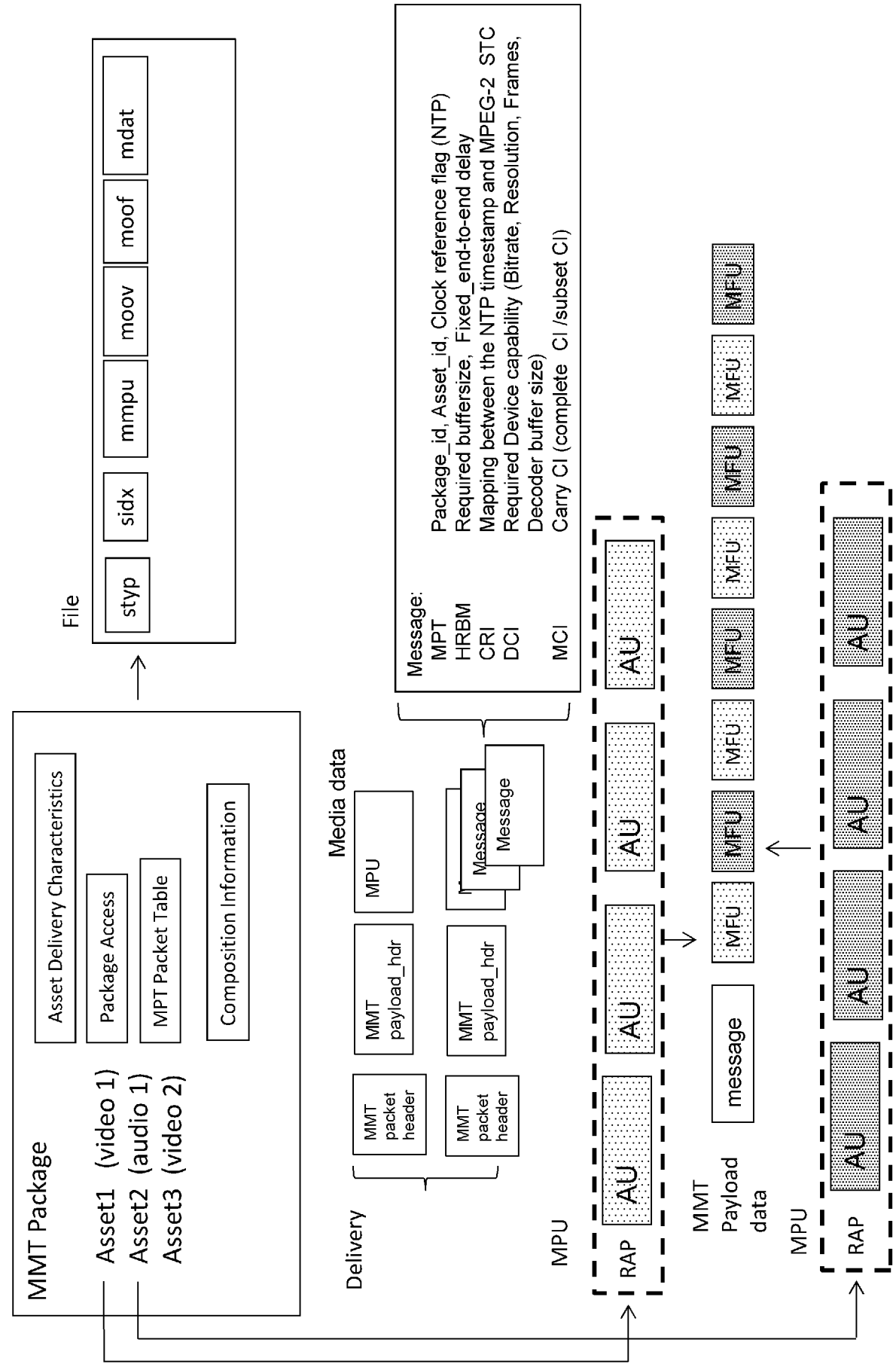


[図1]

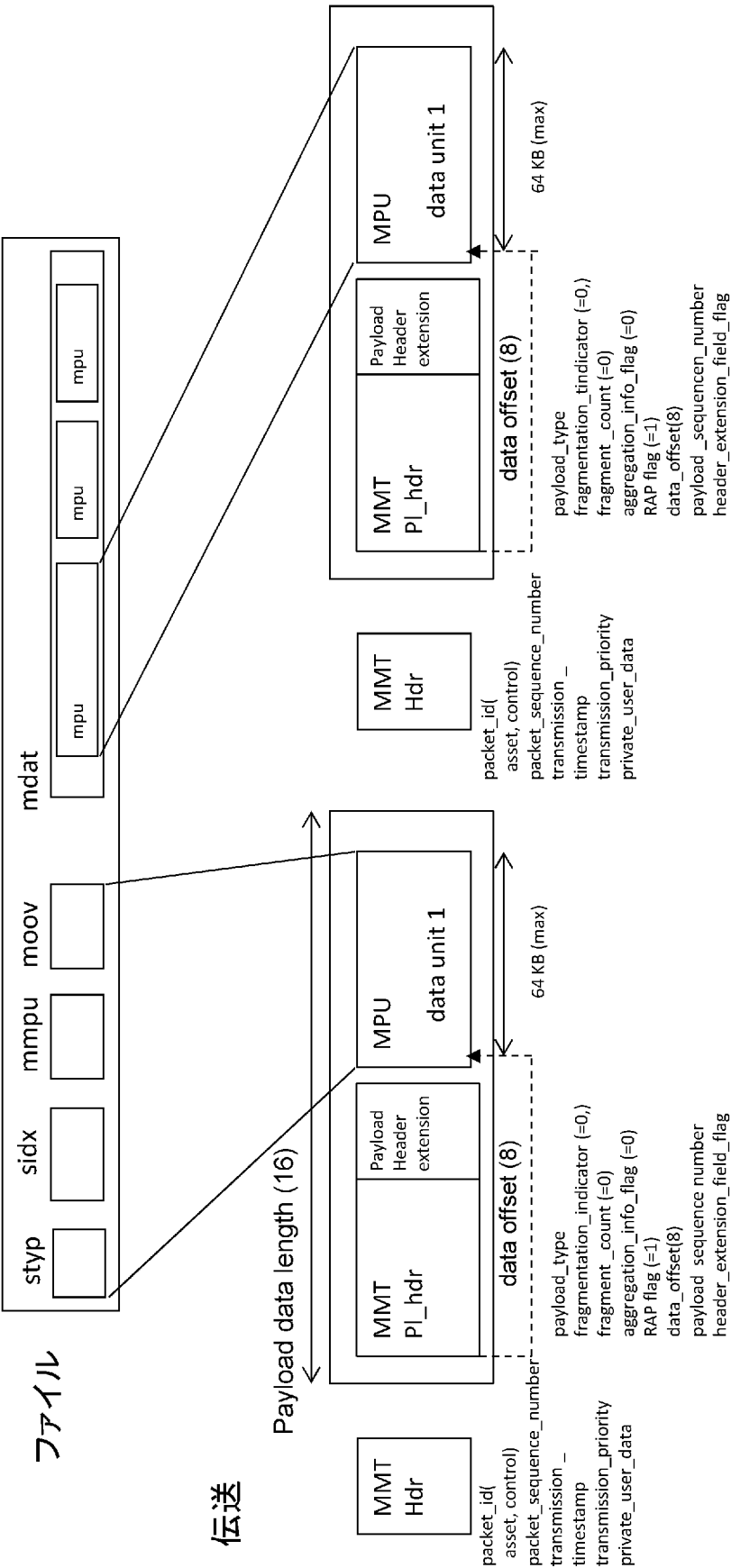


[図2]

