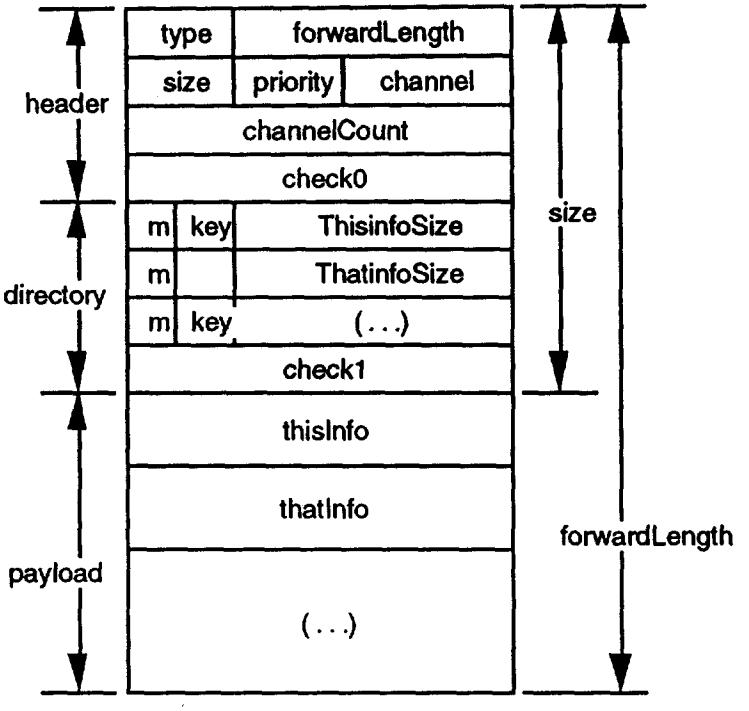
match payload information with associated data such as error correction or encryption information. For example, heavily error protected executable code could be downloaded. In general, the use of the universal header provides the possiblity of accessing a wide variety of algorithms and syntaxes that are rigorously registered and so can be unambiguosly labeled. Apple believes that any system using the SMPTE header should be limited only by hardware and software, and not by the flexibilty of the bitstream specification.



Sample UniPack Packet Format: Complete Packet

UniPack & MPEG2

The issue that we would like to explore is how SMPTE header structures might be usefully included in compliant MPEG-2 streams.

Why?

Such a stream could be playable **both** by a dedicated MPEG-2 player, **and** by a SMPTE-savvy system. An MPEG player would perform as it would with other MPEG streams since it would just ignore the SMPTE header. However, players that understand the SMPTE header would have 2 main advantages.

The first advantage is the variety of data structures it can carry. This would permit the use of any number of registered error correction mechanisms, timing specifications, titling formats, copyright information, etc...

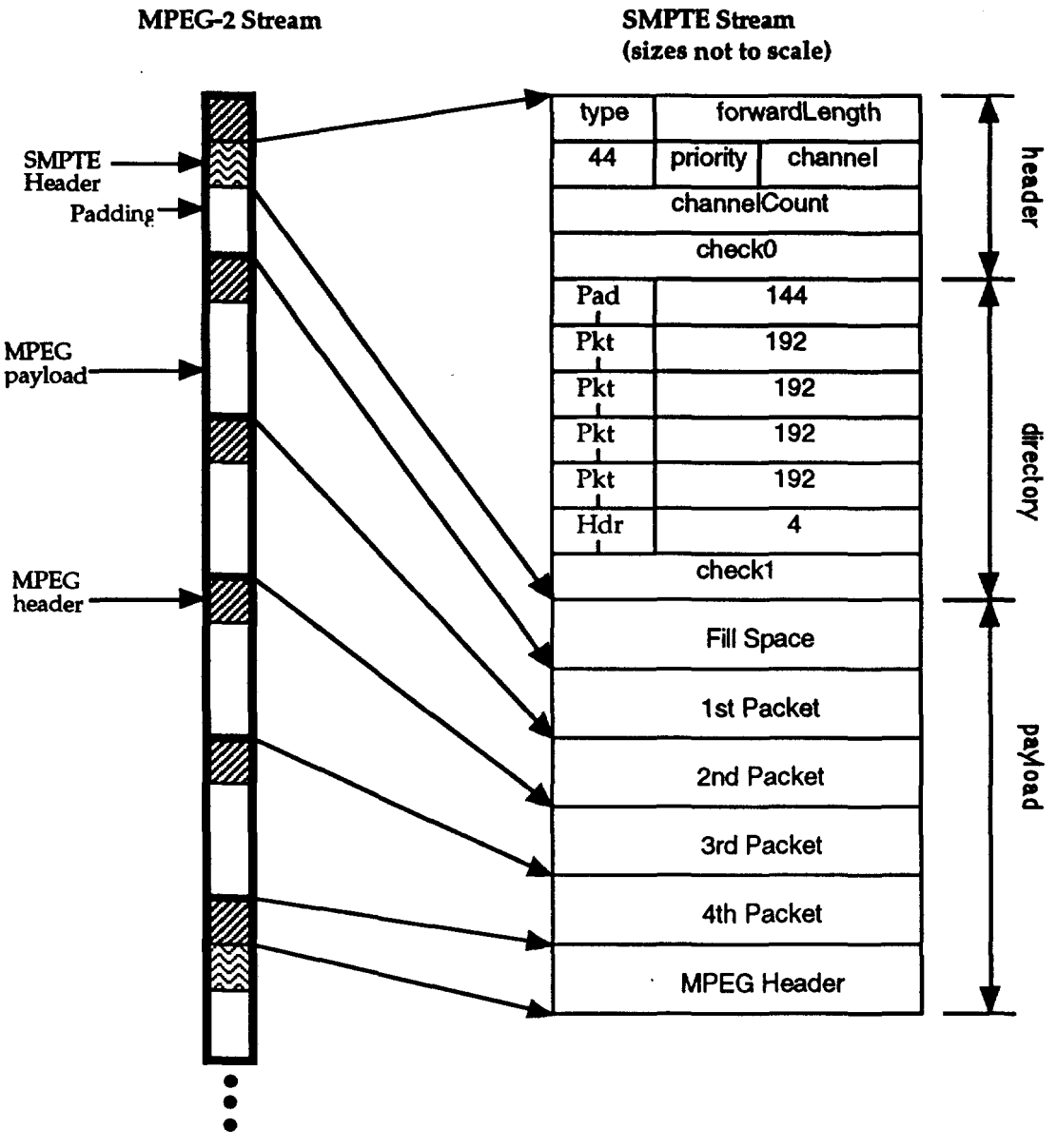
The other main advantage is that the SMPTE header includes information that allows a stream to identify the standard that was used to produce it. This makes it possible for a player to recognize and decode a wide variety of standards. As long as a standard has been registered with a recognized standards body, it will have a unique identifier that will enable it to be unambiguously identified.

