Telecommunication Standardisation Sector
Study Group 15
Experts Group for Video Coding and Systems in
ATM and Other Network Environments

Document AVC-789

15 May 1995

SOURCE: Stuart Dunstan, Siemens Ltd
TITLE: Proposed additions to H.222.1 acknowledged signalling procedures
PURPOSE: Discussion and Proposal

1. Introduction

This document proposes a number of additions to the H.222.1 acknowledged signalling procedures contained in Annex A of Draft Rec. H.222.1, 31 January 1995. A summary of the proposed additions is as follows;

- END PDU and RELEASE indication signal parameter to indicate source of release
- N(CAUSE) values in the BGREJ PDU indicating reason for logical channel establishment refusal
- a signal indicating clock synchronisation of the system clock in the remote terminal
- M-ERROR indication signal and ERRCODE values
- H.222.1 user plane error codes
- PDU coding in the Transport Stream. Two phases of signalling are required.
- N(REQUEST) parameter for the STATREQ PDU to determine format of information to be returned
- Keep alive function in the ESTABLISHED state

2. Source of END PDU

The END PDU may be sent in response to a RELEASE request signal, or in response to a protocol error. It is proposed that a parameter be included in the END PDU, and in the RELEASE indication signal, identifying the source of the END PDU. (This is included in the SSCOP and may also be useful here).

3. CAUSE parameter values

CAUSE values are associated with the BGREJ PDU and indicate the reason for refusal of the remote CM to accept the establishment request. The value should be transferred to the local CM using the RELEASE indication signal. An initial list of cause values is shown in Table 1.

cause	value
unspecified	0000 0000
multiplexing identifier value already in use	0000 0001
maximum number of logical channels exceeded	0000 0010
resource unavailable	0000 0011

Table 1. An initial list of values for the CAUSE parameter associated with the BGREJ PDU.

The RELEASE request signal should also include a cause value. The N(CAUSE) value would be mapped to it.

4. Remote system clock synchronised signal

A signal indicating clock synchronisation of the remote system clock may be useful. Two possible options are;

- when establishing the PCR_PID, inclusion of a parameter in the BGN PDU that requests the BGAK not be returned until clock synchronisation has been achieved.
- a user to user message

5. ERRCODE values

The M-ERROR indication signal is used between the signalling entity and the layer management at the incoming and outgoing H.222.1 entities. The M-ERROR indication signal has a parameter ERRCODE which indicates the error condition. Tables 2 and 3 show the values that the parameter ERRCODE may take at each of the outgoing and incoming signalling entities respectively.

error type	error code	error condition	state
inappropriate PDU	A	BGAK PDU	0,
	B	BGREJ PDU	0, 2,
	C	ENDAK	2
	D	STAT	0,
unsuccessful retransmission	E	VT(RPT) >=max_RPT	1,3
keep alive function failure[1]	F	no response from peer H.222.1 signalling entity	2

Note 1: This is included if either of the last two proposals in clause 9 are implemented.

Table 2. ERRCODE parameter values at outgoing Signalling Entity.

error type	error code	error condition	state
inappropriate PDU	A	STATREQ	0
CRC32 error	B	PDU discarded (CRC32 error)	

Table 3. ERRCODE parameter values at incoming Signalling Entity.

6. User plane errors

There should be some reporting of receive side user plane H.222.1 errors to a H.222.1 management layer. This connection is depicted in Figure 1 of [1]. The receive H.222.1 demultiplexer for example may be able to detect errors. Table 4 shows some possible error conditions and codes at the H.222.1 receive side user plane entity.

error type	error code	comment
undefined multiplex identifier value	A	a logical channel has not been established for this multiplex identifier value
stream type error	B	the stream type differs from that agreed to at logical channel establishment

Table 4. Possible H.222.1 receive side user plane error conditions.

7. PDU coding

The procedures defined in Annex A of H.222.1 require the PDUs to be coded in the TS and the PS. The PSI/PSM tables have been proposed to code these PDUs. The pdu_type_descriptor() is used to distinguish the different PDU types.

Note that the semantic meaning of the PSI/PSM tables is changed from that of H.222.0 in the BGREJ, BGAK, and ENDAK PDUs, since they relate to the reverse direction of transmission. The tables are in Annex A since they have exactly the required functionality.