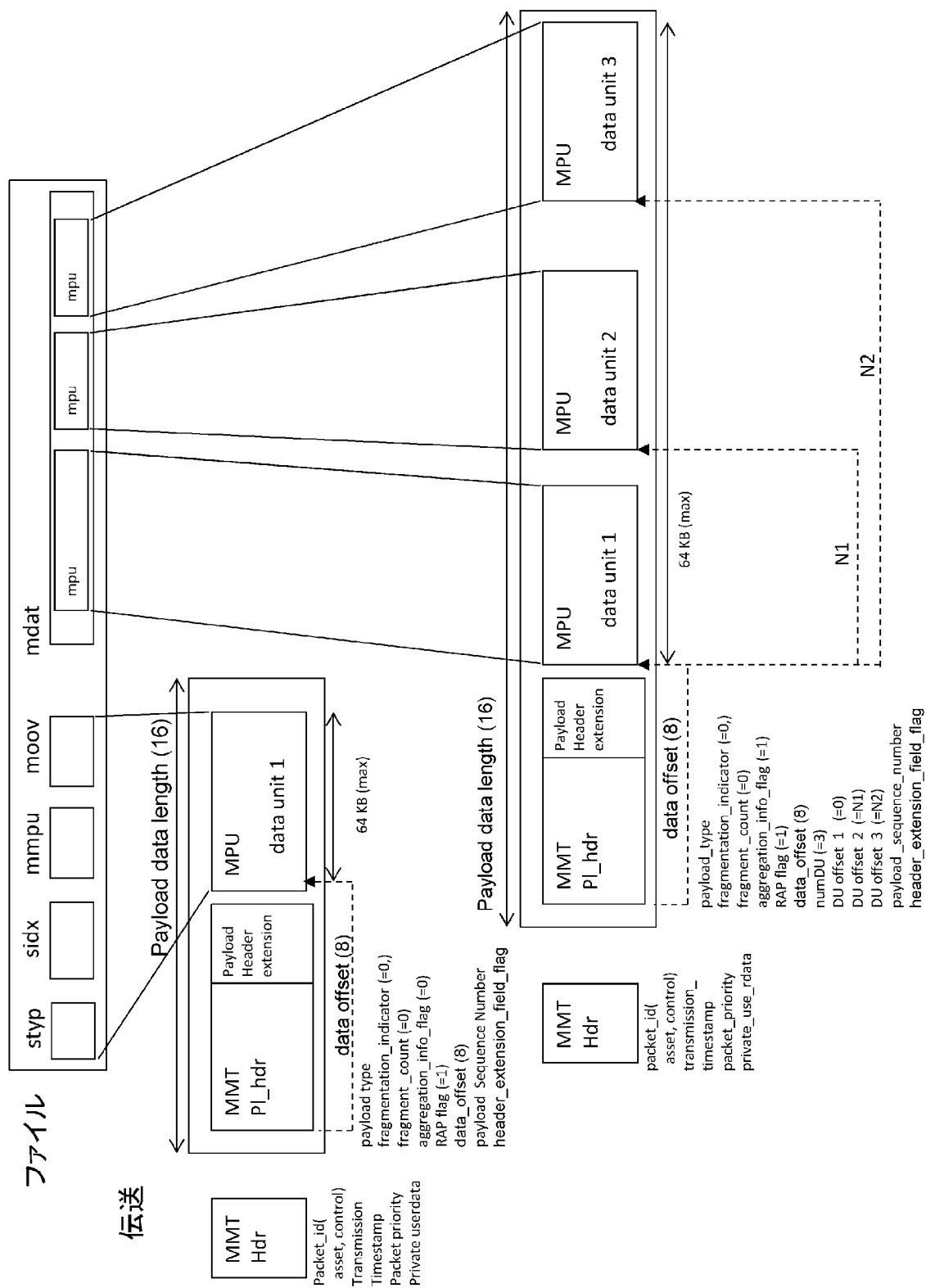


[図3]

MPU packetize case1 “Meta, or MPU data in MMT payload”

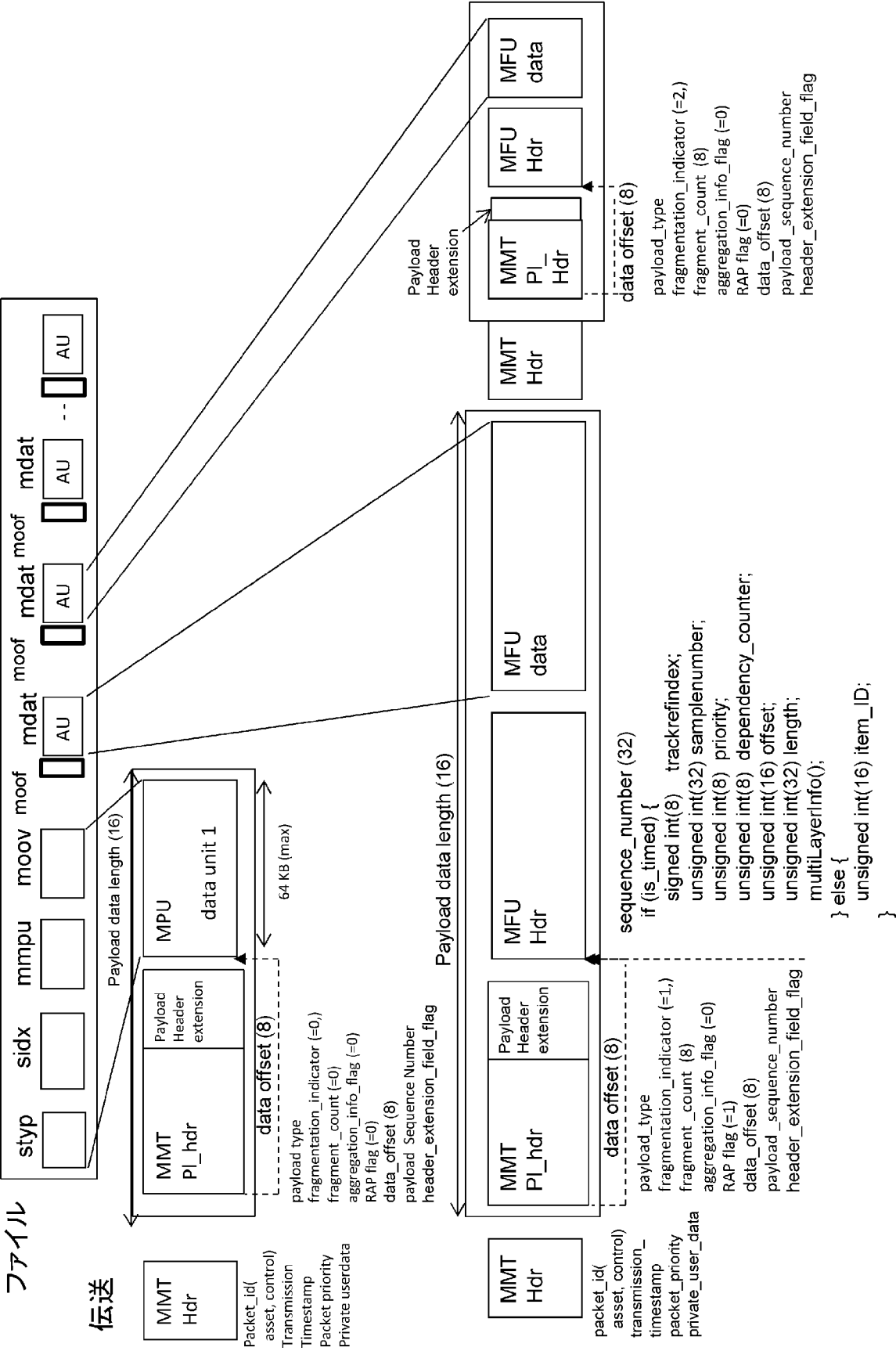


MPU packetize case2 “Aggregated MPUs”

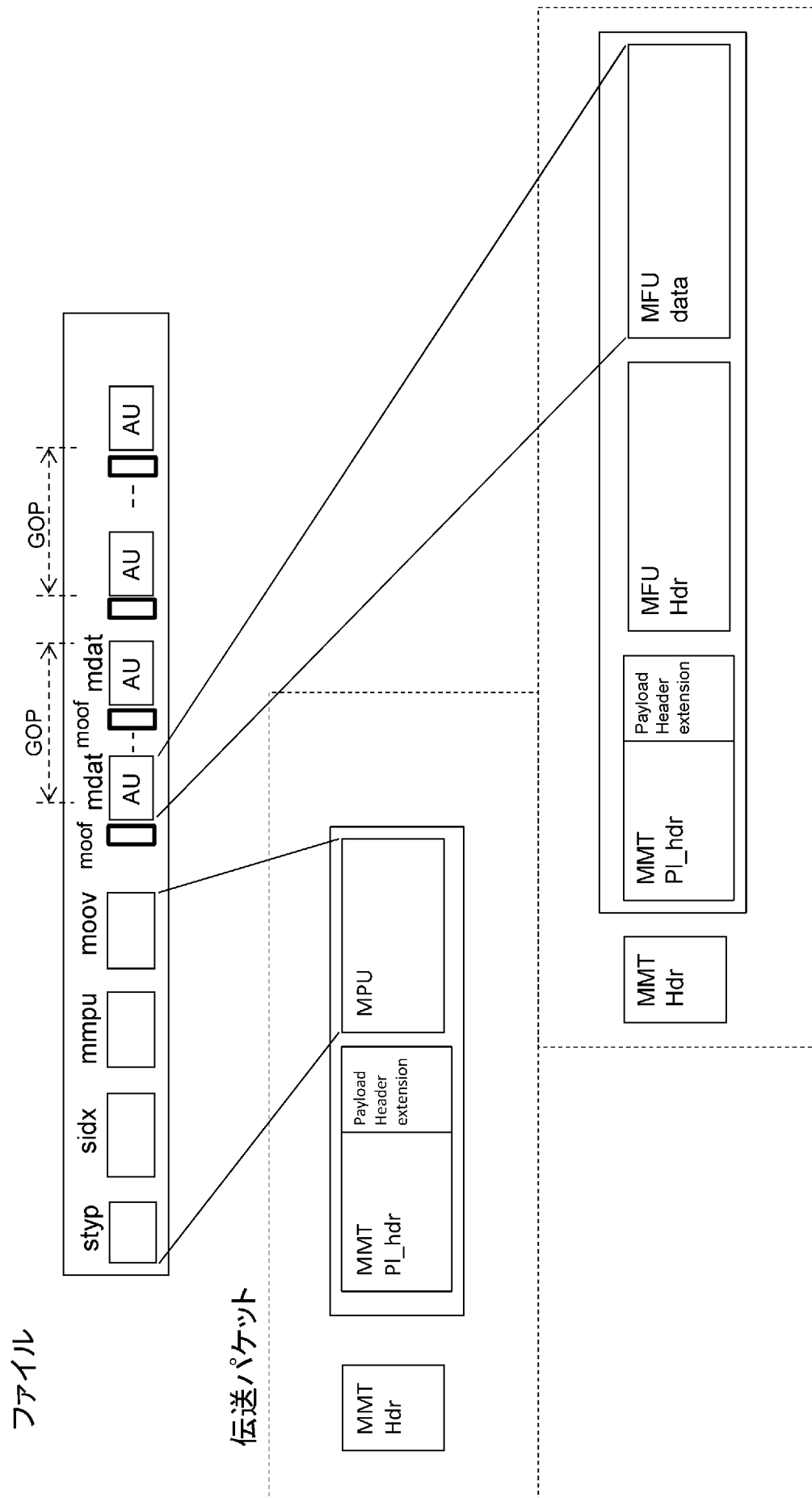


[図5]

