
ATM Forum Technical Committee

ATMF 95-0326 R2

Title: Draft Proposal for Specification of FEC-SSCS for AAL Type 5

Source:

Hiroshi Esaki (Toshiba Corp.)
801 Schapiro Research Building,
c/o CTR, Columbia University,
530 West, 120th Street,
New York, New York, 10027, USA.
Tel:(212)-854-2365, Fax:(212)-316-9068
E-mail: hiroshi@ctr.columbia.edu

Georg Carle
Institute of Telematics, University of Karlsruhe
Zirkel 2
D-76128 Karlsruhe, Germany
Tel:+49-721-608-4027, Fax:+49-721-38809
E-mail: carle@telematik.informatik.uni-karlsruhe.de

Tim Dwight
MCI Telecommunications Corporation,
901 International Parkway
Richardson, Texas, 75081, USA.
Tel:(214)-498-1484, Fax:(214)-498-1300
E-mail:0006078043@mcimail.com

Aloke Guha
Research and Advanced Development,
Network Systems Corporation,
7600 Boone Avenue N.,
Minneapolis, MN., 55428, USA.
Tel:(612)-391-1261, Fax:(612)-391-1191
E-mail:guha@network.com

Keiji Tsunoda, Kumiko Kanai
R&D Center, Toshiba Corporation,
1 Komukai-Toshiba-cho, Saiwai-ku,
Kawasaki, 210 Japan.
Tel:+81-44-549-2238, Fax:+81-44-520-1806
E-mail: tsunoda@csl.rdc.toshiba.co.jp, kumiko@isl.rdc.toshiba.co.jp

Date: October 2 - 6, 1995

Distribution: Plenary, SA&A

Abstract:

This contribution discusses the specification of FEC-SSCS for AAL type 5. The features of the proposed specification are:

1. Compatibility with existing AAL type 5
2. FEC control parameters are negotiable and adjustable
3. Media and service independence
4. AAL-SDUs of variable length are supported
5. Pipelining transmission is supported
6. Minimization of processing costs for lossless transmission case
7. Three operational modes are supported
8. Cell-loss/bit-error occurrence indication is supported.

Notice :

This contribution has been prepared to assist the ATM Forum, and is offered by the above affiliations as a basis for discussion. The above listed affiliations reserves the right to add, amend or withdraw the statements contained herein.

Specification of an FEC Service Specific Convergence Sublayer for AAL5 and AAL 3/4

1 Introduction

This draft specifies the Forward Error Correction Service Specific Convergence Sublayer (FEC-SSCS). The FEC-SSCS is located in the upper part of ATM Adaptation Layer on top of the Common Part Convergence Sublayer (CPCS) of AAL type 5 and AAL type 3/4. The FEC-SSCS is used to obtain high throughput and small latency reliable end-to-end data delivery.

[Editor's note:

In this draft, the detailed data formats are specified assuming the use of AAL type 5. We never preclude the use of AAL 3/4 for the FEC scheme. In order to include the use of AAL 3/4, additional effort may be required for amendments of the specification.

]

2 Service provided by FEC-SSCS

The FEC-SSCS provides the capability to transfer AAL-SDUs from a source AAL-SAP to a destination AAL-SAP through the ATM network.

When a destination CPCS entity detects an error (bit error and/or cell loss) when receiving a CPCS-SDU, the FEC-SSCS entity tries to recover the original data sent from the source CPCS entity using the FEC (Forward Error Correction) algorithm specified in this specification.

FEC-SSCS has three operational modes, associated with three different error types it tries to recover.

a) SEC (Symbol Error Correction) Mode

In the SEC mode, the FEC-SSCS entity only tries to recover bit errors in the symbols. (Symbols are data units defined by the FEC algorithm. Typical symbol lengths are 8 bit, and 16 bit.) Missing symbols, or missing cells, resp., are not subject of error recovery in this mode. When symbols are missing due to cell discard in the network, the received frame will not be recovered.

b) SLC (Symbol Loss Correction) Mode

In the SLC mode, it will be tried to recover only symbols that are missing due to cell loss. Bit errors are not subject of error recovery. When the recovery of missing symbols fails, the originally received data is either delivered to the upper layer entity, together with an error indication, or not delivered to the upper layer entity.

c) SEAL (Symbol Error And Loss correction) Mode

In the SEAL mode, it will be tried to recover both bit errors in symbols, and the absence of the symbols due to cell discarding.

[Editor's note :

The basic data format and required algorithm can be common in the three operational modes. The combination of functionalities of the individual modes differ.

(1) data format

the payload of each cell (more generally: each vertical line defined by the FEC algorithm) contains a CRC-10 field for error detection.

{ rationale:

- the CRC-10 calculation at the sender side does not require very high processing effort.
- the CRC-10 calculation (attaching CRC-10) at the sender side can be skipped in SLC mode.
- bit error check using CRC-10 at the receiver can be skipped in SLC mode

}

(2) Reed-Solomon (RS) error recovery algorithm

It is assumed that only missing symbols are subject of error recovery. It is not assumed that bit errors are subject of error recovery by the RS error recovery algorithm in the receiver.

{rationale:

- In SEAL and SEC modes, the positions of symbol errors are detected by CRC-10. These symbols may be assumed as missing symbols, rather than symbol errors in the RS error recovery algorithm. This simplifies FEC processing at the receiver process.
- In SLC mode, the positions of missing symbols are detected by the SN field.

}

]

All operating modes provide the following services.

1. Cell-loss indication

The primitive from FEC-SSCS to the upper layer includes a cell loss indication. When the received CPCS-SDU experiences cell discarding, it is indicated to the upper layer using the cell loss indication.

[Editor's note:

The cell loss indication may be useful for some transport protocols, e.g. TCP, to use cell loss indication for flow/congestion control.

]

2. Bit error indication

The primitive from FEC-SSCS to the upper layer includes a bit error indication.

When the received CPCS-SDU contains bit errors, it is indicated to the upper layer using the bit error indication.

3. FEC parameter negotiation

The parameters of the FEC algorithm are negotiated after the connection set-up procedure. These parameters can be modified during a session.

[Editor's note :

The method used for FEC parameter negotiation must be defined. In this draft, a method using in-band signaling is assumed. The use of Q.2931 signaling, or the use of a signaling channel are for further study (FFS). The in-band signaling message is identified using the CPCS-UU field.

]

4. Adjustment of the length of actually transmitted redundant data

The length of redundant data is usually negotiated by the FEC parameter negotiation procedure. However, the source entity can decide how much redundant data is actually transmitted to the destination entity.

In other words, the FEC parameter negotiation procedure only specifies the maximum amount of redundant data to be transmitted to the destination entity.

[Editor's note :

This operation is completely localized in the source entity, and allows to modify the amount of redundant data of an FEC frame for cases in which there is no need for negotiation between the source and destination entities.

For efficient processing of frames at the destination entity, the source entity has to indicate the amount of redundant data actually sent to the destination entity. This information will be explicitly given either by the LI (length indicator) of the CPCS trailer, or by the Number of Redundant Cells (#RedCell) field in the CPCS-UU of the CPCS trailer.

By this operation, the data transmission efficiency can be optimized by reduction (and optimization) of the actually transmitted redundant data, without performing the FEC parameter negotiation procedure. The benefits of this operation are as followed:

a) Reduction of redundant data transmission overhead

This option will work well when the end-station must transmit various length of packets, e.g., from 100 bytes to 64 KBytes. For larger packets, a larger amount of redundant data should be appended. If the end-station does not want to renegotiate FEC parameters for the transmission of small packets, and appends the same number of redundancy cells to all frames, the transmission overhead due to the redundant data would be large for small frames.

By adjustment of the actually transmitted redundant data, the number of redundancy cells appended to small frames can be reduced.

b) Avoiding FEC parameter renegotiation overhead

In some cases, the FEC parameter renegotiation procedure would be costly. Moreover, when the generated packet size changes frequently, the end-station may have to perform frequent FEC parameter renegotiations to reduce the transmission performance (i.e., latency and throughput). By using this operation, the frequency of FEC parameter renegotiation can be reduced.

c) Low-cost implementation

Some interface cards may support only one (or few) FEC parameters set(s). This could be the case for low-cost FEC-SSCS implementations. It can be generally assumed that the implementation of this operation will be easier and simpler than the implementation of FEC parameter renegotiation.

]

5. Variable size CPCS-SDU

The size of a CPCS-SDU that is exchanged between an FEC-SSCS entity and a CPCS entity can vary. The maximum size of a CPCS-SDU and the minimum size of a CPCS-SDU are specified by parameters for the FEC-SSCS. Without the occurrence of errors, the CPCS-SDU experiences a smaller data delivery latency from the source FEC-SSCS entity to destination FEC-SSCS entity.

6. Pipelined data transfer in source FEC-SSCS

The source FEC-SSCS entity can transfer FEC-SSCS-PDUs by pipelining. By the use of pipelining, the buffer resources in the source FEC-SSCS entity can be used more efficiently, the transmission latency at the source FEC-SSCS can be reduced, and the burstiness of the resulting ATM traffic can be reduced. Both the pipelining of AAL-SDU transmission and the pipelining of FEC frame transmission should be supported to reduce the peer-to-peer data delivery latency.

[Editor's note :

The fragmentation of AAL-SDU into multiple FEC-SSCS-PDU will be performed for the large size AAL-SDU transmission.

In pipelining mode, the received FEC-SSCS-PDU will be transmitted to the upper layer before whole of the AAL-SDU is completely received by FEC-SSCS entity. This operation mode is similar to the streaming mode defined in I.363. The benefits of this pipelining mode are:

a) Reduction of required receiving buffer space

Some end-stations may not be able to allocate a sufficiently large buffer space to receive a large AAL-SDU. This will typically occur when the product of "number of VCCs" and "Maximum AAL-SDU size" becomes large. By using the pipelining operation (i.e., AAL-SDU fragmentation), buffer space to be allocated to the VCC can be reduced.

