ANNEX D

(Informative)

AN OVERVIEW OF PSIP FOR TERRESTRIAL BROADCAST WITH APPLICATION EXAMPLES

The Program and System Information Protocol (PSIP) is a small collection of tables designed to operate within every Transport Stream for terrestrial broadcast of digital TV. Its purpose is to describe the information at the system and event levels for all virtual channels carried in a particular Transport Stream. Additionally, information for analog channels as well as digital channels from other Transport Streams may be incorporated. The relational hierarchy for the component tables is explained through typical application examples in this document.

PSIP is the result of combining and compacting two existing optional ATSC protocols: A/55 and A/56. Although these protocols were individually efficient and accomplished their purpose, their mutual implementation was difficult due to their structural differences and their overlapping definitions. PSIP solves this problem. The tables defined in PSIP use packet identifiers (PIDs) that are different from those specified by the optional A/55 and A/56 standards. This provision has been included to enable the operation of existing equipment designed or manufactured to support A/55 and/or A/56.

D1. INTRODUCTION

Under the adopted ATSC standard for digital TV, the typical 6 MHz channel used for analog TV broadcast supports about 19 Mbps of throughput for terrestrial broadcast. Since audiovisual signals with standard resolution can be compressed using MPEG-2 to sustainable rates of around 6 Mbps, then around 3 or 4 digital TV channels can be safely supported in a single physical channel without congestion. Moreover, enough bandwidth remains within the same Transport Stream to provide several additional low-bandwidth non-conventional services such as: weather reports, stock indices, headline news, software download (for games or enhanced applications), image-driven classified ads, home shopping, pay-per-view information, and others.

It is therefore practical to anticipate that in the future, the list of services (virtual channels) carried in a physical transmission channel (6 MHz of bandwidth for the U.S.) may easily reach ten or more. What is even more important is that the number and type of services may also change continuously, thus becoming a more dynamic medium than what we have today.

An important feature of terrestrial broadcasting is that sources follow a distributed information model rather than a centralized one. Unlike cable or satellite, service providers are geographically distributed and have no interaction with respect to data unification or even synchronization. It is therefore necessary to develop a protocol for describing system information and event descriptions which is followed by every organization in charge of a physical

transmission channel. System information allows navigation and access to each of the channels within the Transport Stream, whereas event descriptions give the user content information for browsing and selection.

In this document we describe the development of a transport-based implementation of the PSIP protocol using examples. Our hope is to introduce the reader to the most important concepts and components that constitute the protocol.

D2. ELEMENTS OF PSIP

PSIP is a collection of hierarchically-associated tables each of which describes particular elements of typical digital TV services. Figures D.1 and D.2 show the different components and the notation used to describe them. The packets of the base tables are all labeled with the base PID (base_PID) which has been chosen as 0x1FFB. The base tables are: the System Time Table (STT), the Rating Region Table (RRT), the Master Guide Table (MGT) and the Virtual Channel Table (VCT).

A second set of tables are the Event Information Tables (EIT) whose packet identifiers (PIDs) are defined in the MGT. A third set of tables are the Extended Text Tables (ETT), and similarly, their packet identifiers (PIDs) are defined in the MGT.

The System Time Table (STT) is a small data structure that fits in one packet and serves as a reference for time of day. Receivers can use this table as a reference for timing start times of advertised events.

Transmission syntax for the United States' voluntary program rating system is included in this standard. The Rating Region Table (RRT) has been designed to transmit the rating standard in use for each country using the standard. Provisions were made for different rating systems for different countries and multi-country regions as well..

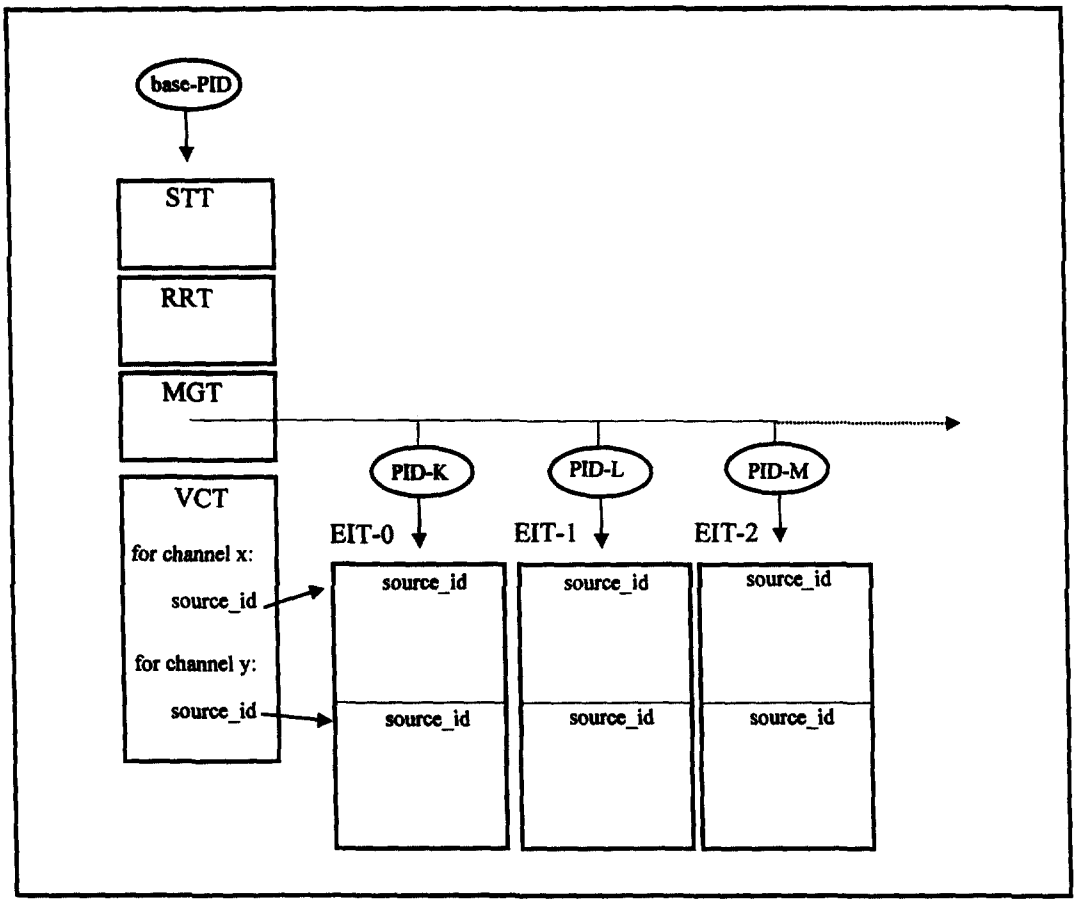

Figure D.1 Main Structure for the PSIP tables

The Master Guide Table (MGT) provides general information about all of the other tables that comprise the PSIP standard. It defines table sizes necessary for memory allocation during decoding; it defines version numbers to identify those tables that need to be updated; and it gives the packet identifiers (PIDs) that label the tables.