MPU packetize case3 “Media Fragmented case”



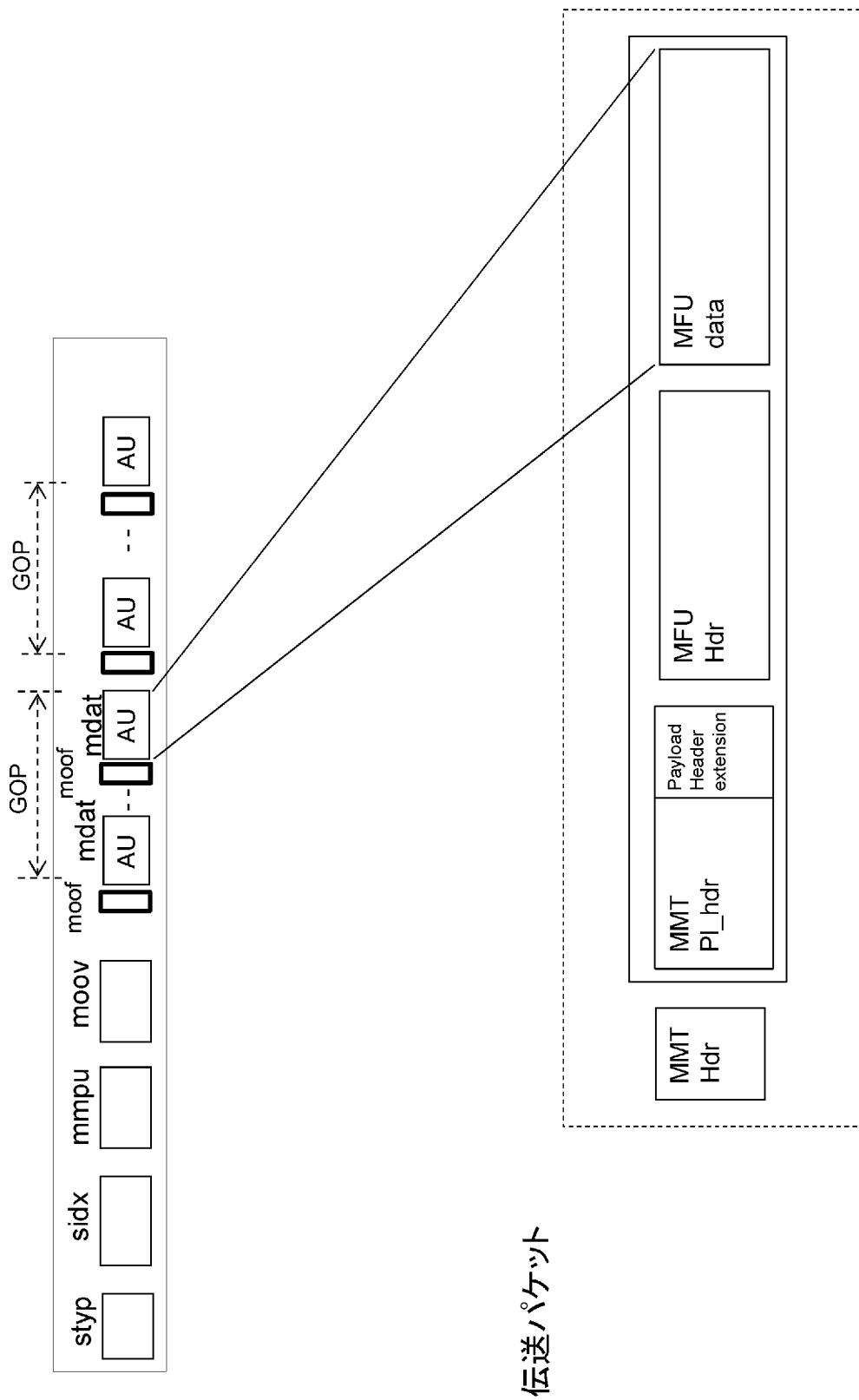
“Media Fragmented case” (1)



[図7]

“Media Fragmented case” (2)

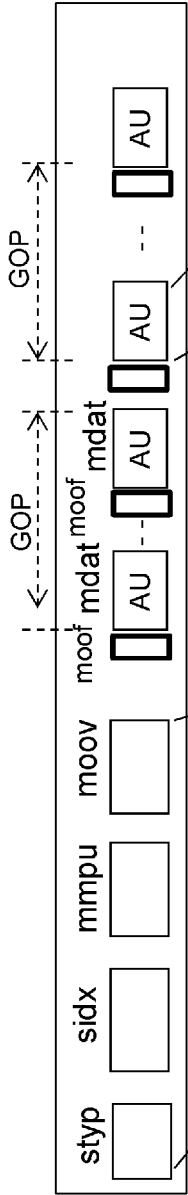
ファイル



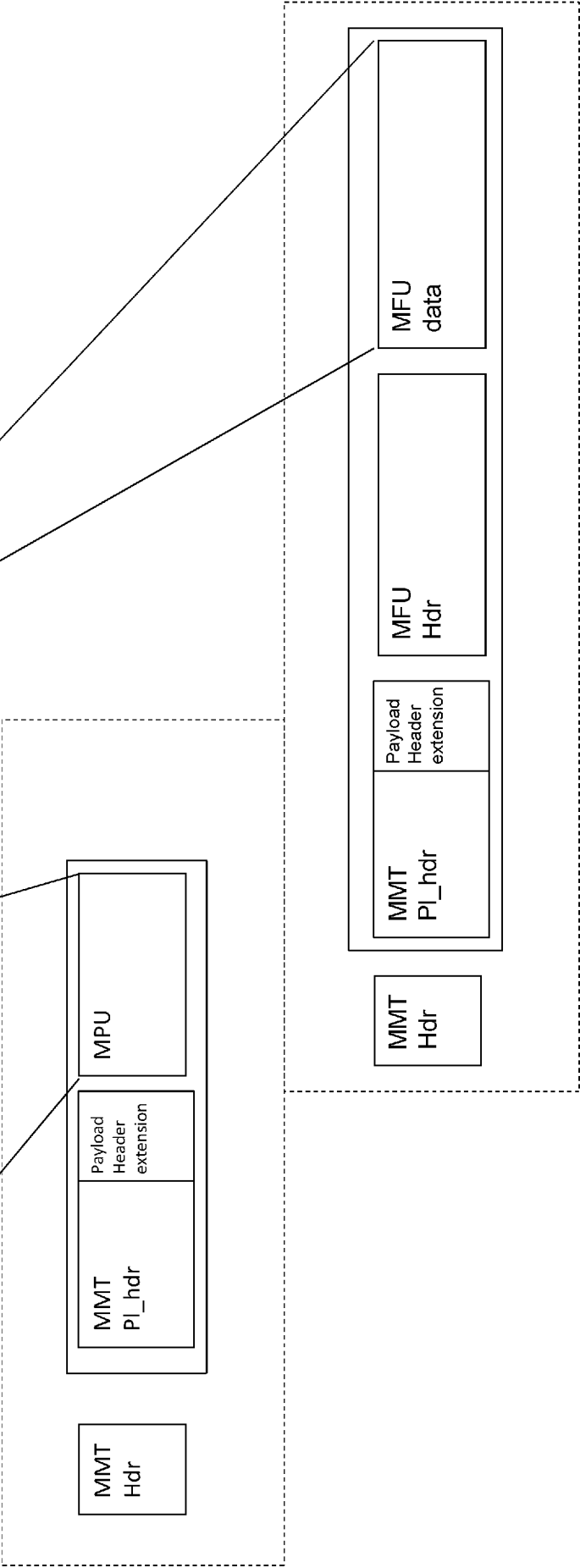
[図8]

“Media Fragmented case” (3)