- b) Reduction of control latency
 Since the error in the received AAL-SDU (e.g., IP packet) will be indicated by the FEC-SSCS entity earlier than in the non-pipelining operation, the upper layer entity can perform the end-to-end control (e.g., fast retransmission) earlier. As a result, the performance (i.e. latency and throughput) of the end-to-end data transmission will be improved.

]

7. Selection of operation mode

The upper layer entity can select which operation mode is applied by the FEC-SSCS (i.e. selection of SEC mode, SLC mode or SEAL mode).

The referenced protocol structure associated with FEC-SSCS is shown below.

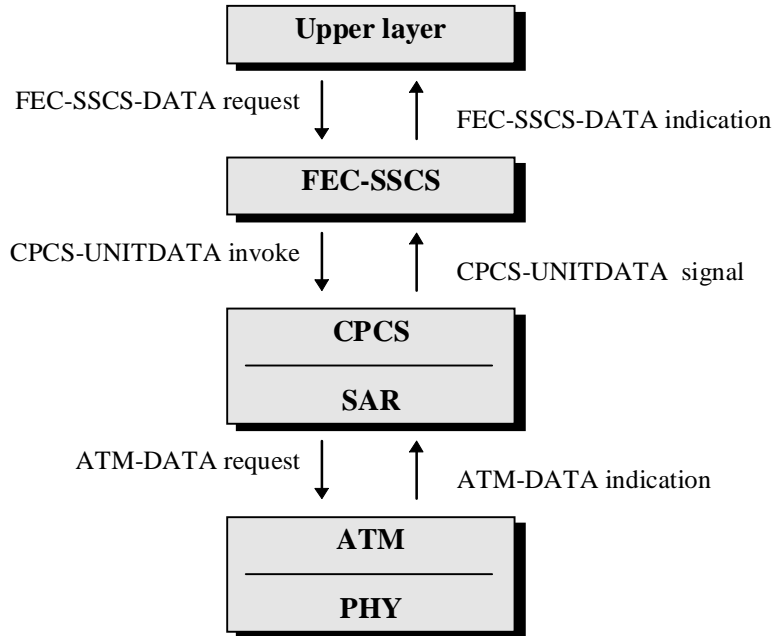


Figure 1 Protocol reference model

[Editor's note :

Relationship of data units:

- Burst of FEC-SSCS-SDU = n FEC-SSCS-SDUs
- FEC-SSCS-SDU (=packet) = m FEC-SSCS-PDUs = m CPCS-SDUs
- FEC-SSCS-PDU (=FEC frame, with data-part and FEC-code-part)=CPCS-SDU

Indication of end of data units:

- Burst of FEC-SSCS-SDU : FEC-SSCS-BI --> CPCS-UU (1 bit)
- FEC-SSCS-SDU (=packet) : FEC-SSCS-M --> CPCS-UU (1 bit)
- FEC-SSCS-PDU (=CPCS-SDU) : ATM-UU --> ATM-PT (2 code points)

]

2.1 Primitives

2.1.1 Primitives between CPCS and FEC-SSCS

The primitives exchanged between CPCS and FEC-SSCS are described in Table 1.

Table 1: Primitives between CPCS and FEC-SSCS

Parameter	Type	Comments
Interface Data (ID)	invoke signal	whole or partial FEC-SSCS-PDU
More (M)	invoke signal	M=0: end of FEC-SSCS-PDU M=1: not end of FEC-SSCS-PDU
CPCS-Loss Priority (CPCS-LP)	invoke signal	- invoke: CPCS-LP=0(high priority) - signal: Mapped from the ATM layer's congestion indication parameter
CPCS-Congestion indication (CPCS-CI)	invoke signal	Mapped to/from the ATM layer's congestion indication parameter CPCS-CI=1: congestion experienced CPCS-CI=0: no congestion experienced
CPCS-User-to-User indication (CPCS-UU)	invoke signal	Transparently transported by the CPCS (include indications: end of burst (FEC-SSCS-BI), FEC-SSCS-SIGI, end of FEC-SSCS-PDU)
CPCS-SDU Reception Status (CPCS-RS)	invoke signal	Indication of corrupted CPCS-SDU

The primitives are called "invoke" and "signal" instead of the conventional "request" and "indication", to have a consistency with the specification of ITU-T I.363.

2.1.1.1 Primitives for the data transfer service

These primitives are CPCS-UNITDATA invoke and the CPCS-UNITDATA signal. They are used for the data transfer.

1. Interface Data (ID)

This parameter specifies the interface data unit exchanged between the CPCS and the FEC-SSCS entity. The interface data is an integral multiple of one octet. It is assumed that the CPCS entity operates in streaming mode.

2. More (M)

This parameter specifies whether the Interface Data Unit is a beginning/continuation of an FEC-SSCS-PDU or the end of an FEC-SSCS-PDU.

3. CPCS-Loss Priority (CPCS-LP)

This parameter indicates the loss priority for the associated CPCS-SDU. It can take only two values, one is for high priority, and the other is for low priority.

The CPCS-LP of invoke shall always indicate high priority.

The CPCS-LP of signal will be mapped from the SAR-LP parameter.

4. CPCS-Congestion Indication (CPCS-CI)

This parameter indicates whether the associated CPCS-SDU has experienced congestion. This indication will be transparently transported to FEC-SSCS Congestion Indication, that will be indicated to the upper layer. This parameter is mapped to and from the SAR-CI parameter.

5. CPCS User-to-User Indication (CPCS-UU)

The parameter is transparently transported by the CPCS between FEC-SSCS entities. This indication shall include at least

- a) the indication of the end of a FEC-SSCS-SDU (the end of a packet) and
- b) the indication of the end of a burst (FEC-SSCS-BI).

6. CPCS-SDU Reception Status (CPCS-RS)

This parameter indicates that the associated CPCS-SDU delivered may be corrupted (i.e. errored). This parameter is used for the indication whether the FEC algorithm is performed. When the RS indicates the reception of a corrupted CPCS-SDU, the FEC process is invoked.

2.1.1.2 Primitives for the abort service

The following primitives are used.

– CPCS-U-ABORT invoke and CPCS-U-ABORT signal

This primitive is used by the FEC-SSCS entity to invoke the abort service. It is also used to signal to the FEC-SSCS entity that a partially delivered CPCS-SDU is to be discarded by instruction from its peer entity (i.e. the source FEC-SSCS entity). These primitives are mapped to and from FEC-SSCS-U-ABORT invoke and FEC-SSCS-U-ABORT indication.

– CPCS-P-ABORT signal

This primitive is used by the CPCS entity to signal to the above FEC-SSCS entity that a partially delivered CPCS-SDU is to be discarded due to the occurrence of some error in the CPCS or below the CPCS. This primitive is mapped to FEC-SSCS-P-ABORT signal.

2.1.2 Primitives between FEC-SSCS and upper layer

The primitives exchanged between FEC-SSCS and upper layer are described in Table 2.

Table 2. Primitives between FEC-SSCS and upper layer

Parameter	Type	Comments
Interface Data (ID)	request indication	whole or partial FEC-SSCS-SDU
More (FEC-SSCS-M)	request indication	FEC-SSCS-M=0: end of FEC-SSCS-SDU FEC-SSCS-M=1: not end of FEC-SSCS-SDU
Burst indication (*) (FEC-SSCS-BI)	request indication	FEC-SSCS-BI=0: end of burst FEC-SSCS-BI=1: not end of burst
FEC-SSCS-Loss Priority (FEC-SSCS-LP)	request indication	- invoke: FEC-SSCS-LP=0(high priority) - signal: Mapped from the ATM layer's congestion indication parameter
FEC-SSCS-Congestion indication (FEC-SSCS-CI)	request indication	Mapped to from the ATM layer's congestion indication parameter FEC-SSCS-CI=1: congestion experienced FEC-SSCS-CI=0: no congestion experncd
FEC-SSCS Cell Loss indication (FEC-SSCS-CLI)	request indication	Semantics of the value is FFS.
FEC-SSCS Bit Error indication (FEC-SSCS-BEI)	request indication	Semantics of the value is FFS
FEC-SSCS Signaling indication (FEC-SSCS-SIGI)	request indication	FEC-SSCS-SIGI=1: signaling data FEC-SSCS-SIGI=0: not signaling data
FEC-SSCS-User-to-User indication (FEC-SSCS-UU)	request indication	Transparently transported by the FEC-SSCS (functions are FFS)
FEC-SSCS Reception Status (FEC-SSCS-RS)	request indication	Indication of corrupted FEC-SSCS-SDU

(*) The necessity of Burst indication is FFS (it may not be necessary).

2.1.2.1 Primitives for the data transfer service

These primitives are FEC-SSCS-DATA request and FEC-SSCS-DATA indication. They are used for the data transfer. The following parameters are defined.

– Interface Data (ID)

This parameter specifies the interface data unit exchanged between the FEC-SSCS and the upper layer entity. The interface data is an integral multiple of one octet.

– More (FEC-SSCS-M)

This parameter specifies whether the Interface Data communicated contains a beginning/continuing of a FEC-SSCS-SDU or the end of FEC-SSCS-SDU. This parameter is mapped to and from the indication contained in CPCS-UU.

[Editor's note:

This parameter is a more bit within a packet.

One bit in the CPCS-UU is required.

]

– Burst Indication (FEC-SSCS-BI)

This parameter specifies whether the transmission of an FEC-SSCS-SDU in the beginning/continuing burst state, or in the end of burst state.