The Virtual Channel Table (VCT), also referred to as the Terrestrial VCT (TVCT), contains a list of all the channels that are or will be on-line plus their attributes. Among the attributes we have the channel name, navigation identifiers, stream components and types, etc.

As part of PSIP there are several Event Information Tables, each of which describes the events or TV programs associated with each of the virtual channels listed in the VCT. Each EIT is valid for a time interval of 3 hours. Since the total number of EITs is 128, up to 16 days of programming may be advertised in advance. EIT-0 always denotes the current 3 hours of programming, EIT-1 the next 3 hours, and so on. As a minimum, the first four EITs must always be present in every Transport Stream

Start times for EITs are constrained to be one of the following UTC times: 0:00 (midnight), 3:00, 6:00, 9:00, 12:00 (noon), 15:00, 18:00, and 21:00. Imposing constraints on the start times as well as the interval duration is necessary for the purpose of re-multiplexing. During re-multiplexing, EIT tables coming from several distinct Transport Streams may end up grouped

together or *vice versa*. If no constraints were imposed, re-multiplexing equipment would have to parse EITs by content in real time, which is a difficult task.

For example, consider a broadcast corporation operating in the Eastern time zone of the U.S. This corporation decides to carry 6 EITs (18 hours of TV program information). If at present, the Eastern time is 15:30 EDT (19:30 UTC), then the coverage times for the EIT tables are:

Table D.1 An Example of EIT Coverage Times

EIT number	Version Num.	Assigned PID	Coverage (UTC)	Coverage (EDT)
0	6	123	18:00 - 21:00	14:00 - 17:00
1	4	190	21:00 - 24:00	17:00 - 20:00
2	2	237	0:00 - 3:00	20:00 - 23:00
3	7	177	3:00 - 6:00	23:00 - 2:00 (nd)
4	8	295	6:00 - 9:00	2:00 (nd) - 5:00 (nd)
5	15	221	9:00 - 12:00	5:00 (nd) - 8:00 (nd)

The abbreviation "nd" denotes next day. Before 17:00 EDT, the MGT will list the currently valid PIDs as: 123, 190, 237, 177, 295, and 221. At 17:00 EDT, table EIT-0 will become obsolete while the other ones will remain valid. At that time, the PID list can be changed to 190, 237, 177, 295, 221, maintaining the version number list as 4, 2, 7, 8, 15. Therefore, by simply shifting the listed PID values in the MGT, table EIT-1 can become EIT-0, table EIT-2 can become EIT-1, and so on.

However, it is also possible to regenerate one or several EITs at any time for correcting and/or updating the content (e.g. in cases where "to be assigned" events become known). Regeneration of EITs is flagged by updating version fields in the MGT. For example, if table EIT-2 needs to be updated at 16:17 EDT, then the new table must be transmitted with a version number equal to 3. Whenever the decoder monitoring the MGT detects a change in the version number of a table, it assumes that the table has changed and needs to be reloaded.

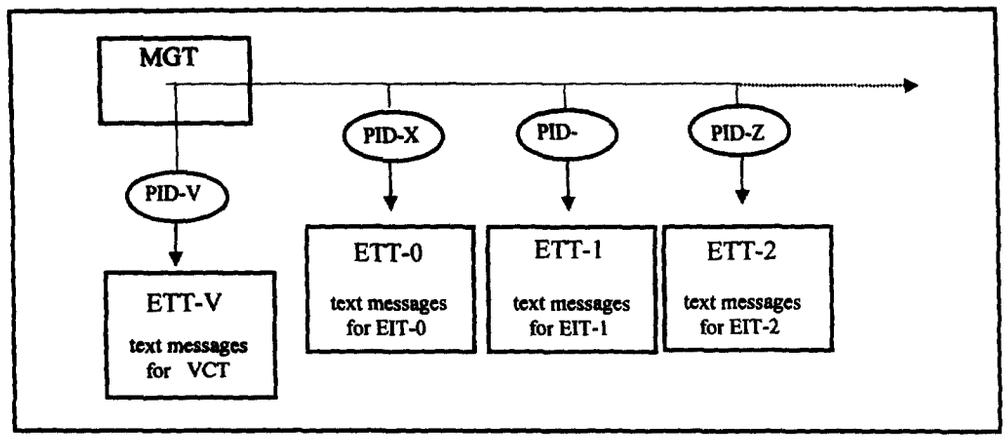

Figure D.2 Extended Text Tables in the PSIP hierarchy.

As illustrated in Fig. D.2, there can be several Extended Text Tables (ETTs), each of them having its PID defined in the MGT. Each Event Information Table (EIT) can have one ETT. Similarly, the Virtual Channel Table can have one ETT. As its name indicates, the purpose of an Extended Text Table (ETT) is to carry text messages. For example, for channels in the VCT, the messages can describe channel information, cost, coming attractions, etc. Similarly, for an event such as a movie listed in the EIT, the typical message is a short paragraph that describes the movie itself. Extended Text Tables are optional.

In this final section paragraph we review once more the requirement list. The minimum amount of information required in an ATSC terrestrial digital Transport Stream is the VCT, the MGT, the RRT, the STT, and the first four EITs. All of the other elements are optional.

D3. APPLICATION EXAMPLE

For the purpose of this example, we assume that a broadcast group, here denominated NBZ, manages the frequency bands for RF channels 12 and 39. The first one is its analog channel whereas the second one will be used for digital broadcast. According to the premises established in this document, NBZ must carry the PSIP tables in the digital Transport Stream of RF channel 39. The tables must describe TV programs and other services provided on RF channel 39 but can also describe information for the analog RF channel 12.

Assume that NBZ operates in the Eastern time zone of the U.S., and that the current time is 15:30 EDT (19:30 UTC). NBZ decides to operate in minimal configuration, therefore only the first four EITs need to be transmitted. As explained previously, EIT-0 must carry event information for the time window between 14:00 and 17:00 EDT, whereas EIT-1 to EIT-3 will cover the subsequent 9 hours. For the first 6 hours, the following scenario applies:

Table D.2 The first 3-hour segment to be described in VCT and EIT-0

		14:00-14:30	14:30 -15:00	15:00 - 15:30	15:30 - 16:00	16:00 - 16:30	16:30-17:00
PTC 12	NBZ	City Life	City Life	Travel Show	Travel Show	News	News
PTC 39 VC #1	NBZ	City Life	City Life	Travel Show	Travel Show	News	News
PTC 39 VC #2	NBZ	Soccer	Golf Report	Golf Report	Car Racing	Car Racing	Car Racing
PTC 39 VC #3	NBZ	Secret Agent	Secret Agent	Lost Worlds	Lost Worlds	Lost Worlds	Lost Worlds
PTC 39 VC #4	NBZ	headlines	headlines	headlines	headlines	headlines	headlines

