International Telecommunication Union

ITU-R

Radiocommunication Sector of ITU

Recommendation ITU-R BT.1869

(03/2010)

Multiplexing scheme for variable-length packets in digital multimedia broadcasting systems

BT Series

Broadcasting service (television)

Foreword

The role of the Radiocommunication Sector is to ensure the rational, equitable, efficient and economical use of the radio-frequency spectrum by all radiocommunication services, including satellite services, and carry out studies without limit of frequency range on the basis of which Recommendations are adopted.

The regulatory and policy functions of the Radiocommunication Sector are performed by World and Regional Radiocommunication Conferences and Radiocommunication Assemblies supported by Study Groups.

Policy on Intellectual Property Right (IPR)

ITU-R policy on IPR is described in the Common Patent Policy for ITU-T/ITU-R/ISO/IEC referenced in Annex 1 of Resolution ITU-R 1. Forms to be used for the submission of patent statements and licensing declarations by patent holders are available from http://www.itu.int/ITU-R/go/patents/en where the Guidelines for Implementation of the Common Patent Policy for ITU-T/ITU-R/ISO/IEC and the ITU-R patent information database can also be found.

Series of ITU-R Recommendations

(Also available online at http://www.itu.int/publ/R-REC/en)

Series	Title
BO	Satellite delivery
BR	Recording for production, archival and play-out; film for television
BS	Broadcasting service (sound)
BT	**Broadcasting service (television)**
F	Fixed service
M	Mobile, radiodetermination, amateur and related satellite services
P	Radiowave propagation
RA	Radio astronomy
RS	Remote sensing systems
S	Fixed-satellite service
SA	Space applications and meteorology
SF	Frequency sharing and coordination between fixed-satellite and fixed service systems
SM	Spectrum management
SNG	Satellite news gathering
TF	Time signals and frequency standards emissions
V	Vocabulary and related subjects

Note: *This ITU-R Recommendation was approved in English under the procedure detailed in Resolution ITU-R 1.*

Electronic Publication
Geneva, 2010

RECOMMENDATION ITU-R BT.1869

Multiplexing scheme for variable-length packets in digital multimedia broadcasting systems*

(Question ITU-R 45/6)

(2010)

Scope

This Recommendation deals with multiplexing schemes for variable-length packets over broadcasting channels. Specifications are given for schemes for transporting IP packets over broadcasting channels: encapsulation format, header compressed IP packet format, and transmission control signals.

The ITU Radiocommunication Assembly,

considering

a) that various kinds of signals for multimedia services may be delivered in digital broadcasting;

b) that multimedia services have also been introduced in telecommunication networks where IP packets including IPv4 and IPv6 packets are used;

c) that those IP packets are variable-length in essence with a maximum length of 65 535 bytes;

d) that an IP-friendly transport mechanism is desirable for multimedia broadcasting services to enable harmonization between broadcasting services and telecommunication services;

e) that an MPEG-2 transport stream has been adopted for digital broadcasting as a means of transporting various kinds of signals;

f) that the MPEG-2 transport stream consists of short fixed-length packets of 188 bytes including a 184-byte payload;

g) that a multiplexing scheme which enables more efficient transport and less complex reception of variable-length packets is desired for multimedia broadcasting,

recommends

1 that for transport of variable-length packets in digital multimedia broadcasting systems, the multiplexing scheme described in Annex 1 should be used;

2 that compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words “shall” or some other obligatory language such as “must” and the negative equivalents are used to express requirements. The use of such words shall in no way be construed to imply partial or total compliance with this Recommendation.

* This Recommendation should be brought to the attention of ITU-T Study Groups 9 and 16.

Annex 1

Multiplexing scheme for variable-length packets

References

Normative references

[1] IETF RFC 791: Internet Protocol.

This IETF standard is available at the following address. http://www.ietf.org/rfc/rfc791.txt

[2] IETF RFC 2460: Internet Protocol, Version 6 (IPv6) Specification.

This IETF standard is available at the following address. http://www.ietf.org/rfc/rfc2460.txt

[3] IETF RFC 768: User Datagram Protocol.

This IETF standard is available at the following address. http://www.ietf.org/rfc/rfc768.txt

[4] ETSI TS 102 606 v1.1.1(2007-10): Digital Video Broadcasting (DVB); Generic Stream Encapsulation (GSE) Protocol.

[5] ETSI EN 301 192 v1.4.2(2008-04): Digital Video Broadcasting (DVB); DVB specification for data broadcasting.

Informative references

[6] ITU-T Recommendation H.222.0, 2006: Information technology – Generic coding of moving pictures and associated audio information: Systems.

