ITU Telecommunications Standardization Sector
Study Group 15
Experts Group for Video Coding and Systems in
ATM and Other Network Environments

Document: AVC-677
Version 2
October 1994

Source: BT Laboratories

Contact: Norman Kenyon
WT2-4
BT Laboratories
Ipswich, IP5 7RE
UK.

Voice: +44 473 642402
Fax: +44 473 648572
Email: kenyon_n_d@bt-web.bt.co.uk

Question: 3/15

Title: Multimedia System Control: a proposal for H.24P and H.24X

Multimedia System Control: a proposal for H.24P and H.24X

Author: Norman Kenyon

Note: some recipients will have seen a previous version of this document, which was input to the LBC meeting of October and distributed to other ITU SG15 colleagues. It has been revised to take account of comments and local discussions: I have used the change-marker fairly consistently, except in Tables 2 and 3.

At this stage the document is not drafted in Recommendation/Standard form, as the emphasis is placed on the technical proposal [Istvan Sebestyen has kindly provided a template for H.24P, which we will use in the next stage of drafting]. The LBC group seems amenable to the approach proposed, and we await the views of Mr Okubo's group.

Summary

In-band procedures and protocol are needed both for ATM-based multimedia systems and for PSTN videophone/multimedia; there is no obvious reason why these two digital systems, despite their bit-rate difference, should not have essentially the same scheme, at least at the "effective" layer - though this may sit on different transport protocols.

This paper makes a proposal, for further consideration. It draws on the experience of H.242, but seeks to avoid the deficiencies thereof: there are transmit-capabilities as well as receive-caps (H.242 had only the latter); multiple audio and video streams and other data can be declared, and the capsets are shorter. There are no commands (on the assumption that audio, video and data have their own headers), but there are content indicators, and one end-point can control what is sent to it by means of "mode requests".

The proposal is based on variable-length (ILC) messages.

1. General

These notes refer to non-H.221-based multimedia communication. Transmission is in packets or cells whose header specifies the content in terms of medium/mode. Thus the conclusions are applicable to the PSTN-multimedia scene as well as to ATM.

The applicability of the procedures should be as wide as possible, taking into account storage/retrieval, messaging and "distribution" services as well as conversational. There must be maximum independence of the two directions of transmission.

There is a need for a logical channel carrying C&I (control and indications) - here meaning all "in-band" signalling which is not part of user-information streams, including the capability exchanges and mode-setting, etc. In the ATM case, the C&I stream would be included in the H.222.0 multiplex; for PSTN, the control stream is applicable, within the H.22P multiplex. The protocol for this C&I channel is the subject of this paper.

Note: there should be no confusion with the T.120 management system, which is carried within the data stream, and covers different functionalities from those described here - the C&I stream and the T.120-data stream are complementary. I had wondered whether all the proposal here should be embedded within the T.120 stream - an extension of the T.GCC, T.AVC etc. But (i) there may not be a T.120 stream at all, and (b) if there is, it could well be carried on a different VC (we have been advocating this separation, when not using H.221 in a CBR connection).

2. In-band C&I

2.1 Message structure

It is proposed that the capability sets ("capsets") and all other control-and-indication ("C&I") signals be structured as variable-length <u>messages</u>, of ILC (identifier - length - content) type thus:

I	L	C1	C2	//	Cn

- I is a header byte specifying the identity/type of message; the first column in Table 1 lists the identities and the second the byte values, structured for convenience as a 3-bit attribute and 5-bit set of values of each attribute, written (000 to 111)[0-31] in a similar way to H.221, H.230 etc.
- L is one byte specifying the length of whole message - the number of bytes C1.....Cn in the range 0-255

It is assumed that <u>error correction is provided at a lower layer</u>, and so no further mention of this important aspect is made in this paper.

2.2 Capability messages

There must be separate capability messages for ability to receive and transmit, {RX-cap}, {TX-cap}; the {TX-cap} is just a short code if they are identical; capsets must be efficient for commonly occurring capabilities.

The capset must provide for more than one stream of a given medium type to be sent "simultaneously" - of course not at exactly the same time, but multiplexed either within packets/cells or in separate ones. For example, a terminal may declare its ability to receive (or send) two independent H.262 video streams and two independent G.722 audio streams at the same time - something which has not been allowed for in H.221. Thus it is important that the capset structure be clear as to what may be <u>simultaneously received</u> (or transmitted). NB: in H.221, failure to do this gave us trouble with respect to MLP/LSD/HSD, and we have not (so far) tackled the case of two or more audio or video signals, except for Au-ISO + G.728/....

