A New Method for Multi-channel Audio Transmission Based on AVB Standard

Liming Shao[1, a], Chao Chen[1, b] and Qi Wang [1, c]

[1]Communication University of China, Beijing, China

[a]shaoliming@cuc.edu.cn, [b]chenchao@cuc.edu.cn, [c]vita1982@cuc.edu.cn

Keywords: AVB, AVTP, AAC, ADTS

Abstract. This paper firstly introduces Ethernet Audio Video Bridging (AVB) standard developed by the IEEE 802.1 Audio Video Bridging Task Group, then focuses on AVTP standard which is aimed to transport the audio and video data, next introduces the new multi-channel network audio transmission protocol that we designed by modifying AVB Transport Protocol Data Unit. In the last part of the paper, the frame encapsulation and performance of the new audio protocol are described.

Introduction

Ethernet Audio Video Bridging (AVB) is a set of technical standards developed by the Audio Video Bridging Task Group of the IEEE 802.1 standards committee. This task group had been renamed to Time-Sensitive Networking Task Group at November 2012 to reflect the expanded scope of work.

On the basis of the traditional Ethernet, AVB supports time-sensitive multimedia applications based on the network audio and video streaming, with some unique features like low latency, precise synchronization, traffic shaping for media streams, admission controls. [1]

The base system of AVB primarily consists of four IEEE 802.1 AVB standards:
a. IEEE 802.1AS: Precision Time Protocol (PTP)
b. IEEE 802.1Qat: Stream Reservation Protocol (SRP)
c. IEEE 802.1Qav: Forwarding and Queuing Enhancements for Time-Sensitive Streams (FQTSS)
d. IEEE 802.1BA: Audio Video Bridging Systems

In addition to the above, there are two high-level protocols:
a. IEEE 1722: Audio Video Bridging Transport Protocol (AVTP)
b. IEEE 1733: Real-time Transport Protocol (RTP)

Fig. 1 is the block diagram of the IEEE 802.1 AVB protocol stacks.

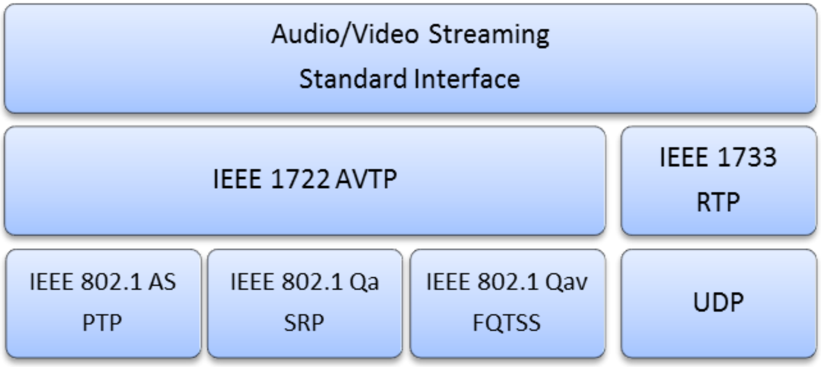

Fig. 1 – Block diagram of the IEEE 802.1 AVB protocol stacks

Precision time protocol (IEEE 802.1AS) defined by AVB provides the perfect low latency, low jitter clock for Ethernet. Stream reservation protocol (802.1Qat) allows network resources to be reserved for specific traffic streams traversing a bridged local area network by using the consultation mechanism. By using 802.1Qav forwarding and priority mechanisms, the AVB architecture implements traffic shaping. AVB

Systems standard specifies the default configuration for AVB network devices, which can help inexperienced users to use AVB devices easily. [1, 2]

Audio Video Bridging Transport Protocol (AVTP) provides time synchronization and latency/bandwidth services by defining the packet format protocols and synchronization mechanisms, which will facilitate interoperability between stations that stream time-sensitive audio and/or video across LANs. [3]

Frame Structure of AVTP

We described the basic framework for AVB standard in the first paragraph, next will talk about Audio Video Bridging Transport Protocol (AVTP) as an important part of AVB standard.

AVTP specifies a protocol standard for audio/video data transported on AVB network. AVTP is designed to take advantage of networks that support gPTP, SRP, FQTSS. What's more, AVTP specifies methods to transport audio/video data and timing information so that AVTP streams transmitted from a transmitter can be received by one or more receivers. [3]

The IEEE 802.3 encapsulation of stream data and control data of Audio/Video Transport Protocol Data Unit (AVTPDU) is described in Fig. 2, and the shaded part is the AVTP data.