Abbreviations

ACM	adaptive coding and modulation
AMT	address map table
ATM	asynchronous transfer mode
CID	context identification
CRC	cyclic redundancy check
DVB	digital video broadcast
ETSI	European Telecommunications Standards Institute
GSE	generic stream encapsulation
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
INT	IP/MAC notification table
IP	internet protocol
MAC	media access control

MLD multicast listener discovery

MPE multi protocol encapsulation

MPEG Moving Pictures Experts Group

NIT network information table

ONU optical network unit

PES packetized elementary stream

RFC Request For Comment (IETF standard)

SN sequence number

TLV type length value

TS transport stream

UDP user datagram protocol

VCM variable coding and modulation

1 Introduction

Various multimedia broadcasting services are expected to be made possible by adopting the multiplexing schemes for fixed-length MPEG-2 TS packets and that for variable-length packets as depicted in Fig. 1.

FIGURE 1

Protocol stack

<table>
<tr><td colspan="4">Multimedia broadcasting</td></tr>
<tr><td colspan="2">Real-time services</td><td colspan="2">IP-based services</td></tr>
<tr><td>Video and audio</td><td>Data and control</td><td>A/V file</td><td>Control</td></tr>
<tr><td>PES</td><td>Section</td><td>IP packet</td><td>Signalling packet</td></tr>
<tr><td colspan="2">MPEG-2 TS</td><td colspan="2">Multiplexing scheme for variable- length packets</td></tr>
<tr><td colspan="4">Transmission slot (channel coding and modulation)</td></tr>
<tr><td colspan="4">Physical layer (terrestrial/satellite)</td></tr>
</table>

BT.1869-01

2 Requirements for multiplexing scheme for variable-length packets

Because broadcasting services use radio spectrum, which is a finite resource, and similar services using the Internet have been launched, a multiplexing scheme for variable-length packets should support the following requirements:

a) variable-length packets of various formats including IPv4 and IPv6 packets can be multiplexed;

b) a maximum 65 535-byte-long packet can be multiplexed without fragmentation;

c) the overhead needed to transmit packets should be small;

d) the receiving process should be simple enough to process received packets at a high packet rate.

3 Encapsulation scheme for variable-length packets

3.1 Format of type-length-value container

The type-length-value (TLV) multiplexing scheme is shown in Fig. 2 and Table 1. This scheme can multiplex variable-length packets of any format unless packet filtering and fragmentation are needed. The type of packet is indicated by the packet_type field, and the length of the packet is indicated by the length field. Header compressed IP packets and transmission control signals can also be encapsulated into TLV containers. This scheme enables multiplexing a maximum 65 535-byte-long packet without fragmentation. The transmission overhead is small and the TLV multiplexing scheme efficiently uses transmission capacity.

FIGURE 2

Format of TLV container

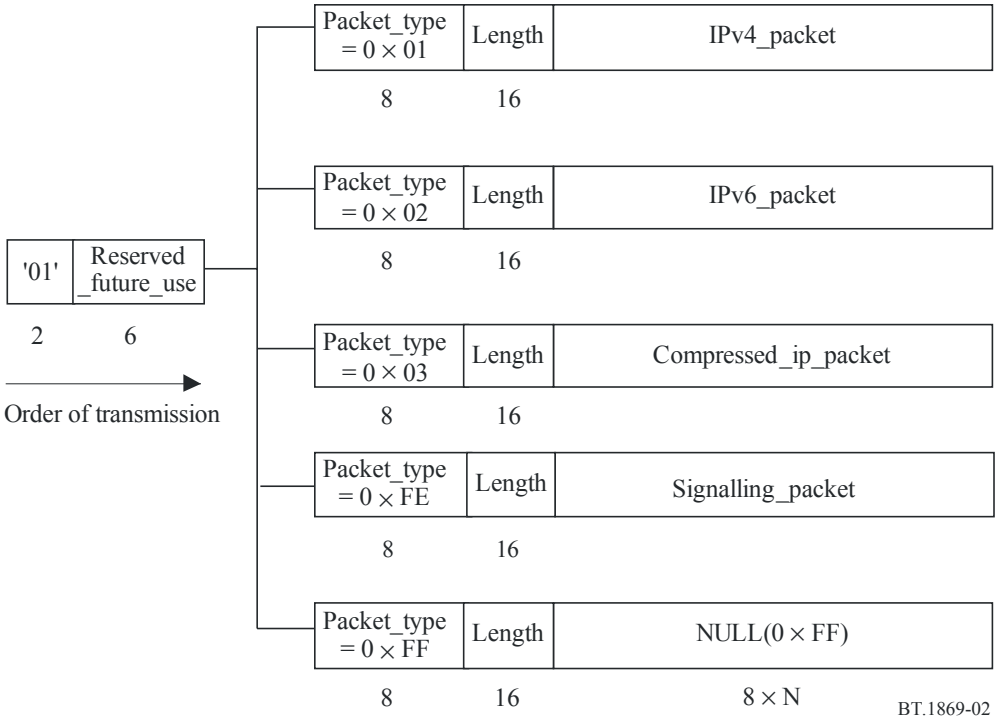


TABLE 1

TLV container

Syntax	No. of bits	Mnemonic
TLV {		
'01'	2	bslbf
reserved_future_use	6	bslbf
packet_type	8	bslbf
length	16	uimsbf
if (packet_type==0x01)		
IPv4_packet ( )		