It is proposed that the capsets consist of a string of <u>records</u> each having a first byte beginning with 0 and any further bytes beginning with 1; each record conveys a single <u>stream-capability</u>, which may itself have a number of options. For example, if an equipment can only decode one audio stream at a time it will send only one audio-cap record; if that single audio stream can be decoded as either G.711 (A or μ) or G.728 then these three options will be stated in the one record - see §2.3 for details. This is the normal case in current H.320 terminals. But if the terminal can handle two audio streams <u>simultaneously</u> there will be a record for each. In general, records will be only one or two bytes each.

The whole message will have the form:

I	L	C	D1	//	Dn	A1	//	An	V1	//	Vn	T1	//	Tn

- C is a single byte stating number of data (0-3), audio (0-7) and video (0-7) records;
- D1....Dn is a string of bytes in one or more data records;
- A1....An is a string of bytes in one or more audio records;
- V1....Vn is a string of bytes in one or more video records;
- T1....Tn form one "transfer" record of two or more bytes (may not be needed for PSTN-MM).

A very reasonable multimedia equipment would require only 8 bytes, even if it has a choice of ~~four~~ audio and video codings~~, default H.261/CIF and/or AV.25Y MPEG-2,~~ and a T.120 data stream~~, and a reasonable range of flow rates~~:

I	L	C	D1	A1	V1	T1	T2

A more elaborate case is exemplified in Table 3.

It has been suggested that, as terminals become more software-oriented, capabilities might be interdependent; for example, they might be able to decode pictures faster (lower MPI) when the audio algorithm is less computation-intensive (G.722) and vice versa. The only suggestion I have to cope with this is the transmission of alternative capsets, written {cap~RX}; any number of these could follow the principal capset {RX-cap} - though I don't pose this suggestion with much enthusiasm!

2.3 Capability Values

Detailed values are presented in Table 2.

Data capabilities: as few as possible within this message - ideally just one stream with T.120 protocol, and all else managed from within ~~it~~that.

Audio and video capabilities in Table 2~~1~~ are hopefully self-explanatory. The "default" values are pre-assigned parameters that can be called up without additional bytes in the record.

Transfer (transmission and management) capabilities: this item mops up all the capabilities that are

- not covered by audio, video and data caps
- not treated by other types of message, such as encryption and non-standard facilities

Thus parameters to do with bitrate/throughput, additional channels or VCs, different networks and interworking etc come in here. There is also included in the transfer record the capability to accept downloaded software (see §2.5).

I assume that a maximum throughput is a valid parameter; it may well be~~, but wonder if it is~~ necessary, after call establishment, to express capabilities to adopt a different AAL from that set up, particularly to add a video enhancement VC and/or a data channel VC~~.... Or should this happen during the out-of-band call-setup signalling~~? [See Mr Chia-Chang Li's paper - refers to Network Adaptation Capabilities). We hope to stipulate that all conversational terminals must be able to do H.32Y for interworking with ISDN-connected H.320 terminals, but not all B-ISDN terminals may have such a conversational intention, so an H.32Y-cap seems reasonable.

~~*What options of multiplexing are available in H.222.0/1, which would require setting out in the transfer record? Likewise H.22P?*~~

2.4 Capability basic procedural rules

Prior to receipt of the first {RX-cap} from the remote end-point, transmission is on the basis of call class as determined in call set-up procedures (see §3).

From receipt of the first {RX-cap} until receipt of End_Session (see below), all transmitted signals shall be in the range indicated by the latest {RX-cap} received and the latest {TX-cap} transmitted - that is, a valid principal capset always replaces the previous one; moreover any alternative capsets are also deleted at this point, and must be retransmitted if still valid. An end-point shall store the latest received capsets~~,~~ as the "currently valid capsets".

There must be an acknowlegment for receipt of each: ACK_TX-cap, ACK_RX-cap. If ACK is not received within T1 from the end of transmission of the capset, transmission of the capset is repeated without change.