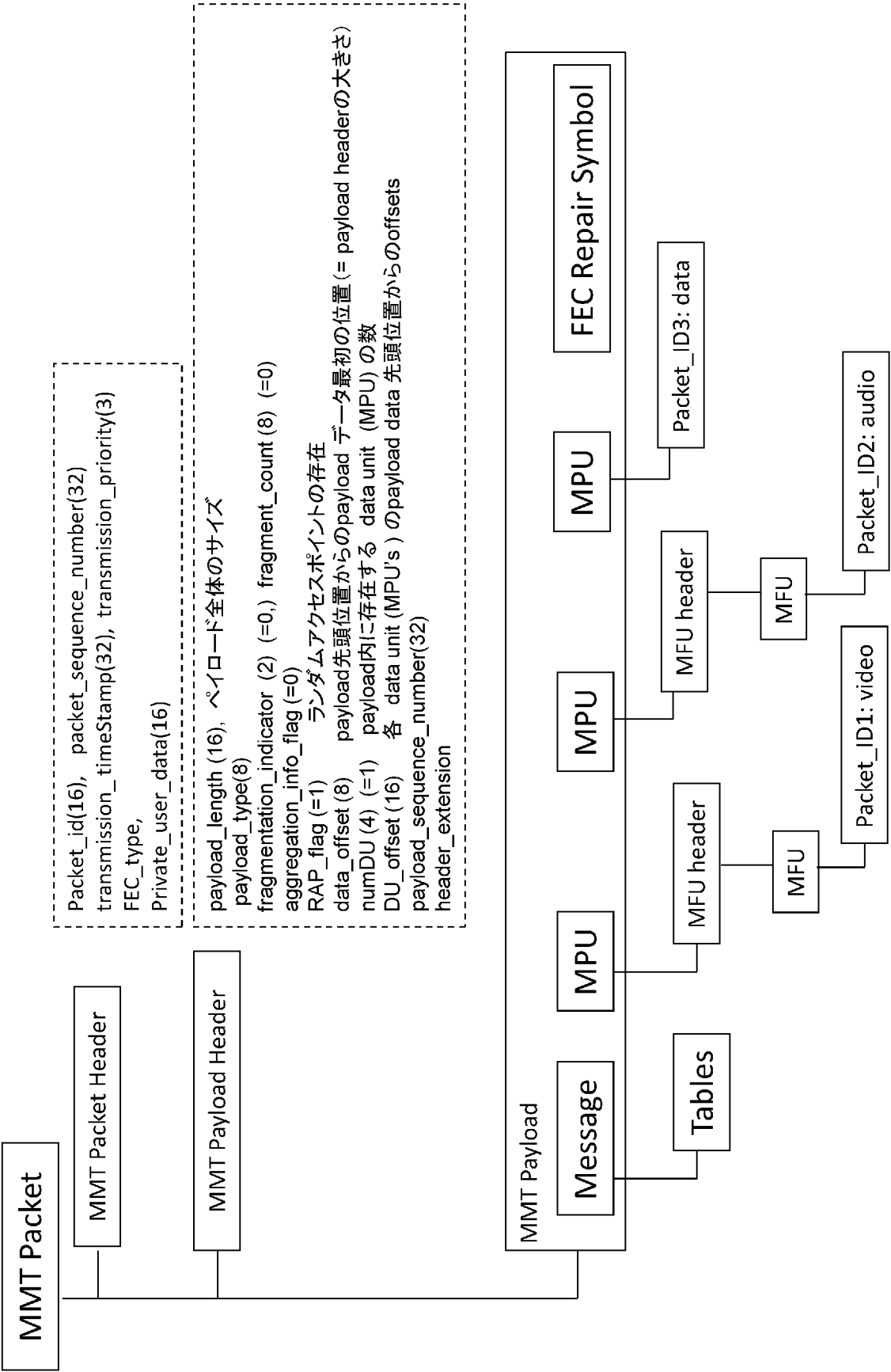
ファイル



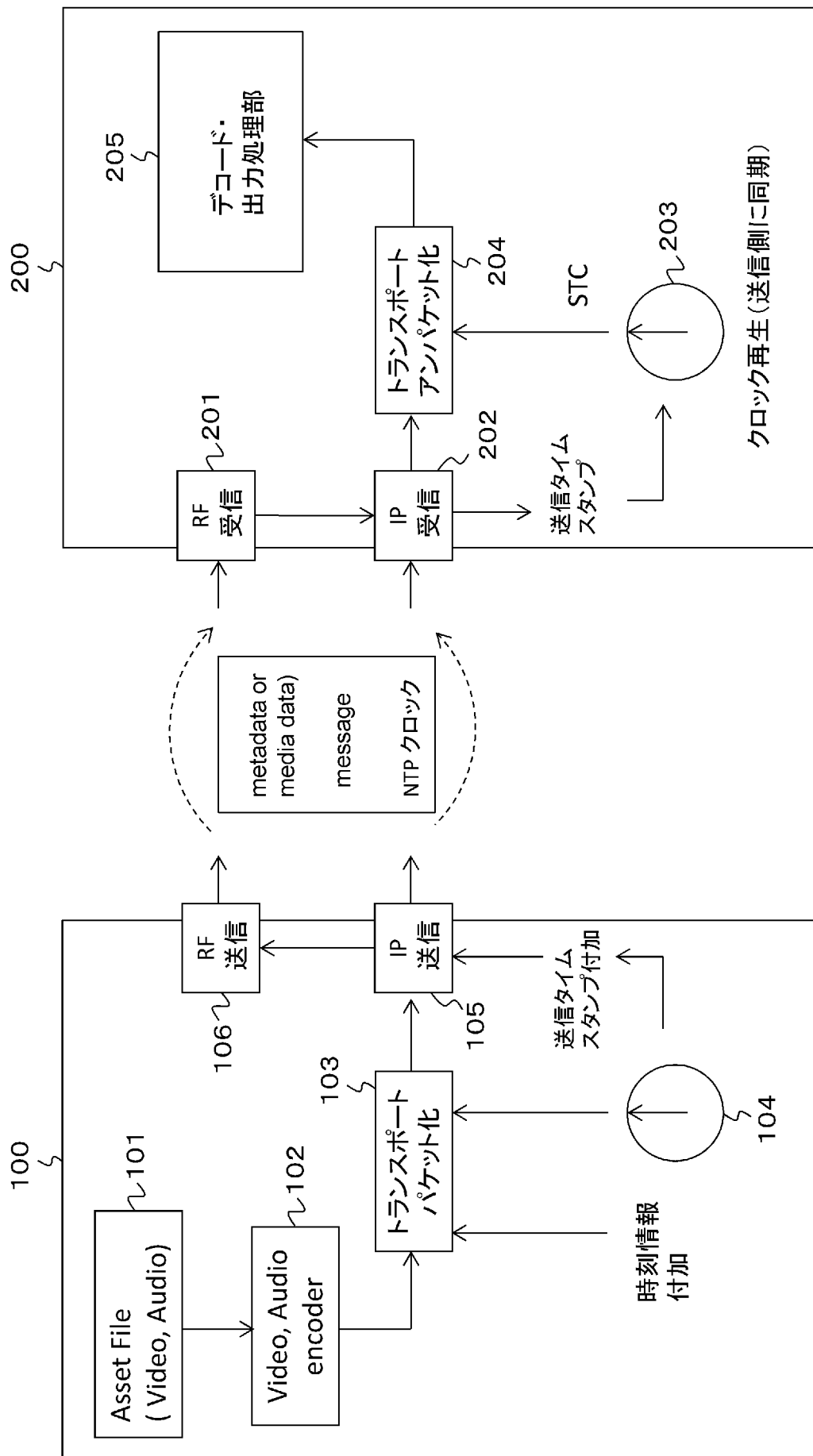
伝送パケット



[図9]

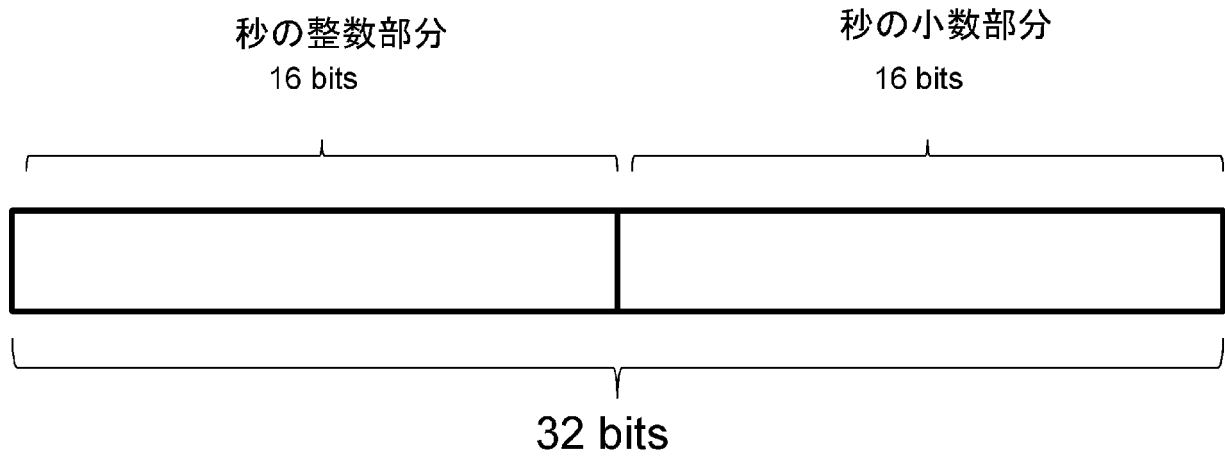


[図10]