7.1. Program Stream

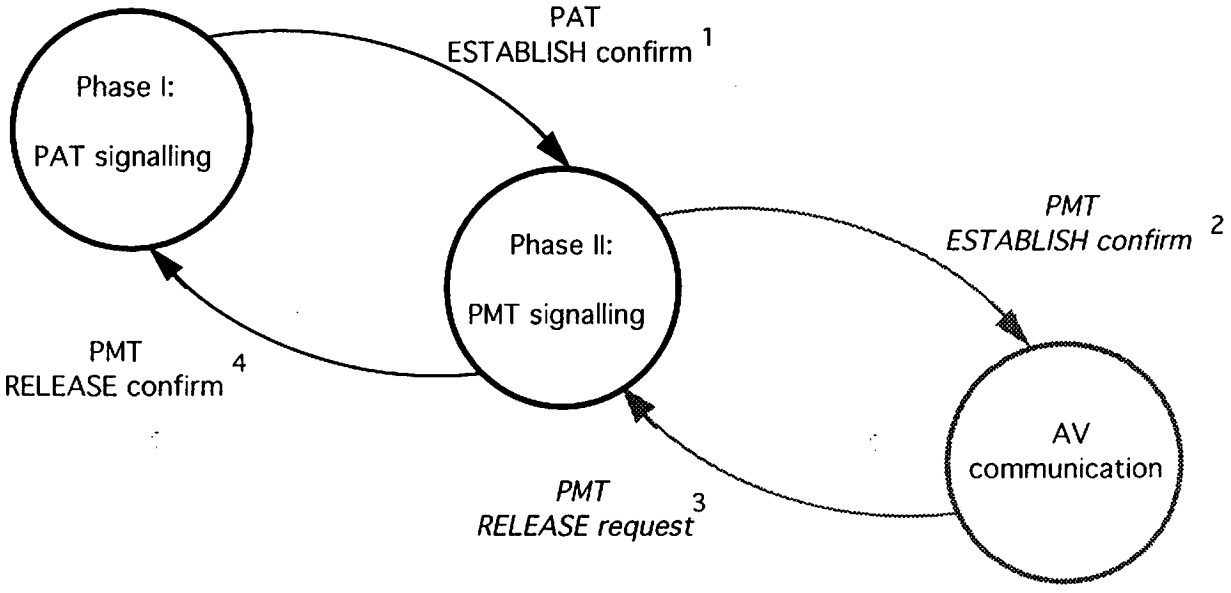
The program_stream_map() is used to code all PDUs in the PS. There appears to be no difficulties in using this syntax; each program_stream_map() sent is one PDU. The CRC32 can be checked and appropriate actions taken. As described in Annex A H.222.1, the current_next_indicator and the program_stream_map_version have no semantic meaning in the acknowledged signalling procedures.

7.2. Transport Stream

The TS requires two phases of signalling. They are;

Phase I:
a bootstrap or metasignalling phase, which uses the program_association_section() to indicate the program_map_PIDs which carrying TS_program_map_section()s

Phase II:
a normal signalling phase which uses the TS_program_map_section() to associate a number of elementary streams, and to convey their attributes.



Notes: 1. Confirmation that the program_map_PID has been understood is recieved. PMT tables may now be sent on that PID.
2. Confirmation that the logical channel has been established has been received. The logical channel may now be used for AV communication.
3. A request to release the logical channel has been given. The logical channel may not be used for AV communications.
4. Confirmation that ALL logical channels have been released has been received. The program_map_PID may now be released if required.

Figure 1. Illustration of two phase acknowledged signalling in Transport Stream.

Annex A H.222.1 currently deals only with the second phase of signalling and not with the bootstrap level.

The acknowledged signalling procedures need to be performed twice; once for the bootstrap phase, Phase I, and once for the normal phase, Phase II. Hence when the Phase I PAT signalling has been acknowledged, the Phase II PMT signalling may begin and logical channels may be established. Audiovisual communication may then begin. The program_map_PID can not be released until the Phase II PMT signalling has released all logical channels. These procedures are illustrated in Figure 1.

7.3. Redundant fields in PSI tables

Annex A PDUs are difficult to code using the H.222.0 program_association_section() syntax, since there is no place to include the pdu_type_descriptor().

It is noted that just as in the PSM table in the PS, some fields in the program_association_section() and the TS_program_map_section() have no semantic meaning in the case of acknowledged signalling procedures, since they relate to the broadcast model of repeated signalling. These fields are shown shaded in Table 5.