Requests REQ_TX-cap and REQ_RX-cap may be sent at any time to elicit these details from the remote end-point. This may contain a "cause" bit to be set if the request follows receipt of an errored capset; other causes may be identified...

The above ACKs and REQs are included under the identifier "Capset_ANR" in Table 1: there may be a case for some NACKs also.

2.5 Information-content transmitted ("Modes")

Action related to any capability value received may not be taken until the complete message is received and validated. Such action may be to transmit a corresponding Mode, or send a request for the remote end-point to do so.

In H.221, the multiplex and coding algorithms etc of the content of the framed signal are conveyed under "commands" - effectively these command the receiving demultiplexer, and other blocks, to adopt a certain configuration. Thus they are controls, not indications, because if they are not obeyed the system falls over.

In packet-based multiplexing, the headers of the various streams carry these commands, so they do not need to be included as such in H.24X/P. However there is a case for an (optional?) indication as to the "mode" being transmitted, this being easier to interpret at the receiver than a dynamic inference from a stream of packets. The structure of the Mode message is the same as for capsets (for "capability absent /present" read "OFF/ON").

No associative information is conveyed by the Mode message. If there are more than one audio and/or video stream, it may be necessary to indicate their relationship; this could be managed at a higher level by the application (using MHEG, for example), *but it may be prudent to provide a default (in the absence of such higher-level communication) as an option in the session start-up.*

Expression of wish to be sent particular mode(s)

Each end-point sends a TX-capset, except where the end-point cannot or does not wish to allow control of its transmitted mode (for example, a broadcast multipoint service similar to H.331). If having sent a TX-capset an end-point wishes to revert to the state where remote control of its transmission is not allowed, it shall send the "no_TX-caps" message - see Table 1.

If the currently valid received TX-capset is other than "no_TX-caps", an endpoint may select a mode which it prefers to have transmitted to it, by sending a REQ_Mode message. An end-point whose latest transmission of TX-capset was other than "no_TX-caps" and which is in receipt of REQ_Mode shall comply; if for any reason it is unable to do so, it shall send NACK_REQ_Mode.

The message REQ_Mode has~~takes into account the incoming TX-capset and chooses the preferred Mode - its~~ structure and value definitions~~ are~~ identical to those for Mode indicators, exept for the identifier.

Downloaded software

Under the identifier "Download_SW" in Table 1 there are messages REQ_RX_SW, Ready_for_SW and NACK_RX_SW to manage such a transfer. Of course, T.120 can do this better, but probably we need something for when T.120 is not in use. The facility is intended for server-to-terminal purposes, but could be applicable also for terminal-to-terminal cases. [There is a major problem to be solved here, in that the complexity of software environments may make something as elementary as downloading a piece of program, for mutual understanding by the two end-points of the control signals exhanged, beyond the wit of Man.]

2.6 Other C&I

Table 1 lists message identifiers for Encryption, Maintenance (only loops, so far), audio C&I (such as AIM etc contained in H.230), video C&I (such as Freeze-picture, Fast-update, etc.), non-standard messages and changing or ending a session (see §3.1). Provision is made under Attribute (011) for keyboard, keypad and other user-input control messages. Do any of these need to be shorter than 3 bytes? If so, allocate value (111)[0-31].

3. Procedures

3.1 Point-to-point

The following sequence takes place for establishment of a point-to-point session. For PSTN, points 1-4 are not relevant, and point 5 begins when completion of V.34 modem training is notified. For B-ISDN, the call-control messages are dealt with in Annex 1.

1. Call request: SETUP including "Desired" message ~~(Note 1)~~ - Table 1, identity (000)[0]

2. Destination end-point judges whether the desire can be met, and how; if it can, and a human user is needed, it rings and awaits an answer before sending CONNECT (Note 1).

3. Destination sends CONNECT, including "Acceded" message (Table 1, identity (000)[1]), which contains call class selected and equipment type (Note 1).

4. On sending/receiving CONNECT, both end-points commence in-band procedure according to the Class of communication notified in Acceded (Note 1). For classes other than conversational and speech, branch to Point 50.

5. For conversational and speech services, both ends send a Capability messages (RX-cap and optionally TX-cap) and one audio stream (PCM encoded?).

6. On receipt of {RX-cap} from the remote end-point, a mode switch may be made - other/changed media streams may be transmitted. A Mode indicator message is sent, stating the parameters selected for transmission.