TABLE 1 (*end*)

Syntax	No. of bits	Mnemonic
else if (packet_type==0x02)		
IPv6_packet ( )		
else if (packet_type==0x03)		
compressed_ip_packet( )		
else if (packet_type==0xFE)		
signalling_packet ( )		
else if (packet_type==0xFF){		
for(i=0;i<N;i++){		
NULL	8	bslbf
}		
}		
}		

reserved_future_use – This indicates that the value may be used for future extensions. Unless otherwise specified within this document, all reserved bits are set to "1".

packet_type – This indicates which type of packet is encapsulated. It is coded according to Table 2.

TABLE 2

Packet type assignment values

Value	Description
0x00	Reserved
0x01	IPv4 packet
0x02	IPv6 packet
0x03	IP packet with header compression
0x04 – 0xFD	Reserved
0xFE	Signalling packet
0xFF	NULL packet

length – This field specifies the number of bytes immediately following the length field to the end of the TLV container.

IPv4_packet ( ) – This indicates an IPv4 packet, which has an IPv4 header defined in RFC 791 [1].

IPv6_packet ( ) – This indicates an IPv6 packet, which has an IPv6 header defined in RFC 2460 [2].

compressed_ip_packet ( ) – This indicates an IP packet having compressed headers presented in § 4.

signalling_packet ( ) – This indicates the transmission control signals presented in § 5.

NULL – These are the fixed 8-bit stuffing bytes with the value "0xFF".

3.2 Format of Generic Stream Encapsulation packet

The Generic Stream Encapsulation (GSE) specified in ETSI TS 102 606 [4] is able to encapsulate variable-length packets, such as IP packets. Each GSE packet may have a label field and a CRC field. Receivers can filter packets they receive by using the label field of each packet. When GSE packets are fragmented into pieces to be set into transmission slots, the integrity of the restored packets can be ensured by checking the CRC.

The GSE protocol has been devised as an adaptation layer to provide network layer packet encapsulation and fragmentation functions over Generic Stream. GSE provides efficient encapsulation of IP packets over variable-length layer 2 packets, which are then directly scheduled on the physical layer into baseband frames.

GSE maximizes the efficiency of IP packet transport reducing overhead by a factor of 2 to 3 with respect to MPE over MPEG-TS. This is achieved without any compromise of the functionalities provided by the protocol, due to the variable-length layer 2 packet size, suited to IP traffic characteristics.

GSE also provides additional features that increase the protocol flexibility and applicability. Some key GSE functions/characteristics are:

1 Support for multi-protocol encapsulation (e.g. IPv4, IPv6, MPEG, ATM, Ethernet, and VLANs).

2 Transparency to network layer functions, including IP encryption and IP header compression.

3 Support of several addressing modes: In addition to the 6-byte MAC address (including multicast and unicast), it supports a MAC addressless mode, and an optional 3-byte address mode.

4 A mechanism for fragmenting IP packets or other network layer packets over baseband frames to support ACM/VCM.

5 Support for hardware filtering.

6 Extensibility: additional link protocols can be included through specific protocol type values (e.g. layer 2 security, IP header compression, etc.).