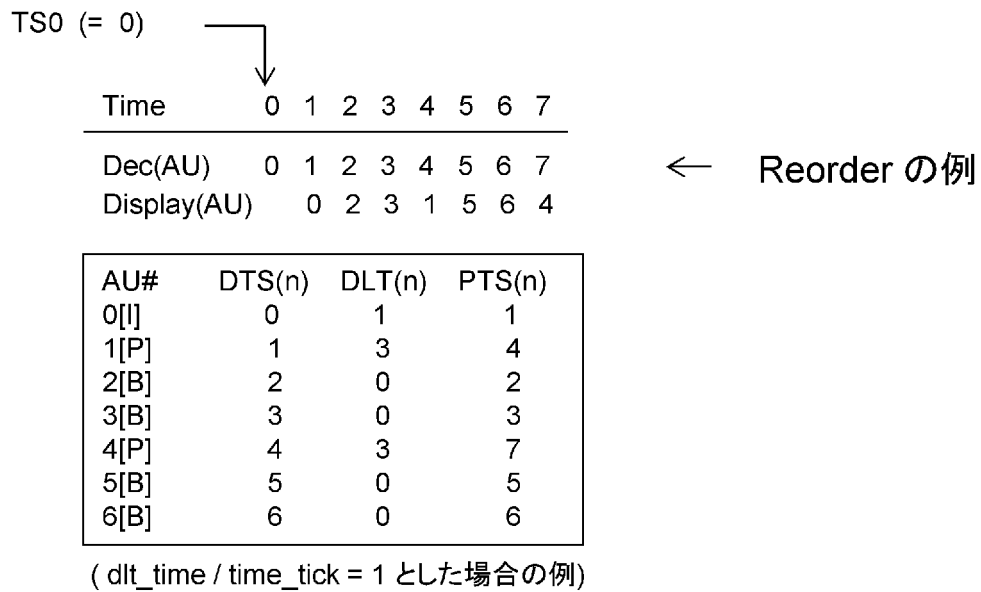
[図11]

NTP Short フォーマット

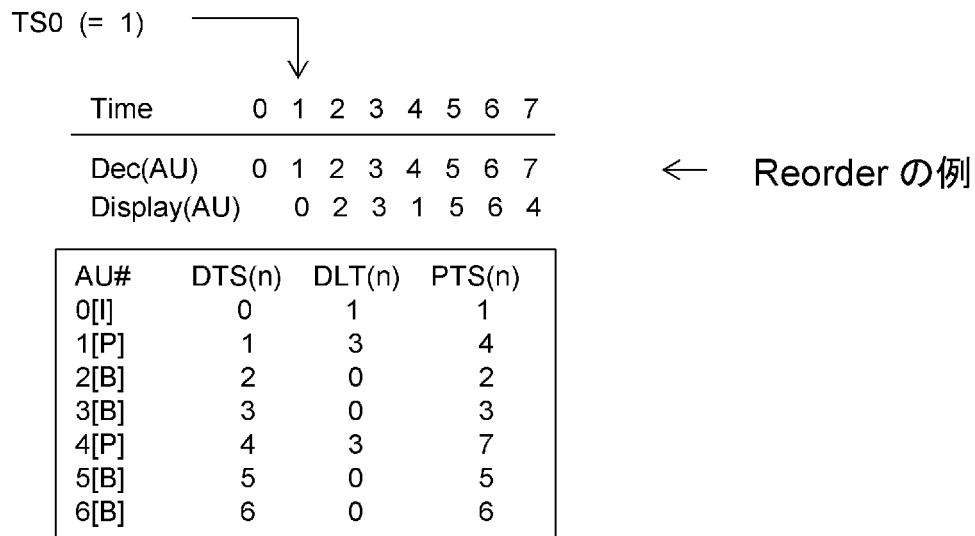


2の補数表現で。Big Endian。

[図12]

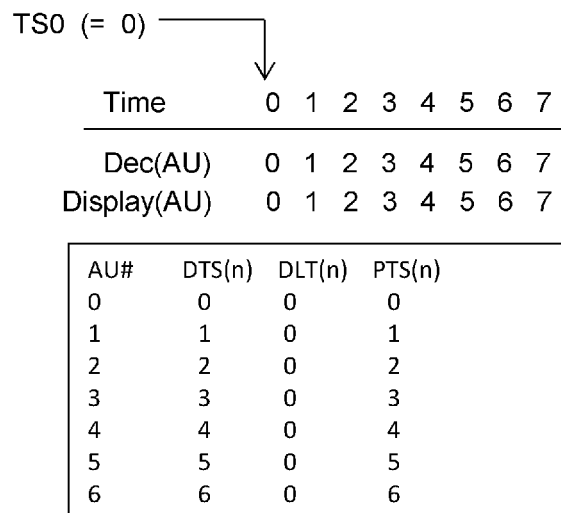


[図13]



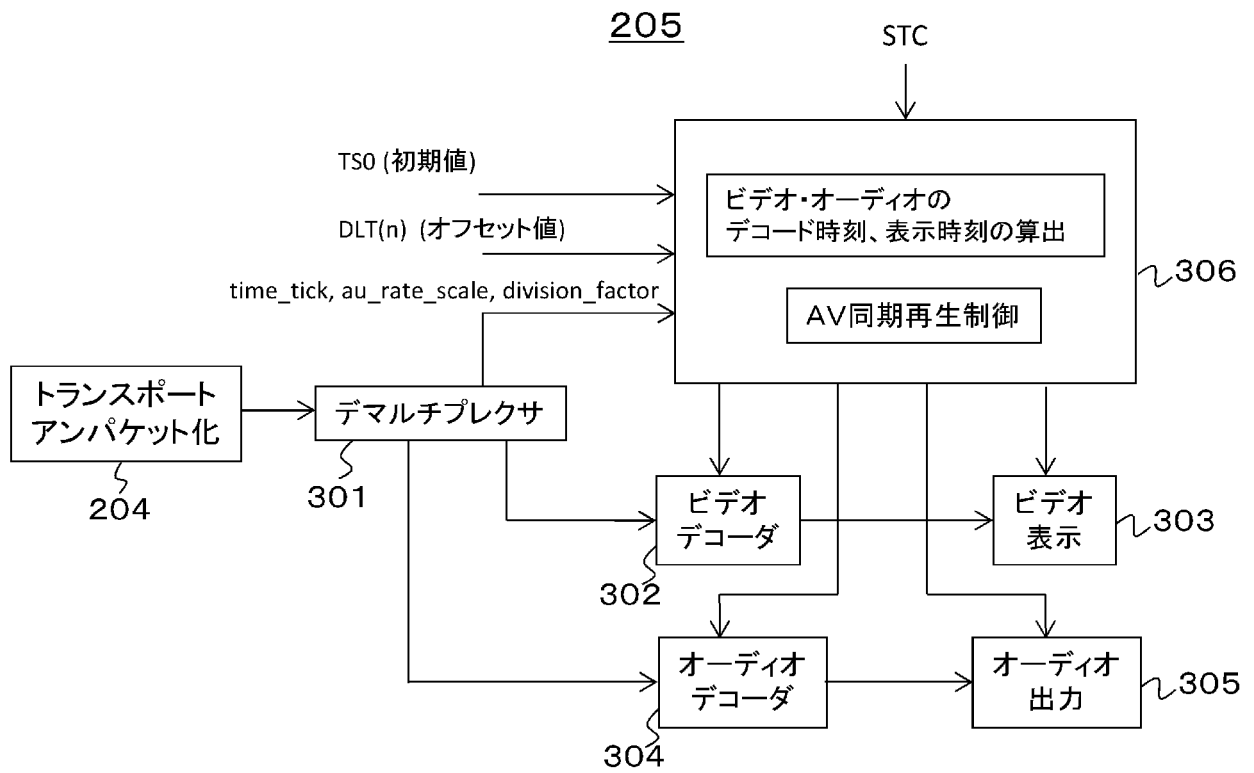
(dlt_time / time_tick = 1 とした場合の例)

[図14]