7. An end-point may send a REQ_Mode message if a TX-cap message has been received; an end-point receiving such a message shall comply if it can, but otherwise must respond with NACK_REQ_Mode.

8. If it is desired to change from conversational to another class, a Change _Session message is used.

9. Prior to clearing down, an end-point transmits End_Session, and thereafter transmits no in-band messages, and ignores any received. *[Or should there be an ACK/NACK process here?]*

50. Storage/retrieval/messaging/distribution classes of service: for further study. Should there be provision for a text welcome while higher things are being negotiated? The presumption here is that, just as speech is always available as starting-point for conversational applications, text should always be available as a default starting point for the other classes; an exception is that a fax-only device will go straight into its preamble.

3.2 Point-to-Multipoint working

Where one source is feeding several receivers, it will be unable (and does not try) to respond to any upstream signals, whether of the RX-capset type or requests to transmit in a particular medium/mode, REQ_TX-capset, etc. The source should transmit its TX-capset periodically (eg less than T1 intervals), including only the values actually in use in that session, together with a zero-length RX-capset. (I+L only).

Table 1: C&I and Call-Control Stream messages

Function	Identifier	Length	Content
Attribute (100)			**Capabilities, etc**
RX-cap (principal)	(100) [0]	0 or many	First byte [ddaaavvv], followed by 0-3 data records, 0-7 audio records, 0-7 video records, and finally one transfer record. Each record has a first byte [0bbb bbbb], but this may be followed by more bytes [1bbb bbbb]. A record [0000 0000] is identical to the preceding record.
TX-cap (principal)	(100) [1]	0,1 or many	If the only content byte of TX-capset is [0000 0000], identity to the RX-capset is indicated. Zero length indicates no capabilities in the given direction ("no RX-caps"; "no TX-caps").
cap~RX (alternative)	(100) [2]	0 or many	If the first content byte of two or more is [0000 0000], then the subsequent bytes are not D/A/V records as above but another scheme (f.f.s.).
cap~TX (alternative)	(100) [3]	0,1 or many	Details of data, audio, video cap records are in Table 2.
Capset_ANR	(100) [2]	1 or 2	ACK/NACK/REQ for RX-cap/TX-cap; NACK for REQ_Mode; cause, etc. (details ffs)
Attribute (000)			**Call control**
Desired	(000) [0]	1 or more	[0bbb bbbb] - preferred call class, including AAL type; (mandatory byte) [1bbb bbbb] - "essential facilities" of preferred call class (optional byte) [0bbb bbbb] - second choice call class, including AAL type (optional byte) [0bbb bbbb] etc - optional further classes
Acceded	(000) [1]	1 or 2	[0bbb bbbb] - selected call class (mandatory byte) [0bbb bbbb] - type of equipment answering (terminal, MCU etc) (optional - default is terminal)
Refused	(000) [2]	1 or more	[0000 0000] - Absent (human user), or.... ....[0bbb bbbb] - preferred or only acceptable call class [1bbb bbbb] - "essential facilities" of preferred/only call class (optional byte) [0bbb bbbb] etc - optional further classes acceptable
Attribute (010)			**Various**
Mode_indicator	(010) [0]	one/few	Like capset ~~+ association of streams~~
REQ_Mode	(010) [1]	few?	Same format as Mode~~RX-capset, but indicating~~.
Download_SW	(010) [8]	one/few	Cause/ACK/NACK/Ready; what_next.....
Encryption_SE	(010) [9]	1 to many	First byte = "message identifier" defined in H.233 and H.234; subsequent bytes = "content" defined in H.233/4.
Encryption _IV	(010) [10]	few	Initialization vectors
Picture_C&I	(010) [11]	one/few	Freeze-pic. (VCF), Fast-update (VCU), Source (VIA, VIA2, VIA3, VIS), MIV, VIR etc
Audio_C&I	(010) [12]	one/few	Mute (AIM/AIA), delay-equalise (ACE, cancel-ACE) etc.