Table D.3 The second 3-hour segment to be described in VCT and EIT-1

		17:00-17:30	17:30-18:00	18:00 - 18:30	18:30 - 19:00	19:00-19:30	19:30 - 20:00
PTC 12	NBZ	Music Today	NY Comedy	World View	World View	News	News
PTC 39 VC #1	NBZ	Music Today	NY Comedy	World View	World View	News	News
PTC 39 VC #2	NBZ	Car Racing	Car Racing	Sports News	Tennis Playoffs	Tennis Playoffs	Tennis Playoffs
PTC 39 VC #3	NBZ	Preview	The Bandit	The Bandit	The Bandit	The Bandit	Preview
PTC 39 VC #4	NBZ	headlines	headlines	headlines	headlines	headlines	headlines

Similar tables can be built for the next 6 hours (for EIT-2 and EIT-3). According to this scenario, NBZ broadcasts four regular digital channels (also called virtual channels and denoted as VC), one matching the analog transmission (simulcast), another for sports, and a third one for movies. The fourth one supports a service displaying headlines with text and images.

D3.1 The Master Guide Table (MGT)

The purpose of the MGT is to describe everything about the other tables, listing features such as version numbers, table sizes, and packet identifiers (PIDs). Fig. D.3 shows a typical Master Guide Table indicating, in this case, the existence in the Transport Stream of a Virtual Channel Table, the Rating Region Table, four EITs, one Extended Text Table for channels, and two Extended Text Tables for events.

The first entry of the MGT describes the version number and size of the Virtual Channel Table. The second entry corresponds to an instance of the Rating Region Table. If some region's policy makers decided to use more than one instance of an RRT, the MGT would list each PID,

version number, and size. Notice that the base PID (0x1FFB) must be used for the VCT and the RRT instances as specified in PSIP.

The next entries in the MGT correspond to the first four EITs that must be supplied in the Transport Stream. The user is free to choose their PIDs as long as they are unique in the MGT list of PIDs. After the EITs, the MGT indicates the existence of an Extended Text Table for channels carried using PID 0x1AA0. Similarly, the last two entries in the MGT signal the existence of two Extended Text Tables, one for EIT-0 and the other for EIT-1.

MGT

table_type	PID	version num.	table size
VCT	0x1FFB (base_PID)	4	485 bytes
RRT - USA	0x1FFB (base_PID)	1	560 bytes
EIT-0	0x1FD0	6	2730 bytes
EIT-1	0x1FD1	4	1342 bytes
EIT-2	0x1DD1	2	1224 bytes
EIT-3	0x1DB3	7	1382 bytes
ETT for VCT	0x1AA0	21	4232 bytes
ETT-0	0x1BA0	10	32420 bytes
ETT-1	0x1BA1	2	42734 bytes

Figure D.3 Content of the Master Guide Table

Descriptors can be added for each entry as well as for the entire MGT. By using descriptors, future improvements can be incorporated without modifying the basic structure of the MGT. The MGT is like a flag table that continuously informs the decoder about the status of all the other tables (except the STT which has an independent function). The MGT is continuously monitored at the receiver to prepare and anticipate changes in the channel/event structure. When tables are changed at the broadcast side, their version numbers are incremented and the new numbers are listed in the MGT. Based on the version updates and on the memory requirements, the decoder can reload the newly defined tables for proper operation.

D3.2 The Virtual Channel Table (VCT)

Figure D.4 shows the structure of the VCT which essentially contains the list of channels available in the Transport Stream. For convenience, it is possible to include analog channels and even other digital channels found in different Transport Streams.

The field number_of_channels_in_section indicates the number of channels described in one section of the VCT. In normal applications, as in the example being considered here, all channel information will fit into one section. However, there may be rare times when most of the physical channel is used to convey dozens of low-bandwidth services such as audio-only and data channels in addition to one video program. In those cases, the channel information may be larger than the VCT section limit of 1 Kbyte and therefore VCT segmentation will be required.

For example, assuming that a physical channel conveys 20 low-bandwidth services in addition to a TV program, and assuming that their VCT information exceeds 1 Kbyte, then two or more sections may be defined. The first section may describe 12 virtual channels and the second 9 if such a partition leads to VCT sections with less than 1 Kbyte.

A new VCT containing updated information can be transmitted at any time with the version_number increased by one. However, since a VCT describes only those channels from a particular Transport Stream, virtual channels added to the VCT at arbitrary times will not be detected by the receiver until it is tuned to that particular Transport Stream. For this reason, it is highly recommended that channel addition be made in advance to give the receivers the opportunity to scan the frequencies and detect the channel presence.

The fields major_channel_number and minor_channel_number are used for identification. The first one, the major channel number, is used to group all channels that are to be identified as belonging to a particular broadcast corporation (or particular identifying number such as 12 in this case). The minor channel number specifies a particular channel within the group.

The field short_name is a seven-character name for the channel and may allow text-based access and navigation. The fields transport_stream_id and program_number are included to link the VCT with the PAT and PMT. A sequence of flags follows these fields. The flags indicate: (1) if the channel is hidden (e.g. for NVOD applications), (2) if the channel has a long text message in the VCT-ETT, and (3) if the channel is visible in general or has some conditional access constraints.

After the flags, a description of the type of service offered is included, followed by the source_id. The source_id is simply an internal index for representing the particular logical channel. Event Information Tables and Extended Text Tables use this number to provide a list of associated events or text messages respectively.

VCT

current_next_indicator = 1
number_channels_in_section = 5

major num.	minor num	short name	carrier freq. (MHz)	channel TS ID	progr. num	flags	service type	source id	descriptors
12	0	NBZ	205.25	0x0AA0	0x0AA0	--	analog	20	ch_name
12	1	NBZD	620.31	0x0AA1	0x00F1	--	digital	21	ch_name serv_locat.
12	5	NBZ-S	620.31	0x0AA1	0x00F2	--	digital	2	ch_name serv_locat.
12	12	NBZ-M	620.31	0x0AA1	0x00F3	--	digital	23	ch_name serv_locat.
12	31	NBZ-H	620.31	0x0AA1	0x00F8	--	digital	24	ch_name serv_locat.

Figure D.4 Content of the Virtual Channel Table

Two descriptors are associated with the logical channels in the example. The first one is extended_channel_name and, as its name indicates, it gives the full name of the channel. An example for channel NBZ-S could be: "NBZ Sports and Fitness". The other one, the service_location descriptor, is used to list the available bit streams and their PIDs necessary to decode packets at the receiver. Assuming that NBZ-M offers bilingual transmission, then the following attributes are tabulated within its service_location descriptor:

PID_audio_1	AC-3 audio	English
PID_audio_2	AC-3 audio	Spanish
PID_video	MPEG-2 video	No lang.