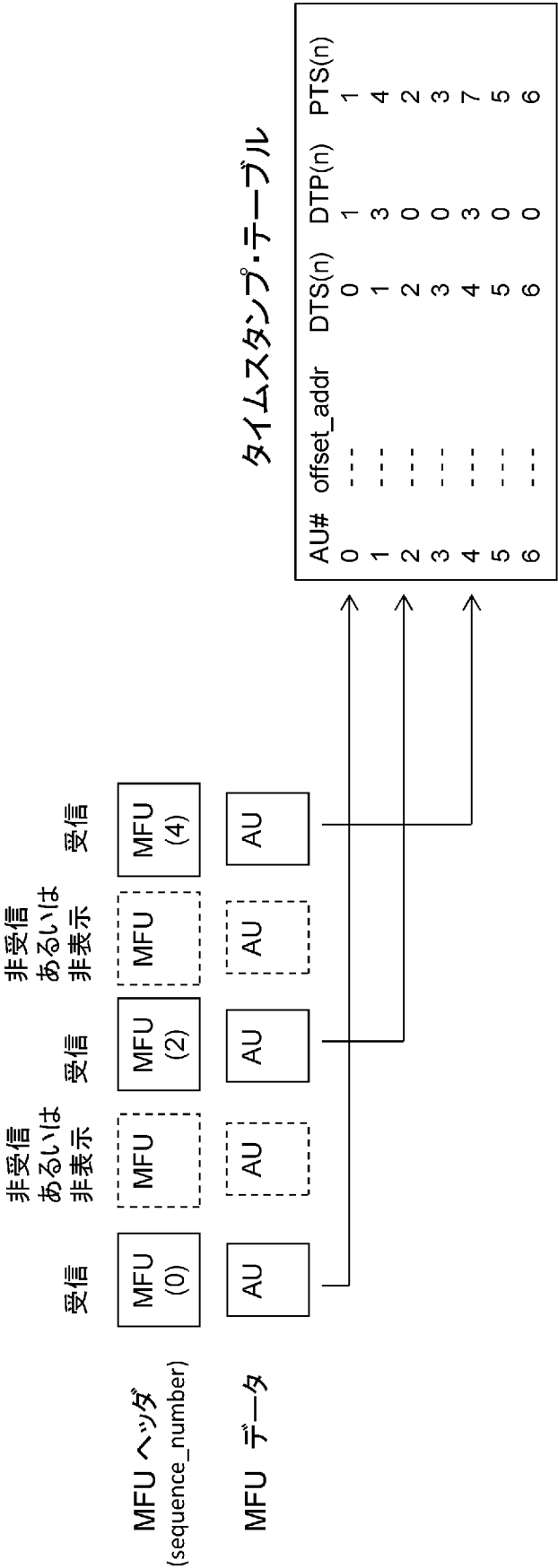


(dlt_time / time_tick = 1 とした場合の例)

[図15]

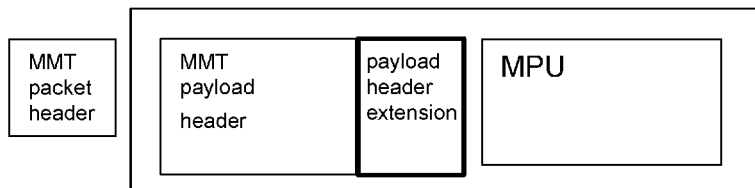


[図17]

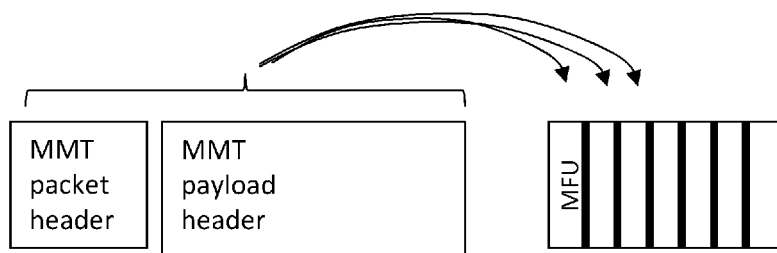


[図18]

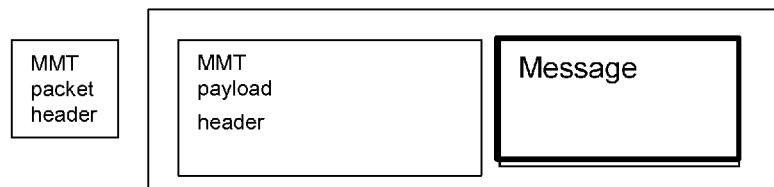
(a) MPU payload header の中で時刻情報を送る場合



(b) MPU payload の中でフラグメント化されたAUごとに時刻情報を送る場合

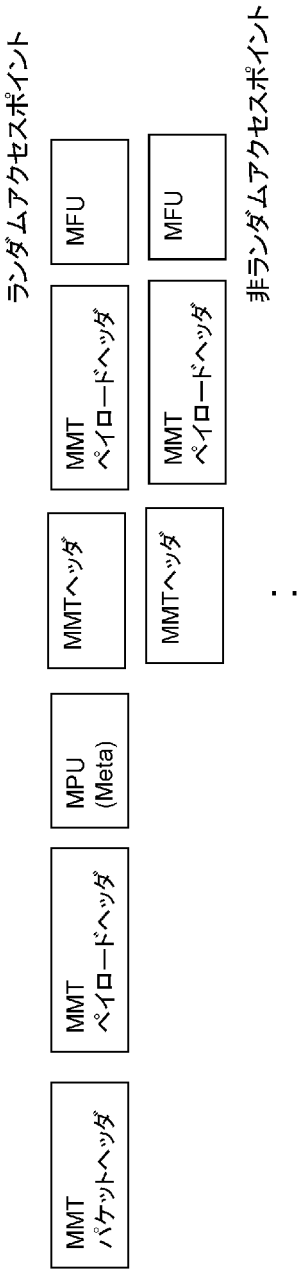


(c) Message で 時刻情報を送る場合

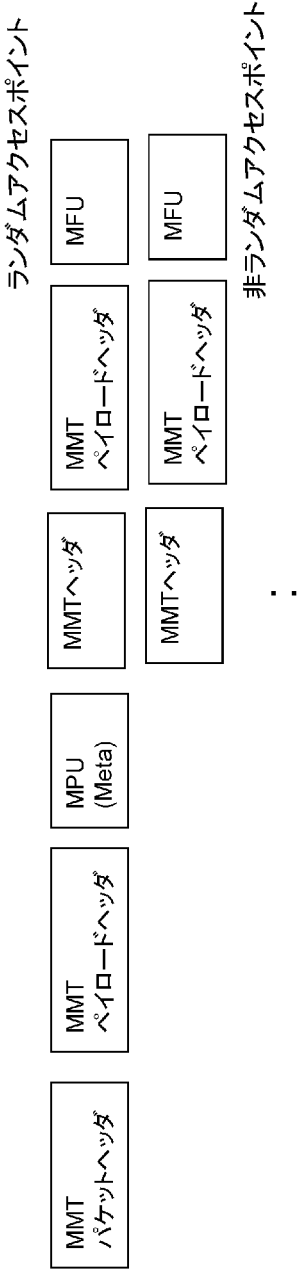


[図19]

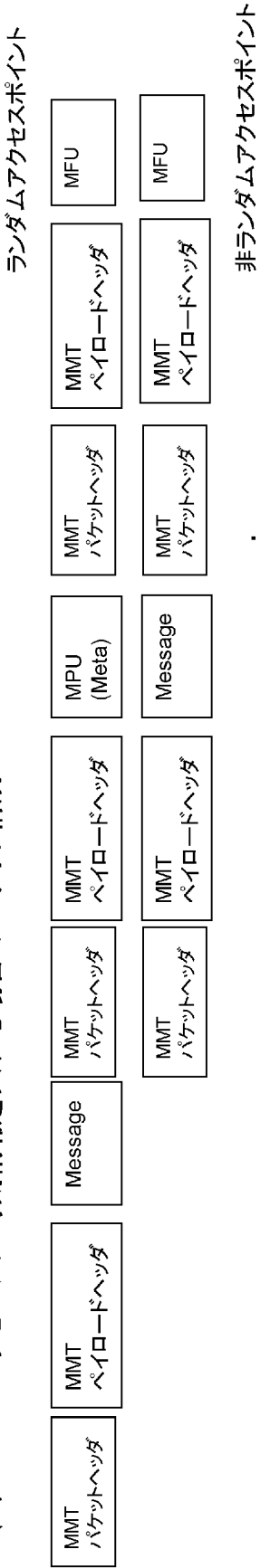
(a) MMT ペイロードヘッダに時刻情報を入れる場合のパケット構成



(b) MFU 内に時刻情報を入れる場合のパケット構成



(c) MMT メッセージに時刻情報を入れる場合のパケット構成



[図20]

MMT packet overall

Syntax	No. of Bits	Format
mmt packet() {		
mmtp_header()		
mmtp_payload() {		
mmtp_payload_header()		
mmtp_payload_data()		
}		
}		

[図21]

MMT packet header

Syntax	No. of Bits	Format
mmtp_header(){		
packet_id	16	uimsbf
packet_sequence_number	32	uimsbf
transmission_timestamp	32	uimsbf
QoS_classifier_flag	1	bslbf
flow_identifier_flag	1	bslbf
private_user_data_flag	1	bslbf
FEC_type	2	bslbf
reserved	3	bslbf
type_of_bitrate	3	bslbf
delay_sensitivity	3	bslbf
reliability_flag	1	bslbf
transmission_priority	3	bslbf
reserved	1	bslbf
flow_label	7	bslbf
extension_flag	1	bslbf
reserved	5	bslbf
private_user_data	16	bslbf
}		

[22]

MMT payload header

Syntax	No. of Bits	Format
mmtp_payload_header(){		
payload_length	16	uimsbf
payload_type	8	bslbf
fragmentation_indicator	2	bslbf
fragment_count_flag	1	bslbf
aggregation_info_flag	1	bslbf
random_access_point_flag	1	bslbf
payload_sequence_number_flag	1	bslbf
header_extension_field_flag	1	bslbf
reserved	1	bslbf
data_offset	8	uimsbf
if(fragment_count_flag == 1)		
fragment_count	8	uimsbf
numDU	8	uimsbf
if (aggregation_info_flag == 1){		
for (l = 0 ; l < numDU ; l++)		
DU_offset	16	uimsbf
}		
if (payload_sequence_number_flag == 1)		
payload_sequence_number	32	uimsbf
if (header_extension_field_flag == 1)		
mmtp_payload_header_extension()		
}		