Maintenance	(010) [13]	one/few	Audio, Video, Digital loop; loop detection random number.....
Non-standard messages	(010) [14]	4 or more	Country code (Byte 1 as in T.35; byte 2*), manufacturer code (2 bytes*) *assigned nationally
Change_Session	(010) [30]	one/few	Cause/ACK/NACK/Ready; what_next.....
End_Session	(010) [31]	one/few	Cause/ACK/NACK/Ready; what_next.....
Attribute (011)			**User input messages**
Alphanumeric	(011) [0]	any	A string of characters coded according to T.61
Numeric	(011) [1]	any	A string of binary-coded numbers, each in range 0-255
X-Y functions	(011) [2]	4	1-byte function (pen up/down, click, double-click, drag...) + 3-byte XY address (0-4095 each way)

Table 2: Capability Records

Record type	Value	Meaning
Data record	0abc wxyz	First or only byte in the record (Note 1) abcdefg=0000000: all data capabilities cancelled a=1: capability for data channel in separate VC using AAL Type 5, maximum rate as specified in bits defg b=0/1: T.120 capability absent/present c=0/1: reserved (set to 0) wxyz: maximum bitrate according to Table 4
Audio record	0abc defg	First or only byte in the record abcdefg=0000000: all audio capabilities cancelled g=0/1: G.711 A-law absent/present f=0/1: G.711 μ-law absent/present e=0/1: G.722 absent/present d=0/1: G.728 absent/present c=0/1: default MPEG audio absent/present (Note 2) b=0/1: AV.25Y audio absent/present a=0/1: reserved (set to 0)
	100j klmn 1pqr wxyz	two bytes, neither being the first of the record, specifying parameters of MPEG audio j=0 (j=1 and kl=11 reserved) kl=00: Layer I; kl=01: Layers I&II; kl=10: Layers I&II&III mn=00/01/10/11: sampling @ 32k/44.1k/48k/all three p=0/1: asynchronous mode capability absent/present qr=00/01/10/11: correction modes 1/2/3/all three wxyz: maximum bitrate according to Table 4

Video record	0abc defg	First or only byte in the record abcdefg=0000000: all video capabilities cancelled a=1: presence of standard video other than H.261 and H.26P, specified in other bytes of this record; a=0: absence of standard video other than H.261 and H.26P bc=00: absence of both H.261 and H.26P video bc=01: H.261/QCIF default for PSTN-Vp (MPI=4/29.97, max rate = 30 kbit/s) bc=10: H.261/QCIF other default value - max rate = transfer rate? bc=11: H.261/CIF default (includes bc=01,10 defaults) def=000: absence of H.26P video def=001/010/011/100/101: H.26P present with MPI=1/2/3/4/5÷29.97, bitrate up to 30 kbit/s (f.f.s.) g=reserved (set to 0)
	100j klmn	One byte specifies H.261 MPI capability beyond and including any defaults set; if present, this is the second byte in the record j=0: QCIF; j=1: QCIF + CIF kl=00/01/10/11: MPI values 1/2/3/4 (÷29.97) for QCIF mn=00/01/10/11: MPI values 1/2/3/4 (÷29.97) for CIF
	100p wxyz	One or two bytes specify H.261 bitrate capability beyond and including any defaults set; if present, they must follow an MPI capability byte p=0: this byte applies to QCIF and also to CIF if present p=1: this byte applies to QCIF and is followed by another byte applying to CIF wxyz: maximum decoder bitrate according to Table 4.
	1010 klmn 1tuv wxyz	Two bytes specify H.262 Low and/or Main profiles; if H.261 cap bytes present, they follow these klmn: see table 5 tuv wxyz: maximum decoder rate according to table 4
	1011 klmn 1tuv wxyz	Two bytes specify H.262 High-1440 and/or High profiles; if H.261 cap and/or H.262 Low/Main bytes present, they follow these klmn: see table 6 tuv wxyz: maximum decoder rate according to table 4
	11ij klmn	reserved

Transfer record	0abc defg	First byte in the record a=0/1: unable/able to conform to H.32Y (Note 4) b=0/1: unable/able to receive (or send) management s/w c=0/1: unable/able to receive (or send) enhancement video in separate VC using AAL Type 2 d=0/1: unable/able to receive (or send) MPEG-2 Program Mux e=0/1: unable/able to receive (or send) MPEG-2 Transport Mux f=0/1: Interleaver A/B (?) (when AAL Type 1 is in use) g = reserved (set to 0)
	1tuv wxyz	Second byte in the record: value of maximum throughput according to Table 4
	1tuv wxyz	Optional third byte: maximum throughput of video enhancement VC, applicable to case c=1 of first transfer record byte

Note 1: Data rates take default values (f.f.s.) if no other bytes in the record specify more closely.

