Telecommunications Standardisation sector
Study Group 15
Experts Group for Video Coding and Systems
in ATM and Other Network environments

Document AVC - 742
Version 1
24-27 January, Tokyo

Source: BT
Title: H.222.1 issues
Purpose: Information & Proposal

1. Introduction

This contribution addresses some issues related to H.222.1. These include bit error correction and timing recovery. Bit error correction is found to be necessary, and it is concluded that timing recovery functionality should be included in H.222.1. A framing structure to perform these functions is proposed.

A number of assumptions are made in order to investigate a method for bit error correction. Firstly, this study only considers real-time conversational services using H.222.0/MPEG-2 (System) multiplex which preclude the use of data re-transmission and require minimal delay. The delay constraint also discounts the use of interleaving, which could otherwise reduce the effect of impulse noise.

2. Bit error correction: is it needed?

ITU-T SG15 Q2 Experts Group has assumed the following bit error conditions, as stated in [1].

	Worst case	Average case	Best case
Bit error rate	10^{-5}	10^{-7}	10^{-9}

It is stated in [2] that bit error correction is needed, as with this bit error performance for a 6 Mbit/s MPEG-2 signal, one error will occur every 2.5 minutes, even in the best case.

3. Bit error correction: how much FEC needed?

To perform calculations on the amount of FEC needed, a FEC block size of 255 octets is assumed as in [2] and by the European Digital Video Broadcast project.

The table below shows estimates of the probability of errors in these blocks, assuming that all bit errors are independent.

Bit error rate	10^{-5} (Worst case)	10^{-7} (Average case)	10^{-9} (Best case)
P(no errors in block)	0.979807	0.999796	0.999998
P(one error in block)	0.019988	0.000204	2.04×10^{-6}
P(two errors in block)	0.000204	2.08×10^{-8}	2.08×10^{-12}

The mean time between uncorrectable errors with no error correction, one bit per block and two bits per block bit error correcting capability are shown below, measured in seconds assuming a bit rate of 6 Mbit/s. It should be noted that the period between errors for the worst case condition using two bit/block FEC is about 4 minutes. However the two bit/block FEC easily meets the normal case criteria and is reasonably low in overhead.

Bit error rate	10^{-5} (Worst case)	10^{-7} (Average case)	10^{-9} (Best case)
No FEC	0.016837	1.666837	166.6668
One bit/block FEC	1.657132	16350.17	$1.68 \times 10^{+8}$
Two bits/block FEC	244.3488	$2.55 \times 10^{+8}$	Very long time

It is clear that the no FEC case is unacceptable: this is in agreement with [2]. The capability of correcting one bit per block seems to be acceptable in all but the worst case where an error occurs every 1.6 secs. The capability of correcting two bits per block gives an improvement for all bit error rates over the no FEC best case.

Considering the desire to have a small percentage of bits for FEC (quite large FEC blocks) and the wide-spread use of Reed Solomon (RS) codes for FEC, we propose that RS FEC be used.

In general, RS codes require one FEC symbol (octet in the above case) for each required piece of information about errored symbols. To correct a symbol, it is necessary to know the position of the errored symbol and how to correct it, so two FEC symbols are needed.

The use of two FEC octets would allow one errored octet to be corrected, that is, up to eight consecutive bit errors in one octet.

If correction of short bursts of errors is required, then four octets of FEC would be needed, as the bursts of errors would not in general be aligned with octets.

Although we have no strong preference, we suggest the selection of a RS FEC frame of length 255 octets, and having four octets of FEC. This allows two random bit errors, or any two octets, or any burst of nine bit errors to be corrected.

4. Bit error correction: where should it be?

Bit error correction should be included either in the AAL or in H.222.1.

ITU-T SG15 Experts Group has concluded that for conversational services cell loss correction is not as critical as bit error correction. As AAL1 can only provide bit error correction by using the optional interleaver, it would seem necessary to provide some other means of bit error correction.

AAL5 provides CRC, but not bit error correction.

The need for bit error correction, and the precise requirements of FEC, are specific to ITU-T H.32x. Given the fact that the existing AALs (1 and 5) do not provide suitable FEC, and that the ITU-T SG15 Experts Group has agreed to seek generic AALs, it would appear that FEC belongs to H.222.1, and not to the AAL layer.

5. Bit error correction: a multiplex for H.222.1

ITU-T SG15 Experts Group considered a number of scenarios for error correction using AAL 1 and AAL 5. Given the above, the best solution is of the type 2-1 of [2], namely using Reed Solomon FEC with a sync-word at the beginning of each error correcting frame as shown below:

```
+----------------------------------------+
| RS   | MPEG  | RS  | RS   | MPEG  | RS  |     PS = Program Stream
| Sync | PS/TS | FEC | Sync | PS/TS | FEC |     TS = Transport Stream
+----------------------------------------+     RS = Reed Solomon
|<----- N bytes ---->|
```

6. Jitter removal and timing recovery in H.222.1

Functionality for jitter removal and timing recovery could be located in H.222.1. Currently, AAL5 does not support this functionality, and AAL1 only provides it for constant bit rate services.