[Editor's note:

This parameter is a 'more' bit for a burst. A burst may correspond to a block of information that the application wants to transmit. For the example of a file transfer, the burst may correspond to the file to be sent. One bit in CPCS-UU is required. The necessity of this parameter is FFS. This parameter may be unnecessary.]

– FEC-SSCS-Loss Priority (FEC-SSCS-LP)

This parameter indicates the loss priority for the associated EC-SSCS-SDU. It can take only two values, one is for high priority and the other is for low priority. The FEC-SSCS-LP invoke shall always indicate high priority. The FEC-SSCS-LP signal will be mapped from the CPCS-LP parameter.

– FEC-SSCS-Congestion Indication (FEC-SSCS-CI)

This parameter indicates whether the associated FEC-SSCS-SDU has experienced congestion. This indication will be transparently transported to the upper layer entity. This parameter is mapped to and from the CPCS-CI parameter.

– FEC-SSCS Cell Loss indication (FEC-SSCS-CLI)

This parameter indicates whether the associated FEC-SSCS-SDU has experienced cell discarding. The usage of this parameter is up to the upper layer's decision. The semantic of the parameter value is FFS.

– FEC-SSCS Bit Error indication (FEC-SSCS-BEI)

This parameter indicates whether the associated FEC-SSCS-SDU did contain bit error(s). The use of this parameter is up to the upper layer's decision. The semantic of the parameter value is FFS.

– FEC-SSCS Signaling Indication (FEC-SSCS-SIGI)

This parameter indicates whether the associated FEC-SSCS-SDU contains a signaling message that is to be indicated to the peer entity (FEC-SSCS entity).

[Editor's note :

This information must be transparently transported between FEC-SSCS entities. Therefore, one bit in FEC-SSCS-UU is required for the use of this parameter.

– FEC-SSCS User-to-User Indication (FEC-SSCS-UU)

The parameter is transparently transported by the FEC-SSCS between upper layer entities.

– FEC-SSCS Reception Status (FEC-SSCS-RS)

This parameter indicates that the associated FEC-SSCS-SDU delivered may be corrupted (i.e. failure of error recovery). This parameter is used for the indication whether the FEC algorithm was performed. When the CPCS-RS indicates the reception of a corrupted CPCS-SDU, the FEC process is invoked.

2.1.2.2 Primitives for the abort service

For the abort service, the following primitives are used.

– FEC-SSCS-U-ABORT request and FEC-SSCS-U-ABORT indication

This primitive is used by the upper layer entity to invoke the abort service. It is also used by the upper layer entity to indicate that a partially delivered FEC-SSCS-SDU is to be discarded by instruction from its peer entity (i.e. the source upper layer entity). These primitives are mapped from and to the CPCS-U-ABORT invoke and CPCS-U-ABORT signal.

– FEC-SSCS-P-ABORT indication

This primitive is used by the FEC-SSCS entity to signal to the upper layer entity that a partially delivered FEC-SSCS-SDU is to be discarded due to the occurrence of an error in the FEC-SSCS or below the FEC-SSCS.

When the FEC-SSCS entity receives a CPCS-P-ABORT signal from the CPCS entity below, the CPCS-P-ABORT signal is mapped to the FEC-SSCS-P-ABORT signal.

2.2 Description of connections

Multiple data flows may be associated with a single FEC-SSCS connection. Within the FEC-SSCS connection, different data flows are not explicitly identified by an identifier. The method how to multiplex the multiple data flows of the upper layer into a single FEC-SSCS connection depends on the upper layer's decision. A negotiation protocol is required between the associated upper layer entities in order to support this type of multiplexing.

[Editor's note:

However, FEC-SSCS-UU may be used to provide a flow identifier for each data flow.

]

3 Interaction with the management and control plane

3.1 Management plane

for further study

3.2 Control plane

The parameters of the FEC algorithm can be negotiated both during FEC-SSCS connection set-up phase and during a connection. The parameter negotiation during a connection must be after/before the transmission of a burst, that is identified by FEC-SSCS-BI (Burst indication).

The inband signaling (i.e. control) message is generated by the upper layer. The inband signaling message is identified by the FEC-SSCS entity using FEC-SSCS-SIGI. There would be a set of default parameters for the FEC algorithm, but the parameters can also be specified and negotiated using signaling messages.

Before exchanging the signaling messages (i.e. inband control messages) between peer FEC-SSCS entities, the peer FEC-SSCS entities have to recognize that both entities provide FEC-SSCS functionality.

There are two alternatives for this recognition procedure.

- (a) Using an SSCS protocol identifier sub-field in Q.2931
- (b) Using a user-to-user message provided by Q.2931

The transmission of (inband) signaling messages is not protected by FEC.

[Editor's note :

When the (inband) signaling message should be protected by FEC, either default FEC parameters or a parameter negotiation with Q.2931 is required. In this draft, it is assumed for simplicity that the signaling messages are not protected by FEC.

]

Other service functions are FFS.

In order to allow control functions to use inband signaling messages, the semantic (i.e. the message type) of the signaling message has to be specified. This parameter is defined as SID (i.e. signaling semantics ID).

[Editor's note :

We may be able to define the assured mode operation as optional. The current specification does not provide an assured mode service (i.e. providing error-free data delivery). When the FEC algorithm can not recover the errored information, the corrupted data is delivered (or not

delivered) to the upper layer with error indication by RS parameter. When the FEC-SSCS entity contains sufficient buffer space, it may be possible to operate retransmission based error correction for the case where the FEC algorithm fails to recover the errored information. This optional retransmission scheme requires substantial implementation complexity for the SSSCS. For an upper layer with an additional retransmission scheme, detailed assessment of a retransmissions scheme in the SSSCS is required. The specification defined in this draft does allow future extensions of an optional retransmission mode.

]

3.2.1 Primitives for FEC parameter negotiation

The primitives for FEC parameter negotiation messages are the same as the primitives specified in 2.1.1 and in 2.1.2.

In addition to the exchange of Interface Data (ID) between the upper layer and the FEC-SSCS entity, the following primitives are defined. Both the upper layer and the FEC-SSCS entity may use these parameters for negotiation of the FEC parameters.

Table 3: Primitives for FEC parameter negotiation

Parameter	Type	Comments
Symbol Length indication (SYL)	request indication	8xn bits (n>0) are specified Format is FFS
Vertical Length indication (VL)	request indication	Usually indicating 46 octets length Format is FFS.
Maximum Length of Horizontal Line (MHL)	request indication	Must be less than $[2^{(8n)} - FCPL]$ (in symbol)
FEC-Code-Part Length indication (FCPL)	request indication	$2^{(SNL)}$ must be larger than the maximum length of FEC-code-part.
Selection of Operation Mode (SOM)	request indication	SOM=1 : Cell Loss is not recovered SOM=2 : Bit Error is not recovered SOM=3 : Both are recovered
Negotiation Status indication (NS)	request indication	NS=0h : hello polling NS=1h : hello positive acknwldgmnt NS=2h : hello negative acknwldgmnt NS=3h : polling parameters NS=4h : polling positive acknwldgmnt NS=5h : polling negative acknwldgmnt NS=6h : confirmation

(#): SNL is the length of sequence number field in the FEC-frame-header field.

These primitives are in the Interface Data (ID) of the FEC-SSCS-DATA request and the FEC-SSCS-DATA indication. They are used for the transmission of negotiation (signaling) messages associated with FEC parameters.

– Symbol Length indication (SYL)

This parameter specifies the length of a symbol defined in the FEC algorithm. The symbol is a vector containing N bits (e.g. $N=8m$, $m>0$). When the symbol length is N bits, the maximum length of user data and FEC's redundant data together must be less than 2^N symbols. A SYL of either 8 bits or 16 bits is recommended.

- Vertical Length indication (VL)
This parameter specifies the length of a vertical line in the FEC frame. VL may be represented as an octet. Usually, VL will indicate 46 octets.
- Maximum Length of Horizontal Line indication (MHL)
This parameter specifies the maximum number of vertical lines associated with user data in the FEC frame. MHL is represented as an integer, and must be less than $[2^{(N)}-FCPL]$. The sender or receiver process can specify this parameter, e.g. due to the requirement of buffer resource in receiver entity.
- FEC-Code-Part Length indication (FCPL)
This parameter specifies the number of vertical lines associated with the FEC redundant information part in the FEC frame. FCPL is represented as an integer.
- Selection of Operation Mode (SOM)
This parameter specifies which operational mode is applied. When SOM=1 (i.e. cell losses are not recovered), FEC is only used for bit error recovery (SEC: Symbol Error Correction Mode). When SOM=2 (i.e. bit errors are not recovered), only symbols missing due to cell loss are subject of FEC error recovery. (SLC: Symbol Loss Correction Mode) When SOM=3, both bit errors and cell loss are subject of FEC error recovery. (SEAL: Symbol Error and Loss Correction)
- Negotiation Status indication (NS)
This parameter specifies the status of negotiation between the associated peers. A Three way hand-shaking method is adopted to obtain a secure negotiation.

[Editor's note :

The following length in bits seem to be appropriate for these parameters: SYL=6 bits (for symbols of up to 64 bits), VL=6 bits (up to 64 symbols per vertical line), MHL>11 bits (for a frame size of up to 64KB for 64 octets of vertical line), FCPL=4 bits (16 vertical lines), SOM=2 bits and NS= 3 bits.

When we have an octet alignment, it would be SYL=8bits, VL=8bits, MHL=16bits, FCPL=8bits, SOM+NS=8 bit. The total length of this parameters would be 6 octets.

]

4 Functions, structure and coding of FEC-SSCS

The FEC-SSCS has the following service characteristics.