[23]

MMT payload header extension

Syntax	No. of Bits	Format
mmtp_payload_header_extension() {		
payload_header_extension_type	16	bslbf
payload_header_extension_length	16	uimsbf
asset_type	2	bslbf
delta_sequence_type	1	bslbf
au_count_in_mpu	10	uimsbf
au_rate_scale	3	uimsbf
division_factor	2	bslbf
time_tick	2	bslbf
timestamp_type	1	bslbf
reserved	3	bslbf
timestamp_for_processing	32	uimsbf
if(delta_sequence_type) {		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_variable_length_code	variable	bslbf
trailing_filler()		
} else {		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_fixed_length_code	8	uimsbf
}		
}		

[図24]

semantics	
payload_header_extension_type 0x0001	16bits providing timestamp for processing
payload_header_extension_length	16bits
asset_type	2bits 01 video 10 audio others reserved
time_tick	2bits 01 90000 Hz 精度の値 (LSB) 10 NTP shorttime others reserved
au_rate_scale	3bits if(asset_type == 01) //video 000 3750 24Hz 001 3600 25Hz 010 3000 30Hz 011 1800 50Hz 100 1500 60Hz 101 900 100Hz 110 750 120Hz 111 reserved
	else if (asset_type == 10) // audio 000 1920 48KHz * 1024 001 2089.8 44.1KHz * 1024 010 2880 32KHz * 1024 others reserved
division_factor	2bits if(asset_type == 01) //video 00 1 01 1.001 others reserved else if (asst_type == 10) // audio 00 1 01 2 others reserved
timestamp_type	1bit 1 timestamp is decode_timestamp 0 timestamp is display_timestamp
timestamp_for_processing	32bits if time_tick is "01", timestamp is in 90KHz precision. if time_tick is "10", timestamp is NTP short timestamp specified in RFC5059.
au_count_in_mpu	10bits
delta_sequence_type	1bit 1 delta value in variable length coding 0 delta value in fixed length coding
delta_fixed_length_code	8bits
delta_variable_length_code	variable

[図25]

MMT payload header extension

Syntax	No. of Bits	Format
mmtp_payload_header_extension() {		
payload_header_extension_type	16	bsbf
payload_header_extension_length	16	uimsbf
if (payload_header_extension_type == 0x01)		
presentation_timestamp /* in NTP format*/	32	uimsbf
else if (mfu_payload_header_type == 0x02){		
decoding_timestamp /* in NTP format*/	32	uimsbf
presentation_timestamp /* in NTP format*/	32	uimsbf
}		
else if (mfu_payload_header_type == 0x03)		
presentation_timestamp /* in 90KHz */	32	uimsbf
else if (mfu_payload_header_type == 0x04){		
decoding_timestamp /* in 90KHz */	32	uimsbf
presentation_timestamp /* in 90KHz */	32	uimsbf
}		
else{		
}		
}		

[図26]

semantics

payload_header_extension_type (16bits)

MMT payload header extensionのタイプを示す。

0x01	NTP short time フォーマットの表示タイムスタンプを供給する。
0x02	NTP short time フォーマットの表示タイムスタンプと デコードタイムスタンプとを供給する。
0x03	90KHz 精度での表示タイムスタンプを供給する。
0x04	90KHz 精度での表示タイムスタンプとデコードタイムスタンプとを 供給する。
others	reserved

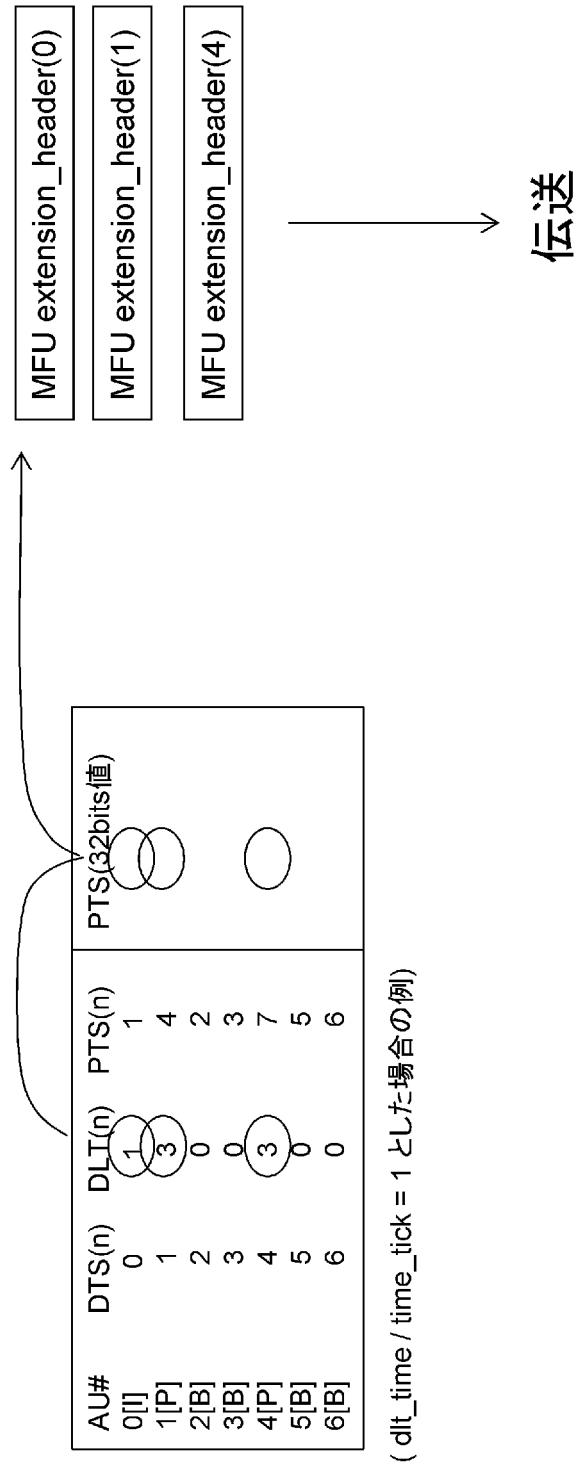
presentation_timestamp (32bits)

表示タイムスタンプの値

decoding_timestamp (32bits)

デコードタイムスタンプの値

[図27]



[図28]

MFU

Syntax	No. of Bits	Format
mfu(){		
mfu_header()		
mfu_media_data()		
}		

[図29]

MFU header

Syntax	No. of Bits	Format
mfu_header(){		
sequence_number	32	uimsbf
if (is_timed) {		
trackref_index	8	uimsbf
sample_number	32	uimsbf
priority	8	bslbf
dependency_counter	8	bslbf
offset	16	uimsbf
length	32	uimsbf
multiLayer_flag	1	bslbf
mfu_header_extension_flag	1	bslbf
reserved	6	bslbf
if(multilayer_flag == 1){		
dependency_id	3	bslbf
depth_flag	1	bslbf
reserved	4	bslbf
temporal_id	4	bslbf
quality_id	4	bslbf
priority_id	6	bslbf
view_id	10	bslbf

continued

[図30]

Continued from previous page

Syntax	No. of Bits	Format
} else {		
layer_id	6	bslbf
temporal_id	3	bslbf
reserved	23	bslbf
}		
if (mfu_header_extension_flag == 1) {		
mfu_header_extension()		
}		
} else {		
item_ID	16	bslbf
}		
}		

[図31]

Semantics

sequence_number	MPU内のMFUのシーケンス番号
trackref_index	メディアトラックの番号
sample_number	当MFUが属するサンプル(アクセスユニット)のmoof box内の順番
priority	MPU内の当MFUの優先度
dependency_counter	当MFUにデコードが依存しているMFUの数
offset	mdat boxからのオフセット
length	MFUの大きさ
multiLayer_flag	マルチレイヤ情報の有無
mfu_header_extension_flag	mfu_extensionの有無
dependency_id	MFU同士の種々の依存関係を示すID群
depth_flag	
temporal_id	
quality_id	
priority_id	
view_id	MFU同士の種々の依存関係を示すID群
layer_id	
Item_ID	ファイルのID

[図32]

Syntax	No. of Bits	Format
mfu_header_extension(){		
mfu_header_extension_type	8	bslbf
mfu_extension_length	8	uimsbf
asset_type	2	bslbf
au_rate_scale	3	uimsbf
division_factor	2	bslbf
time_tick	2	bslbf
timestamp_type	1	bslbf
reserved	6	bslbf
timestamp_for_processing	32	uimsbf
delta_fixed_length_code	8	uimsbf
}		

[図33]

semantics

asset_type	01	2bits	
	10	video	
	others	audio	
		reserved	
Time tick		2bits	
	01	90000 Hz	
	10	NTP shorttime	
	others	reserved	
AU_rate_scale	3bits		
	if(asset_type == 01) //video		
	000	3750	24Hz
	001	3600	25Hz
	010	3000	30Hz
	011	1800	50Hz
	100	1500	60Hz
	101	900	100Hz
	110	750	120Hz
	111	reserved	
	else if (asset_type == 10) // audio		
	000	1920	48KHz * 1024
	001	2089.8	44.1KHz * 1024
	010	2880	32KHz * 1024
	others	reserved	
	others	reserved	

division_factor	2bits	
	if(asset_type == 01) //video	
	00	1
	01	1.001
	others	reserved
	else if (asst_type == 10) // audio	
	00	1
	01	2
timestamp_type	1bit	
	1	timestamp is decode_timestamp
	0	timestamp is display_timestamp
timestamp_for_processing	32bits	
	if time_tick is "01",	
	timestamp is in 90KHz precision.	
	if time_tick is "10",	
	timestamp is NTP short timestamp	
	specified in RFC5059.	
delta_fixed_length_code	8bits	