Considering the ITU-T SG15 Experts Group (Grimstad) agreement to seek generic AALs, and that jitter removal and timing recovery may not be considered to be generic, these functions may best be placed in H.222.1.

The requirements for these functions were considered in [3] where it was concluded that they could be provided by the inclusion of timestamps, relative to a common clock, and values of cell (or bit) rate in the transmitted bitstream. The cell or bit rate field was used to indicate bit rate changes for variable bit rate transmission (i.e. piece-wise constant bit rate transmission).

H.222.1 could provide jitter removal and timing recovery by inclusion of timestamps and/or bit rates.

```
+-------------------------------------------------------------------+
| RS   |       |          |             |              | MPEG  | RS  |
| Sync | Flags | <Length> | <Timestamp> | < BitRate >  | PS/TS | FEC |
+-------------------------------------------------------------------+
|<---------------------------N bytes------------------------------>|
```

The definition of Timestamp and BitRate could be the same as in [3]:

The Timestamp is two octets long. The first two bits are reserved for future use. The remaining 14 bits contain a sample of a counter, Ct, driven by a clock, fnx, where fnx = fn / x, and fn is the network clock frequency, e.g. 155.52MHz, and x is an integer that is a power of two. Note: this is the same mechanism as used in AAL1 for SRTS, except that in this case the full timestamp is transmitted, while in AAL1 only the residual part is transmitted. When fn = 155.52 MHz, and x = 128, fnx = 1.22 MHz, and the counter will wrap around to zero after 13.48 ms.

The BitRate is two octets long. The coding of BitRate field is defined on a logarithmic scale, so that the percentage error due to coding of the actual value into a finite length field is independent of the magnitude of the bit rate. This also allows a large range of bit rates to be encoded in a 16 bit field. A value of zero is defined to mean that the bit rate is zero and no more PS/TS bits will be transmitted until a non-zero value of BitRate is received.

The Length field is one octet long. This is required for the case of the bit rate going to zero: the data may stop in the middle of the FEC block, and the Length field indicates how many octets contain valid data and how many are stuffing.

Three single bit Flags are needed to indicate the presence of the three fields defined above, assuming that their presence is independent - this may not need to be the case. To retain octet alignment and have respect for efficiency, these flags and the sync field are put into one octet. This leaves five bits for the Sync field: the choice of these five bits is arbitrary.

The optional fields need not be in every FEC frame, but a minimum frequency does need to be defined: 10Hz would be in alignment with the minimum frequency of PCRs in MPEG.

7. The value of N

There are three obvious possible choices for the value of N:

1) N=255. Reed Solomon codes usually have length $2^m - 1$.

2) N=8*47=376. This would align the FEC frames to the AAL1 and AAL5 PDUs (assuming that AAL5 PDUs are made from 8 cells).

3) N=188+X. The X would be such that 188 bytes of TS/PS are present in each FEC frame.

Choice 1 would maximise the efficiency of FEC.

Choice 2 couples the FEC block size to the AAL PDU size. There is no obvious gain from doing this.

Choice 3 couples the FEC block size to the TS packet size. This would confine uncorrectable FEC blocks to single TS packets. It would also couple the timing information, Timestamp and BitRate, to TS packets. This type of coupling requires the overhead per FEC block to be constant, that is, the optional fields would need to be present in every block. This seems unnecessary and wasteful.

It appears that choice 1 is the most suitable.

Assuming that the optional fields are only included infrequently and have no effect on efficiency, the overhead of the above scheme is (1 sync octet + 4 FEC octets) / 255 octets = 1.96%. Note: the overhead with H.261 is 20 / 512 bits = 3.9%.

8. Conclusion

It has been shown that error correction is necessary. The following proposal was made for a FEC block structure providing Reed Solomon FEC and jitter removal and timebase recovery functionality.

```
+-------------------------------------------------------------------+
| RS   |       |          |             |            | MPEG  | RS  |
| Sync | Flags | <Length> | <Timestamp> | < BitRate > | PS/TS | FEC |
+-------------------------------------------------------------------+
|<---1 byte--->|<-1 byte->|<--2 bytes-->|<--2 bytes-->|       |<-4->|
|<---------------------------255 bytes----------------------------->|
```


References

[1] ITU-T SG15 AVC-635: ATM Performance Assumptions, AVC Experts Group.

[2] ITU-T SG15 AVC-657: Consideration on the error correction functionality in H.222.1/AAL, Japan.

[3] ATM Forum / 94-0780 | ITU-T SG15 AVC-686: An AAL for class B services, Mike Nilsson, Dave Beaumont, Geoff Morrison, BT.

END