- In SEC mode, the receiving entity tries to recover bit errors in the user data part of the FEC SSSCS PDU. When bit errors are detected, the FEC error recovery procedure is not performed. The originally received data is either delivered or not delivered to the upper layer entity.
- In SLC mode, the receiving entity tries to recover cell loss in the user data part of the FEC-SSCS-PDU.
When bit errors in the user data part of an FEC-SSCS-PDU are detected, the originally received data is either delivered or not delivered to the upper layer entity.
- In SEAL mode, the receiving entity tries to recover both bit errors and cell losses in an FEC-SSCS-PDU.
- The source FEC-SSCS entity can adjust the actually transmitted redundant data length, without the FEC parameter renegotiation procedure.
This redundant data length adjustment can be performed by local decision of the FEC-SSCS source entity.

- The FEC-SSCS destination entity indicates to the upper layer whether the FEC-SSCS-SDU has experienced cell loss.
- The FEC-SSCS destination entity indicates to the upper layer whether the FEC-SSCS-SDU did contain bit error(s).
- FEC parameters can be negotiable both connection-by-connection and burst-by-burst.
- The FEC-SSCS source entity can transfer the CPCS-SDU (as well as the FEC-SSCS-SDU) by pipelining.

4.1 Functions of FEC-SSCS

Three operating modes are provided.

(1) SEC (Symbol Error Correction) mode

The receiving entity tries to recover bit errors.

When missing symbols due to cell discarding are detected, the error recovery process is not performed.

(2) SLC (Symbol Loss Correction) mode

The receiving entity tries to recover lost symbols. When either bit errors in the symbols are detected, or the recovery of missing symbols failed, the originally received data is either discarded or delivered to the upper layer entity.

(3) SEAL (Symbol Error And Loss correction) mode

The receiving entity tries to recover both bit errors and cell losses in an FEC-SSCS-PDU.

The functions provided by FEC-SSCS include :

(a) Preservation of FEC-SSCS-SDU

This function provides for the delineation and transparency of FEC-SSCS-SDUs.

(b) Preservation of FEC-SSCS-User-to-User information

This function provides for the transparent transfer of FEC-SSCS user-to-user information.

(c) Error correction, and error handling

This function provides for the correction and handling of FEC-SSCS-PDU (=CPCS-SDU) corruption.

Errors of an FEC-SSCS-PDU are detected using the error detection capability of the CPCS entity. When the received FEC-SSCS-PDU contains bit errors, the FEC-SSCS entity tries to recover errors using the FEC algorithm. When the received FEC-SSCS-PDU does not contain any error, the FEC-SSCS entity transfers the received FEC-SSCS-PDU while deleting the redundant information of the FEC-SSCS-PDU.

If the FEC algorithm cannot correctly recover the original FEC-SSCS-SDU sent by the source FEC-SSCS entity, the FEC-SSCS-SDU is either discarded or is optionally delivered to the upper layer entity with error indication.

(d) Creation of FEC frame

This function provides for the creation of an FEC frame, that includes some redundant information to recover the error at the destination FEC-SSCS entity.

The FEC frame has a two dimensional matrix structure. Each FEC contains header fields for every vertical line and FEC's redundant information field.

(e) Abort

The function provides for the means to abort a partially delivered FEC-SSCS-SDU.

(f) Handling of congestion indication

This function provides for the passing of congestion information between the layers above the FEC-SSCS and the layer below in both direction.

- (g) Handling of loss priority information
This function provides for the passing of loss priority information between the layer above FEC-SSCS and the layer below in both direction.
- (h) Handling of cell loss indication
This function provides for the indication whether the received FEC-SSCS-SDU has experienced cell loss.
- (i) Handling of bit error indication
This function provides for the indication whether the received FEC-SSCS-SDU did contain bit error(s).
- (j) FEC parameter negotiation
This function provides for the negotiation of FEC parameters. The FEC parameters are negotiated burst-by-burst during a session, as well as connection-by-connection. The three way hand-shaking is adopted for the negotiation.

Other functions are FFS.

4.2 FEC-SSCS-PDU and FEC-SSCS-SDU structure and coding

4.2.1 Abstraction of FEC-SSCS-PDU and FEC-SSCS-SDU Structure

The FEC-SSCS-SDU has a trailer, which includes a FEC-CPCS-UU field. However, the FEC-SSCS-SDU does not have padding, length indicator, or error detection field. 48 octet alignment is provided using the functionality of the CPCS entity.

One FEC-SSCS-SDU may correspond to one IP packet, and one burst of FEC-SSCS-SDUs may correspond to a bulk of data (e.g. a user file) to be transmitted.

An FEC-SSCS-SDU generally generates multiple FEC frames from one FEC-SSCS-SDU. It attaches the FEC-code-part and the header fields to every vertical lines. One FEC frame may be separated into two CPCS-SDUs. Then, one of two CPCS-SDUs contains the user-data-part, and the other contains the FEC-code-part.

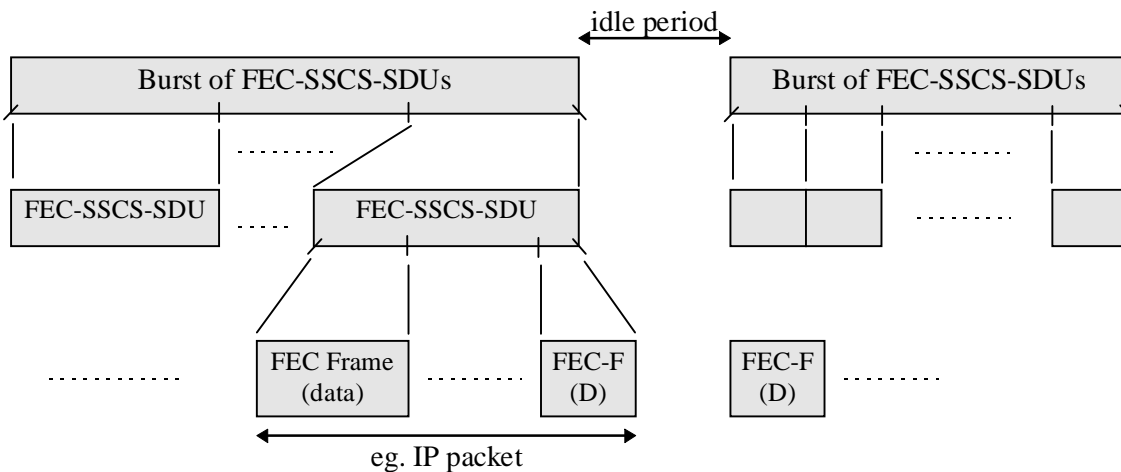


Figure 2 Data Structure Reference Model
(Burst / FEC-SSCS-SDU / FEC-Frame)

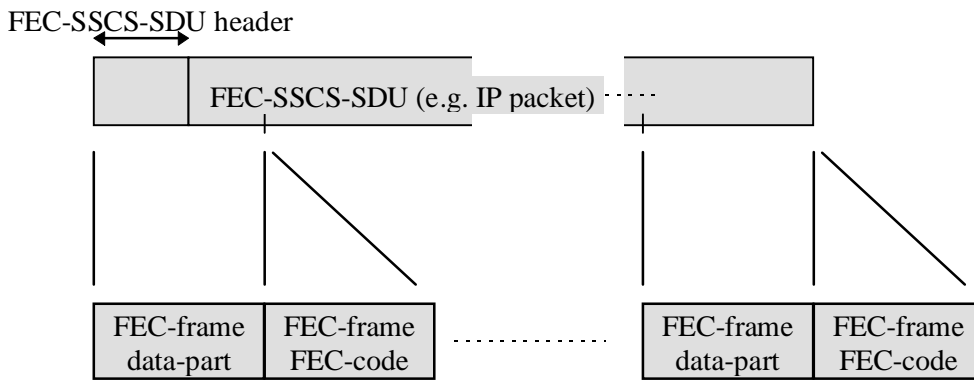


Figure 3: Data Structure Reference Model
(FEC-SSCS-SDU / FEC-Frame)

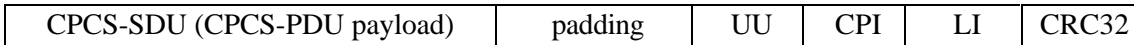


Figure 4: Data Structure Reference Model (CPCS-PDU)

4.2.2 Frame format of FEC-SSCS-PDU

The FEC-SSCS-PDU consists of the FEC-SSCS-SDU and the header that contains the FEC-SSCS-UU parameter. The maximum size of an FEC-SSCS-SDU is for further study.

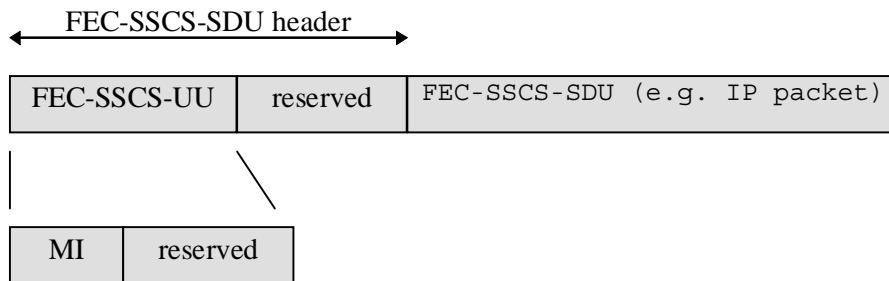
[Editor's note :

There would be no upper limit associated with the maximum size of an FEC-SSCS-SDU. Since the FEC-SSCS-SDU could correspond to IP packet in TCP/IP, the maximum size of an FEC-SSCS-SDU would be equal to IP's MTU size.