Example

One way in which a SMPTE header might be included in an MPEG stream would be to include 2 bits in the MPEG header that would indicate that the packet's payload is universal header information for the identified channel as opposed to coded data. Two bits would be used to distinguish between the first header-carrying transport packet and subsequent ones. Please see the diagram on next page.

Another way of including a universal header might be to include the header using a mechanism similar to the manner in which private data is included in the current working draft.

Another method might be to put the header in the as-yet-undefined PID0 packets.



Example MPEG-2 transport stream with imbedded SMPTE header

Both diagrams are different views of the same bitstream. The one on the left shows an MPEG-2 bitstream that has a packet containing a SMPTE header referencing several MPEG packets. The diagram on the right shows a SMPTE packet whose payload consists of padding (for alignment purposes), packets, and the MPEG header data that precedes the next SMPTE header.

MPEG-2 suggestions

We strongly urge that MPEG-2 make specific provisions for including SMPTE universal headers in all of the system syntax specifications. We also urge that the current byte-alignment between header data and payload data be maintained, and further suggest that quadword alignment be considered if possible.

We believe that the universal header should be identified by using a mechanism OTHER THAN private data. We urge that the mechanism by which this is done entail as little overhead as possible.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG93 / ___
September 1993

Source: Apple Computer - Gary Demos, David Singer, Joseph Stampleman
Title: MPEG-2 Transport Stream Mapping and Adaptation Layer Fields for ATM
Committe: Systems

Introduction

The choice of the AAL that is used to carry MPEG-2 data will greatly impact the quality of the delivered services. It is not anticipated that there will necessarily be discussions on this topic at the Brussels meeting. This document is intended to present some ideas on this topic so that whatever discussions do take place and then carry over to the appropriate forums might consider these proposals.

This document supports an AAL similar to that proposed by Sandy MacInnis & Randy Sharp (5 bits sequence & 3 bits checksum in each ATM packet). The only suggested change would be to reduce the number of bits in the sequence counter to 4 and add a bit that would be used to signal packets carrying optional extra Reed-Solomon information.

Sequence Counters & Checksums

The 188 byte size of the transport stream packets is meant to allow 4 bytes to be used with 4 ATM cells of 48 bytes each. It appears that no existing Adaptation Layers of ATM (AAL 1,3,4,&5) are optimal for this purpose. The mapping of the 188 bytes of MPEG-2 transport packets onto the 192 bytes of 4 ATM cells should be considered carefully in order to provide tools for appropriate reconstruction of MPEG-2 data in the presence of the kinds of losses which may occur on an ATM connection.

The kinds of losses which might be expected on an ATM connection are as follows:

1) If one or more bits are dropped in an ATM header (five bytes preceding the 48 byte cell payload), then the ATM 48-byte cell will not be received at the destination. The cell will be missing.

2) Congestion can also result in missing cells.

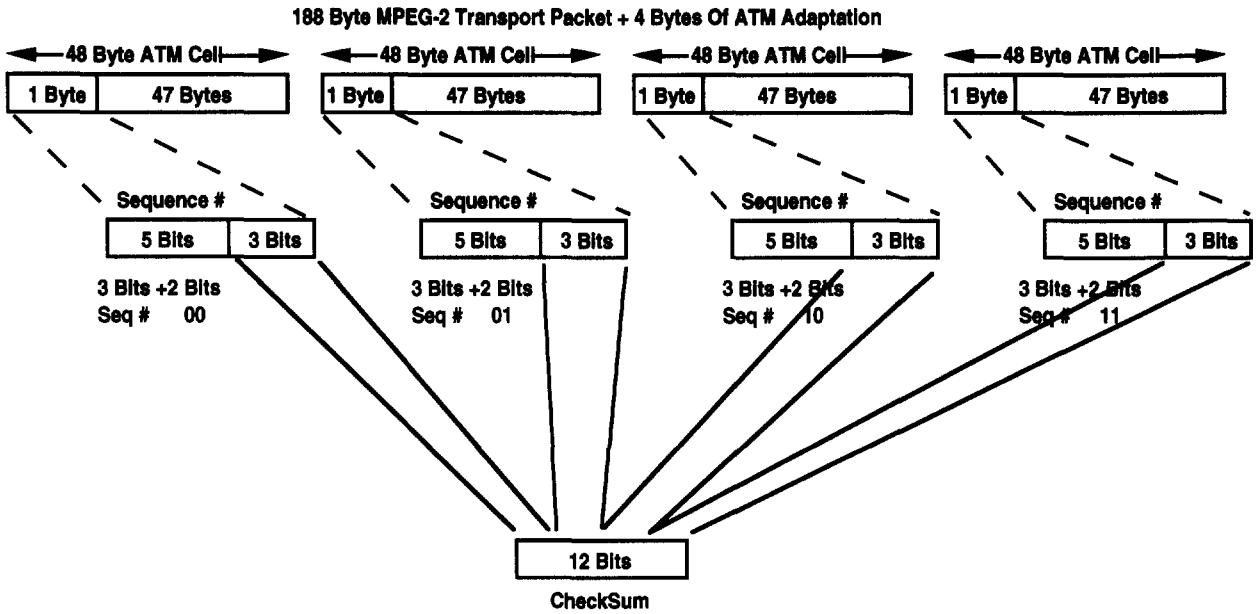
3) If one or more bits are dropped in the ATM 48-byte payload, the cell will be delivered, but its contents will be damaged.

There are two features which we feel are necessary in the Adaptation Layer Fields in order to assist in adapting to these likely error types.

1 - A sequence number is needed in each ATM cell in order to indicate which cells have not been delivered (if any).

2 - A checksum is needed in order to determine whether the data is valid in the four 48-byte cells making up the MPEG-2 188-byte cell.

It was proposed by Alexander (Sandy) MacInnis of Kalieda and Randy Sharpe of BBT in the July meeting that the following configuration be used for the ATM adaptation layer for MPEG-2 transport:



MPEG-2 ATM Adaptation Layer

The exact placement of the bits in each cell may not be accurate to their proposal, but for the purposes of discussion, the size of the fields is the main issue.