Two VCTs may exist simultaneously in a Transport Stream: the current and the next VCT. The current VCT is recognized by having the flag current_next_indicator set to 1, while the next one has this flag set to 0. Although carrying the next VCT is optional, its use is recommended to give receivers advance notification of the new parameters that become operational during a VCT update.

Assume for example that a Transport Stream contains a VCT with a version number of 6 which has been operational for 20 hours. At 10:00 p.m., a football game using much more bandwidth will be broadcast, and for this reason, the number of available channels and PIDs will

be redefined. Around 9:30 p.m., simultaneous transmission of the next VCT can start with a version number of 7. By continuously monitoring the MGT, a receiver can be informed that a next VCT is available. The receiver may want to cache the new VCT for future use. The receiver continues monitoring the MGT and when this table signals a version change for the current VCT (from 6 to 7), then the cached information can be used.

When the VCT refers to an analog service type, the channel_TSID cannot refer to the identifier of a "Transport Stream" in the MPEG-2 sense. Analog NTSC broadcast signals can, however, carry a 16-bit unique identifier called a "Transmission Signal Identifier."[8] For the example VCT in Figure D.4, the Transmission Signal Identifier for channel 12.0 is 0x0AA0. A receiver can use the Transmission Signal ID given in the analog channel's channel_TSID field to verify that the NTSC signal received at the frequency given in the VCT is actually the desired signal. In the case that the Transmission Signal ID is not known or not available, the channel_TSID field may contain 0xFFFF to indicate "unknown."

D3.3 The Event Information Tables (EITs)

The purpose of an EIT is to list all events for those channels that appear in the VCT for a given time window. As mentioned before, EIT-0 describes the events for the first 3 hours, EIT-1 for the next 3 hours, and so on. EIT-i and EIT-j have different PIDs as defined in the MGT. In PSIP, tables can have a multitude of instances. The different instances of a table share the same table_id value and PID but use different table_id_extension values.

In PSIP, an instance of EIT-k contains the list of events for a single virtual channel with a unique source_id. For this reason, the table_id_extension has been renamed as source_id in the EIT syntax. Figure D.5 shows, for example, the NBZ-S instance for EIT-0. Following similar procedures, the NBZD, NBZ-M, and NBZ-H instances of EIT-0 can be constructed. The process can be extended and repeated to obtain all of the instances for the other tables in the time sequence: EIT-1, EIT-2, etc.

The three events programmed for the 3-hour period for NBZ-S are listed in Figure D.5. The field event_id is a number used to identify each event. If an event time period extends over more than one EIT, the same event_id has to be used. The event_id is used to link events with their messages defined in the ETT, and therefore it has to be unique only within a virtual channel and a 3-hour interval defined by EITs. The event_id is followed by the start_time and then the length_in_seconds. Notice that events can have start times before the activation time (14:00 EST in this example) of the table. The ETM_location specifies the existence and the location of an Extended Text Message (ETM) for this event. ETMs are simply long textual descriptions. The collection of ETMs constitutes an Extended Text Table (ETT).

[8] A method to include such a unique 16-bit "Transmission Signal ID" in the NTSC VBI is specified in the EIA-752 specification.

EIT-0
source id = 22 (NBZ-S instance)

num_events_in_section = 3

event ID	local start time	length (seconds)	ETM location	title	descriptors
51	12:30	7200	01	Soccer Live	content_advisory
52	14:30	3600	00	Golf Report	closed_caption
53	15:30	9000	01	Car Racing	content_advisory

Figure D.5 Content of EIT-0 for NBZ-S

An example of an ETM for the Car Racing event may be:

> "Live coverage from Indianapolis. This car race has become the largest single-day sporting event in the world. Two hundred laps of full action and speed."

Several descriptors can be associated with each event. The most important is the content advisory descriptor which assigns a rating value according to one or more systems. Recall that the actual rating system definitions are tabulated within the RRT. When a closed caption descriptor is included, it signals the existence of closed captioning and lists the necessary parameters for decoding.

D3.4 The Rating Region Table (RRT)

The Rating Region Table is a fixed data structure in the sense that its content remains mostly unchanged. It defines the rating standard that is applicable for each region and/or country. The concept of table instance introduced in the previous Section is also used for the RRT. Several instances of the RRT can be constructed and carried in the Transport Stream simultaneously. Each instance is identified by a different table_id_extension value (which becomes the rating_region in the RRT syntax) and corresponds to one and only one particular region. Each instance has a different version number which is also carried in the MGT. This feature allows updating each instance separately.

Figure D.6 shows an example of one instance of an RRT, defined as the first rating region and carrying the MPAA standard rating system. Changes in the content of the RRT must be defined and approved by the ATSC. Each event listed in any of the EITs may carry a content advisory descriptor. This descriptor is an index or pointer to one or more instances of the RRT.

RRT
rating_region = 1 (first instance)

rating_region_name_text= "US (50 states+possesions)"
dimensions = 1

dimension_name = "MPAA Theatrical Rating System"
values_defined = 7

value	abbrev	rating value
0	" "	" "
1	"G"	"Suitable for All Ages"
2	"PG"	"Parental Guidance Suggested"
3	"PG-14"	"Parents Strongly Cautioned"
4	"R"	"Restricted, under 17 must be accompanied by adult"
5	"NC-17"	"No One 17 And Under Admitted to Theater"
6	"NR"	"Not Rated by MPAA"

Figure D.6 An instance of a Rating Region Table (RRT).

D4. PACKETIZATION AND TRANSPORT

In the previous sections, we have described how to construct the MGT, VCT, RRT, and EITs based on the typical scenario described in Tables D.1 and D.2. The number of virtual channels described in the VCT is 5 and therefore, each EIT will have 5 instances.

For the example, the size of the MGT is less than a hundred bytes and the VCT ranges between 300 to around 1500 bytes depending on the length of the text strings. Similarly, each EIT instance can have from 1 to about 3 Kbytes depending again on the text length.

Typically, the MGT, STT, VCT, and each instance of the RRT and EIT will have one or at most a few sections. For each table, the sections are appended one after the other, and then segmented into 184-byte packets. After adding the 4-byte MPEG-2 TS header, the packets are multiplexed with the others carrying audio, video, data, and any other components of the service. Figure D.7 illustrates this process.