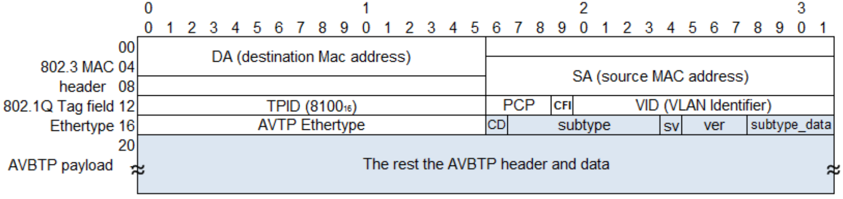

Fig. 2. - AVTP control or stream data frame in an IEEE 802.3 frame with an IEEE 802.3Q Tag field (Source: IEEE Std 1722-2011 p. 43)

The first 18 octets in Fig. 2 are IEEE 802.3 Ethernet header information with an IEEE 802.3Q Tag field. The 17-18th octets are Ethernet type flag (Ethertype) field, and the Ethertype field of AVTP is 22F0$_{16}$.

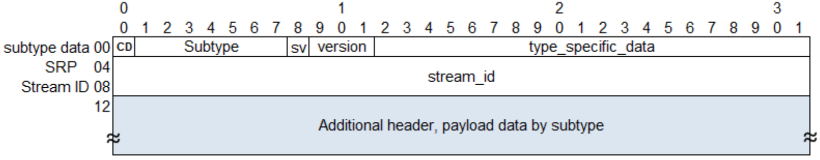

Fig. 3. - AVTPDU common header fields (Source: IEEE Std 1722-2011 p. 8)

Fig. 3 shows the common header fields of stream data and control data of AVTPDU.

The *cd* (control/data) field is 1bit. If the cd bit is zero (0), then this AVTPDU is a stream data AVTPDU. In the case of control data, the cd bit is one (1).

The *subtype* field indicates the protocol type carried by AVTPDU. Currently defined subtype values are listed in Table 1.

Table 1. – AVTP subtype values

Hexadecimal value	Function	Meaning
00$_{16}$	61883_IIDC_SUBTYPE	IEC 61883/IIDC over AVTP
01$_{16}$	MMA_SUBTYPE	MMA payload over AVTP
02$_{16}$-7D$_{16}$	----	Reserved for future protocols
7E$_{16}$	MAC address acquisition protocol (MAAP)	MAAP
7F$_{16}$	Experimental subtype	Experimental over AVTP

It has been mentioned above that Fig. 3 shows the common header fields of stream data and control data of AVTPDU. If the cd bit is zero (0), then this AVTPDU is a stream data AVTPDU. Fig. 4 shows the common header fields of stream data that we reference primarily.

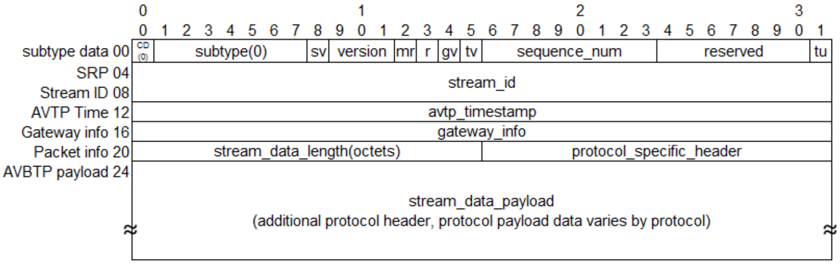

Fig. 4. - AVTP common stream header format (Source: IEEE Std 1722-2011 p. 11)

There are some brief explanations about the fields in Fig. 4:

The *mr* field indicates that there has been a change in the source of the media clock. The *r* field is reserved for future use. The *gv* field is not defined by this standard. The field should be set "0" by transmitter and ignored by receiver. The *tv* field is used to indicate the validity of the *attp_timestamp* field time value.

The *sequence_num* field indicates the sequence of AVTPDUs in a stream, and listeners can use the sequence number to detect AVTPDUs lost in transmission. The *tu* field shall be set by transmitter when the attached gPTP system reports any significant problem with its synchronization system. The *avtp_timestamp* field contains the AVTP presentation time. The field shall be in units of nanoseconds and has a maximum value of 232-1 ns.

The *gateway_info* field is not defined by this standard. The field should be set "0" by transmitter and ignored by receiver. The *stream_data_length* field indicates the length of the payload following the *protocol_specific_header* field. The *protocol_specific_header* field is specified by the protocol carried by AVTPDU. The *stream_payload_data* field consists of additional protocol specific data. [3]

New Structure Based on AVTP for Multi-channel Audio Transmission

Now many occasions require high-performance network audio transmission system, such as concert, large stadium, and large aircraft. In these cases the transmission of multi-channel audio and low latency is very important.