]

1. FEC-SSCS-UU field

The FEC-SSCS-UU field is used to transparently transfer the FEC-SSCS user-to-user information.



(1)

Figure 5 FEC-SSCS-PDU format

The MI (Message Identifier) field has length of one bit. This MI is mapped from FEC-SSCS-SIGI, that indicate the message type of the FEC-SSCS-SDU, i.e. the peer-to-peer signaling message or user-data message. MI=0 indicates that the FEC-SSCS-SDU contains a signaling message, and MI=1 indicates that the FEC-SSCS-SDU contains a user-data message.

The size and coding rule of FEC-SSCS-UU and the reserved field is FFS.

4.2.3 Frame format of FEC frame

The format of FEC frame, that is actually transmitted from the source entity to the destination entity is described in Figure 6.

The FEC frame consists of a user-data-part and FEC-code-part. Each part has a FEC-frame-header field. The user-data-part contains a part (or whole) of the FEC-SSCS-SDU and the attached FEC-frame header field.

The FEC-code-part contains the FEC redundant information field and the attached header field.

(1) user-data field

The user-data field is used to carry a part (or whole) of an FEC-SSCS-SDU. "n", the length in vertical direction, is determined so that the length of the vertical line in the FEC frame, $\{(t,1),(t,2),(t,3),\dots,(t,n)\}$, is $46/a$ octet. Here, "a" is a integer larger than zero which is usually one. In other words, the length of the vertical line in FEC frame is usually 48 octet, that is the payload length of AAL5 SAR-PDU.

The maximum length of the horizontal line in the user-data field is specified by the MHL parameter in the signaling message specified in section 3.2. The maximum length of the horizontal line in the user-data must be less than $[2^{\text{SYL}} - \text{FCPL}]$ symbols. Here, SYL is the symbol size in bit, and FCPL is the horizontal length of FEC-code-part (s).

(2) FEC redundant data field

The FEC redundant data field is used to carry the appended data to protect the user-data field information against bit error and cell loss. The data in the FEC redundant data field is calculated based on the defined FEC algorithm with the negotiated parameter (i.e. SYL and FCPL). The calculation algorithm of the FEC redundant data vector, $\{(M+1,t),(M+2,t),\dots,(M+s,t)\}$, is specified in section 4.3. The length of the FEC redundant data vector ("S" symbols), is given by the FCPL parameter in the signaling message specified in section 3.2.

(3) FEC-frame-header field

The FEC-frame-header field is used to identify the position in the associated x-axis of a missing or errored vertical line and to detect bit errors in each vertical line.

The length of the FEC-frame-header is 2 octets in each vertical line.

The detailed format and coding rule of the FEC-frame-header field is specified in section 4.2.4.

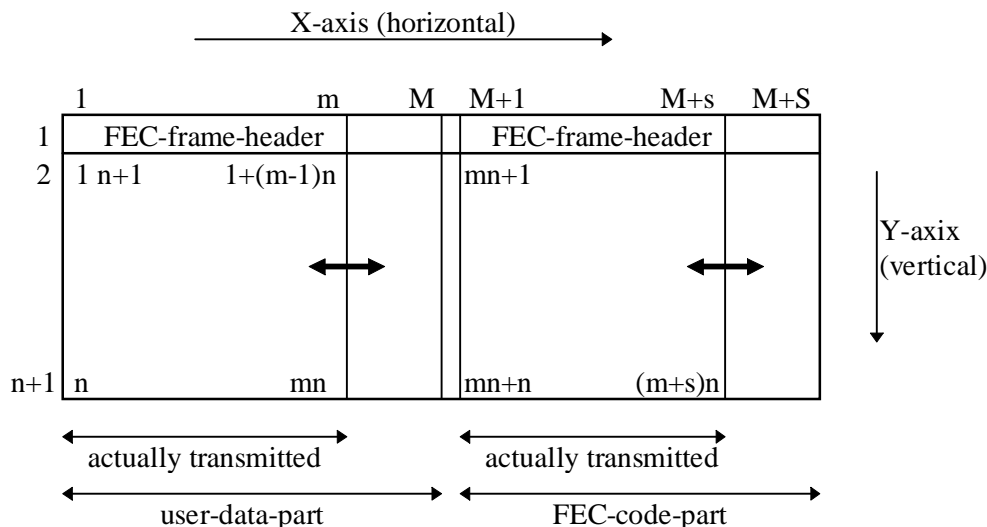


Figure 6 Structure of FEC frame

* The matrix element in the FEC frame is presented by (x,y). And each element corresponds to a symbol (e.g. 8 bits) specified in the FEC coding algorithm (i.e. the Reed Solomon coding scheme).

a) Writing and reading order in FEC frame

In order to provide the above functionality, the following field and coding rule is provided. All fields exist in each vertical line in the FEC frame. The length of the FEC-frame-header field of each vertical line is 2 octets.

(i) Sequence Number (SN) field

The SN field is used to identify the position of the vertical line associated with its horizontal direction.

When the length of the SN field is "q", 2^q must be larger than FCPL, the number of vertical lines in the FEC-code-part. The SN of the first vertical line both in user-data-part and in FEC-code-part always starts from zero (0).

[Editor's note:

SN would be either 5 bits or 4 bits.

]

[Editor's note:

The difference of the following two methods is the numbering rule of SN.

]

(ii) User/FEC (U/F) field

The U/F field is used to identify which part, i.e. user-data-part or FEC-code-part, the received vertical line belongs to. The U/F field has one bit and takes two values. "U" represents user-data-part, and "F" represents FEC-code-part. The U/F field in user-data-part is always "U", and the U/F field in FEC-code-part is always "F".

(iii) Parity (P) field

The P field is used to perform the parity check for the FEC-frame-header field except the CRC field. The purpose of the P bit is to detect bit errors in the FEC-frame-header field. A single bit error in the FEC-frame-header field except the CRC field can be detected.

[Editor's note :

The use of Parity bit is FFS.]

(iv) CRC field

The CRC field is used to detect bit errors in the vertical lines.

Below, the format of the FEC-frame-header field and an example of field pattern is described.

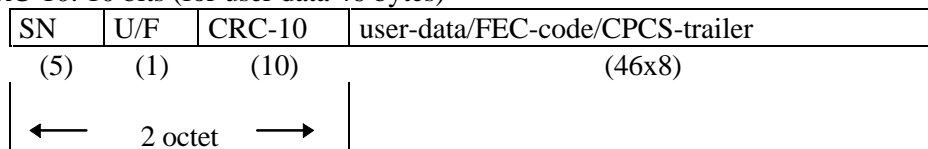
[Editor's note :

At this time, two solutions are described.

]

[1] Solution #1

- SN: 5 bits (32 cycles sequential number)
- U/F: 1 bit (user-data-part/FEC-code-part)
- CRC-10: 10 bits (for user data 46 bytes)

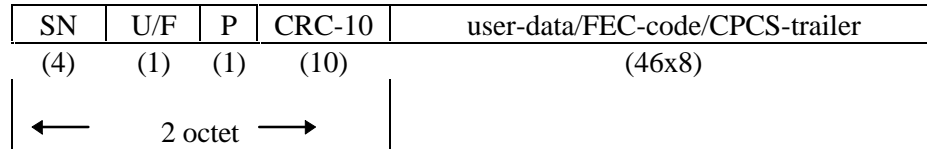


The sequence number SN is not reset during the transmission of FEC-SSCS-SDU.

CPCS-SDU	FEC	user-data-part	FEC	data-part
SN	2 3 4	5 6 7 --- 31 0 1 --- 30 31	0 1 2 3	4 5 6 7 --
U/F	F F F	U U U --- U U U --- U U	F F F F	U U U U --

[2] Solution #2

- SN : 4 bits (16 cycles sequential number)
- U/F : 1 bit (user-data-part/FEC-code-part)
- P : 1 bit (parity check for SN+U/F)
- CRC-10 : 10 bits (for user data 46 bytes)



The sequence number SN always starts from zero at the beginning of user-data-part.

CPCS-SDU	FEC	user-data-part	FEC	data-part
SN	2 3 4	0 1 2 --- 15 0 1 --- 15 0 1	2 3 4 5	0 1 2--
U/F	F F F	U U U --- U U U --- U U	F F F F	U U U --

4.2.5 Frame format of CPCS-PDU

The method to transport the FEC-frame is specified in this section. In order to avoid the large padding field in the CPCS-PDU for the transmission of FEC-code part field, the following FEC frame transmission method is applied.

If the FEC frame (which will have 48 octets alignment) is transmitted as a single CPCS-SDU, each CPCS-SDU has 40 octets padding field.

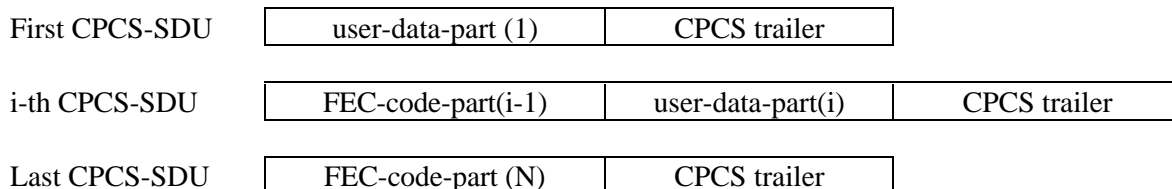
[Editor's note:

At this time, three alternatives are proposed.

One method should be selected.

]

(a) Format A



The length of FEC-code-part(i) is indicated by the #RedCell field at CPCS-UU in (i-1)-th CPCS trailer.

[Editor's note:

Since the length of FEC-code-part(i-1) can be adjusted by the source entity without a FEC parameter negotiation procedure, the length of FEC-code-part(i-1) must be indicated in every CPCS-SDU transmission.

