTU (maximum transmission unit) due to the small amount of data. They are generally a single NALU and are transmitted in a single RTP packet encapsulation mode. The NAL header type field is the same as the original NAL unit type.

2) Aggregate packet mode: used to pack multiple NAL units into a single RTP data packet. Due to the large video image data, the amount of image data in one frame is generally greater than the length of the RTP data packet, so this encapsulation format is less widely used.

3) Fragmented packet mode: Pack an independent NALU into multiple RTP packets. An independent NALU is divided into multiple consecutive NAL unit bytes. Fragments of the same NALU must be sent sequentially using an increasing RTP sequence number, and the RTP data packet loss condition can be judged by the sequence number. Similarly, NAL units must be encapsulated in the order of RTP sequence number.

3. Overall design
The functional structure diagram of the airborne network video playback system is shown in Figure 1.
Airborne network video playback system

For the airborne record data processing module, the specific functions are as follows:

1) Data shunting: By identifying data packets, the airborne recorded network data is shunted and extracted into multiple data files such as ordinary low-capacity data, network video data, and high-capacity vibration data.

2) Video data preprocessing: Process network video data files, decapsulate and group RTP video data packets, and restore them to a complete frame of image, analyze RTCP data packets, and add before each frame of image Information such as identifier, time stamp, and data length of the frame image.

For the video data playback module, the specific functions are as follows:

1) Synchronous playback of multi-channel video: realize the synchronous playback of multi-channel video, keeping the time basically the same.

2) Play frame by frame: Play the key data segment frame by frame, including the previous frame and the next frame.

3) Data interception: intercept key data segments and save them as separate video files.

4) Time jump: realize fast jump to the video screen at the specified time.

5) Screenshot: save the current image as a BMP file.

6) Video decoding: decode a frame of video image into YUV data.

7) Image display: Use SDL to display YUV image data.

8) Image processing: A certain algorithm is used to realize functions such as image rotation and 1/4 partial display.

The airborne network video data playback flowchart is shown in Figure 2.

First, the data files recorded on the airborne are processed and the network video data is streamed out. Then preprocess the network data packet such as unpacking and calculating time to form a data file that can be used for fast playback. The video data packet format recorded by the network telemetry is consistent with the preprocessed video data file, and both can be played back. Read a frame of data and send it to FFmpeg for decoding, and then call SDL for display.
4. Software Implementation

4.1. Video playback technology based on FFmpeg and SDL

FFmpeg\cite{7}\cite{8}\cite{9} is an open source, cross-platform audio and video streaming solution that can record, convert, and stream audio and video. Libavcodec is a set of advanced audio/video codec libraries for the coding and decoding of various types of sounds/images. Libavformat is used to parse audio and video encapsulated in various formats, including functions such as obtaining key information required for audio and video decoding. Libswscale implements conversion of various image pixel formats and image size scaling functions. Libavutil is a public utility function library. After obtaining the complete NALU, send it to the FFmpeg buffer for decoding. The decoding process is: 1) initialize the decoder; 2) register the container and CODEC; 3) find the H.264 decoder; 4) open the H.264 decoder; 5) Allocate memory for the decoder; 6) Decode a frame of video image to obtain YUV data.

SDL\cite{10}\cite{11} (Simple Direct Media Layer) is an open source, cross-platform, multimedia development library. SDL itself does not have the function of character display, and the external extension library SDL_ttf needs to be used to display characters. SDL shows that YUV\cite{12}\cite{13} data flow is: 1) Initialize SDL and TTF libraries; 2) Create window and open font; 3) Create renderer based on window; 4) Create image texture and set text surface; 5) Settings Image texture data; 6) Create text texture data; 7) Copy image texture and text texture to the renderer.