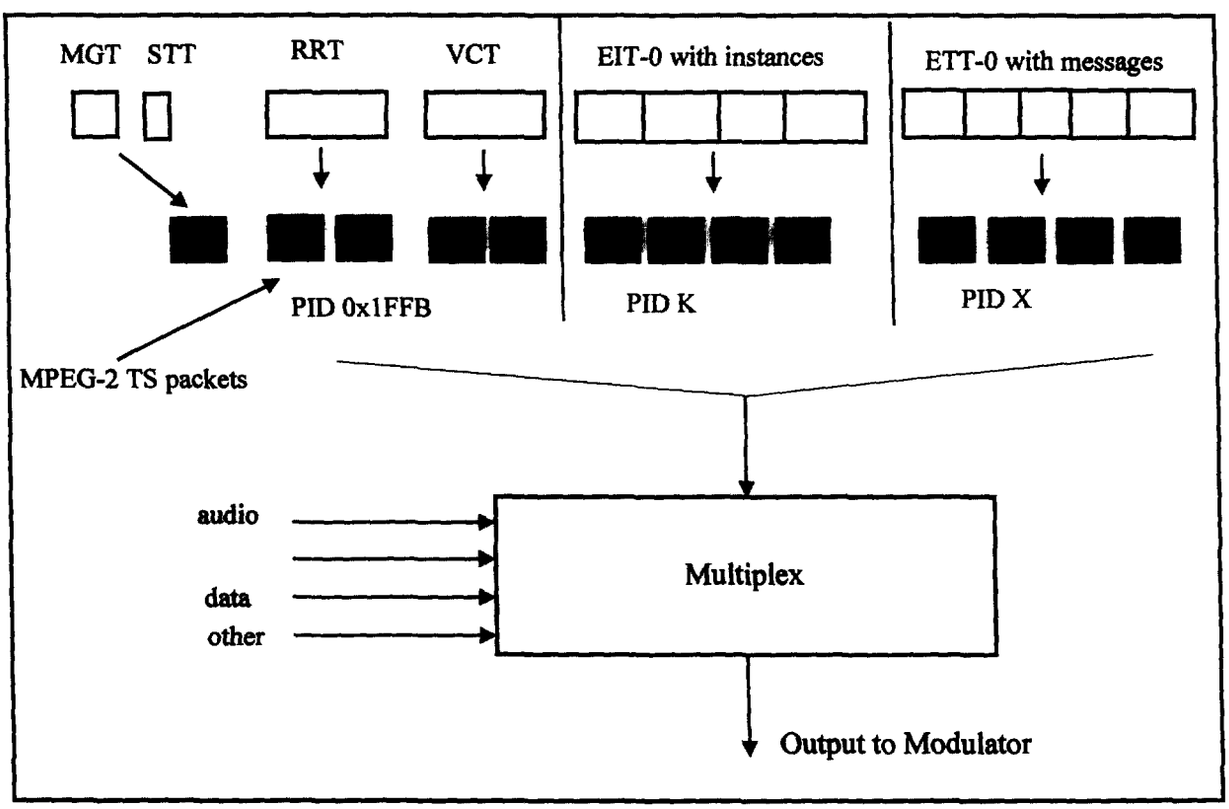

Figure D.7 Packetization and transport of the PSIP tables

D.5 TUNING OPERATIONS AND TABLE ACCESS

As described by the PSIP protocol, each Transport Stream will carry a set of tables describing system information and event description. For channel tuning, the first step is to collect the VCT from the Transport Stream which contains the current list of services available. Figure D.8 shows this process.

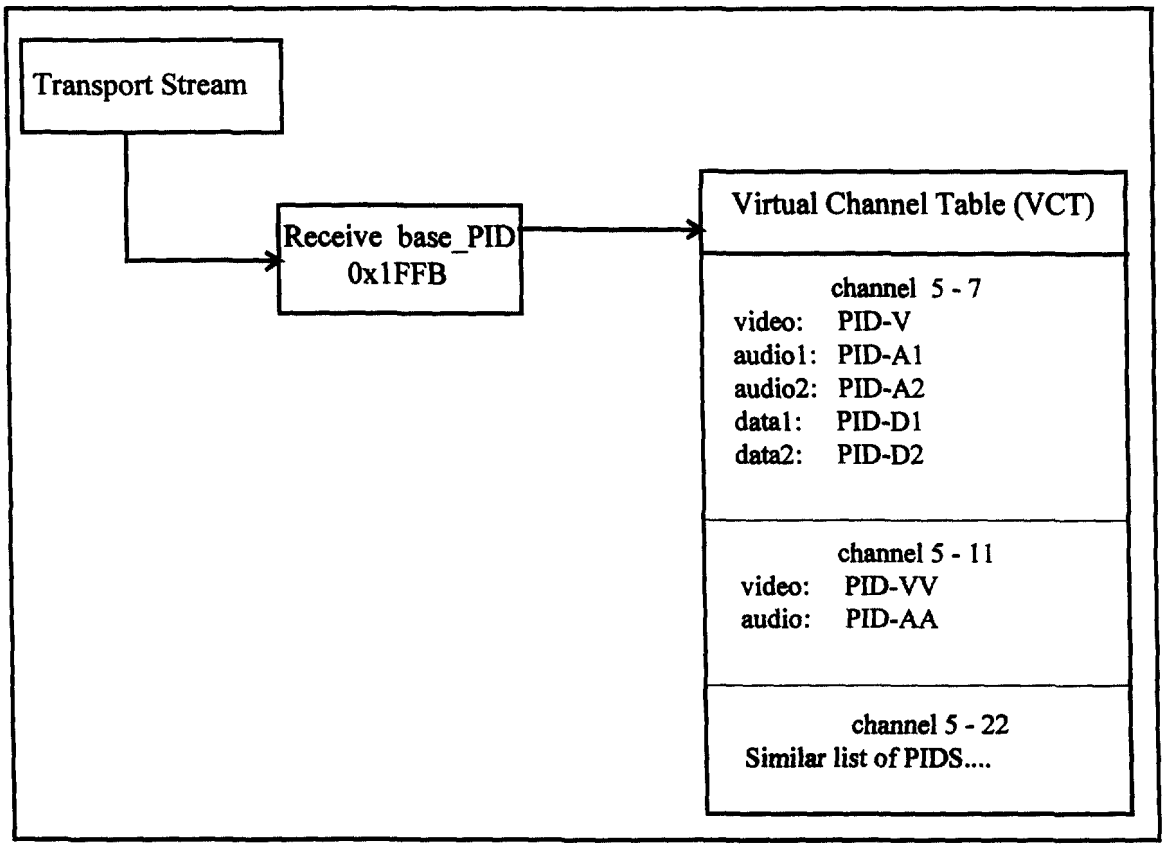

Figure D.8 Extraction of the VCT from the Transport Stream

Once the VCT has been collected, a user can tune to any virtual channel present in the Transport Stream by referring to the major and minor channel numbers. Assuming that in this case, the user selects channel 5 - 11, then the process for decoding the audio and video components is shown in Fig. D.9. For terrestrial broadcast, the existence of a service location descriptor in the VCT is mandatory and therefore there is no need to access the PAT or PMT for tuning. This feature has been included in PSIP to minimize the time required for changing and tuning to channels. However, PAT and PMT information must be present in the Transport Stream to support the general MPEG-2 compliance