In this configuration, the four byte difference between 192 bytes (four 48-byte ATM cells) and 188 bytes (MPEG-2 packet) is applied as one byte in each of the four cells. Each byte is split into a sequence number field and a checksum field. Their proposal was to use 5 bits of sequence number and 3 bits of checksum. The 5 bits of sequence number will support counting up to 32 cells, and can therefore detect cell losses up to 31 in a row. The 5 bits is further naturally split into 2 bits of MPEG-2 cell position, and 3 bits of MPEG-2 cell sequence number. This allows the framing of the first, second, third, and fourth ATM cells in the MPEG-2 transport packet to be defined by the low 2 bits. Eight MPEG-2 transport packets can be sequenced.

The 3 bits of checksum can be combined over the 4 ATM cells to create a 12 bit checksum for the MPEG-2 transport packet. This supports bit checking accuracy to roughly one part in 4k, which is significantly more robust than a one byte checksum, which is only one in 256 in robustness.

It is certainly possible to adjust the allocations of these relative numbers of bits. It appears that 4 bits of sequence number and 4 bits of checksum would be workable, providing a 16 cycle sequence count, and a 16-bit checksum. The use of 6 bits of sequence number and 2 bits of checksum has the weakness that only a single byte would be available for the checksum. The use of 4 bits of sequence number and 3 bits of checksum provide adequate sequence counting and checksum.

By providing sequence numbers and a checksum, the location of errors and missing packets would be provided. This can be used with Reed-Solomon codes or other error correct codes to provide correction for lost or damaged ATM cells. Reed-Solomon codes, and other codes, can correct with twice the effectiveness if they can be informed of which cells are errored or missing, which can then be treated as "erasures".

If the checksum is incorrect, and it identifies a given packet as correct when it is flawed, then the Reed-Solomon (or other) code will still correct the packet, but with half the effectiveness. That is, it can be corrected, but only half as many cells can be corrected in this case with a given number of extra Reed-Solomon (or other) cells. The robustness of the checksum must be taken into account when designing the number of extra error correcting cells.

We therefore believe that the effectiveness of the adaptation layer is maximized by being able to provide sequence numbers and checksum validity information. Since a simple form of checksum would cover 4 ATM cells, the location of the error may not be identified to the cell, but rather to the group of 4 cells. However, with an appropriate checksum construction, it appears to be possible to identify which of the four ATM cells have error(s). Some care should be taken in the design of the checksum to ensure this cell identification capability. This capability does not appear feasible with a single byte checksum. Since the Reed-Solomon code, or other error correction code, will probably be arranged to correct entire lost cells, having a knowledge of which cells contain errors, to the individual cell, is the optimal goal.

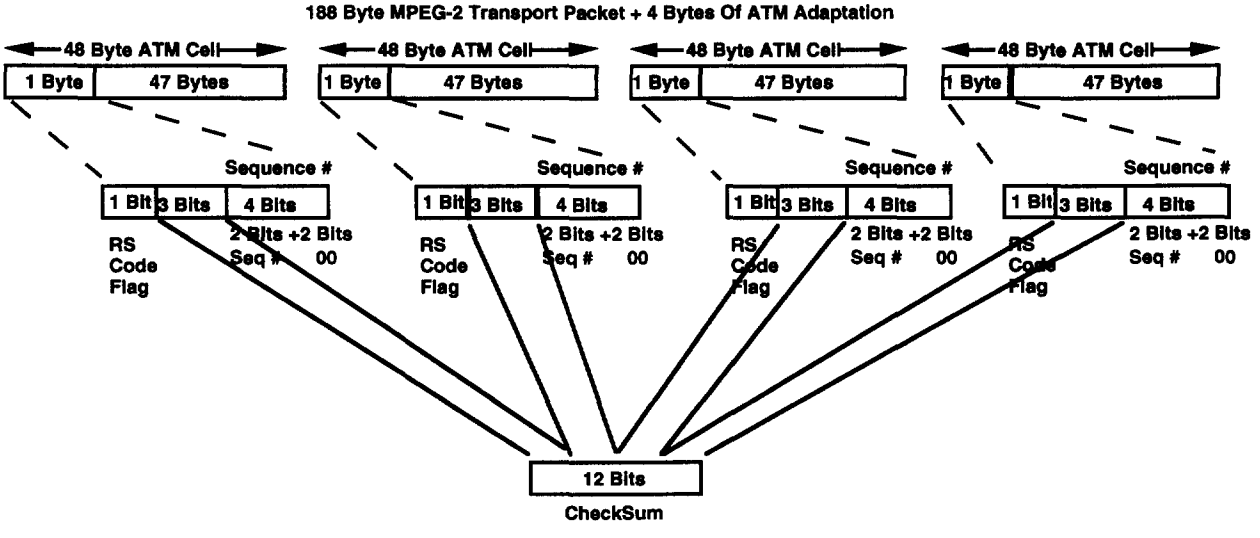
Alteration To Include RS Code Flag

Packet retry models, as are used in TCP/IP, and ATM Adaptation Layer 5 (AAL 5), may not be suitable for MPEG-2 transmission. The only viable alternatives to packet-retry, in the presence of errored or lost cells, are to attempt to suppress the bad data, or to attempt to correct the data. The use of an error correcting code is often more desirable.

When using Reed-Solomon error correction codes, or other error correction techniques, it would be desirable to be able to identify which cells contain the error correction codes, and which cells contain the MPEG-2 transport data. Assuming that an "in the clear" form of Reed-Solomon (or other error correcting code) is being used, it would be desirable to be able to identify and differentiate the clear cells from the error correcting cells. In this way, a lower cost decoder could ignore the error correcting cells, and thus avoid any need to decode the Reed-Solomon (or other error correcting) code. A more capable decoder which contains a Reed-Solomon (or other error correction) decoder would be able to correct for lost or errored cells up to the limits of the code being used.

It appears that the appropriate place to identify which cells are involved in error correcting codes is in the MPEG-2 transport adaptation to ATM. Thus, it may be appropriate to consider modifying the proposal by Alexander (Sandy) MacInnis of Kalieda and Randy Sharpe of BBT to add a bit for an error correction code flag. This bit would indicate whether the particular ATM 48-byte cell contains a quarter of an MPEG-2 transport packet, or whether it contains 47 bytes of Reed-Solomon (or other error correcting) code.

The following modification to the above proposal adds this RS Code flag, and reduces the number of bits in the sequence counter from 5 to 4:



Possible MPEG-2 ATM Adaptation Layer Including RS-Code Flag

The inclusion of the RS-Code flag in the high bit allows easy byte sign detection of the presence of absence of RS-Code data in each ATM cell. Maintaining 4 bits of sequence number and 3 bits (for 12 bits) of CheckSum still provides adequate ranges.