7 Low complexity.

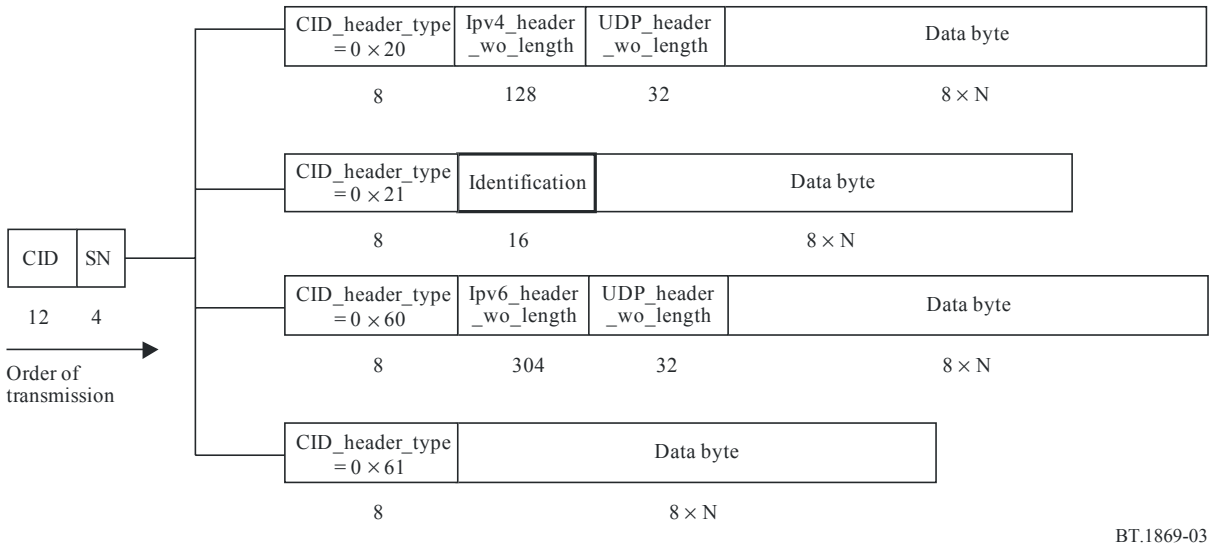
4 IP packet header compression (Header Compression for Broadcasting: HCfB)

When IP packets are to be conveyed as variable-length packets, it is convenient for broadcasting services to have much compatibility with various services using telecommunication networks. Each IP packet generally has at least 20 bytes of IPv4 header or 40 bytes of IPv6 header, besides 8 bytes of UDP header. Based on these headers, routers in telecommunication networks need to decide which way each packet is to be transferred. Hence, these headers are very important in telecommunication networks. On the other hand, they are never necessary in broadcasting channels, since all packets in broadcasting channels are just transferred to receivers. Transfer throughput can be increased if this unused header information is compressed.

The format of a header compressed IP packet is shown in Fig. 3 and Table 3. This reduces IP and UDP headers to 3 or 5 bytes of compressed header for most packets. When content is transferred on IP packets, most fields in these headers are constant during connection. Once an uncompressed header is sent, these fields with the same values in the following packets may not necessarily be sent. Based on this principle, IP and UDP headers with all the information are sent at long intervals, and the compressed headers are sent for almost all packets. The compressed headers are restored at a receiver by filling them with the header of a preceding packet that has all the information.

FIGURE 3

Format of header compressed IP packet



BT.1869-03

TABLE 3

Header compressed IP packet

Syntax	No. of bits	Mnemonic
compressed_ip_packet ( ) {		
CID	12	uimsbf
SN	4	uimsbf
CID_header_type	8	uimsbf
If (CID_header_type==0x20) {		
IPv4_header_wo_length ( )		
UDP_header_wo_length ( )		
for(i=0;i<N;i++){		
packet_data_byte	8	bslbf
}		
}		
else if (CID_header_type==0x21) {		
Identification	16	bslbf
for(i=0;i<N;i++){		
packet_data_byte	8	bslbf
}		
}		
else if(CID_header_type==0x60) {		
IPv6_header_wo_length ( )		
UDP_header_wo_length ( )		
for(i=0;i<N;i++){		

TABLE 3 (*end*)

Syntax	No. of bits	Mnemonic
packet_data_byte	8	bslbf
}		
}		
else if(CID_header_type==0x61) {		
for(i=0;i<N;i++){		
packet_data_byte	8	bslbf
}		
}		
}		

CID – Context IDentification – This indicates the IP flow, which is identified by the combination of the following fields. For IPv4, this is source IP address, destination IP address, protocol, source port number, and destination port number. For IPv6, this is source IP address, destination IP address, next_header, source port number, and destination port number.

SN – Sequence Number – This is a 4-bit field incrementing with each packet with the same CID. The SN wraps around to 0 after its maximum value.

CID_header_type – This field indicates which type of header the packet has. It is coded according to Table 4.

TABLE 4

CID_header_type assignment value

Value	Description
0x00 – 0x1F	Reserved
0x20	Full header of packet with IPv4 and UDP headers
0x21	Compressed header of packet with IPv4 and UDP headers
0x22 – 0x5F	Reserved
0x60	Full header of packet with IPv6 and UDP headers
0x61	Compressed header of packet with IPv6 and UDP headers
0x62 – 0xFF	Reserved

Identification – This field contains the IP identification of the IPv4 header.

IPv4_header_wo_length ( ) – This is an IPv4 header without either the total_length field or the header_checksum field shown in Fig. 4 and Table 5.