For example, on large-scale concerts, many different audio, like vocals and instruments, need to be transported to different speakers fast. This process requires low latency and performance to change audio routing quickly, while convention analog audio system is not well positioned to meet these requirements.

AVB has the ability to transmit audio/video media stream simultaneously and efficiently, like MPEG2-TS, SD-DVCR. Furthermore, AVB supports low latency. The maximum latency per hop is 250 µs in AVB networks so that the latency of media data transmitted is less 2 ms in 7-hop LAN.

Based on the above advantages of AVB, Audio Video Bridging Transport Protocol (AVTP) can be redesigned to transport more channels audio efficiently. The *protocol_specific_header* field and the *stream_data_payload* field in AVTP data unit are redefined.

Fig. 5 shows the modifications what we have done for frame structure of AVTP stream data.

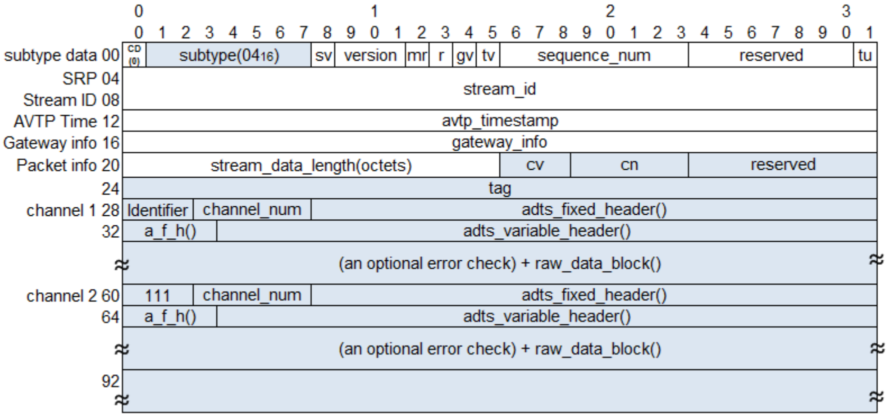

Fig. 5. – The new protocol common header fields

There are some brief explanations about the fields in Fig. 5:

The *subtype* field indicates the protocol type carried by AVTPDU. We use 04_{16} in the range of reserved values as the protocol type that we designed. The *cv* field indicates the version of new protocol and should be set 000_2. Other values are reserved for future version. The *cn* field is 5 bits and indicates number of channel whose value has a maximum 32.

The *reserved* field should be set 0 by transmitter and be ignored by receiver. The *tag* field is the label of original channel and is used to specify which channel is transmitted. The value of 1 represents transmitting while 0 is not. For example, when the first 8 bits of the field is 10110001, the 1/3/4/8th channel in first 8 channels are transmitted.

The *identifier* field should be set 111_2 as an identifier of the channel's beginning. The *channel_num* field is channel number (0-32). The *a_v_h()* field is abbreviation of *adts_variable_header().*

AVTP payload data starts from the 28th octet. The payload data in the protocol we designed is AAC-ADTS data. Advanced Audio Coding (AAC) is an audio compression algorithm with high compression ratio and also is currently the most widely used low-bit-rate audio coding standard. AAC has lower bit rate at the same quality compared with MP3, WAV, and can save bandwidth effectively.

Audio data should be sampled and quantized before compression and coding, and be stored in the form of sample values. The output stream after compression and coding exists in the form of audio frames. Each audio frame contains a number of compressed audio sample data and is called Raw Data Block. The Raw Data Block whose length is variable forms ADTS frame after encapsulated. ADTS audio compression stream is suitable for network transmission. [5]

ADTS data is composed of adts_variable_header(), adts_variable_header(), the optional error_check, raw_data_block(). Adts_fixed_header() is fixed header of ADTS. The information in this header does not change from frame to frame. It is repeated every frame to allow random access into a bitstream. Adts_variable_header() is variable header of ADTS. This header is transmitted every frame as well as the fixed header, but contains data that changes from frame to frame. [4]

Fixed Header of ADTS

Table 2. - Syntax of adts_fixed_header()

Syntax	No. of bits	Mnemonic
adts_fixed_header()		
{		
syncword;	12	bslbf
ID;	1	bslbf
layer;	2	uimsbf
protection_absent;	1	bslbf
profile;	2	uimsbf
sampling_frequency_index;	4	uimsbf
private_bit;	1	bslbf
channel_configuration;	3	uimsbf
original/copy;	1	bslbf
home;	1	bslbf
}		

There are some explanations about the fields in Table 2:

The *syncword* field shall be set '1111 1111 1111'. The ID field is MPEG identifier, and shall be set to '1'. The *layer* field indicates which layer is used and shall be set to '00'. The *protection_absent* field indicates whether *error_check()* data is present or not.