Note 2: default MPEG~~J.42~~ could be all Layers + all sampling freqs. + all correction modes + asynchronous mode, with maximum rate equal to that given in the transfer record~~; default PSTN/Mobile audio is ffs~~.

~~Note 3: H.261 and H.262 take default values if no other bytes in the record specify more closely: H.261 default is CIF, MPI=1/29.97; H.262 default is f.f.s. (how about one of the DVB values?).~~

Note 4: it would be nice if H.32Y were to include also non-video (audio+data) equipments.

Table 3: **Example of RX-capset and TX-capset**

Identifier	Length	Content	Meaning
(100) 0000	12	(01 010 010)	one data record; two audio; two video
		(0100 1001)	data record: T.120 capable
		(0001 0011)	audio record #1: default MPEG-audio and G.711 A/μ capabilities
		(0000 0011)	audio record #2: G.711 A/μ only
		(0111 0000)	video record #1: H.262, H.261/CIF;
		(1001 0010)	MPI=1/29.97 for QCIF, 3/29.97 for CIF
		(1000 1011)	Max rate for CIF and QCIF: 512 kbit/s
		(1010 0100)	MPEG Main profile A, Low profile B
		(1001 1111)	Maximum MPEG rate ~6 Mbit/s
		(0000 0000)	video record #2 identical to video #1
		(0110 1010)	transfer record: H.32Y capable; able to receive management s/w; no video enhancement; MPEG Program Mux; AAL-1 interleaver B
		(1001 1111)	transfer record: max rate ~6 Mbit/s
(100) 0001	1	(0000 0000)	same TX-capset as RX-capset

Table 4: bitrate coding

Four-bit coding [wxyz] or seven-bit coding [tuv wxyz] available. In the seven -bit case, add the [wxyz] bitrate to the [tuv] bitrate.

000	zero	0000		1000	
001	4 Mbit/s	0001	8 kbit/s	1001	256 kbit/s
010	8 Mbit/s	0010		1010	
011	12 Mbit/s	0011	16 kbit/s	1011	512 kbit/s
100	16 Mbit/s	0100		1100	
101		0101	32 kbit/s	1101	1024 kbit/s
110		0110		1110	
111		0111	64 kbit/s	1111	2048 kbit/s

Table 5: MPEG-2 Main and Low Profile coding

	Main Profile*	Low Profile*		Main Profile*	Low Profile*
0000	none	none	1000	B	C
0001	none	B	1001	C	none
0010	none	C	1010	C	B
0011	A	none	1011	C	C
0100	A	B	1100	E	none
0101	A	C	1101	E	B
0110	B	none	1110	E	C
0111	B	B	1111		

* A = Simple 4:2:0 single-layer; B = Main 4:2:0 single-layer; C = SNR scalable 4:2:0 scalable; E = High 4:2:2 scalable

Table 6: MPEG-2 High-1440 and High Profiles

	High-1440 Profile*	High Profile*		Main Profile*	Low Profile*
0000	none	none	1000	E	none
0001	none	B	1001	E	B
0010	none	D	1010	E	D
0011	none	E	1011	E	E
0100	B	none	1100		
0101	B	B	1101		
0110	B	D	1110		
0111	B	E	1111		

* B = Main 4:2:0 single-layer; D = spatially scalable 4:2:0 scalable; E = High 4:2:2 scalable

Annex: Call Control for ATM

[NB: this section is not targetted at the PSTN case!]

I assume there shall always be a clear distinction between those messages to do with call set-up, clear down, hold, transfer, etc and those dealing with in-band matters, even if the two types of message have the same format and are carried on the same logical channel. In Table 1 I have used (000) attribute for call control, and have left (001) free also for this - if it turns out that call control is in a different logical channel, there is no need to share the attributes between in-band and call-control functions.

We need some kind of "hand-over" point between phases: I use here the CONNECT message as handing over to in-band at session start, and End_Session handing back to call control.

Call-setup exchanges need not involve the full capset exchanges [reasons: arguably too much detail involved, not relevant to the setup itself; in any case, the provision for capset changes during a session is necessary, so it may as well be used in the same way at the start of the session]. Enough must be included to:

(a) ensure that the calling user's wishes with respect to "fallback" are respected;

(b) avoid setting up the call if the low level or total lack of interworking is such that the user would not want it set up;

(c) ensure that if the call is set up the (default) transmitted signals during intialization (capset exchange) are receivable.