]

(a) Format B

CPCS-SDU(user-data)	user-data-part	CPCS trailer
---------------------	----------------	--------------

CPCS-SDU(FEC-code)	FEC-code-part	CPCS trailer
--------------------	---------------	--------------

**The last cell of CPCS-SDU(user-data)

user-data-part	padding	UU	CPI	LI	CRC-32
----------------	---------	----	-----	----	--------

**The last cell of CPCS-SDU(FEC-code)

FEC-code-part	padding	UU	CPI	LI	CRC-32
---------------	---------	----	-----	----	--------

The FEC-code-part in the last cell of CPCS-SDU (FEC-code) will be null, when the vertical line of FEC frame is exactly the same length as the cell length.

This means this format always requires an additional cell that includes 40 cell padding field (i.e. no user data in the cell).

The length of FEC-code-part is indicated by the LI field in in CPCS trailer.

[Editor's note :

Since the length of FEC-code-part can be adjusted by the source entity without FEC parameter negotiation procedure, the length of FEC-code-part must be indicated in every CPCS-SDU transmission.

]

(a) Format C

CPCS-SDU(user-data)	user-data-part	FEC-code(#)	CPCS trailer
---------------------	----------------	-------------	--------------

CPCS-SDU(FEC-code)	FEC-code-part	CPCS trailer
--------------------	---------------	--------------

**The last cell of CPCS-SDU(user-data)

user-data-part	FEC-code(#)	padding	UU	CPI	LI	CRC-32
----------------	-------------	---------	----	-----	----	--------

*The last cell of CPCS-SDU(FEC-code)

FEC-code-part	padding	UU	CPI	LI	CRC-32
---------------	---------	----	-----	----	--------

FEC-code(#) has a length of exactly 8 bytes, which is the length of the AAL5 CPCS trailer field. The FEC-code(#) is the last 8 bytes of the FEC-code-part : the last 8 bytes in the last vertical line in the FEC-code-part. This means that, when the vertical line of the FEC frame is exactly the same length as the cell length, the FEC-code-part in the last cell of CPCS-SDU(FEC-code) is 40 bytes and there is no padding field.

The data transmission order of the FEC-code part is little changed in this solution.

After the transmission of FEC-code(#), which is the last 8 bytes in the FEC-code-part, the data transmission is started from the first vertical line in the FEC-code-part. This means that only the last 8 bytes of the FEC-code-part are transmitted in the CPCS-SDU(user-data). Therefore, the vertical line will be exactly in one cell for CPCS-SDU(FEC-code), when the length of the vertical line and the cell length is exactly the same length. This solution would require additional implementation complexity, but it allows to avoid having a always 40 bytes padding cell at the end of an FEC-frame, even when it is defined that the vertical line length is exactly the same as the cell length.

Additionally, this allows to keep the alignment between the vertical line and cell in the FEC-code-part.

The length of FEC-code-part is indicated by the LI field in in CPCS trailer.

[Editor's note :

Since the length of FEC-code-part can be adjusted by the source entity without FEC parameter negotiation procedure, the length of FEC-code-part must be indicated in every CPCS-SDU transmission.

]

The coding rule of CPCS-UU is specified below.

0	1	2	3	4	5	6	7
M	BI	Res		#RedCell			

Figure: Coding of CPCS-UU

(1) FEC-SSCS-SDU More Indication (M) field

This field is used to indicate the end of the FEC-SSCS-SDU from the source FEC-SSCS entity to the destination FEC-SSCS entity.

The M indication is transparently transported. M=0 represents the end of an FEC-SSCS-SDU, and M=1 represents the beginning or continuation of an FEC-SSCS-SDU.

(2) Burst Indication (BI) field

This field is used to indicate the end of a burst from the source upper layer entity to the destination upper layer entity.

The BI indication value is transparently transported. BI=0 represents the end of the burst, and BI=1 represents the beginning or continuing of burst. BI field value may be examined in FEC-SSCS entity.

[Editor's note:

The necessity of the BI bit is FFS. The BI bit may be unnecessary.

]

(3) Number of Redundant Cells (#RedCell) Field

This field is used to indicate the number of redundant cells in the subsequent CPCS-SDU. The #RedCell indication value is transparently transported.

[Editor's note:

The necessity of #RedCell is FFS. A length of 4 bits would be preferable.

]

4.2.6 Frame format of Signaling Message for FEC parameter negotiation

The following fields are specified as a signaling message (FEC-SSCS-SDU) between the source upper layer entity and the destination upper layer entity, in order to negotiate the FEC parameters.

The BI field of CPCS-UU for the FEC-SSCS-SDU must be 0, and the MI field must be 1.

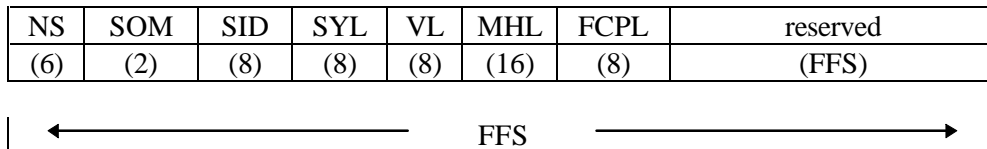
These are the fields used for the transmission of negotiation (signaling) messages associated with FEC parameters.

– Message Semantics identifier (SID)

This field specifies the semantics (i.e. message type) of the following field information.

The SID field has a length of one octet. The value in the SID field shall be zero (0).

- Symbol Length (SYL) indication field
This field specifies the length of a symbol defined in the FEC algorithm. The symbol is a vector containing $8 \times n$ bits ($n > 0$).
The value of "n" is indicated in the SYL field. The SYL field has a length of one octet.
- Vertical Length (VL) indication field
This field specifies the length of a vertical line in the FEC frame. VL may be represented by one octet. Usually, VL may indicate 46 octets to have an alignment with cell length.
The VL field has a length of one octet.
- Maximum Length of Horizontal Line indication (MHL)
This parameter specifies the maximum number of vertical lines associated with user data in the FEC frame. MHL is represented as an integer, and must be less than $[2^N - 1 - FCPL]$. The sender or receiver process can specify this parameter, e.g. according to buffer requirements in the receiver entity.
- FEC-Code-Part Length (FCPL) indication field
This field specifies the number of vertical lines associated with the FEC redundant information part in the FEC frame. FCPL is represented by an integer.
The FCPL field has a length of one octet.
- Selection of Operation Mode (SOM) indication field
This field specifies which operational mode is applied.
When SOM=1 (i.e. cell loss is not recovered), only symbols with bit errors are subject to FEC's error recovery.
When SOM=2 (i.e. bit error is not recovered), only missing symbols due to cell loss are subject to FEC's error recovery.
When SOM=3, both missing symbols and symbols with bit errors are subject to FEC's error recovery.
The SCLP field has a length of two bits.
- Negotiation Status (NS) indication field
This field specifies the status of negotiation between the associated peers. A three way hand-shaking method is adopted to obtain a secure negotiation.
The NS field has a length of six bits.



4.3 FEC algorithm

4.3.1 Symbol length, FEC-frame size, and Correction capability

In the FEC frame specified in section 4.2.3, one Reed-Solomon block (RS-block) corresponds to one horizontal line. Therefore, the FEC frame has n RS-blocks. The use of other FEC schemes is FFS.

[Editor's note :

When we support more than one FEC scheme, we must have some protocol and message to negotiate which scheme is used associated with the establishing connection.

The method of FEC scheme negotiation is also FFS.

]

The FEC scheme uses a Reed-Solomon ($M, M+S$) code which is able to correct up to $[S/2]$ errored symbols or s erased (missing) symbols. Here, m is equal to or smaller than M , and s is equal to or smaller than S , i.e., $m \leq M$ and $s \leq S$.

m and s are the number of vertical lines in FEC frame defined in 4.2.3. m is the number of actually transmitted vertical lines for the data-part, and s is the number of actually transmitted vertical lines for the FEC-code-part.

Therefore, m+s vertical lines are actually transmitted from the source entity to the destination entity.

The RS(M, M+S) code can correctly recover the received data when the following equation is satisfied.

$$S \geq a + 2b$$

Here, "a" is the number of erased (missing) symbols, and "b" is the number of errored symbols, whose position in the FEC frame is not identified.

Since the specified FEC frame has the CRC's bit error detection capability in each vertical line, the actual error correction capability of the FEC algorithm based on an RS(M,M+S) code is as followed.

$$S \geq a + b$$

The symbols in a vertical line are assumed to be erased when bit errors occurred in this cell.

This means that the erasure mode is used for the correction of dummy symbols corresponding to cell loss locations.

If the total number of vertical lines with bit errors is larger than "s", then the RS-Code may be used for bit error detection in every row of symbols. In this case, the FEC frame can be recovered as long as not more than s/2 symbols per row are errored. Implementation of the RS-Code's bit error detection is not mandatory and depends on a vendor's decision. Also, the receiving entity of FEC-SSCS may use the error correction capability of CRC-10 to correct a single bit error per cell. (This as well would be an implementation issue and not subject to standardization.)

Reed-Solomon codes to be used are built over Galois Field (2^N), and the generator polynomial G(X) is given by :

$$G(X) = \prod_{i=0}^{S-1} X - \alpha^{i+k}$$

where alpha is a root of the primitive polynomial, N is the size of a symbol in bits, and k is the base exponent of the generator polynomial.

The specific generator polynomials to be used for the Reed-Solomon code are FFS. The generator polynomial depends on the size of the symbols and the number of appended redundant symbols.

The symbol length is determined by (i) feasibility of implementation, and (ii) the allowable maximum number of (M+S).