The profile field indicates AAC coding level. There are three profiles identified in the MPEG-2 AAC standard: Main Profile (00), Low Complexity Profile (01), Scalable Sampling Rate Profile (10).

The *sampling_frequency_index* field indicates the sampling frequency used according to Table 3.2. The *channel_configuration* field indicates the channel configuration used. [4]

Table 3. ADTS sampling frequency index

sampling_frequency_index	sampling_frequency [Hz]
0x0	96000
0x4	44100

Variable Header of ADTS

Table 4. - Syntax of adts_variable_header()

Syntax	No. of bits	Mnemonic
adts_variable_header()		
{		
copyright_identification_bit;	1	bslbf
copyright_identification_start;	1	bslbf
frame_length;	13	bslbf
adts_buffer_fullness;	11	bslbf
number_of_raw_data_blocks_in_frame;	2	uimsbf
}		

There are some explanations about the fields in Table 3.2:

The *copyright_identification_bit* field is one bit of the 72-bit copyright identification field. The copyright identifier is given by a Registration Authority as designated by SC29. The *copyright_identification_start* field indicates whether copyright identification is transmitted or not. If copyright identification is transmitted, this bit should be kept '1'.

The *frame_length* field indicates the length of the frame including headers and error_check in bytes. The *adts_buffer_fullness* field indicates state of the bit reservoir in the course of encoding the ADTS frame.

The *number_of_raw_data_blocks_in_frame* field is number of raw_data_block()'s that are multiplexed in the adts_frame() is equal to number_of_raw_data_blocks_in_frame + 1. The minimum value is 0 indicating 1 raw_data_block().[4]

Conclusions

The new audio transport protocol is based in AVB protocol. In addition to the payload data different with AVB protocol, the new protocol also utilizes Precision Time Protocol (PTP), Stream Reservation Protocol (SRP), FQTSS.

Each frame of the new protocol contains up to 32 audio channels. Each channel contains 32 octets of audio information. The channels share the same sampling rate and other parameters. The number of AVTP data frame transmitted per second depends on the bit rate of one audio channel. The larger the bit rate, the more data frames transmitted per second. There are about 24 octets of audio data and 8 octets of audio header information in every channel.

For example, the bit rate of an audio channel is M bps, and the number of audio frame transmitted per second is N. M and N meet the following formula:

$$n \approx \frac{M}{24*8} \tag{1}$$

The new protocol supports Precision Time Protocol (PTP), Stream Reservation Protocol (SRP), FQTSS, so it is compatible with AVB. The new protocol has the similar latency with AVB.

The transmitter shall transmit a valid timestamp value in the avtp_timestamp field once every audio frame period. The timestamp shall be transmitted in an AVTPDU that meets the following conditions:

packet_arrival_time_L ≤ timestamp value
timestamp value – transmission_delay_limit ≤ packet_arrival_time_F

where

packet_arrival_time_F is the gPTP time when the first bit of the packet that has the timestamp has arrived at the receiver.
packet_arrival_time_L is the gPTP time when the last bit of the packet that has the timestamp has arrived at the receiver.
transmission_delay_limit is equal to AVTP Max Transit Time.[3]

Acknowledgements

This work was financially supported by the National Key Technology Support Program (2012BAH51F02).

References

[1] Information on https://en.wikipedia.org/wiki/Audio_Video_Bridging
[2] Dong He, Sheng Huang: Next-generation Network Audio/Video Real-time Transmission Technology: Ethernet AVB (in Chinese)
[3] IEEE 1722™-2011: IEEE Standard for Layer 2 Transport Protocol for Time-Sensitive Applications in Bridged Local Area Networks
[4] ISO/IEC 13818-7: Information technology — Generic coding of moving pictures and associated audio information: Part 7: Advanced Audio Coding (AAC)
[5] Xiaoqiang Guo, Guangtao Fu: Format Analysis of ADTS and LATM in Audio Coding Standard. J. Modern TV technology (2008) (in Chinese)
[6] IEEE 802.1Qat: IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment: Stream Reservation Protocol (SRP)
[7] IEEE 802.1Qav: IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment: Forwarding and Queuing Enhancements for Time-Sensitive Streams
[8] IEEE 802.1AS: IEEE Standard for Local and metropolitan area networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks
[9] IEEE 802.1BA: IEEE Standard for Local and metropolitan area networks— Audio Video Bridging (AVB) Systems