FIGURE 4

Structure of IPv4_header_wo_length ( )

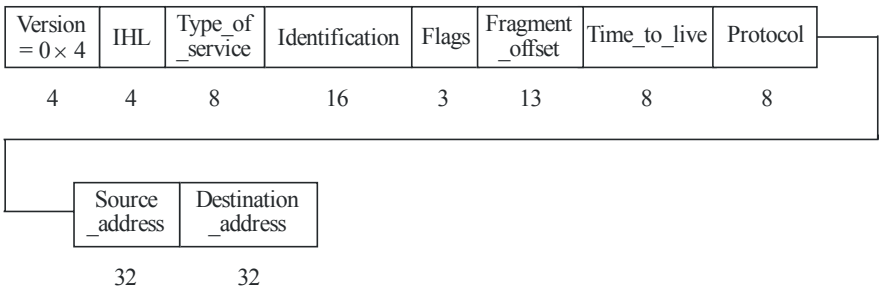


TABLE 5

IPv4_header_wo_length

Syntax	No. of bits	Mnemonic
IPv4_header_wo_length ( ) {		
version	4	uimsbf
IHL	4	uimsbf
type_of_service	8	bslbf
identification	16	bslbf
flags	3	bslbf
fragment_offset	13	uimsbf
time_to_live	8	uimsbf
protocol	8	bslbf
source_address	32	bslbf
destination_address	32	bslbf
}		

IPv6_header_wo_length ( ) – This is an IPv6 header without the payload_length field shown in Fig. 5 and Table 6.

FIGURE 5

Structure of IPv6_header_wo_length ( )

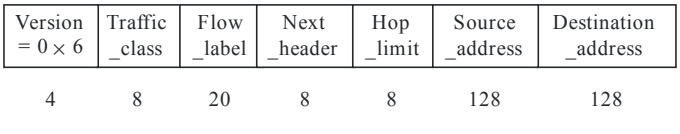


TABLE 6

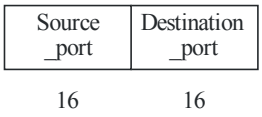
IPv6_header_wo_length

Syntax	No. of bits	Mnemonic
IPv6_header_wo_length ( ) {		
version	4	uimsbf
traffic_class	8	bslbf
flow_label	20	bslbf
next_header	8	bslbf
hop_limit	8	uimsbf
source_address	128	bslbf
destination_address	128	bslbf
}		

UDP_header_wo_length ( ) – This is a UDP header [3] without either the length field or the checksum field shown in Fig. 6 and Table 7.

FIGURE 6

Structure of UDP_header_wo_length ( )



BT.1869-06

TABLE 7

UDP_header_wo_length

Syntax	No. of bits	Mnemonic
UDP_header_wo_length ( ) {		
source_port	16	uimsbf
destination_port	16	uimsbf
}		

5 Control signals for multiplexing IP packets

A receiver needs to identify a desired IP data stream to demultiplex in the broadcasting signals.

5.1 Control signals for IP packets conveyed over MPEG-2 TS packets

For IP packets conveyed over MPEG-2 TS packets by means such as multi protocol encapsulation, the IP/MAC notification table (INT) as per ETSI EN 301 192 [5] can be used to accomplish IP address resolution. With INT, receivers are able to identify the desired IP data stream in the broadcasting signals.

5.2 Control signals for IP packets conveyed over TLV containers

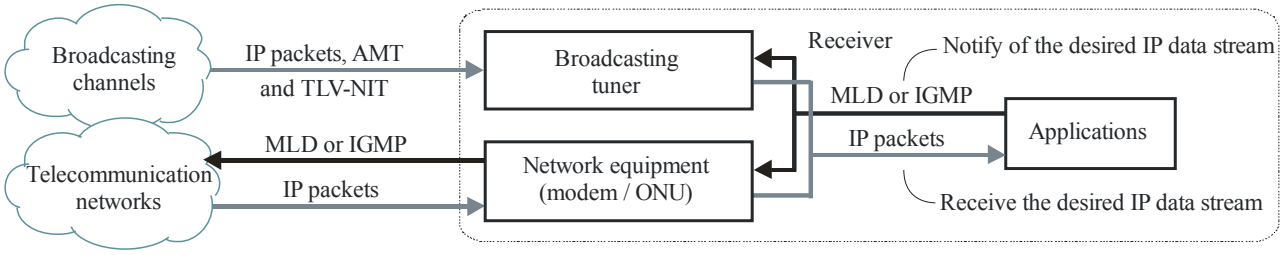
For IP packets not conveyed over MPEG-2 TS packets but over TLV containers, an Address Map Table (AMT) and a TLV-Network Information Table (TLV-NIT) are defined.

The AMT is used to list IP multicast group addresses associated with a **service_id** identifying the service that broadcasting channels are offering. The TLV-NIT is used to associate the **service_id** with the **TLV_stream_id** or other physical organizations of the signals carried via a given network and the characteristics of the network itself. The TLV-NIT is the same as the NIT in MPEG-2 systems, except that it is transmitted by the signalling packet in the TLV container.