It may be desirable to establish a relationship between the sequence numbers and the positions of Reed-Solomon (or other) error correcting code blocks. For example, it may be desirable to continue incrementing the sequence number at the transition from the data cells to the error correction cells. However, it may be useful to set the sequence number of the next data cell following the last error correction cell to be the same sequence number as the first error correction cell. In this way, for decoders which cannot process the Reed-Solomon (or other) error correction codes, the sequence number of the MPEG-2 transport data will continuously increase without gaps.

For example:

Cell Type:	MPEG-2 data	MPEG-2 data	RS-Code	RS-Code	MPEG-2 data	MPEG-2 data
Sequence #	11	12	13	14	13	14

Since the RS-Code cells have the RS Code Flag set (=1), they can be identified and used by RS-capable decoders, and they can be ignored by decoders not capable of decoding the RS-Code.

For those cells involved in the RS-Code (or other code), the checksum will come in 3-bit increments. Thus, if two 48-byte cells are used for RS-Code (or other code), there will be 6-bits of checksum. If four bytes are used, there will be 12-bits, and if eight bytes are used, there will be 24 bits available for checksum. The checksum helps validate the sequence counter and the RS-Code bit, as well as the RS-Code (or other code) data. However, it is obvious that a 6-bit checksum is not robust. Thus, the use of the checksum field may need to be tailored to the particular RS-Code parameters and the particular use. For RS-Codes involving 4-cells or more of additional RS-data, a minimum of 12-bits of Checksum will be available.

Reed-Solomon Groups

In order to correct lost or errored cells, a typical configuration of Reed-Solomon code would be to add two, four, or eight extra cells to a cell group. Depending upon error conditions, up to two, four, or eight cells can be corrected, with one, three, and seven being correctable under other conditions. A cell group size might be 32, 48, 64, 96, 128, or 192 cells, depending upon the particular requirements. Smaller groups provide smaller latency, but start-up overheads in Reed-Solomon code pipelines suggest longer groups for effiency.
The following table indicates the resulting group sizes:

Group Size In Cells	Group Size In Bytes
32 + 2 = 34	34 * 48 = 1,632
32 + 4 = 36	36 * 48 = 1,728
48 + 2 = 50	50 * 48 = 2,400
48 + 4 = 52	52 * 48 = 2,496
48 + 8 = 56	56 * 48 = 2,688
64 + 2 = 66	66 * 48 = 3,168
64 + 4 = 72	68 * 48 = 3,264
64 + 8 = 72	72 * 48 = 3,456
96 + 8 = 104	104 * 48 = 4,992
128 + 8 = 136	136 * 48 = 6,528
192 + 8 = 200	200 * 48 = 9,600

These group sizes can be considered in relation to various typical network packet sizes:

Ethernet/IP	1.5k Bytes
FDDI	up to 4k Bytes
Fibre Channel	up to 2.1k Bytes

Since network packet sizes are related to potential uses of MPEG-2 transport on computer networks, it is relevant to consider the relationship of these ATM group sizes to network packet sizes. It should also be noted that ATM systems will often wish to carry network packets having these sizes, such as IP packets. Often such transport will use ATM Adaptation Layer 5 (AAL 5). However, a packet-retry model, such as AAL-5, may not be suitable for real-time media transport, such as MPEG-2. Thus, a model of transport wherein lost or errored cells are corrected may be more suitable.

It is thus suggested that the relationship of ATM, MPEG-2 transport, and computer networks be considered together. The inter-relationship of these transport issues suggests that a set of interoperability relationships may be suggested for how to group and transport MPEG-2 using ATM and computer networks.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG93 / ___
September 1993

Source: Apple Computer - Gary Demos, David Singer, Joseph Stampleman
Title: Use of adaption_field_control reserved code
Committee: Systems

Introduction

A goal of MPEG systems from the outset of the design of the transport bitstream has been extensibility. This document raises two questions related to this issue.

One question is the exact mechanism by which extensibility can be achieved. For example, when the adaption_field_control is coded with the reserved value, some currently defined values cease to have meaning. Could a packet using the reserved code use these fields? This and related questions need to be resolved.

Another question is the procedure by which reserved values can be "unreserved". What procedures need to be followed in order for MPEG to discuss, reach agreement on, and incorporate extensions?

Clarifications needed from members of the MPEG-2 Systems Committee:

1) How many of the fields other than the PID will be checked or needed in the transport_packet when the 2 bits of adaptation_field_control are set to the reserved code of 0? The draft document is unclear on this point. The document reads: " In current MPEG-2 decodes, a transport packet with the adaptation_field_control set to '00' shall be discarded." If this is true, we desire clarification of the requirements on other fields in this packet under this condition:

a) Is the sync byte required, or is this field available for other uses?
b) Is the transport_packet_error_indicator field available for use?
c) Can the PES_packet_start_indicator bit be given a different meaning?
d) Is transport priority needed or used in this context? Is it available?
e) Can any of these bits be used to extend the number of bits of the PID?
f) Is the transport_scrambling_control field available for use?

2) When the 2 bits of adaptation_field_control are set to the reserved code of 0, is it acceptable to make use of the PES_packet_start_indicator bit for purposes other than the start of a PES packet? For example, could this bit be used to indicate that this transport_packet contains a UniPack header when using the reserved code?

3) Would it be acceptable to use the PES_packet_start_indicator being 0 to indicate that this transport_packet contains payload for a previous UniPack header on this PID? Since the adaptation_field_control is set to the reserved code of 0, the data would not be MPEG-2 data in this context, since the packet would be discarded by MPEG-2 (only) decoders.

4) Is the transport_scrambling_control field available for other uses when the adaptation_field_control is set to the reserved code of 0? For example, could this 2 bit field be used as a continuity counter for UniPack payload packets when using the reseved code?

5) The sync_byte does not seem to be defined. Is it possible to specify the sync byte as being a CRC_8 over the subsequent 3 bytes? This would provide a more robust sync than a defined static sync code value, since the continuity_counter would yield a varying value. Such a varying value would ensure that

repetitive computer data would not perpetuate the static value within the data. If a static sync code value is required or already specified, what is this value? Could common values such as 0, 1, and 255 be avoided?