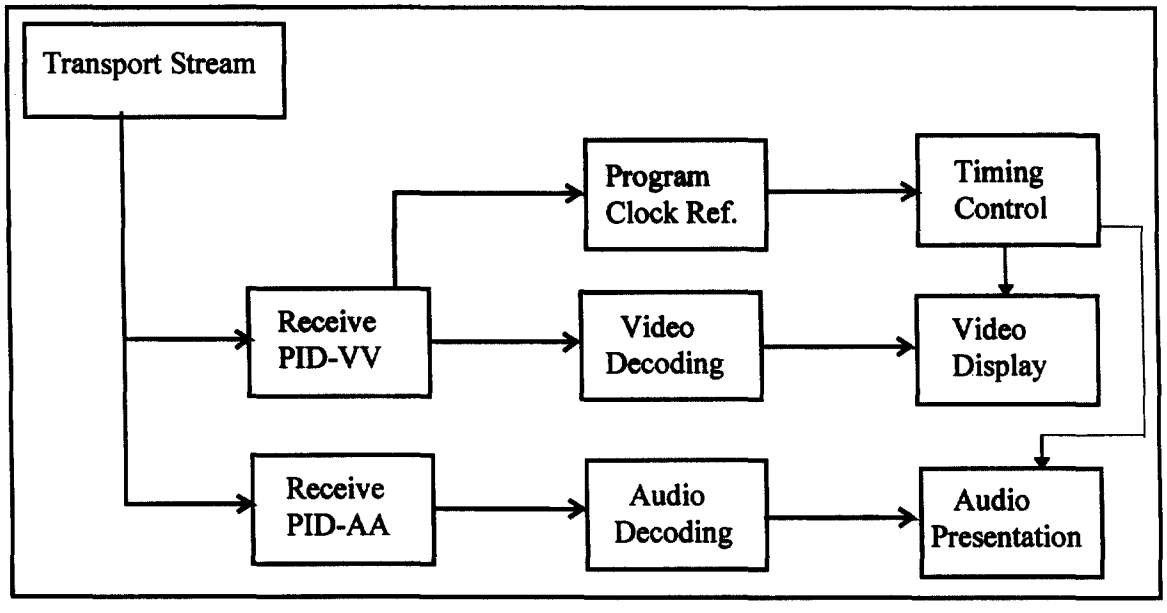

Figure D.9 Acquisition of audiovisual components

ANNEX E

(Informative)

TYPICAL SIZE OF PSIP TABLES

The typical sizes for the PSIP tables (STT, MGT, VCT, RRT, EIT and ETT) are calculated in this Section. The notation used here for the different equations is listed in the Table E.1.

Table E.1 Symbols

Symbol	Description
P	number of EITs (4 to 128)
C	number of virtual channels (analog and digital) per EIT
Cd	number of digital channels per EIT
E	number of events per virtual channel
R	number of rating regions
D	average number of rating dimensions per rating region
L	average number of rating values per rating dimension

E1. SYSTEM TIME TABLE (STT)

The typical size for the STT is 20 bytes, with the assumption of having no descriptors.

E2. MASTER GUIDE TABLE (MGT)

The typical size for the MGT (in bytes), based on the assumptions listed in the column "Assumption", is shown in Table E.2

Table E.2 Typical size (bytes) of MGT

Part	Size (bytes)	Assumption
PSI header and trailer	12	
message body	38+22*P	1. With one Terrestrial VCT, one channel ETT, one RRT instance, P EITs and P event ETTs 2. No descriptors
Total	50+22*P	

E3. TERRESTRIAL VIRTUAL CHANNEL TABLE (TVCT)

The typical size of the TVCT (bytes), based on the assumptions listed in the column labeled "Assumption" is shown in Table E.3.

Table E.3 Typical TVCT size (bytes)

Part	Size (bytes)	Assumption
PSI header and trailer	12	1. All TVCT messages are carried in one section.
message body	4+32*C	
extended channel name descriptor	20*C	2. One string and one segment per string for long channel name text. 3. Long channel name text is compressed by Huffman coding with a standard table, and the text length after compression is 10 bytes
service location descriptor	23*Cd	4. Three elementary streams per virtual channel for digital channels.
Total	16+52*C+23*Cd	

E4. RATING REGION TABLE (RRT)

The typical size (in bytes per rating region) of the RRT, based on the assumptions listed in the column "Assumption", is shown in Table E.4.

Table E.4 Typical size (in bytes per rating region) of RRT

Part	Size (bytes per rating region)	Assumption
PSI header and trailer	12	1. One section only.
message body	25+D*(14+ 26*L)	2. One string and one segment per string for all text. 3. Rating region name text is compressed by Huffman coding with a standard table, and the size after compression is 12 bytes. 4. Dimension name text is compressed by Huffman coding with a standard table, and the size after compression is 4 bytes. 5. Abbreviated rating value text is compressed by Huffman coding with a standard table, and the size after compression is 2 bytes. 6. Rating value text is compressed by Huffman coding with a standard table, and the size after compression is 6 bytes. 7. No descriptors.
Total	37+D*(14+26*L)	

E5. EVENT INFORMATION TABLE (EIT)

The typical size of the EIT (in bytes per virtual channel per EIT), based on the assumptions listed in the column "Assumption", is shown in Table E.5.

Table E.5 Typical size (bytes per virtual channel per EIT) of EIT

Part	Size (bytes per virtual channel per EIT)	Assumption
PSI header and trailer	12	1. One section only
message body	2+30*E	2. One string and one segment per string for title text. 3. Title text is compressed by Huffman coding with a standard table, and the size after compression is 10 bytes. 4. No AC-3 and service location descriptors.
closed captioning service descriptor	9*E	5. number_of_services = 1.
content advisory descriptor	(3+R*(3+2*D))*E	6. No rating_description_text.
Total	14+(42+R*(3+2*D))*E	

E6. EXTENDED TEXT TABLE (ETT)

The typical size for the ETT (in bytes per virtual channel per EIT, or bytes per event per EIT), based on the assumptions listed in the column labeled "Assumptions", is shown in Table E.6.

Table E.6 Typical size (bytes per virtual channel or bytes per event) of ETT

Part	Size (bytes per virtual channel per EIT, or bytes per event per EIT)	Assumptions
PSI header and trailer	12	
message body	508	1. A virtual channel or an event can have one text string and one segment per string for the extended text message. 2. Extended text message is compressed by Huffman coding with a standard table, and the size after compression is 500 bytes.
Total	520	

E7. AN EXAMPLE FOR TERRESTRIAL BROADCAST

Suppose that a TV provider is in charge of two physical transmission channels, one for analog and the other for digital services. Assume that the digital Transport Stream carries five virtual channels, each with 6 events in EIT-0, EIT-1, EIT-2 and EIT-3. For each virtual channel and each event an extended text message is available.

Regarding the Rating Region Table, suppose that a single rating region is defined with six dimensions and five values per dimension. Based on these assumptions, typical sizes for every PSIP table can be calculated. The results are listed in Table E.7 and Table E.8.