When a receiver is notified of the desired IP data stream, it identifies the broadcasting signal in which that IP data stream is multiplexed by referring to the AMT and TLV-NIT, and it then tunes in to that signal. To notify the desired IP data stream, applications can use MLD or IGMP, which are widely used in telecommunication networks to control receiving IP multicast packets. Because of the mechanism using AMT and TLV-NIT, applications can acquire the intended IP data stream without having to distinguish whether it comes from broadcasting channels or telecommunication networks, as illustrated in Fig. 7.

FIGURE 7

Applications acquire content without distinguishing between channels



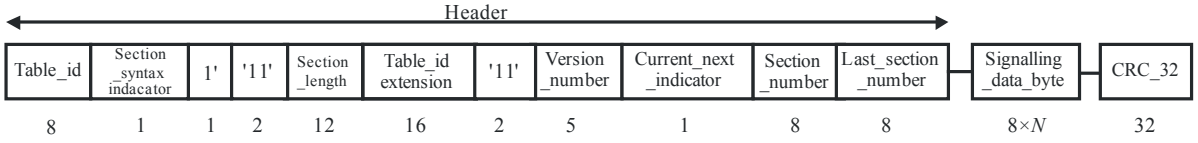
BT.1869-07

5.2.1 Structure of section extended format

The structures of the transmission control signals comply with the section-extended format shown in Fig. 8 and Table 8.

FIGURE 8

Structure of section-extended format



BT.1869-08

TABLE 8

Section-extended format

Syntax	No. of bits	Mnemonic
signalling_packet ( ) {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'1'	1	bslbf
'11'	2	bslbf
section_length	12	uimsbf
table_id_extension	16	uimsbf
'11'	2	bslbf
version_number	5	umisbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0; i<N; i++) {		
signalling_data_byte	8	bslbf
}		
CRC_32	32	rpchof
}		

table_id – This is an 8-bit field identifying the table to which the section belongs. The value of this field is as shown in Table 9.

TABLE 9

Table_id assignment values

Value	Description
0x00 – 0x3F	Reserved
0x40	TLV-NIT (TLV-Network Information Table) (actual network)
0x41	TLV-NIT (TLV-Network Information Table) (any other network)
0x42 – 0xFD	Reserved
0xFE	Table is indicated by the value of table_id_extension field
0xFF	Reserved

section_syntax_indicator – This is a field determining whether a normal or extension format is used and represents normal and extension formats, respectively, when this field contains “0” and “1”.

section_length – The section_length is a field that writes the number of data bytes following this field, and does not exceed 4093.

table_id_extension – This is a field extending the table identifier. When the value of the table_id field is 0xFE, this field is used to identify the table, as shown in Table 10.

TABLE 10

Table id extension assignment values

Value	Description
0x0000	AMT (Address Map Table)
0x0001 – 0xFFFF	Reserved

version_number – This is a field that writes the table version number.

current_next_indicator – This field contains "1" and "0", respectively, when the table is currently used and when the table cannot be used at present, but will be valid next.

section_number – This is a field that writes the number of the first section comprising the table.

last_section_number – This is a field that writes the number of the last section comprising the table.

signalling_data_byte – This field is used to contain transmission control signals.

CRC_32 – This field complies with ITU-T Recommendation H.222.0.

5.2.2 Structure of transmission control signals

All signals multiplexed with TLV containers are controlled by the following transmission control signals.

– A TLV-NIT that carries information correlating modulation frequencies and other information on transmission channels with broadcast programmes.

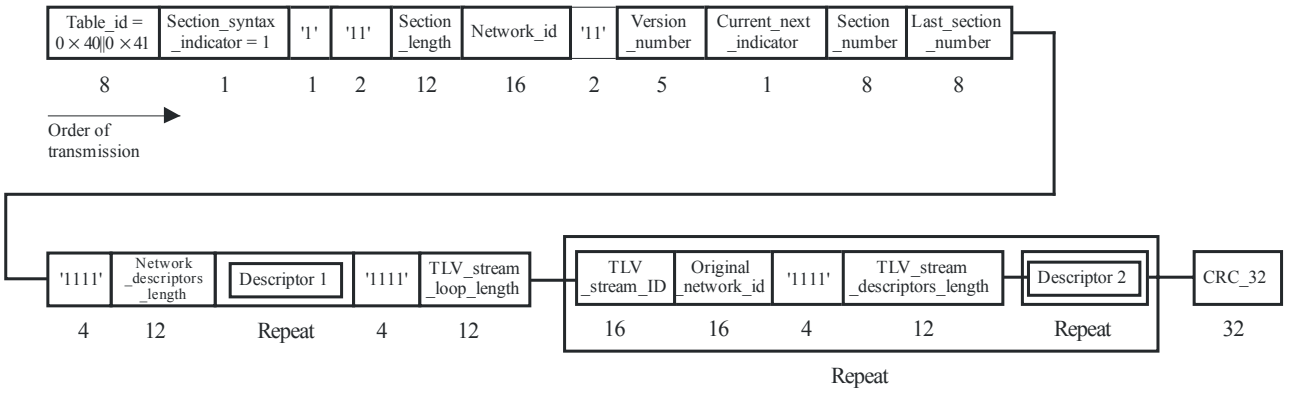
– An AMT that associates IP addresses specifying IP data flows with their broadcast services.