From the view point of implementation feasibility, the symbol length should be 4 bits, 8 bits, 16 bits, 32 bits and 64 bits. The allowable maximum number of symbols in an RS-block is given by $2^N - 1$. Here, N is the symbol length in bit.

Therefore, when the user-data in the vertical line is 46 octet, the maximum allowable size of data (i.e. user-data-part + FEC-code-part) is given by the following table.

Symbol length	Max. of M+S	Max of User-data + FEC-code part
4 bits	15	690 Byte
8 bits	255	1.73 KByte
16 bits	65,535	3.015 MByte
32 bits	4.30x10 ⁹	197.57 GByte
64 bits	1.85x10 ¹⁹	8.51x10 ²⁰ Byte

Table 4. Symbol Length and FEC Frame Maximum Size

[Editor's note:

For the current IP (IPv4), 4 bit symbols or 8 bit symbols would be preferable.

For an ATM environment, 8 bit symbols or 16 bit symbols would be preferable.

16 bit symbols can cover the maximum SDU size of AAL type 5 by a single FEC frame. 8 bit symbols can cover the default MTU for IP by a single FEC frame.

]

[Editor's note:

Some default value or well-known value should be specified, in order to reduce the implementation complexity.

A likely default number of appended redundant symbols would be one for small FEC-SSCS-PDUs (=FEC frames), and would be four for larger FEC-SSCS-PDUs.

]

4.3.2 Calculation of FEC redundant symbols

The FEC redundant symbols are calculated by the specified generator polynomial and user-data symbols.

Generator Polynomial: $G(X)$

RS-block: $\{C[M + S - 1], C[M + S - 2], \dots, C[S], C[S - 1], \dots, C[1], C[0]\}$

User-data: $\{U[m], U[m - 1], \dots, U[1]\}$

FEC-data: $\{F[s - 1], F[s - 2], \dots, F[0]\}$

The transmitted data is User $\{U[m], U[m - 1], \dots, U[1]\}$ and FEC-data $\{F[s - 1], F[s - 2], \dots, F[0]\}$ m is smaller than or equal to M , and s is smaller than or equal to S .

$C[M + s - i]$ ($1 \leq i \leq M$) is given by the following equation.

$$C[M + s - i] = \begin{cases} U[m - i + 1] & (1 \leq i \leq m) \\ P & (m \leq i \leq M) \end{cases}$$

Here, symbol P is a predetermined symbol that is shared between the source and the destination FEC-SSCS entity.

The symbols $C[i]$ ($m < i \leq M$) are only used for the calculation of FEC-data, and are not transmitted to the destination FEC-SSCS entity.

$\{C[s - 1], C[s - 2], \dots, C[0]\}$ is given by the following equation.

$$M(X) * X^S = Q(X) * G(X) + C[S - 1] * X^S + C[S - 2] * X^{S-1} + \dots + C[1] * X + C[0]$$

Here,

$$M(X) = C[M + S - 1] * X^{(M-1)} + C[M + S - 2] * X^{(M-2)} + \dots + C[S + 1] * X + C[S]$$

$F[i]$ ($0 \leq i \leq s - 1$) is given by the following equation.

$$F[s - i] = C[S - i] \quad (1 \leq i \leq s)$$

4.3.3 FEC Error Recovery Algorithm

The correct symbols are obtained by the following calculation, when the received symbol vector is $\{R[m+s-1], R[m+s-2], \dots, R[s], R[s-1], \dots, R[1], R[0]\}$. The received symbol vector have missing symbols and symbols with bit errors. The positions of missing symbols and symbols with bit errors can be identified by the sequence number (SN).

Here, the errored/missing symbols in the user-data part are represented by

$\{EU[1], EU[2], \dots, EU[q]\}$ ($1 \leq q \leq m$), and the errored/missing/not transmitted symbols in FEC-code part are represented by $\{EF[1], EF[2], \dots, EF[r]\}$ ($S-s \leq r \leq S$). $EF[i]$ ($1 \leq i \leq S-s$) are the symbols in FEC-code part that are not actually transmitted from the source entity, and $EF[i]$ ($S-s < i \leq r$) are the symbols in FEC-code part, that are errored/missed during cell transmission.

[Editor's note:

The symbols in FEC-code part, that are not actually transmitted from the source entity, $EF[i]$ ($1 \leq i \leq S-s$) are assumed as the missing/errored symbols in FEC error recovery algorithm.

Execution of the FEC error recovery algorithm is not triggered when there are no errored/missing symbols in user-data part, even when there are errored/missing/not transmitted symbols in the FEC-code part.

]

When $q+r$ is equal to or smaller than S , the errored user data $\{EU[1], EU[2], \dots, EU[q]\}$ can be correctly recovered. The errored/missing symbols, ($R[x_i]$ ($1 \leq i \leq q+r$)), are the unknown variables, and are obtained by the following equation:

$$R(X) = Q(X) * G(X)$$

Here,

$$R(X) = \sum_{i=1}^m R[m+s-i] * X^{(M+S-i)} + \sum_{i=m+1}^M P * X^{(M+S-i)} - \sum_{i=1}^s R[s-i] * X^{(S-i)} - \sum_{i=s+1}^S EF(i-s) * X^{(S-i)}$$

The generator polynomial ($G(X)$) and the pre-determined symbol (P) are shared between source and destination FEC-SSCS entities. Since the $G(X)$ has " S " independent solutions, $\{a[1], a[2], \dots, a[S]\}$, the following " S " equations are obtained.

$$R(a[i]) = 0 \quad [\text{where, } G(a[i]) = 0, \quad 1 \leq i \leq S]$$

[Editor's note:

How to implement this algorithm is an implementation issue.

]

Since the length of the user-data-part is indicated by LI (Length Indicator) in the CPCS-trailer, the destination FEC-SSCS entity can identify the originally transmitted user-data $\{U[m], U[m-1], \dots, U[1]\}$ ($m \leq M$).

5 Procedure for the FEC-SSCS sublayer

5.1 State Variables

FFS.

5.2 Procedure in User-Data Transfer Phase

The procedure in user data transfer phase is specified by pseudo-code.

[Editor's note:

The pseudo-code is assuming Format A.

]

5.2.1 Procedure at the FEC-SSCS Source Entity

The pseudo-code at the sender process of the FEC-SSCS entity is specified.

```
<<START>>
Receive FEC-SSCS-SDU (partial part of SDU) ;
While (BI==0) # in the burst status
{
  While [(number of vertical line < MHL) .AND. (FI=0) == true]
  {
    Calculate_FEC-code-part #1 (user-data, FEC-code-part);
    Calculate_AAL5-Trailer (vertical line, CRC-32, LI);
    Create/send_vertical-line_#1 (SN,U/F,P,CRC-10,user-data);
    Receive FEC-SSCS-SDU (next partial part of SDU);
  }
  Calculate_FEC-code-part_#2 (user-data, FEC-code-part);
  Calculate_AAL5-Trailer (vertical line, CRC-32, LI);
  Create/send_vertical-line_#2 (SN,U/F,P,CRC-10,user-data,CRC-32,LI);
  While (number of vertical line < FCPL)
  {
    Calculate_AAL5-Trailer (vertical line, CRC-32, LI);
    Create/send_vertical-line_#1 (SN,U/F,P,CRC-10,FEC-code);
  }
  if (FI == 1)
  {
    Calculate_AAL5-Trailer (vertical line, CRC-32, LI);
    Create/send_vertical-line_#2 (SN,U/F,P,CRC-10,FEC-code);
  }
}
if [(BI == 1) .AND. (FI == 0)]
{
  ERROR;
}
<<STOP>>
```

[Procedure of Functions]

(a) Create/send_vertical-line_#1 (SN,U/F,P,CRC-10,data);

Input is either user-data or FEC-code in the full length of the single vertical line (i.e. typically 46 octet).

Output is the vertical line with complete header field (i.e. SN,U/F,P and CRC-10).

In the SLC mode, the calculation of CRC-10 can be skipped.

(b) Create/send_vertical-line_#2 (SN,U/F,P,CRC-10,data);

Input is either user-data or FEC-code in the full or partial length of the single vertical line.

Output is the full/partial length of vertical line with complete header field (i.e. SN,U/F,P and CRC-10). In the SLC mode, the calculation of CRC-10 can be skipped.

- (c) Calculate_FEC-code-part_#1 (user-data, FEC-code-part) ;
 Input is the user-data in the full length of single vertical line (i.e. typically 46 octet).
 Output is the vertical line of FEC-code-part (i.e. FCPL of vertical lines).
- (d) Calculate_FEC-code-part_#2 (user-data, FEC-code-part) ;
 Input is the user-data in either the full or partial length of a single vertical line.
 When the input user-data is not the full length of the vertical line, the symbol pattern of the remaining part is assumed to be some predefined pattern for the FEC calculation.
 Output is the vertical line of FEC-code-part (i.e. FCPL of vertical lines).

5.2.2 Procedure at the Receiver FEC-SSCS Entity

The pseudo-code at the receiver process in FEC-SSCS entity is specified.