Depending upon the resolution and clarification of these issues, various UniPack universal packet header mappings onto MPEG-2 are possible. If we make the following set of assumptions concerning the clarifications, we can provide an example UniPack mapping onto MPEG-2 transport. These assumptions for this example, using the adaptation_field_control reserved code of 0, are:

*) The sync_byte must be present
*) The PID cannot be extended
*) The transport_packet_error_indicator must retain its meaning
*) The PES_packet_start_indicator is available as a UniPack header start
*) transport_priority retains its meaning
*) transport_scrambling_control is available as a UniPack continuity counter
*) The adaptation_field_control is set to the reserved code of 0
*) The continuity counter is not available, and obeys the behavior specified in the draft document which reads: "The continuity_counter shall not be incremented when the adaptation_field_control of the packet equals '00'".

Under these assumptions, the example intended mapping of UniPack onto MPEG-2 transport would be as follows. The existing MPEG-2 transport packet layer is shown first. Then an example UniPack header packet is shown, using the "complete packet" as the example. Next is shown a UniPack payload packet, where the PID is used to associate the data with a previous UniPack header packet on the same PID. The next drawing shows how a common data stream can contain data which is readable by both MPEG-2 readers as well as UniPack-savvy readers. For the UniPack-capable reader, extra information can be provided about the MPEG-2 packets on the same PID. Also, ECC and other support and check information can be interspersed using UniPack payload packets.

MPEG-2 Transport Packet Layer (from MPEG-2 Systems working draft):

Syntax	No. of bits	Identifier
transport_packet(){		
sync_byte	8	**bslbf**
transport_packet_error_indicator	1	**bslbf**
PES_packet_start_indicator	1	**bslbf**
transport_priority	1	**bslbf**
PID	13	**uimsbf**
transport_scrambling_control	2	**bslbf**
adaptation_field_control (!=0)	2	**bslbf**
continuity_counter	4	**uimsbf**
if(adaptation_field_flag=='10' || adaptation_field_flag=='11') {		
adaptation_field()		
}		
else adaptation_field_length = 0		
if(adaptation_field_flag=='01' || adaptation_field_flag=='11') {		
N = 188 - adaptation_field_length - 4		
for (i=0;i<N;i++){		
data_byte	8	**bslbf**
}		
}		
}		

The above table shows the unmodified MPEG-2 transport packet. The following section from the draft document describes the use of the fields:

Semantic Constraints

1. Packet_data_bytes consists of contiguous segments of Packetized Elementary Stream (PES) packets.

2. If a PES packet starts within a Transport Packet, the PES packet's Packet Start Code Prefix occurs immediately following the continuity_counter, or the adaption_field() if one is present.

3. The PID values 0 and 1 are reserved for Program Association Table and Conditional Access Table respectively.

4. PID value 1FFF is reserved for null packets.

5. PID values 2 to 7 and 1FF8 to 1FFE will be reserved for future use.

Transport_error_indicator: The transport_error_indicator is a 1 bit flag. When set to '1' it indicates that at least one uncorrectable bit error exists in the associated transport packet. This bit may be changed by entities external to the transport layer.

PES_packet_start_indicator: The PES_packet_start_indicator is a one bit flag. A '1' indicates that the payload of this transport packet will commence with a PES packet header. A '0' indicates there is no PES header in the transport packet payload.

transport_priority: The transport_priority is a one bit indicator. When set to '1' it indicates that the associated packet is of greater priority than packets within the same PID stream which do not have the bit set to '1'.

PID: The PID is a 13 bit field, indicating the type of the data stored in the packet payload. PID value 0x0000 is reserved for transport table (refer to ...). PID value 0x0001 is reserved for conditional access table (refer to ...). PID values 0x0002-0x0007 are reserved. PID values 0x1FF8-0x1FFE are reserved. PID value 0x1FFF is reserved for null packets.

scrambling_control: The 2 bit scrambling_control indicate the scrambling of the transport packetpayload.

00	not scrambled
01	user defined
10	user defined
11	user defined

adaptation_field_control: This 2 bit field indicates whether this transport packet header is followed by an adaptation field and/or payload.

00	reserved
01	no adaptation_field, payload only
10	adaptation_field only, no payload
11	adaptation_field followed by payload

In current MPEG-2 decoders, a transport packet with the adaptation_field_control set to '00' shall be discarded.

continuity_counter: The continuity_counter is a 4 bit field incrementing with each transport packet with the same PID. The continuity_counter wraps around to 0 after its maximum value. The continuity_counter shall not be incremented when the adaptation_field_control of the packet equals '00' or '10'. If two consecutive transport packets of the same PID have the same continuity_counter value and the adaptation_field_control equals '01' or '11', the two transport packets shall be considered duplicate.

Proposed Future UniPack Transport Packet Layer, Complete Packet Example:

Syntax	No. of bits	Identifier
transport_packet(){		
sync_byte	8	bslbf
transport_packet_error_indicator	1	bslbf
PES_packet_start_indicator (==1, UniPack Header)	1	bslbf
transport_priority	1	bslbf
PID	13	uimsbf
UniPack_continuity_counter (==0)	2	bslbf
adaptation_field_control (==0, indicating UniPack)	2	bslbf
continuity_counter (frozen)	4	uimsbf
Additional UniPack Header:		
UniPack_c	2	bslbf
UniPack_header_type *(this example = complete packet)*	6	bslbf
total_packet_length (= total_data_length+header_length)	24	uimsbf
header_length (start of transport_packet to data_bytes)	8	uimsbf
priority_byte	8	uimsbf
channel_number (= MPEG-2 program number)	16	uimsbf
channel_running_byte_count	32	uimsbf
CRC_32 (over first 16 bytes of header)	32	bslbf
Optional Directory (Depending on UniPack_header_type):		
N = (header_length - 24)/4		
for (i=0;i<N;i++){		
UniPack_f	2	bslbf
UniPack_key	6	bslbf
directory_item_packet_length	24	uimsbf
}		
CRC_32 (over directory)	32	bslbf
N = 188 - header_length - 4		
for (i=0;i<N;i++){		
data_bytes	8	bslbf
}		
}		

The above table shows the UniPack mapping onto the MPEG-2 transport_packet layer. See the UniPack document for details on the meaning of the UniPack fields. Note that the first CRC_32 has been extended to cover the first 4 bytes of the transport_packet header, in addition to the additional 12 bytes of the UniPack first header block. This augments other checks on the PID and other key fields in the transport_packet header to ensure validity. CRC_32 should be sufficiently robust for all data uses of a universal header.