5.2.2.1 TLV-Network Information Table (TLV-NIT)

Figure 9 and Table 11 show the structure of TLV-NIT.

FIGURE 9

Structure of TLV-NIT



BT.1869-09

TABLE 11

TLV-NIT

Syntax	No. of bits	Mnemonic
TLV_network_information_table ( ){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'1'	1	bslbf
'11'	2	bslbf
section_length	12	uimsbf
network_id	16	uimsbf
'11'	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
network_descriptors_length	12	bslbf
for(i=0;i<N;i++){		
descriptor ( )		
}		
reserved_future_use	4	bslbf
TLV_stream_loop_length	12	uimsbf
for(i=0;i<N;i++){		
TLV_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
TLV_stream_descriptors_length	12	uimsbf
for(j=0;j<N;j++){		
descriptor ( )		
}		
}		
CRC_32	32	rpchof
}		

table_id – This is an 8-bit field identifying the table to which the section belongs. The value of this field is as shown in Table 9.

section_syntax_indicator – This field is set to "1", which represents the section-extended format.

section_length – This is a 12-bit field, the first two bits of which is "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length does not exceed 1021, so that the entire section has a maximum length of 1 024 bytes.

network_id – This is a 16-bit field that serves as a label to identify the delivery system, which the TLV-NIT informs about, from any other delivery system.

version_number – This is a field that writes the table version number.

current_next_indicator – This field contains "1" and "0", respectively, when the table is currently used and when the table cannot be used at present, but will be valid next.

section_number – This is a field that writes the number of the first section comprising the table.

last_section_number – This is a field that writes the number of the last section comprising the table.

network_descriptors_length – The value of the first two bits of this field is "00". The remaining 10 bits is a field that writes the number of bytes in the descriptor that follows the network_descriptors_length.

TLV_stream_loop_length – The value of the first two bits of this field is "00". The remaining 10 bits is a field that writes the number of data bytes following this field.

TLV_stream_id – This field represents the identification number of the applicable TLV stream.

original_network_id – This field represents the identification number of the original network of the applicable TLV stream.

TLV_stream_descriptors_length – This field represents the number of bytes in all descriptors of the applicable TLV stream immediately after this field. Note that the value of the first two bits is "00".

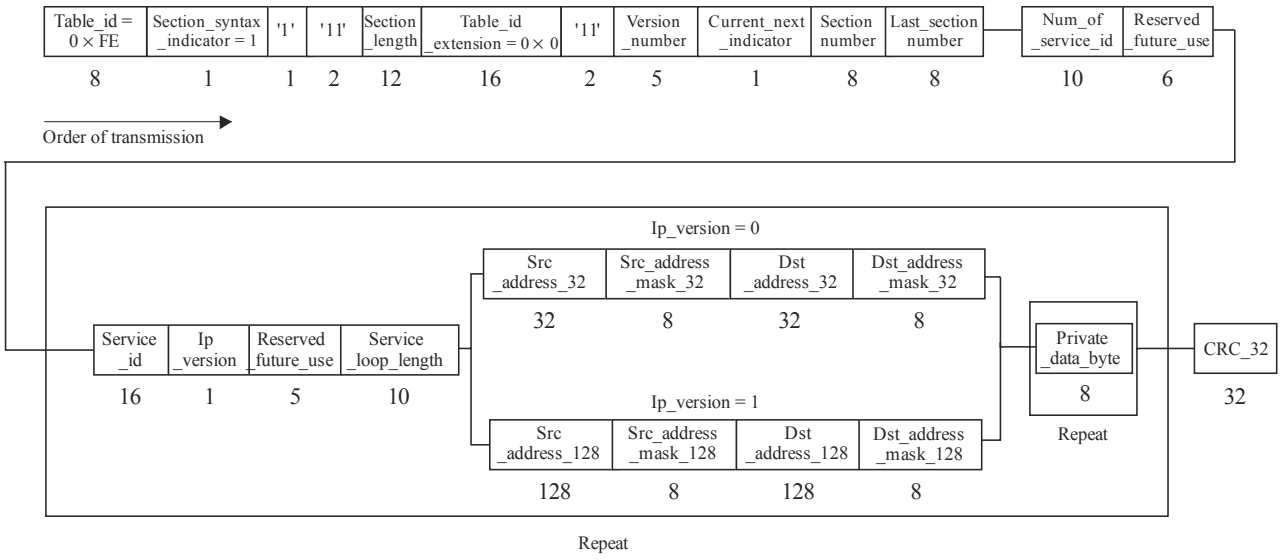
CRC_32 – This field complies with ITU-T Recommendation H.222.0.