For these purposes it is proposed that a number of call classes be identified, each having its own initialization procedure (though I would hope the procedures turn out to be as similar as possible).

A.1 Originating (calling) end

In the call request message, one, two or more of these classes may be listed, in order of preference. It may be that, under special circumstances, a user originating a call does not want it to go through unless the destination has a more specific set of capabilities; this could be conveyed by one or more "essential facilities" bytes as an optional extension field to that call class, within the call request message.

Table A1: Call Classes

Class	**Code**	**Caller's intention** (as preferred class)	**Caller's intention** (as fallback)
Conversational (persons)	Cv	Would like to speak to a person, not a machine	Prepared to speak to a person if the expected machine does not come on line
Speech (machines)	Sp	Would like to communicate by speech with a server or answering machine	Prepared to speak to a machine if (eg) called person does not answer
Storage/retrieval	Tm	Would like to communicate by text or higher with a server or answering machine	Prepared to communicate by text or higher, if preferred class unavailable
Facsimile	Fx	Would like to communicate by fax	Prepared to send a fax, if preferred class unavailable
Messaging	Mg	Would like to communicate with message centre, by text or higher	Prepared to be connected to a text message centre, if preferred class unavailable
Multimedia	Mm	Would like to communicate: can do any class; class to be chosen by destination end-point	Prepared to communicate in any other class, chosen by the destination end-point, if preferred class unavailable

It is not suggested that a user be required to take on complex decisions every time he wants to make a call! In fact, using Mm alone will take care of most PC/workstation situations: calling a number which is a person's terminal, the latter will answer (the person, or his answering machine...); if the destination is a server of some kind it will also answer appropriately. A simple videophone would normally send {(1) Cv; (2) Sp}, but if the user for some reason wanted to preclude the call maturing if there is no video capability at the destination then he would have to insert Video in the "essential facilities" optional field after the Cv code.

If a call request were being made by a server, it should use one or more of the limited codes Tm, Mg, Sp or Fx, according to its I/O media (unless it is actually able to carry on a conversation and a human destination user would be prepared for this!).

A.2 Destination (called) end

The destination end-point considers these classes in turn, responding to the first whose minimum capability (including any "essential facilities") it can match. In Table A24, the capabilities listed for an answering destination are the minimum necessary to justify answering - any higher capabilities will be indicated subsequently in the capset. There is a distinction between text+ (text and the usual other character-based blobs...) capabilities for databases and for messaging: at a terminal the difference may not matter, but a database server is not necessarily able to handle messaging, and *vice versa*.

Table A2: Call Classes answered

Class	Code	Answering destination	Non-answering destination
Conversational (persons)	Cv	has user ready to speak	has no user or no speech capability
Speech (machines)	Sp	has speech machine	has no speech machine
Storage/retrieval	Tm	has text+ for database	has no text+ for database
Facsimile	Fx	has fax capability	has no fax capability
Messaging	Mg	has text messaging	has no text messaging
Multimedia	Mm	has speech or text or fax	has none of the above

When answering, the destination returns CONNECT, including an Acceded message containing the code for the call class which has been selected (if the SETUP contained only one class code in its Desire message, this Acceded could be omitted?). Of course, Mm should not be used here. Initialization follows:

- Cv and Sp use the same procedure - see "Procedures" 1-4 and 5e-9e
- Tm and Mg could well use the same procedure, ffs - see "Procedures" 1-4 and 50t-
- Fax probably has its own procedure, ffs - see "Procedures" 1-4 and 50t-

If an end-point does not answer it should if possible send a REJECT, including a Refused message containing the reason; this may either be "Absent" (alerted but no human user answered), or in the form of a string (similar to a Desired message) of classes/essential_facilities which it would have answered.

In the (unlikely) event that a destination user does not want to accept calls unless there are "essential facilities" in addition to those (speech, text, fax) implied by the class code, the terminal could be set to reject; then the calling end-point would detect the presence of this restriction in the Refused message, and make another call request using the same, if it can.

A.3 Handover

End_Session is effectively an "in-band" signal which hands over control to the "out-of-band" signalling process.