Program Association Section			Program Map Section		
program_association_section() {			TS_program_map_section() {		
table_id	8	uimsbf	table_id	8	uimsbf
section_syntax_indicator	1	bslbf	section_syntax_indicator	1	bslbf
'0'	1	bslbf	'0'	1	bslbf
reserved	2	bslbf	reserved	2	bslbf
section_length	12	uimsbf	section_length	12	uimsbf
transport_stream_id	16	uimsbf	program_number	16	uimsbf
reserved	2	bslbf	reserved	2	bslbf
version_number	5	uimsbf	version_number	5	uimsbf
current_next_indicator	1	bslbf	current_next_indicator	1	bslbf
section_number	8	uimsbf	section_number == '0'	8	uimsbf
last_section_number	8	uimsbf	last_section_number == '0'	8	uimsbf
for (i=0; i<N;i++) {			reserved	3	bslbf
program_number	16	uimsbf	PCR_PID	13	uimsbf
reserved	3	bslbf	reserved	4	bslbf
if(program_number == '0') {			program_info_length	12	uimsbf
network_PID	13	uimsbf	for (i=0; i<N; i++) {		
}			descriptor()		
else {			}		
program_map_PID	13	uimsbf	for (i=0;i<N1;i++) {		
}			stream_type	8	uimsbf
}			reserved	3	bslbf
CRC_32	32	rpchof	elementary_PID	13	uimsnf
}			reserved	4	bslbf
			ES_info_length	12	uimsbf
			for (i=0; i<N2; i++) {		
			descriptor()		
			}		
			}		
			CRC_32	32	rpchof
			}		

Table 5. Transport Stream PSI syntax with undefined semantic meaning in case of acknowledged signalling procedures; shown shaded.

7.4. Sections and PAT

The PAT table may be transmitted in multiple sections. Sections may be transmitted in any order.

It is noted that the PMT table may only be transmitted in one section. This is favourable here, since one BGN PDU, for example, is deemed to have been received when that one section is received.

For the purposes of the acknowledged signalling procedures and the PAT, one of the following is required to identify when the PDU is deemed to have been received;

- the last section to be transmitted must be the last section of the table i.e. the final section has section_number == last_section_number
- the PDU is deemed to have been received when one program_association_section() has been received

The latter is appropriate in the case of the BGN PDU i.e acknowledge individual program_association_section()s. section_number and last_section_number are not used at the decoder. However in order not to constrain the number of programs (a section is constrained in length to 1024 bytes), in the case of a STAT PDU the former alternative above may be desirable. These coding rules shown be clarified in H.222.1 Annex A.

Note that one of the justifications for sections was to deal with error robustness in the case of repeated transmission (See H.222.0 Annex C); but procedures here deal with error robustness through acknowledgement, and the justification for sections is not required.

7.5. Phase I program_map_PID establishment

The possibilities for dealing with Phase I program_map_PID establishment are shown in Table 6.

no.	option	characteristics
1	define a fixed program_map_PID e.g 0x0020[1]	- no Phase I signalling required. - does not prohibit multiple programs.
2	signal the one or more program_map_PIDs out of band	- phase I signalling is out of band. - does not prohibit multiple programs.
3	use current program_association_section() in PID=0, table_id=0, allow only one section for PAT, and redefine semantic meaning of some section number fields	- two phases of signalling - uses bytes which have no other use - have changed H.222.0 semantics in PID=0.
4	use a private_section() with program_association_section() like syntax in an ITU-T bootstrap PID e.g. PID=0x0020	- two phases of signalling - have not violated H.222.0 PID=0 semantics

Note 1: It is noted that PID values 0x0010-0x001F have been reserved by DVB.

Table 6. Proposals for dealing with Phase I program_map_PID establishment

With respect to proposal No. 4, Table 7 shows suitable syntax for a Logical Channel Signalling Entity Program Association Section i.e a PDU suitable for Phase I acknowledged signalling. The syntax is similar to program_association_section().

In Table 7a) the section_number field has been changed to pdu_type, and the last_section_number field has been changed to pdu_parameter. An alternative is to insert the pdu_type and pdu_parameter fields following the last_section_number field, where these later fields are required.

Table 7b) shows a descriptor field following the program_map_PID field. The pdu_type_descriptor() would be inserted here. This is attractive since this is the same method as used in the PMT and PSM tables.