5.2.2.1 SEC (Symbol Error Correction) mode operation

```

<<START>>
Error[p] = no-error;
Missing[p] = not-missing
Receive CPCS-SDU;
While (BI==1) # in the burst status
{
  While (FEC-SSCS-M==1)
  {
    Check_CRC-32;
    Check_LI;
    If [(check_CRC-32) == no-error]
    {
      Error[c]=non-error;
      Send_User-Data (user-data-part);
      if (Error[p] == error)
      {
        FEC_Recovery_Process (SEC);
      }
    }
  }
  else
  {
    Error[c]=error
    if [(Check_LI) == no-error]
    {
      if (Error[p] == error)
      {
        FEC_Recovery_Process (SEC);
      }
    }
  }
  else
  {
    Check_SN-User-Data;
    if (Missing[p] == not-missing)
    {
      Check_SN-FEC-Code;
    }
  }
  if (Check_SN-User-Data == missing)
  {
    FrameError [p];
    Missing[c]=missing;
  }
  else
  {
    Missing[c]=not-missing;
    Check_Error-User-Data;
    if (Check_Error-User-Data == no-error)
  }
}

```

```

        {
            Send_User-Data (user-data-part);
            Error[c]=non-error;
        }
        else
        {
            Error[c]=error;
        }
    }
    if (Missing[p] == not-missing)
    {
        Check_SN-FEC-Code;
        if (Check_SN-FEC-Code == not missing)
        {
            FEC_Recovery_Process (SEC);
        }
    }
}
Error [p]=Error [c];
Missing [p]=Missing [c];
Receive CPCS-SDU;
}
}

```

<<STOP>>

[Procedure of Functions]

- a) Error [i] (i=p,c)
This parameter indicates whether the FEC frame has an error or not. i=p represents the previous FEC frame, and i=c represents the current FEC frame.
- b) Missing[i] (i=p,c)
This parameter indicates whether the FEC frame has a missing symbol or not. i=p represents the previous FEC frame, and i=c represents the current FEC frame.
- c) FEC_Recovery_Process (SEC)
This function represents the FEC error recovery process operating in SEC mode.
- d) Check_SN-User-Data
This function indicates whether the user-data part has a cell loss using SN check.
- e) Check_SN-FEC-code
This function indicates whether the FEC-code-part has a cell loss using SN check.
- f) Check_Error-User-Data
This function indicates whether the user-data-part has a bit error by using CRC-10's error check sequence.

5.2.2.2 SLC (Symbol Loss Correction) mode operation

```

<<START>>
Error [p] = no-error;
Missing [p] = not-missing
Receive CPCS-SDU;
While (BI==1) # in the burst status
{
    While (FEC-SSCS-M)
    {
        Check_CRC-32;
        Check_LI;
        If [(check_CRC-32) == no-error]
        {
            Error [c]=non-error;
        }
    }
}

```

```

    Send_User-Data (user-data-part);
    if (Missing[p] == missing)
    {
        FEC_Recovery_Process (SLC);
    }
}
else
{
    Error[c]=error
    if [(Check_LI) == non-error]
    {
        FrameError[p];
    }
    else
    {
        Check_SN-User-Data;
        if [(Error[p] == no-error).AND.(Missing[p] == missing)]
        {
            Check_SN-FEC-Code;
        }
        if (Check_Error-User-Data == error)
        {
            FrameError[p];
            Error[c]=error;
        }
        else
        {
            Missing[c]=missing;
            Check_SN-User-Data;
            if (Check_SN-User-Data == not-missing)
            {
                Send_User-Data (user-data-part);
                Missing[c]=not-missing;
            }
            else
            {
                Missing[c]=missing;
            }
        }
        if (Error[p] == no-error)
        {
            Check_Error-FEC-Code;
            if (Check_Error-FEC-Code == no-error)
            {
                FEC_Recovery_Process (SLC);
            }
        }
    }
}
Error[p]=Error[c];
Missing[p]=Missing[c];
Receive CPCS-SDU;
}
}
<<STOP>>

```

[Procedure of Functions]

- a) Error [i] (i=p,c)
This parameter indicates whether the FEC frame has an error or not. i=p represents the previous FEC frame, and i=c represents the current FEC frame.
- b) Missing[i] (i=p,c)
This parameter indicates whether the FEC frame has a missing symbol or not. i=p represents the previous FEC frame, and i=c represents the current FEC frame.
- c) FEC_Recovery_Process (SLC)

This function represents the FEC error recovery process operating in SLC mode.

d) Check_SN-User-Data

This function indicates whether the user-data part has a cell loss using SN check.

e) Check_SN-FEC-code

This function indicates whether the FEC-code-part has a cell loss using SN check.

f) Check_Error-User-Data

This function indicates whether the user-data-part has a bit error by using CRC-10's error check sequence.

5.2.2.3 SEAL (Symbol Error And Loss Correction) mode operation

```
<<START>>
Error[p] = no-error;
Missing[p] = not-missing
Receive CPCS-SDU;
While (BI==1) # in the burst status
{
  While (FEC-SSCS-M==1)
  {
    Check_CRC-32;
    If [(check_CRC-32) == no-error]
    {
      State[c]=no-error;
      Send_User-Data (user-data-part);
      if (State[p] == error)
      {
        FEC_Recovery_Process(SEAL);
      }
    }
  }
  else
  {
    Check_SN-User-Data;
    Check_Error-User-Data;
    if { [(Check_SN-User-Data).OR.(Check_Error-User-Data)] == error}
    {
      FEC_Recovery_Process(SEAL);
      State[c]=error;
    }
    else
    {
      State[c]=no-error;
      Send_User-Data (user-data-par);
    }
  }
  if [(State[p] == error)
  {
    FEC_Recovery_Process(SEAL);
  }
}
State[p]=State[c];
Receive CPCS-SDU;
}

<<STOP>>
```

[Procedure of Functions]

a) State [i] (i=p,c)

This parameter indicates whether the FEC frame has an error (i.e. either contains a symbol with bit errors, or missing symbols) or not. i=p represents the previous FEC frame, and i=c represents the current FEC frame.

- b) FEC_Recovery_Process (SEAL)
This function represents the FEC error recovery process operating in SEAL mode.
- c) Check_SN-User-Data
This function indicates whether the user-data part has a cell loss using SN check.
- d) Check_SN-FEC-code
This function indicates whether the FEC-code-part has a cell loss using SN check.
- e) Check_Error-User-Data
This function indicates whether the user-data-part has a bit error by using CRC-10's error check sequence.

5.2.3 Procedure of FEC Error Recovery

```

<<< Start >>>
  Identification_of_errored_symbol_position (FEC-frame, positions);
  RS_Recovery_Process (FEC-frame, positions, result, User-Data);
  if (result == recovered)
  {
    Check_CRC-32 (#optional);
    if (Check_CRC-32 == no-error)
    {
      Send_User-Data (user-data-part[r]);
    }
    else
    {
      Fail_Recovery(user-data-part[o]);
    }
  }
  else
  {
    Fail_Recovery(user-data-part[o]);
  }
<<< Stop >>>

```

[Procedures and Functions]

- a) Identification_of_errored_symbol_position(FEC-frame, positions)
Input is the FEC-frame that is subject for error recovery.
Output is the set of errored (i.e. symbol error) positions.
- b) RS_Recovery_Process (FEC-frame, position, result, User-data)
Input is the FEC-frame that is subject for error recovery and the set of errored positions.
Output is result of recovery and User-Data. When the result is not-failed, the recovered user-data-part is returned as User-Data.
- c) Fail_Recovery(user-data-part[o])
The original user-data-part is returned with the indication of recovery failing.
- d) Send_User-Data(user-data-part[r])
The user-data-part is returned with the indication of completion of error recovery. The user-data-part[r] represents the recovered user-data-part.

5.2.4 Identification Procedure of Symbol missing position

FFS.

5.3 Procedure in Signaling Phase

5.3.1 Procedure of FEC parameter negotiation

FFS.

5.3.2 Procedures of Other Functions

FFS.

6 Annex A: List of acronyms

AAL	ATM Adaptation Layer
AAL-PDU	AAL Protocol Data Unit
AAL-SAP	AAL Service Access Point
AAL-SDU	AAL Service Data Unit
BI field	Burst Indication field
CPCS	Common Part Convergence Sublayer
CPCS-CI	CPCS Congestion Indication
CPCS-LP	CPCS Loss Priority
CPCS-PDU	CPCS Protocol Data Unit
CPCS-SDU	CPCS Service Data Unit
CPCS-RS	CPCS-SDU Reception Status
CPCS-UU	CPCS User-to-User
FCPL	FEC-Code-Part Length
FEC	Forward Error Correction
FEC frame	FEC-SSCS Protocol Data Unit
FEC-SSCS	FEC Service Specific Convergence Sublayer
FEC-SSCS-CI	FEC-SSCS Congestion Indication
FEC-SSCS-LP	FEC-SSCS Loss Priority
FEC-SSCS-M	FEC-SSCS More Indication
FEC-SSCS-PDU	FEC-SSCS Protocol Data Unit (i.e., FEC frame)
FEC-SSCS-RS	FEC-SSCS-SDU Reception Status
FEC-SSCS-SDU	FEC-SSCS Protocol Data Unit
FEC-SSCS-SIGI	FEC-SSCS SIGNaling message Identifier
FFS	For Further Study
FI field	FEC-SSCS-SDU Indication field
MHL	Maximum length of Horizontal Line
NS	Negotiation Status
P field	Parity field
RS algorithm	Reed Solomon algorithm
SEC	Symbol Error Correction
SEAL	Symbol Error And Loss
SID	Message Semantics IDentifier
SLC	Symbol Loss Correction
SN field	Sequence Number field
SNL	Length of Sequence Number field
SOM	Selection of Operation Mode
SYL	Symbol Length
U/F field	User-data-part/FEC-code-part field
VL	Vertical Length
#RedCell field	Number of Redundant Cells field

7 Appendix

7.1 Appendix A: Generator Polynomial

One example of a generator polynomial is described below.

When the symbol length is 8 bits, Reed-Solomon codes to be used are built over the Galois Field (256).

Then, the generator polynomial is given by :

$$\prod_{i=0}^{s-1} X - \alpha^{i+k}$$

where alpha is a root of the primitive polynomial $X^8 + X^7 + X^2 + X + 1$, k is the base exponent of the generator polynomial, which may be k=120.