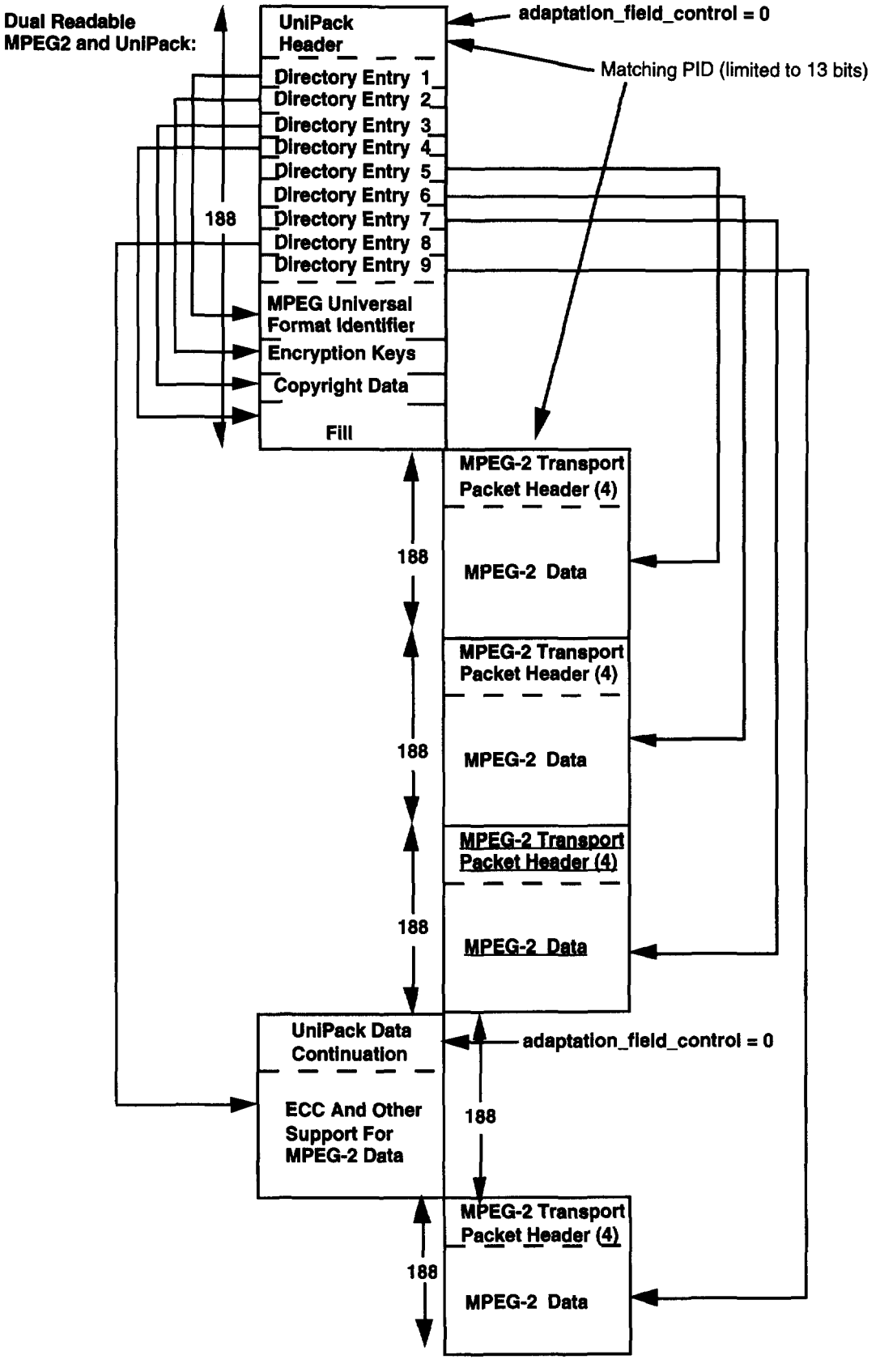
The remainder of the transport_packet following the CRC_32 after the optional directory contains data as described by the UniPack header (and its optional directory).

Proposed UniPack Transport Packet Layer, Continuation of UniPack Data:

Syntax	No. of bits	Identifier
transport_packet(){		
sync_byte	8	**bslbf**
transport_packet_error_indicator	1	**bslbf**
PES_packet_start_indicator (==0, UniPack Payload))	1	**bslbf**
transport_priority	1	**bslbf**
PID	13	**uimsbf**
UniPack_continuity_counter (increments each pkt)	2	**bslbf**
adaptation_field_control (==0, indicating UniPack)	2	**bslbf**
continuity_counter (frozen)	4	**uimsbf**
N = 188 - header_length - 4		
for (i=0;i<N;i++){		
data_bytes	8	**bslbf**
}		
}		

The above table shows a continuation packet for payload for the UniPack header which was issued previously on the same PID. The data_bytes may contain other information such as larger continuity counters, a 16 bit channel number (program number), and other information, as described in the UniPack header's initial directory.

On the following page, the combination of UniPack and MPEG-2 is shown, with the data stream readable from both MPEG-2 and UniPack. The directory entries describe each MPEG-2 packet individually. It is also possible to have a single identifier, descriptor, or directory entry for a collection of MPEG-2 transport_packets.


Dual Readable
MPEG2 and UniPack:
UniPack
Header
Directory Entry 1
Directory Entry 2
Directory Entry 3
Directory Entry 4
Directory Entry 5
Directory Entry 6
Directory Entry 7
Directory Entry 8
Directory Entry 9
MPEG Universal
Format Identifier
Encryption Keys
Copyright Data
Fill
188
adaptation_field_control = 0
Matching PID (limited to 13 bits)
MPEG-2 Transport
Packet Header (4)
MPEG-2 Data
188
MPEG-2 Transport
Packet Header (4)
MPEG-2 Data
188
MPEG-2 Transport
Packet Header (4)
MPEG-2 Data
188
UniPack Data
Continuation
ECC And Other
Support For
MPEG-2 Data
adaptation_field_control = 0
188
MPEG-2 Transport
Packet Header (4)
MPEG-2 Data
188

Interoperability Concerns Beyond Broadcast Television

~~29.97, 59.94 Frame Rates~~
- Awkward conversion

~~Interlace Scan~~
- Unnecessary complication
- Exacerbated artifacts
- "Interim" standards persist

~~Non-Square Pixels~~
- Inefficient image manip.

~~Quasi-Scalable Hierarchy~~
- Unrelated spatial and temporal formats

24, 30, 60 Frame Rates
- 25, 50, 75 as alternate

Progressive Scan
- Achievable now
- Facilitates image processing
- Market enabler

Square Pixels
- Efficient image manipulation

Scalable Hierarchy
- True future extensibility
- 1280x720, 1920x1080 to start

RECEIVED
NOV 29 1993
FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

RECEIVED

NOV 2 9 1993

JEGI Doc.-039

FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

Comments on Interoperability Aspects of the Grand Alliance HDTV System

Robert F. Sproull, VP & Fellow, representing Wayne E. Rosing, Exec. VP
Sun Microsystems Laboratories, Inc.