Syntax		
LCSE_program_association_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	8	bslbf
pdu_type	8	uimsbf
pdu_parameter	8	uimsbf
for (i=0; i<N;i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if(program_number == '0') {		
network_PID	13	uimsbf
}		
else {		
program_map_PID	13	uimsbf
}		
}		
CRC_32	32	rpchof
}		

a) where section fields not required

Syntax		
LCSE_program_association_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	8	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i=0; i<N;i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if(program_number == '0') {		
network_PID	13	uimsbf
}		
else {		
program_map_PID	13	uimsbf
reserved	4	bslbf
length	12	uimsbf
for (i=0; i<N2; i++){		
descriptor()		
}		
}		
}		
CRC_32	32	rpchof
}		

b) use of a descriptor

Table 7. Suitable syntax for Phase 1 acknowledged signalling PDU. New/renamed fields shown shaded.

8. N(REQUEST) parameter and the STATREQ PDU

The STATUS signals, and the STATREQ and STAT PDUs, are meant to return a multiplexing map and list of descriptors of the incoming signalling entity at the remote terminal. Figure 2 illustrates the procedures defined in Annex A H.222.1.

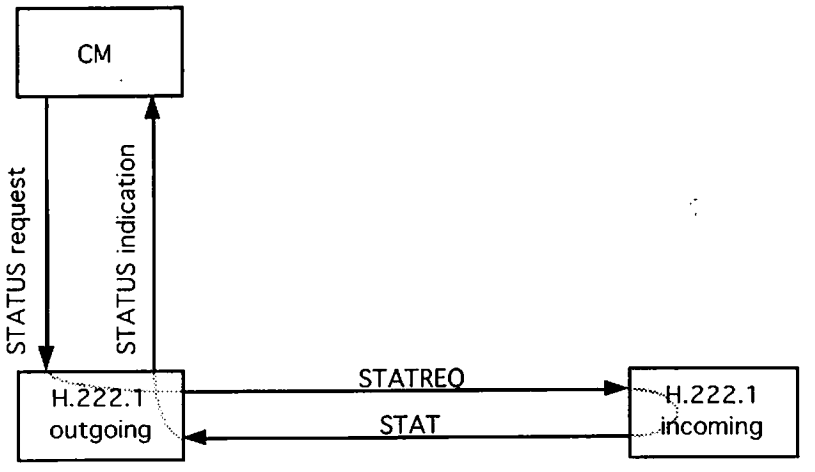


Figure 2. Procedures for status signals and PDUs.

Exactly what information should be returned in the STAT PDU? A parameter N(REQUEST) is proposed that is associated with the STATREQ PDU, which instructs the peer H.222.1 signalling entity on what to return. Table 8 illustrates possible values for the N(REQUEST) parameter

description		coding
programs/elementary streams[1]	descriptors	
only those specified	do not include	0000 00000
only those specified	include	0000 00001[2]
all	do not include	0000 00010
all	include	0000 00011[2]

Note 1: The status signals and PDUs are used in both Phase I and Phase II signalling. In Phase I signalling the PDUs are coded using program_association_section() and information is returned on program_map_PIDs and program numbers. In the case of Phase II signalling the PDUs are coded using TS_program_map_section() and information is returned on elementary stream PID values in use, their stream type, and descriptors, if requested.

2. These codes are not used in the case of Phase I signalling, as their are no descriptors.

Table 8. Coding of N(REQUEST) parameter in STATREQ PDU.

9. Keep alive function

The question is raised as to whether a function to detect a broken connection between peer H.222.1 signalling entities during the ESTABLISHED state is required. The STATUS signals and related PDUs may be suitable for this use. Alternatively new PDUs could be introduced.

A keep alive function may currently be implemented under the control of the outgoing CM as follows;

- the CM sends a STATUS request and starts a timer. If the timer expires before the STATUS indication signal is received, then another STATUS request is sent. If after N times of doing this a STATUS indication has not been received the connection to the peer H.222.1 entity is deemed to have been lost.

The following modifications are possible;

- move the timer and counting function to the outgoing H.222.1 signalling entity. The procedure is then initiated by CM issuing the STATUS request signal.
- leave the status signals and status PDUs unchanged (no timer). Define new timers and PDUs to allow the outgoing H.222.1 to automatically and periodically poll the remote H.222.1 while in the ESTABLISHED state. An M-ERROR indication signal is issued whenever the connection is deemed to have been lost.

The later proposal is advocated. The H.222.1 signalling entities are then responsible for determining their own status.

10. References

[1] AVC-787, "Explanation of current H.222.1 acknowledged signalling procedures", ITU-T Study Group 15 Experts Group, 15 May 1995.

- end -