4.2. Airborne network video preprocessing technology

The onboard record data format is a standard Ethernet packet, the format is [MAC Header][IP Header][UDP Header][Payload]. The recorder records all data packets passing through the onboard network switch, including normal data, video data, vibration data and other network data. Therefore, the data shunt technology is first used to extract the webcam video data. The data flow chart is shown in the figure 3.
After the video data is extracted, in order to realize the fast drag playback of the video data, the H.264 video data encapsulated by the RTP protocol will be unpacked and restored into a complete frame of NALU. Add the identifier, time stamp and data length before each frame of image data. The identifier represents the beginning of an image, the time stamp represents the time of the current image, and the data length represents the number of bytes occupied by the current image.

4.3. Airborne multi-channel network video fast synchronized playback technology
When multi-channel video is played back synchronously, time stamp synchronization technology is adopted to make the video picture accurate to the frame. Generally, the video frame rate is 25fps, and the time difference between two adjacent video frames is 40ms. Therefore, the time of starting to play the two video frames is the same, and the frame-by-frame sequential playback can keep the synchronization between multiple channels.

When dragging the progress bar, you only need to move the file pointer to the specified position and search for the identifier until a frame of image starts. Then use a certain channel of video as a benchmark to search for other channels of video. When the time stamp difference is the smallest, start playing frame by frame to maintain synchronization between multiple channels. First, take video 1 as the time stamp benchmark, read and display the current frame of image, and then the thread enters the waiting state and informs the video 2 thread. The video 2 thread reads a frame of image in turn and calculates the current time difference. If the time difference between the frame and the reference frame
is the smallest, the image is displayed and the video 1 thread is notified to read the next frame of
image.

The logic flow chart of time jump processing is shown in the figure below. First read the current
frame of data and compare the current time Tn(s) with the specified time Td(s). If Tn<Td, it means that
the specified time has not been reached, and the file pointer must be moved forward; otherwise, the
specified time has been missed and the file pointer must be moved backward. In general, every time a
frame of data is read, it is necessary to determine whether the frame time is equal to the specified time,
which is time-consuming. Since the frame rate (fps) of the video is fixed, the number of image frames
in the time period |Tn-Td| is known, which is |Tn-Td|*fps. Therefore, the method of moving |Tn-
Td|*fps frame number instead of comparing time is adopted to improve the search speed and realize
the rapid positioning of the video frame at the specified time.

![Figure 4. Time jump logic processing flowchart](image)

4.4. Screenshot function based on BMP bitmap file

BMP format bitmap file is a standard image file format in Windows operating system, and it is a
device-independent bitmap. In order to realize the screenshot function, the YUV data is converted to
RGB data, and then a 54-byte file header and bitmap information header are added to the BMP file.
The color space conversion relationship between YUV and RGB is as follows:

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 1.402 \\
1 & -0.34414 & -0.71414 \\
1 & 1.772 & 0
\end{bmatrix} \begin{bmatrix}
Y \\
U \\
V
\end{bmatrix}
\]  (1)

If the RGB value is greater than 255, the value is 255; if it is less than 0, the value is 0.

A BMP image file is composed of 4 parts: bitmap file header, bitmap information header, palette
and bitmap data [14]:

```c
struct stcBITMAPFILEHEADER
{
    INT16 FileType; //must be "BM"
```
INT32 FileSize; //File size
INT16 Reserved1; // is 0
INT16 Reserved2; // is 0
INT32 DataOffset; //Start position of bitmap data

(2) The data structure of the bitmap information header. Contains information such as the size and compression type of the BMP image, and its structure is defined as follows:

```
struct stcBITMAPINFOHEADER
{
    INT32 Size;//Number of bytes occupied
    INT64 PicWidth; //Width
    INT64 PicHeight; //Height
    INT16 Planes; //is 1
    INT16 Bits; //The number of bits required for each pixel
    INT32 CprssType; //Compression type
    INT32 ImageSize;
    INT64 XDPI; //Horizontal resolution
    INT64 YDPI; //Vertical resolution
    INT32 lRUsed;
    INT32 lRmajor;
};
```

When the number of bits required for each pixel is 1, it means that 1 byte is 8 pixels; When the number of bits required for each pixel is 4, it means that 1 byte table is 2 pixels; When the number of bits required for each pixel is 8, it means that 1 byte is 1 pixel; When the number of bits required for each pixel is 24, it means that 3 bytes are 1 pixel.