October 6, 1993

Interoperability of broadcast material with computers and workstations are most strongly influenced by the following requirements:

1. An all-digital implementation (PS-WP4 rec 1)
2. A well-designed layered communication architecture (PS-WP4 recs 2,4,5,8)
3. Compliance with international standards (PS-WP4 rec 10)
4. Progressive scan (PS-WP4 rec 3)
5. Square pixel aspect ratio (PS-WP4 rec 9)

With minor exceptions, the current GA proposal does an excellent job of meeting these requirements.

Data type independence. The success of this broadcast standard over a long period will require that the modulation and data transport layers be able to support a wide variety of data carriage. In addition to transmitting motion pictures, this broadcast standard will be used to transmit wholly different information types such as high-resolution static images representing magazines and newspapers; text data that is part of news feeds, consumer information, government publications; and many others. The burgeoning Internet and developing National Information Infrastructure exemplify the country's appetite for data transport.

To accommodate such a range of data types, the "packet type" field of the transport layer is used to label each type uniquely. It is essential that the assignment of packet types be administered so as to meet these diverse needs. These assignments should be controlled by a standards group, distinct from groups responsible for the format of the data types themselves.

While the method of administering packet type codes might appear unimportant, it is vital to the aggressive evolution of broadcast material. Product development cycles for digital decoders, algorithms, and equipment are short compared to those for modulation and broadcast equipment. Suitably decoupling the modulation and digital domains will permit rapid evolution of content and services within the broadcast framework.

JEGI Doc.-040

RECEIVED

NOV 2 9 1993

FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

October 6, 1993

To: Joint Experts Group on Interoperability
Advisory Committee on Advanced Television

From: Mike Liebhold
Apple Computer, Inc.

Re: ATV in the Context of the National Information Infrastructure (NII)

Recently, President Clinton, and Vice President Gore announced a series of initiatives supporting a National Information Infrastructure (NII). The NII will ultimately enable all Americans improved communications, collaboration, access to libraries of multimedia information and educational resources, thereby improving national learning and productivity.

Lifelong learning is a necessity, not a luxury for America. There is a direct link between the skills of the nation's workforce and the resultant standard of living of that nation. Education is the foundation for the quality of life and the proper functioning of the social and political process.

The NII is a major national issue that demands an unprecedented level of collective commitment and cooperation of all stakeholders -- students, parents, teachers, doctors, administrators, local communities, hospitals,, business labor, and many agencies of state local federal government.

Advanced Television and HDTV offer great promise to serve as a catalyst for interoperable image communications thereby enabling telecommuting, extension of the formal education into homes and communities extensionof the informal lifelong learning process and enabling the delivery a wide range of new electronic information services including public health information.

As we review the Grand Alliance proposal, we must answer two critical questions:

- How will digital HDTV accelerate information and media services?

- How would an interlaced-HDTV inhibit market formation for electronic education and publishing? (There is no disagreement that interlace is not suitable for text and fine graphics, maps and diagrams.)

Currently the Grand Alliance proposal includes interlace transmission formats. If approved by the FCC, the first generation low-end interlace-only HDTVs will not be able to enjoy the benefits of the NII. Unfortunately, those elements of our society that most need the benefits of the NII, can least afford more than one high resolution display.

There is, in fact, no migration strategy that will permit these first generation interlace-only receivers to EVER display NII services. Further, once these receivers are in place, broadcasters will become captive to the installed base of low-end TVs and will therefore be unable to move from an interim interlace standard to a standard that will enable all Americans to enjoy the benefits of the NII.

Proposal for an Explicit 1080x1920, 1:1, 24 Hz HDTV format

RECEIVED
NOV 29 1993
FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

JEGI Doc.-041
RECEIVED
NOV 29 1993
FEDERAL COMMUNICATIONS COMMISSION
OFFICE OF THE SECRETARY

Kenneth A. Parulski
Representing: Motion Picture and Television Imaging Division
Eastman Kodak Company, 4/81 Kodak Park, Rochester NY 14650-2015

Eastman Kodak Company supports the explicit inclusion in the U.S. terrestrial HDTV standard of a high line-count, progressively scanned, square pixel, 24 Hz image source format[1]. Specifically, we favor the use of 1080 active lines, consistent with the recommendations of the (FCC) Advisory Committee on Advanced Television Service, to maximize compatibility with the anticipated television production standard as well as to encourage development of an international HDTV image format agreement.

Abundant, ultra-high definition source material exists at 24 Hz in the form of filmed programming. To optimize both the efficiency of the intraframe compression used for post-production and the efficiency of the motion-compensated interframe coding used for distribution, the system should adapt to the source frame rate[2]. It appears technically feasible to deliver a 1080x1920, 1:1, 24 Hz format using the GA system design at U.S. HDTV service initiation. This 24 Hz source format provides 49.8 million luminance pixels / sec. The 720x1280, 1:1, 60 Hz format proposed by the GA provides 55.3 million luminance pixels / sec. The 960x1728, 2:1, 60 Hz format proposed by the GA provides 49.8 million luminance pixels / sec. Note that the proposed 24 Hz format rate equals the lower of these two GA 60 Hz rates.

To implement this proposal, all compression encoders must be designed to directly accept the 24 Hz format as one allowable input format. The quantizers used to code the DCT coefficients of the motion compensated residuals, in the 24 Hz mode, should be specifically optimized for motion picture film material. All receivers must be capable of receiving this 24 Hz format, although they could convert to a different format for the final display, if desired. If all receivers are designed to accept a 1080x1920, 1:1 format from the initiation of HDTV service, the migration path to a 60 Hz version of this format will be simplified. The 1080x1920, 1:1, 24 Hz format would provide superior spatial resolution for film originated material, compared to other 24 Hz formats suggested by the GA. It would also allow more efficient and effective image processing during post-production, since there would be no required scanning standard conversions between the telecine output and the GA compression encoder input. A 1080x1920, 1:1 format is also ideal for still images and computer graphics.