[図34]

MFU header

Syntax	No. of Bits	Format
mfu_header(){		
sequence_number	32	uimsbf
if (is_timed) {		
trackref_index	8	bslbf
sample_number	32	uimsbf
priority	8	bslbf
dependency_counter	8	bslbf
offset	16	uimsbf
length	32	uimsbf
multiLayer_flag	1	bslbf
timestamp_type	2	bslbf
time_tick	1	bslbf
reserved	4	bslbf
if(multilayer_flag == 1){		
dependency_id	3	bslbf
depth_flag	1	bslbf
reserved	4	bslbf
temporal_id	4	bslbf
quality_id	4	bslbf
priority_id	6	bslbf
view_id	10	bslbf

continued

[図35]

Continued from previous page

Syntax	No. of Bits	Format
} else {		
layer_id	6	bslbf
temporal_id	3	bslbf
reserved	23	bslbf
}		
} else {		
item_ID	16	bslbf
}		
}		

[図36]

semantics

Timestamp_type (2bits)	
MFUペイロードの先頭にタイムスタンプが供給されるかどうかを示す。	
00	タイムスタンプは供給されない。 MFU ペイロードの1stバイトがフラグメントされたMPU ペイロードの先頭バイトと一致する。
01	MFUペイロードの先頭に表示タイムスタンプが供給される。表示タイムスタンプの直後にフラグメントされたMPU ペイロードの先頭バイトが供給される。
10	MFUペイロードの先頭にデコードタイムスタンプと表示タイムスタンプの双方が供給される。
11	タイムスタンプの直後にフラグメントされたMPU ペイロードの先頭バイトが供給される。
reserved	
Time tick (1bit)	
タイムスタンプの値の精度を記述する。	
0	90000 Hz 精度
1	NTP short time フォーマット

[図37]

Syntax	No. of Bits	Format
mfu_payload(){		
if(timestamp_type == 01)		
presentation_timestamp	32	uimbsbf
else if(timestamp_type == 10){		
decoding_timestamp	32	uimbsbf
presentation_timestamp	32	uimbsbf
}		
else{		
}		
payload_data		
}		

semantics

presentation_timestamp (32bits) 表示タイムスタンプの値
decoding_timestamp (32bits) デコードタイムスタンプの値

[38]

Syntax	No. of Bits	Format
Mfu_header(){		
sequence_number	32	uimsbf
if (is_timed) {		
trackref_index	8	bslbf
sample_number	32	uimsbf
priority	8	bslbf
dependency_counter	8	bslbf
offset	16	uimsbf
length	32	uimsbf
multiLayer_flag	1	bslbf
payload_header_flag	1	bslbf
reserved	6	bslbf
if(multilayer_flag == 1){		
dependency_id	3	bslbf
depth_flag	1	bslbf
reserved	4	bslbf
temporal_id	4	bslbf
quality_id	4	bslbf
priority_id	6	bslbf
view_id	10	bslbf

Continued to next page

[図39]

Continued from previous page

Syntax	No. of Bits	Format
} else {		
layer_id	6	bslbf
temporal_id	3	bslbf
reserved	23	bslbf
}		
} else {		
item_ID	16	bslbf
}		
}		

[図40]

semantics

Payload_header_flag (1bit)	
MFUペイロードの先頭に MFU ペイロードヘッダが存在するかどうかを示す。	
0	MFU ペイロードヘッダは存在しない。 MFUペイロードの1st バイトはフラグメントされたMPU ペイロードの先頭バイトと一致する。
1	MFU ペイロードヘッダが存在する。フラグメントされたMPU ペイロードの先頭バイトがMFUペイロードヘッダの直後に供給される。

[図41]

Syntax	No. of Bits	Format
mfu_payload(){		
if (payload_header_flag == 1){		
mfu_payload_header_type	8	bslbf
mfu_payload_header_length	8	uimsbf
if (mfu_payload_header_type == 0x01)		
presentation_timestamp /* in NTP format*/	32	uimsbf
else if (mfu_payload_header_type == 0x02){		
decoding_timestamp /* in NTP format*/	32	uimsbf
presentation_timestamp /* in NTP format*/	32	uimsbf
}		
else if (mfu_payload_header_type == 0x03)		
presentation_timestamp /* in 90KHz */	32	uimsbf
else if (mfu_payload_header_type == 0x04){		
decoding_timestamp /* in 90KHz */	32	uimsbf
presentation_timestamp /* in 90KHz */	32	uimsbf
}		
else{		
}		
}		
payload_data		
}		

[図42]

semantics

mfu_payload_header_type (8bits)	
MFU ペイロードヘッダのタイプを示す。	
0x01	NTP short time フォーマットの表示タイムスタンプを供給する。
0x02	NTP short time フォーマットの表示タイムスタンプとデコードタイムスタンプとを供給する。
0x03	90KHz 精度での表示タイムスタンプを供給する。
0x04	90KHz 精度での表示タイムスタンプとデコードタイムスタンプとを供給する。
others	reserved
mfu_payload_header_length (8bits)	
本エレメント以後のmfu ペイロードヘッダのサイズをbyte数で示す。	
presentation_timestamp (32bits)	
表示タイムスタンプの値	
decoding_timestamp (32bits)	
デコードタイムスタンプの値	

[図43]

Timestamp Message

Syntax	No. of Bits	Format
timestamp_message() {		
message_id	16	
version	8	
length	16	
extension {		
timestamp_table()		
}		
}		

[図44]

Syntax	No. of Bits	Format
Timestamp_table() {		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
packet_id	16	uimsbf
asset_type	2	bslbf
delta_sequence_type	1	bslbf
au_count_in_mpu	10	uimsbf
au_rate_scale	3	uimsbf
division_factor	2	bslbf
time_tick	2	bslbf
timestamp_type	1	bslbf
reserved	3	bslbf
timestamp_for_processing	32	uimsbf
if(delta_sequence_type) {		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_variable_length_code	variable	
trailing_filler()		
} else{		
for (i = 0; i < au_count_in_mpu ; i++)		
delta_fixed_length_code	8	uimsbf
}		
}		

[図45]

Package Access Message

Syntax	No. of Bits	Format
Package_Access_message() {		
message_id	16	uimsbf
version	8	uimsbf
length	32	uimsbf
extension {		
number_of_tables ; N1	8	uimsbf
for (i=0; i < N1; i++) {		
table_id	8	uimsbf
table_version	8	uimsbf
table_length	16	uimsbf
}		
message_payload {		
for (i=0; i < N1; i++) {		
table()		
}		
}		
}		

[図46]

Time	0	1	2	3	4	5	6	7
Dec(AU)	D(0)	D(1)	D(2)	D(3)	D(4)	D(5)	D(6)	D(7)
Display(AU)		R(0)	R(2)	R(3)	R(1)	R(5)	R(6)	R(4)
DLT(AU)	1	3	0	0	3	0	0	3

[図47]

可変長符号テーブル

符号化入力値	可変長符号	符号語長 (bit)
0 ::	1	1
1 ::	010	3
2 ::	011	3
3 ::	00100	5
4 ::	00101	5
5 ::	00110	5
6 ::	00111	5
7 ::	0001000	7
8 ::	0001001	7
9 ::	0001010	7
10 ::	0001011	7
11 ::	0001100	7
12 ::	0001101	7
13 ::	0001110	7
14 ::	0001111	7
15 ::	000010000	9

“0” の連続する数が符号長を示す。

“0”が無ければ、符号長は1. “0”が1つなら符号長は3.

“0”が2つなら、符号長は5. “0”が3つなら符号長は7.

[図48]

符号長の見積もり

GOP length (M) -----	総 bit 長	4bit固定の語長
M = 3 (I P B B) I B B P B_probability: 2/4	$3 + 5 + 2 = 10$	(3+7) bits 16 bits
M = 4 (I P B B B) I B B B P B_probability: 3/5	$3 + 5 + 3 = 11$	(3+8) bits 20bits
M = 5 (I P B B B B) I B B B B P B_probability: 4/6	$3 + 5 + 4 = 12$	(3+9) bits 24bits
M = 6 (I P B B B B B) I B B B B B P B_probability: 5/7	$3 + 5 + 5 = 13$	(3+10) bits 28bits
M = 7 (I P B B B B B B) I B B B B B B P B_probability: 6/8	$3 + 7 + 6 = 16$	(3+13) bits 32bits
M = 8 (I P B B B B B B B) I B B B B B B B P B_probability: 7/9	$3 + 7 + 7 = 17$	(3+14) bits 36bits
M = 15 (I P B B B B B B B B B B B B) I B B B B B B B B B B B B B P B_probability: 14/16	$3 + 9 + 14 = 26$	(3+23) bits 64bits