5.2.2.2 Address map table

The AMT provides a flexible mechanism for carrying information about the services that IP data flows offer within TLV transferred networks. This table provides a list of the IP addresses which make up each service. Figure 10 and Table 12 show the structure of AMT.

FIGURE 10

Structure of AMT



BT.1869-10

TABLE 12

AMT

Syntax	No. of bits	Mnemonic
address_map_table ( ) {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'1'	1	bslbf
'11'	2	bslbf
section_length	12	uimsbf
table_id_extension	16	uimsbf
'11'	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
num_of_service_id	10	uimsbf
reserved_future_use	6	bslbf
for (i=0; i<num_of_service_id ; i++) {		
service_id	16	uimsbf
ip_version	1	bslbf
reserved_future_use	5	bslbf

TABLE 12 (*end*)

Syntax	No. of bits	Mnemonic
service_loop_length	10	uimsbf
if (ip_version=='0'){ /*IPv4*/		
src_address_32	32	bslbf
src_address_mask_32	8	uimsbf
dst_address_32	32	bslbf
dst_address_mask_32	8	uimsbf
}		
else if (ip_version=='1') { /*IPv6*/		
src_address_128	128	bslbf
src_address_mask_128	8	uimsbf
dst_address_128	128	bslbf
dst_address_mask_128	8	uimsbf
}		
for (j=0; i<N; j++) {		
private_data_byte	8	bslbf
}		
}		
CRC_32	32	rpchof
}		

table_id – The value of this field is set to 0xFE, representing that the table is identified by the value of the table_id_extension.

section_syntax_indicator – This field is set to "1", which represents the section-extended format.

section_length – The section_length is a field that writes the number of data bytes following this field, and does not exceed 4093.

table_id_extension – The value of this field is set to 0x0000, representing the Address Map Table.

version_number – This is a field that writes the table version number.

current_next_indicator – This field contains "1" and "0", respectively, when the table is currently used and when the table cannot be used at present but will be valid next.

section_number – This is a field that writes the number of the first section comprising the table.

last_section_number – This is a field that writes the number of the last section comprising the table.

num_of_service_id – This field indicates the number of service_id listed in this Address Map Table.

service_id – This is a 16-bit field which identifies the service the IP data flow provides.

ip_version – This field indicates the version of the IP, and represents IPv4 and IPv6, respectively, when this field contains "0" and "1".

service_loop_length – This field represents the number of bytes following this field to the next listed service_id field or to just prior CRC_32 field.

src_address_32 – This field specifies an IPv4 source address. The IPv4 address is fragmented into 4 fields of 8 bits where the first byte contains the most significant byte of the IPv4 source address.

src_address_mask_32 – This field specifies an IPv4 mask to define which bits of the IPv4 source address are used for comparison. The specified number of bits from the most significant bit is compared against the bits in the equivalent position of the src_address_32.

dst_address_32 – This field specifies an IPv4 destination address. The IPv4 address is fragmented into 4 fields of 8 bits where the first byte contains the most significant byte of the IPv4 destination address.

dst_address_mask_32 – This field specifies an IPv4 mask to define which bits of the IPv4 destination address are used for comparison. The specified number of bits from the most significant bit is compared against the bits in the equivalent position of the dst_address_32.

src_address_128 – This field specifies an IPv6 source address. The IPv6 address is fragmented into 8 fields of 16 bits where the first byte contains the most significant byte of the IPv6 source address.

src_address_mask_128 – This field specifies an IPv6 mask to define which bits of the IPv6 source address are used for comparison. The specified number of bits from the most significant bit is compared against the bits in the equivalent position of the src_address_128.

dst_address_128 – This field specifies an IPv6 destination address. The IPv6 address is fragmented into 8 fields of 16 bits where the first byte contains the most significant byte of the IPv6 destination address.

dst_address_mask_128 – This field specifies an IPv6 mask to define which bits of the IPv6 destination address are used for comparison. The specified number of bits from the most significant bit is compared against the bits in the equivalent position of the dst_address_128.

private_data_byte – The value of this field is privately defined.