Table E.7 Typical sizes of PSIP tables (except ETT) for the example

Part	Size in bytes (excluding Transport Stream packet header)	Size in Transport Stream packets
STT	20	1
MGT	138	1
TVCT	443	3
RRT	901	5
Subtotal for tables identified by the base_PID	**1502**	**10**
EIT-0	2136	12
EIT-1	2136	12
EIT-2	2136	12
EIT-3	2136	12
Total	**10046**	**58**

Table E.8 Typical sizes of ETTs for the example

Part	Size in bytes (excluding Transport Stream packet header)	Size in Transport Stream packets
Channel ETT	3120	17
Event ETT-0	18720	102
Event ETT-1	18720	102
Event ETT-2	18720	102
Event ETT-3	18720	102
Total	**78000**	**425**

ANNEX F

(Informative)

AN OVERVIEW OF HUFFMAN-BASED TEXT COMPRESSION

This section describes the Huffman-based text compression and coding methods supported in the Program and System Information Protocol. In particular, this section:

- Describes the partial first-order Huffman coding used to compress PSIP text data.
- Provides background description of finite-context Huffman coding. The mechanisms for generating and parsing Huffman codes are described.
- Describes the decode tree data structure.
- Defines the character set supported by this Standard.

F1. DATA COMPRESSION OVERVIEW

Program and System Information data may use partial first-order Huffman encoding to compress English-language text. The Huffman-table based approach has the following features:

- A typical firmware-resident Huffman decode table requires less than 2K of storage.
- The encode and decode algorithms are relatively simple and fast.
- Since first-order Huffman codes are significantly influenced by language phonetics, codes produced from a sample of current program titles produce reasonable compression ratios for future program titles, even though the future program titles may be significantly different from current titles. Therefore, hard-coded tables stored in receiver non-volatile memory are helpful.

  The data compression approach has the following implementation characteristics:

- Program descriptions and program titles may use different Huffman codes. Titles and descriptions have significantly different text characteristics; for example, program titles usually have an upper-case character following a space character, whereas program descriptions usually have a lower-case character following a space-character.
- Hard-coded decode tables, one optimized for titles and one for descriptions, must reside in the receiver's non-volatile memory.

F2. OVERVIEW OF CONTEXT-SENSITIVE HUFFMAN CODING

F2.1 Overview

Each and every character does not occur with the same frequency in program titles and program descriptions. For example, the character "e" occurs more often than the character "x." With Huffman coding, the number of bits used to represent a character is inversely proportional to the character's usage frequency.

The Huffman coding compression ratio depends upon the statistical distribution of the characters being compressed. When character usage is uniformly distributed, no compression is achieved with Huffman coding. To achieve satisfactory compression, the Huffman codes are generated using statistics that match the data being compressed. For example, Huffman codes generated from Pascal computer programs would be less than ideal for compressing C programs. For text strings in the PSIP, program descriptions and program titles may be compressed with different sets of Huffman codes

Context-sensitive Huffman coding recognizes that a character's usage statistics are context dependent. For example, the character "u" has a high probability of occurrence after the character "q". The "order" of the Huffman code defines the "look-back" context by which a character is coded. With order-0, each character is coded independently of the previous character. With order-1, the Huffman code used to represent a given character depends upon the previous character. In zero-order Huffman compression, the occurrence probability of the alphabet elements is used to develop an optimal encoding tree. In first-order Huffman, the conditional probability of a character, given that the previous character is known, is used as the basis of a decoding tree. For this reason, while zero-order Huffman has typically a single tree, first-order Huffman has many, one for each character.

Huffman compression involves the following steps:

- Determine the statistical distribution of the characters or symbols in the source data.
- Create Huffman codes from this statistical information.
- Encode the source data: Translate each character into its corresponding Huffman code.

To decompress the coded data, the data string is parsed bit-by-bit and translated to the original characters. To do this, the decompressor must have the correct decode table, which maps the Huffman codes to their corresponding characters. The following example illustrates the generation and decoding of Huffman codes.

F2.2 Example

Huffman codes are mapped to their corresponding characters using a binary tree structure. The leaves of this tree are the alphabet elements to be coded. The tree is produced by recursively summing the two nodes in the tree with the lowest usage frequency. For the following example, assume that an alphabet contains the following twelve characters which occur a certain number of times in the sample database:

Table F.1 Example Character Set and Frequency of Character Occurrence

Character	Occurrence Number
'a'	144
'b'	66
'c'	30
'd'	30
'e'	18
'f'	12
'g'	6
ESC	arbitrary

The "escape" character is inserted into the table to handle input characters which rarely occur, and have no corresponding Huffman codes. In this example, no Huffman codes will be generated for the characters 'h', 'i', and 'j'. Instead, their frequencies will be summed into the ESC character. Whenever one of these characters occur in the input stream, the encoder inserts the ESC Huffman code, then inserts the original ASCII value for that character.

Figure F.1 shows the construction of the Huffman tree from the character frequencies. The two nodes with the lowest frequencies, ('ESC' and 'g'), are joined together, with a resulting node weight of (9). The next two lowest nodes, ('f' and the intermediate node), are then joined together, with the combined weight of (21). This process continues until the tree's root node is formed. Once the tree is completed, the bit (1) is assigned to all right-hand branches, and the bit (0) is assigned to all left-hand branches.

Decoding a Huffman string is straight-forward. Starting at the Huffman tree root, the decoder parses the string, bit by bit, until it reaches a leaf node. The leaf node is the decoded character. The decoder then moves back to the root of the Huffman tree to continue decoding the bit string. For example, the input string 10111011100010 would be decoded into 'beeaab'.

This example uses order-0 Huffman codes. With order-1, each character in the alphabet has an associated tree of Huffman codes for possible succeeding characters. The ESC character would be inserted into each of these order-1 tables to handle statistically unlikely character pairs.