HD Telecines capable of producing this 24 Hz format have been demonstrated and will soon be commercially available. Kodak also believes that it is feasible to develop and manufacture CCD-based, progressively scanned, square pixel cameras, initially at 30 Hz, and ultimately at 60 Hz. Kodak presently manufactures a progressive scan, square pixel, 1000x1000, 30 Hz sensor[3], and is marketing a monochrome industrial camera using this sensor. Kodak has recently developed an interlaced 1080x1920, 2:1, 60 Hz image sensor[4], which is available as an off-the-shelf product[5]. The sensor was designed using interlaced readout so that it could be interfaced to commercially available HDTV recorders and displays. However, a 30 Hz progressive scan version of this sensor, using existing technology and offering similar performance specifications, could be developed immediately if there was a commercial need. Although a 60 Hz version of this sensor requires improved technology, Kodak believes that a 1080x1920, 1:1, 60 Hz camera could be developed before significant market penetration of consumer HDTV receivers occurs.

References

1. "Considerations for an Explicit, 1080x1920, 1:1, 24 Hz, HDTV Format", Letter from Henri Petit, Vice President, Motion Picture and Television Imaging, Eastman Kodak Company, October 5, 1993.
2. Parulski, K. A., et. al., "Source-Adaptive Encoding Options for HDTV and NTSC", SMPTE *Journal*, 101:674-683, Oct. 1992.
3. Steven, E. G., et. al., "A 1-Megapixel, Progressive-Scan Image Sensor with Antiblooming Control and Lag-Free Operation, *IEEE Trans. on Electron Devices*, 38:981-988, May 1991.
4. Nichols, D. N., "Single-Chip Color HDTV Image Sensor with Two Polysilicon Levels and with WSi_x Lightshield Used for Strapping Vertical Gates", *IEDM Tech. Dig.*, pp. 101-104, Dec. 1992.
5. Kodak KAI-2090CM Image Sensor Specifications, July 1993.

"Film mode" format recommendation

Abundant ATV source material exists at 24 Hz

- **Format parameters**
  - 24 frames per second
  - Progressive scanning
  - Square pixels
  - 1080 lines x 1920 pixels per line
  - 49.8 million luminance pixels / sec
- **System requirements**
  - All compression encoders would directly accept this format as one allowable input format
  - Encoder quantizers used for the film mode would be optimized for motion picture film material
  - All receivers would be capable of decoding this format
  - Display format decided by the receiver manufacturer
- **Reasons for including this format in the GA system**
  - Provides higher image quality for motion picture film, the source of most initial ATV programs
  - Facilitates efficient intraframe compression during post-production and interframe compression during delivery
  - Allows post-production processing and storage to operate more efficiently for 24 Hz sources
  - Helps ensure receivers will migrate to the final 1080 x 1920, 60 Hz, progressive scan format
- **Other sources could use the 1080 line x 1920 pixel format**
  - Progressive scan, square pixel, high resolution format is ideal for still images and computer graphics
  - If desired, electronic cameras could be developed immediately using 30 Hz progressive scanning
  - Future electronic cameras could use 60 Hz progressive scanning

October 5, 1993

To:

Robert Hopkins—Chairman, ACATS Experts Group on Scanning Format and Compression

Robert Sanderson—Chairman, ACATS Joint Experts Group on Interoperability

George Vradenburg III—Chairman, ACATS Expert Group on Production & Receiver/VCR Impact

Joseph Flaherty, Irwin Dorros—Co-Chairmen, ACATS Technical Subgroup

Richard Wiley—Chairman, ACATS

Robert Rast—Grand Alliance

James McKinney—Chairman, ATSC

Subject: Considerations for an Explicit, 1080x1920, 1:1, 24 Hz HDTV Format

HDTV is targeted to provide consumers with a greatly improved image quality experience. For decades, the theatrical experience has been achieved with dramatic success with ultra-high definition filmed images at 24 Hz. It makes good business and practical sense to provide solutions that are as close as possible to that level of performance in any initial HDTV service.

Eastman Kodak Company supports the explicit inclusion in the U.S. terrestrial HDTV standard of a high line-count, progressively-scanned, square pixel, 24 Hz image source format. Specifically, we favor the use of 1080 active lines (versus 960 or 720), consistent with the recommendations of the (FCC) Advisory Committee on Advanced Television Service. We favor 1080, as does ACATS, to maximize compatibility with the anticipated television production standard as well as to encourage development of an international HDTV image format agreement.

Henri D. Petit, Vice President
General Manager, Motion Picture & Television Imaging
EASTMAN KODAK COMPANY · ROCHESTER, NEW YORK 14650-0310 · 716 724-7687 · Fax 716 724-9702

Rationale for our recommendation for inclusion of the 1080x1920, 1:1, 24 Hz format as an explicit image source format is as follows:

- There appears to be agreement that achieving a family of source image formats of 1080x1920, 1:1, 24/30/60 Hz is the long term goal for U.S. HDTV, even though technology is not yet available to deliver the 30 or 60 Hz variants over a 6 MHz channel, with NTSC present, as required by the FCC for terrestrial broadcast.

- The 24 Hz variant **can** be delivered at U.S. HDTV service initiation:

  - Abundant, ultra-high definition source material exists at 24 Hz in the form of filmed programming.

  - HD telecines capable of producing this format have been demonstrated and will soon be commercially available.

  - By taking advantage of the lower temporal rate of 24 Hz, current technology appears to be available to encode this format.

- Where necessary, higher frame rate (60 and possibly 30 Hz) source material could be handled in an interlaced and/or lower horizontal resolution format (say, 1440 pixels per line), until technology advances sufficiently to achieve the long term goal.

- Kodak now believes it is feasible to develop and manufacture CCD-based, progressively-scanned, square pixel cameras that would operate initially at 30 Hz (and ultimately at 60 Hz) with 1080x1920 image format. We further believe that, with sufficient business incentive, such cameras could be commercially available and cost effective in just a few years, much earlier than prior forecasts. Inclusion of the 1080x1920, 1:1, 24 Hz format, consistent with and on the way to the long term target, should increase the business incentive to develop these systems.

From all expert opinion on this topic, 1080x1920, 1:1, 24 Hz appears feasible at HDTV introduction and simultaneously defines a high level of viewer experience within current electronic, compression and transmission technologies. This format should be explicitly

included in the initial service offering in the U.S. as a delivery option for broadcasters, receiver manufacturers, and other creative media service providers. We believe this to be in the best interest of our company, the customers we serve, and the general viewing public.

Please feel free to contact William W. Peck, Manager, Advanced Technology Products, Motion Picture and Television Imaging, if you have any comments or questions.

Sincerely,

H.D. Lotts

SMPTE Journal

Source-Adaptive Encoding Options for HDTV and NTSC

By Kenneth A. Parulski, C. Bradley Hunt and LeRoy E. DeMarsh

Reprinted from the *SMPTE Journal*
October 1992 Issue, Volume 101, Number 10

Copyright © 1992 by the Society of Motion Picture
and Television Engineers, Inc.