(3) Palette. True color images (24-bit BMP) do not need a palette;

(4) Bitmap data. Scan each pixel value of the bitmap in the order from left to right in the line and bottom to top between the lines.

5. Application effect

The airborne multi-channel network video data synchronous playback interface is shown in the figure below, which can realize the synchronous playback of multi-channel network video data files, with functions such as pause, previous frame, next frame, screenshot, image rotation, and time jump.

![Figure 5. Airborne multi-channel network video data synchronization playback software interface](image)

6. Conclusion

This article discusses the design and implementation of a multi-channel airborne network video data synchronization playback system. By splitting and extracting network packet data, preprocessing the
RTP video data packet such as unpacking and time calculation to form a pre-processing file. Using FFmpeg and SDL to decode and display a frame of image. Using a time stamp synchronization mechanism to achieve multiple networks. Precise synchronized playback of videos. The actual application effect shows that the system can realize the fast synchronous playback of multi-channel airborne network video data, and effectively improve the flight test efficiency.

Acknowledgments
Authors wishing to acknowledge assistance or encouragement from colleagues, this work is supported by the National Defense Science and Technology Industry Administration National Defense Basic Research Program Key Project under grants JCKY2016205B006.

References
[1] Lichter M, Bateman A, Balas G. (2009) Flight Test Evaluation of a Run-Time Stability Margin Estimation Tool. In: AIAA Guidance, Navigation, and Control Conference.
[2] Z Wu, N Zhao, S Li, G Ren. (2009) A Novel PCM/FM Multi-symbol Detection Algorithm for FPGA Implementation. Information Technology, 8(4):580-890.
[3] Z Neing, Q Song, L Guo, Z Chen, A Jamalipour. (2016) Integration of scheduling and network coding in multi-rate wireless mesh networks: Optimization models and algorithms[J]. Ad Hoc Networks, 36(1):386-397.
[4] Akyildiz I F, Wang X, Wang W. (2005) Wireless mesh networks: a survey. Computer Networks, 47(4):445-487.
[5] P Cheng, H Qu. (2015) The research of Media streaming technology based on RTP protocol. In: International Conference on Information Sciences, Machinery, Materials and Energy. Paris: Atlantis Press,1234-1237.
[6] Kristensen T, Wang Y K, Even R. (2011) RTP Payload Format for H.264 Video. Ietf Rfc Feb. wenger Et Al. Transport & Signaling of Svc in Ip Networks, 76(12):792.
[7] Z Hao, F Yuan. (2013) Implementation of video transcoding client based on FFMPEG[J]. Advanced Materials Research, (3), 756-759.
[8] Roitzsch M. (2007) Slice-balancing H.264 video encoding for improved scalability of multicore decoding. In: Acm & IEEE International Conference on Embedded Software. ACM.
[9] Stockhammer T, Hannuksela M M. (2005) H.264/AVC video for wireless transmission[J]. IEEE Wireless Communications, 12(4):6-13.
[10] Y Wang, M Gao, Z Yu. (2008) Chinese Display Technology on Embedded Platform Using Simple DirectMedia (SDL). In: International Workshop on International Workshop on Education Technology & Training. IEEE.
[11] Li K, Li L, Lan S Y. (2014) Multiple video streams real-time conversion and playback based on FFmpeg and SDL. Computer Technology and Development, 24(4): 65-68.
[12] J Mukherjee, M K. Lang, S.K. Mitra. (2005) Demosaicing of images obtained from single-chip imaging sensors in YUV color space - ScienceDirect. Pattern Recognition Letters, 26(7):985-997.
[13] J Mukherjee, M K. Langb, S.K.Mitra. (2005) Demosaicing of images obtained from single-chip imaging sensors in YUV color space[J]. Pattern Recognition Letters,26(7): 985-997.