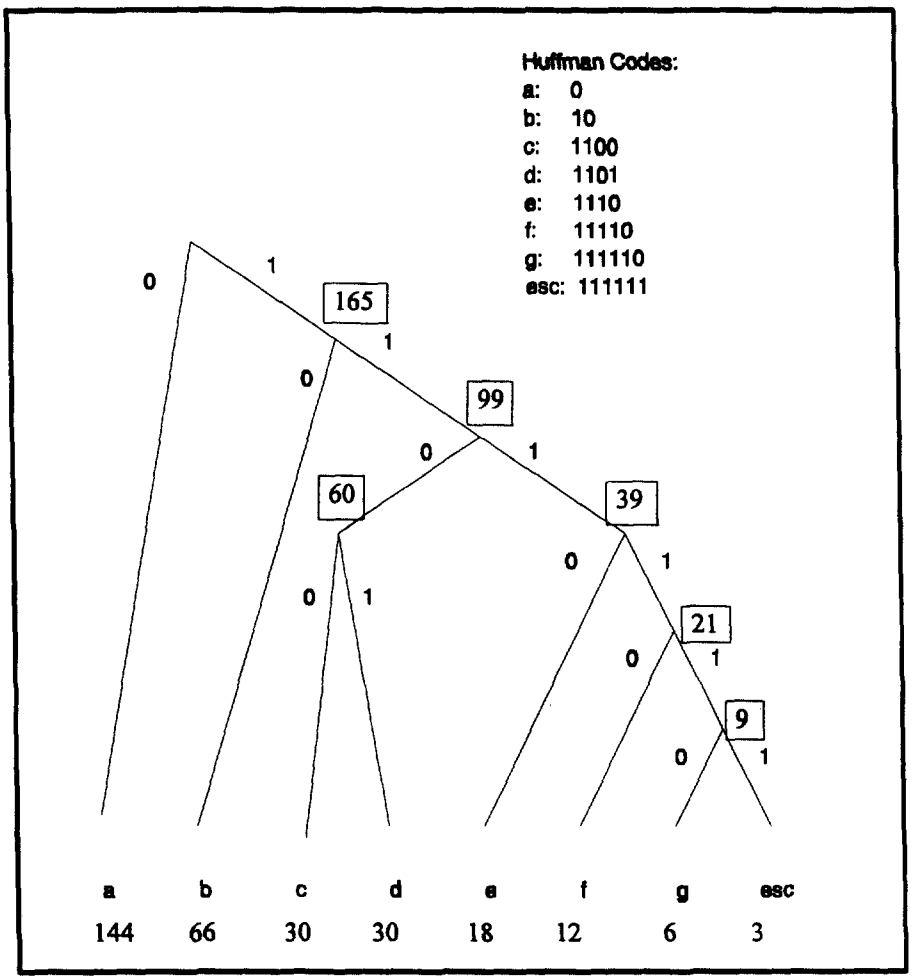

Figure F.1 Example Huffman Tree

F2.3 Decode Tree Example

Actual implementations of Huffman decoders need to map the trees into a suitable data structure that can be used by a computer or processor to traverse the tree top-down. In Annex C, a possible method for representing the trees was described and explicitly defined. Such a method is used here to build the decoding tree data for the example given in Figure F.1. Although an order-0 tree, this table is representative of order-1 decode trees, except that the bytes of each order-1 tree start at a byte location specified by the corresponding tree root offset (rather than starting at location 0).

Table F.2 Decode Tree Example

Byte #	Left/Right Child Word Offset or Character Leaf	
0 (tree root)	225	(ASCII "a" + 128)
1	1	(word offset of right child)
2 (tree node)	226	(ASCII "b" + 128)
3	2	(word offset of right child)
4 (tree node)	3	(word offset of left child)
5	4	(word offset of right child)
6 (tree node)	227	(ASCII "c" + 128)
7	228	(ASCII "d" + 128)
8 (tree node)	229	(ASCII "e" + 128)
9	5	(word offset of right child)
10 (tree node)	230	(ASCII "f" + 128)
11	6	(word offset of right child)
12 (tree node)	231	(ASCII "g" + 128)
13	155	(ASCII "ESC" + 128)

F2.4 Encoding/Character Decoding Examples with 1st-order Huffman tables

As an example of using the Huffman table defined in Table C.4 in Annex C, here we show the procedure to encode and decode the string "The next" using the tables optimized for titles. The coding sequence that generates the bit stream for "The next" is described in Figure F.2.

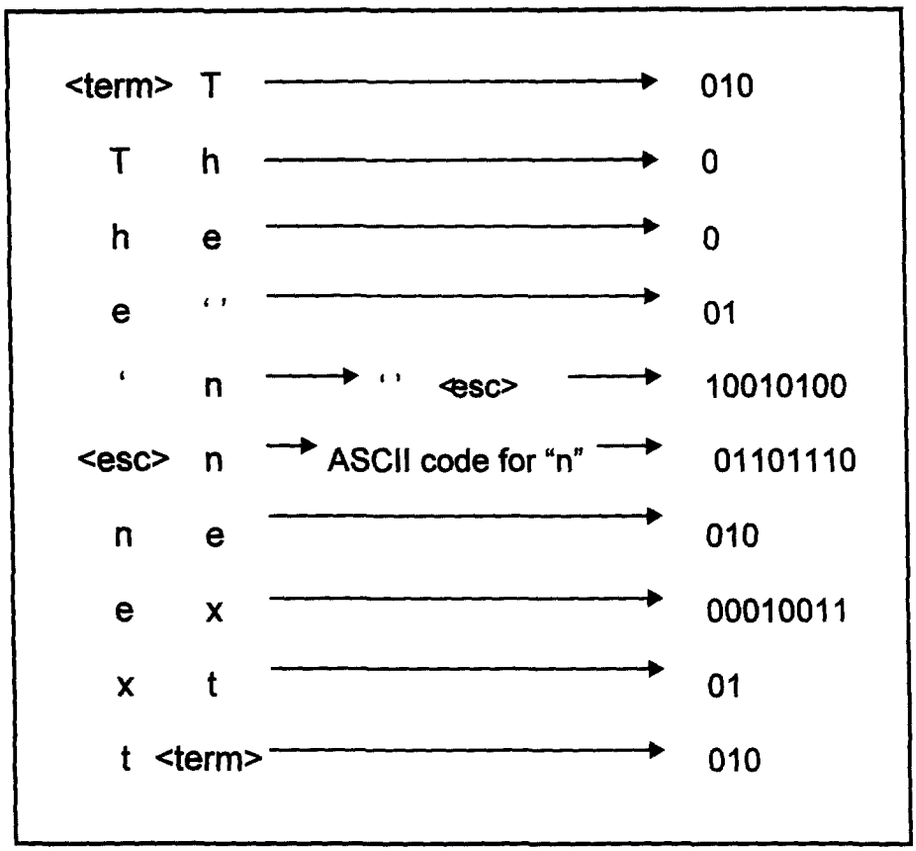

Figure F.2 Coding Example for the string "The next"

The first character 'T' is encoded assuming that the previous one was a *terminate* character. The second letter 'h' is encoded based on the Huffman tree corresponding to the prior symbol 'T.' The sequence proceeds as shown in the Figure. The combination blank-space followed by an 'n' is not listed in the tree, thus the escape character is used to switch the coding process to uncompressed mode. Once in this mode, the 'n' is encoded using its standard 8-bit ISO Latin-1 value. After the 'n', an 'e' is encoded using the appropriate n-tree and the algorithm continues until reaching the final letter followed by a string-terminate character. Uncompressed transmission of this string requires 9 bytes, while after compression, only 39 bits, equivalent to 5 bytes, are needed.

Decoding requires traversing the different trees top-down. As an example, Figure F.3 shows the tree when the prior character is 'x'. From our example, after decoding the letter 'x', the remaining bit sequence is '01010'. Traversing the x-tree top-down using this sequence shows that '01' corresponds to 't', a newly decoded character. The process now jumps to the t-tree and so on, to decode the remaining bits until the terminate code results. Notice that the trees can be obtained by examining the encoding tables or by following the semantics of the provided decoding tables.